TKK Dissertations 149
Espoo 2008

METHODOLOGY FOR UTILISING PRIOR KNOWLEDGE IN CONSTRUCTING DATA-BASED PROCESS MONITORING SYSTEMS WITH AN APPLICATION TO A DEAROMATISATION PROCESS

Doctoral Dissertation

Mikko Vermasvuori

Helsinki University of Technology
Faculty of Chemistry and Materials Sciences
Department of Biotechnology and Chemical Technology

TKK Dissertations 149
Espoo 2008

METHODOLOGY FOR UTILISING PRIOR KNOWLEDGE IN CONSTRUCTING DATA-BASED PROCESS MONITORING SYSTEMS WITH AN APPLICATION TO A DEAROMATISATION PROCESS

Doctoral Dissertation

Mikko Vermasvuori

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Chemistry and Materials Sciences for public examination and debate in Auditorium KE2 (Komppa Auditorium) at Helsinki University of Technology (Espoo, Finland) on the 12th of December, 2008, at 12 noon.

Helsinki University of Technology
Faculty of Chemistry and Materials Sciences
Department of Biotechnology and Chemical Technology

Teknillinen korkeakoulu
Kemian ja materiaalitieteiden tiedekunta
Biotekniikan ja kemian tekniikan laitos

Distribution:
Helsinki University of Technology
Faculty of Chemistry and Materials Sciences
Department of Biotechnology and Chemical Technology
P.O. Box 6100
FI - 02015 TKK
FINLAND
URL: http://chemtech.tkk.fi/
Tel. +358-9-4511
E-mail: mikko.vermasvuori@nesteoil.com

ISBN 978-951-22-9683-5
ISBN 978-951-22-9684-2 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)
URL: http://lib.tkk.fi/Diss/2008/isbn9789512296842/

TKK-DISS-2547

Picaset Oy
Helsinki 2008

ABSTRACT OF DOCTORAL DISSERTATION	HELSINKI UNIVERSITY OF TECHNOLOGY P. O. BOX 1000, FI-02015 TKK http://www.tkk.fi

Author Mikko Vermasvuori

Name of the dissertation

Methodology for utilising prior knowledge in constructing data-based process monitoring systems with an application to a dearomatisation process

Manuscript submitted 23.9.2008	Manuscript revised
Date of the defence 12.12.2008	
[X] Monograph	[ ] Article dissertation (summary + original articles)

Department	Biotechnology and Chemical Technology
Laboratory	Process Control and Automation
Field of research	Process monitoring, fault detection
Opponent(s)	Prof. Christophe Aubrun and Prof. Yrjö Hiltunen
Supervisor	Prof. Sirkka-Liisa Jämsä-Jounela
Instructor	

Abstract

Global competition is forcing the process industry to optimise the production processes. One key factor in optimisation is effective process state monitoring and fault detection. Another motivator to improve process monitoring systems are the substantial losses of revenue resulting from abnormal process conditions. It has been estimated that the petrochemical industry in the US alone loses 20 billion dollars per year because of unoptimal handling of abnormal process situations. Traditionally, the monitoring systems have been based on first principle models, constructed by specialists with process specific expertise. In contrast, the use of data-based modelling methods require less expertise and offers the possibilities to build and update the monitoring models in a short period of time, thus allowing more efficient development of monitoring systems.

The aims of this thesis are to augment data-driven modelling with existing process knowledge, to combine different data-based modelling methods, and to utilise calculated variables in modelling in order to improve the accuracy of fault detection and identification (FDI) and to provide all necessary diagnostic information for fault tolerant control. The suggested improvements are included in a methodology for setting up FDI systems. The methodology has been tested by building FDI systems for detecting faults in two online quality analysers in a simulated and in a real industrial dearomatisation process at the Naantali oil refinery (Neste Oil Oyj). In developing an FDI system, background information about the user requirements for the monitoring system is first acquired. The information is then analysed and suitable modelling methods are selected according to the guidelines given in the methodology. Second, the process data are prepared for the modelling methods and augmented with appropriate calculated variables. Next, the input variable sets are determined with the introduced method and the models are constructed. After the estimation accuracy of the models is validated, the values of the fault detection parameters are determined. Finally, the fault detection performance of the system is tested. The system was evaluated during a period of one month at the Naantali refinery in 2007. The monitoring system was able to detect all the introduced analyser faults and to provide the information needed for a fault tolerant control system, thus validating the methodology. The effects of a number of suggested improvements in data-based modelling are analysed by means of a comparison study.

Keywords Process monitoring, fault detection, data-based modelling	
ISBN (printed) 978-951-22-9683-5	ISSN (printed) 1795-2239
ISBN (pdf) 978-951-22-9684-2	ISSN (pdf) 1795-4584
Language English	Number of pages 212
Publisher Helsinki University of Technology, Department of Biotechnology and Chemical Technology	
Print distribution Helsinki University of Technology, Department of Biotechnology and Chemical Technology	
[X] The dissertation can be read at http://lib.tkk.fi/Diss/2008/isbn9789512296842	

VÄITÖSKIRJAN TIIVISTELMÄ	TEKNILLINEN KORKEAKOULU PL 1000, 02015 TKK http://www.tkk.fi

Tekijä Mikko Vermasvuori

Väitöskirjan nimi

Methodology for utilising prior knowledge in constructing data-based process monitoring systems with an application to a dearomatisation process

Käsikirjoituksen päivämäärä 23.9.2008	Korjatun käsikirjoituksen päivämäärä
Väitöstilaisuuden ajankohta 12.12.2008	
☒ Monografia	☐ Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)

Osasto	Biotekniikan ja kemian tekniikan laitos
Laboratorio	Prosessien ohjauksen ja automaation laboratorio
Tutkimusala	Prosessien ohjaus
Vastaväittäjä(t)	Prof. Christophe Aubrun ja Prof. Yrjö Hiltunen
Työn valvoja	Prof. Sirkka-Liisa Jämsä-Jounela
Työn ohjaaja	

Tiivistelmä

Yleismaailmallinen kilpailu pakottaa prosessiteollisuutta optimoimaan tuotantoprosessejaan pysyäkseen kilpailukykyisenä. Eräs optimoinnin tärkeä osa-alue on prosessien ja prosessilaitteiden vikatilojen tunnistaminen. Prosessien monitoroinnin lisäämistä motivoi myös epänormaalien operointitilanteiden aiheuttamat kustannukset. Yhdysvaltojen petrokemianteollisuuden on arvioitu menettävän 20 miljardia dollaria vuosittain epänormaalien prosessitilanteiden epäoptimaalisesta käsittelystä johtuen. Perinteisesti monitoroinnissa on käytetty fysikokemiallisia malleja, joiden tekeminen on työlästä ja vaatii erittäin hyvää prosessien tuntemusta. Datapohjaisilla mallinnusmenetelmillä mallien laatiminen ja päivittäminen on nopeaa, mikä mahdollistaa monitorointijärjestelmien aikaisempaa tehokkaamman kehittämisen.

Tämän väitöskirjan tarkoitus on parantaa datapohjaisen mallinnuksen tarkkuutta käyttämällä prosessitietämystä, yhdistelemällä erityyppisiä malleja sekä hyödyntämällä laskennallisia muuttujia. Monitoroinnista saatavaa tietoa käytetään edelleen vikasietoisessa säädössä. Ehdotetut parannukset datapohjaiseen monitorointiin on esitetty monitorointijärjestelmien kehittämiseen käytettävän metodologian muodossa. Metodologian toimivuutta on tarkasteltu kehittämällä sen avulla monitorointijärjestelmä Neste Oil Oyj:n Naantalin jalostamon liuotinaromaattien poistoprosessille. Monitorointijärjestelmän tarkoitus oli havaita tuotteen laatua mittaavien online-analysaattoreiden vikaantuminen. Järjestelmän laadinnan ensimmäisessä vaiheessa kartoitettiin käyttäjien vaatimukset järjestelmälle. Tietojen perusteella ja metodologiaa hyödyntäen valittiin kohteeseen sopivat datapohjaiset mallinnusmenetelmät. Mallien selittävät muuttujat valittiin mitattujen ja kehitettyjen laskennallisten muuttujien joukosta metodologiassa esitetyllä tavalla. Mallien suorituskyky määritettiin käyttämällä erillistä validointidataa. Tämän jälkeen vianhavaitsemisjärjestelmän viritysparametreille etsittiin optimaaliset arvot. Metodologiassa ehdotettujen datapohjaiseen mallinnukseen liittyvien parannusten vaikutuksia arvioitiin vertailututkimuksessa. Järjestelmän toiminta varmistettiin ensin simuloidussa prosessiympäristössä, jonka jälkeen se testattiin online Naantalin jalostamolla toukokuussa 2007. Tulosten tarkastelu vahvisti järjestelmän kykenevän havaitsemaan analysaattorivikoja ja antamaan vikasietoiselle säädölle sen tarvitsemat tiedot täten todistaen metodologian toimivuuden.

Asiasanat Prosessimonitorointi, vikojen havaitseminen, data-pohjaiset mallinnusmenetelmät	
ISBN (painettu) 978-951-22-9683-5	ISSN (painettu) 1795-2239
ISBN (pdf) 978-951-22-9684-2	ISSN (pdf) 1795-4584
Kieli Englanti	Sivumäärä 212
Julkaisija Teknillinen korkeakoulu, Biotekniikan ja kemian tekniikan laitos	
Painetun väitöskirjan jakelu Teknillinen korkeakoulu, Biotekniikan ja kemian tekniikan laitos	
☒ Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2008/isbn9789512296842	

Preface

The research work presented in this thesis has been carried out in the Laboratory of Process Control and Automation, Helsinki University of Technology (TKK), between 2004 and 2008. I would like to thank Professor Sirkka-Liisa Jämsä-Jounela for encouraging me to start my post-graduate studies, for her comprehensive academic guidance during the years and for her help in the writing of the thesis. Professor Raimo Ylinen is thanked for his valuable insights and advices.

I would like to thank the pre-examiners of the thesis: Prof. Christopher Aubrun from the University of Henri Poincare, France and Prof. Kauko Leiviskä from the Oulu University for their thorough review of the thesis and for their inspiring comments.

Most of the research presented in this thesis was done within the Networked control systems tolerant to Faults (NeCST) project. I would like to thank all participants of the project, and especially Mauri Sourander from Neste Jacobs Oy for sharing his expertise and for his comments on the thesis. I would also like to express my gratitude for the personnel of the Neste Oil Oyj's Naantali refinery for providing the interesting industrial fault detection problem for the study. I'd like to thank all my former colleagues at the process control laboratory, especially Nikolai Vatanski and Cheng Hui, for the interesting discussions during the coffee breaks. I thank Walter Ahlström foundation for supporting my research.

And most of all, I thank my wife Raisa for her invaluable part in taking care of our daughter Kiira when I was writing the thesis.

Espoo, 17.08.2008

Mikko Vermasvuori

Contents

List of Abbreviations

AC	Architecture Constraint
ANN	Artificial Neural Network
ARIMA	Auto Regressive Integrated Moving Average
ART	Adaptive Resonance Theory
ARTMAP	Adaptive Resonance Theory Neural Network combined with Gaussian classifier
ASM	Abnormal Situation Management
BE	Backward Elimination
BMU	Best Matching Unit
BSS	Blind Source Separation
CO	Constraint Optimisation
CSTR	Continuously Stirred Tank Reactor
CUSUM	Cumulative Sum
CV	Controlled variable
CVA	Canonical Variate Analysis
DC	Data Constraint
DCS	Distributed Control System
DDQI	Data Driven Quality Improvement
DISSIM	Process monitoring method based on dissimilarity of data
DPCA	Dynamic Principal Component Analysis
EIV	Errors-In-Variables
EKF	Extended Kalman Filter
FDD	Fault Detection and Diagnosis
FDI	Fault Detection and Identification
FIR	Finite Impulse Response
FLANN	Functional Link Artificial Neural Network
FP	Flashpoint temperature
FPLS	Fuzzy Partial Least Squares

FS	Forward Selection
FTC	Fault Tolerant Control
GA	Genetic Algorithm
GHSOM	Growing Hierarchical Self Organising Map
GMDH	Group Method of Data Handling
GP	Genetic Programming
GPCA	Generalized Principal Component Analysis
GTW	Process Computer Interface Gateway
GUI	Graphical User Interface
IBP	Initial Boiling Point Temperature
IC	Independent Component
ICA	Independent Component Analysis
IDEF0	Integration Definition (language 0) for Function modelling
ITNN	Input Training Neural Network
IVS	Input Variable Selection
J.PF	Joerding's Penalty Function
JYPLS	Joint-Y Partial Least Squares
KF	Kalman Filter
KPCA	Kernel Principal Component Analysis
KPLS	Kernel Partial Least Squares
LDA	Linear Discrete Analysis
LM	Levenberg-Marquardt
LOOCV	Leave-One-Out Cross Validation
LRGF	Locally Recurrent Globally Feedforward
LV	Latent Variable
LVQ	Learning Vector Quantization
MBPCA	Multi-Block Principal Component Analysis
McorR	Multiway Covariates Regression
MLP	Multilayer Perceptron
MMSOM	Marginal Median Self Organising Map

MOESP	Multivariable Output Error State Space
MPC	Model Predictive Controller
MPCA	Moving Principal Component Analysis
MPLS	Multi-way Partial Least Squares
MSPC	Multivariate statistical process control
MSPCA	Multiscale Principal Component Analysis
MTL	Multitask Learning
MV	Manipulated Variable
MWPCA	Multi-way Principal Component Analysis
NAPCON	Neste Advanced Process Controller
NeCST	Networked Control Systems Tolerant to Faults
N-PLS	Multilinear N-way Partial Least Squares
N4SID	Numerical algorithms for Subspace State Space System Identification
NLPCA	Nonlinear Principal Component Analysis
NLPLS	Nonlinear Partial Least Squares
NMPC	Nonlinear Model Predictive Control
OMS	Oil Management and Storage
OSS	Operator Support System
PC	Principal Component
PCA	Principal Component Analysis
PLS	Partial Least Squares / Projection to Latent Structures
P-SOM	Predictive Self-Organizing Map
QPLS	Quadratic Partial Least Squares
QTA	Qualitative Trend Analysis
RBFN	Radial Basis Function Network
RecBFN	Rectangular Basis Function Network
RMSE	Root Mean Squared Error
RMSPCA	Robust MultiScale Principal Component Analysis
RPCA	Recursive Principal Component Analysis
RPLS	Recursive Partial Least Squares

SFFS	Sequential Floating Forward Selection
SFS	Sequential Forward Selection
SISO	Single Input Single Output
SMI	Subspace Model Identification
SNR	Signal-To-Noise ratio
SPE	Squared Prediction Error
SOM	Self-Organizing Map
TASOM	Time Adaptive Self-Organizing Map
TKK	Helsinki University of Technology
TSK	Takagi-Sugeno-Kang fuzzy model
VMSOM	Vector Median Self Organising Map
WC	Weight Constraint

List of Symbols

Greek alphabet

Λ	Diagonal matrix with eigenvalues on the diagonal
λ	Eigenvalue
λ_f	Detection threshold
ν	Detection limit
Ψ	Fourier transform of a mother wavelet
Σ	Diagonal matrix with singular values on the diagonal
σ	Singular value
Θ	Sum of singular values

Latin alphabet

A	State space model state matrix
B_c	Coefficient matrix
B	State space model input matrix
C_c	Coefficient matrix
C	State space model output matrix
D	State space model feedthrough matrix
d	Delay
E	Residual matrix
F	A Flow
F_{est}	Estimated magnitude of a fault
F_{rel}	Reliability of a fault
f^0	State of the monitored signal (normal or faulty)

$\hat{f}$	Estimated fault magnitude
$\hat{f}^0$	Estimated state of the monitored signal (normal or faulty)
J_{MSE}	Mean squared error
k	Time instant
l	Order of a system
M	Coefficient matrix
m	Number of principal components
m_n	Minimum value of cumulative sum
n	Number of data samples or number of rows in a data matrix
P	Loading matrix
p	Loading value
Q	Squared Prediction Error (SPE) index
S	Sample covariance matrix
T	Score matrix
t	Score value
T^2	Hotelling T^2 index
U	Output matrix of singular value decomposition
U_n	Cumulative sum
u^{dc}	Delay compensated data sample
u_k^f	Filtered data sample
u_{tq}	Total quantity of a substance
X_s	Auto-scaled process data matrix
X_{sd}	Auto-scaled process data matrix also containing values of past measurements
y	Output of a system
y_d	Fault detection residual
$\hat{y}$	Estimated output of a system
$\hat{y}_{rel}$	Reliability of an estimated output value

List of Figures

List of Tables

1 Introduction

1.1 Background

In the highly competitive process industry continuous optimisation of the production processes is a crucial factor in keeping the operation economically viable. Another motivation for optimisation is the growing public awareness of environmental issues, leading to stricter legislation concerning emissions and waste handling. These demands for profitability and cleaner production can be met in part through effective process state monitoring and fault detection. Early detection of faults and process disturbances provides operators with possibilities to handle the abnormal situations more efficiently. According to a study by Lennox and Sandoz (2002), the petrochemical industry in the US alone loses 20 billion dollars per year because of un-optimal handling of abnormal process situations. The substantial size of the estimated losses has also motivated the academic community to develop fault detection methods and fault tolerant control systems to deal with process disturbances and equipment malfunctions.

The field of fault detection and diagnosis (FDD) is considered with the problem of making fault decisions, on the basis of measured process data. A fault decision is an indication of a fault, including information about its type, size, location and time of detection. FDD systems have hierarchical structures for extracting relevant process condition information from measurement data. According to Isermann (1997), the FDD systems consist of feature generation, symptom generation and fault diagnosis blocks. The feature generation includes methods, e.g. limit value checking, signal analysis or model-based process analysis, for creating characteristic values of the process. These values are compared with the corresponding values representing non-faulty operation in order to create analytical symptoms. The symptoms are

analysed by means of fault diagnosis methods. If no a-priori knowledge is available, then the fault diagnosis is performed with a classification method. If fault-symptom causalities are known, diagnostic reasoning strategies can be utilised. Venkatasubramanian *et al.* (2003a) define the four parts of the FDD structure as a measurement space, a feature space, a decision space and a class space. The common factor between this and the previously described structure is the step-wise extraction of process condition related information from measurement data. The extraction of the information, i.e. the different mappings from the measurement space to the class space can be performed with a wide variety of methods. The three main categories of the methods are: quantitative model-based, qualitative model-based and process history based methods. The groups are distinguished by the type of prior knowledge that is needed to use the methods. Quantitative model-based methods require a deep causal understanding of the quantitative relationships between the variables. Qualitative model-based methods rely on shallow knowledge about the qualitative interactions between different units of the process. Process history based, i.e. data-based methods, can be used without process-specific prior knowledge, although large quantities of process data, from which the essential knowledge about the process behaviour is extracted, are required. (Venkatasubramanian *et al.*, 2003a)

The quantitative model-based methods include observers, parity relations, Kalman filters and parameter estimation. These methods are used to achieve analytical redundancy: the inconsistencies between the measured and the estimated process behaviour are presented as symptoms called residuals. Depending on the method, the residuals can be deviations between measured and estimated outputs, parameter estimates or state estimates (Isermann, 1997). However, the applicability of the quantitative model-based methods is limited to linear processes (Venkatasubramanian *et al.*, 2003a). The second group of methods consists of the qualitative model-based methods. Qualitative methods are used when there is no deep understanding of the process and precise numerical models are not available (Lo *et al.*,

2004). Because the relationships between variables are less accurately described by the models, the qualitative models are less prone to modelling errors than the quantitative models. The drawback of the qualitative modelling is the occasional generation of spurious results. Qualitative models are most suitable for finding the root causes of faults in very complex or large processes. The most common model-based qualitative methods are signed digraphs, fault trees and qualitative physics (Venkatasubramanian *et al.*, 2003b).

The third major group of FDD methods are the qualitative and quantitative process history based methods. The most important qualitative data-based methods are expert systems and qualitative trend analysis (QTA). The expert systems are easy to develop and the transparency of the reasoning is a significant benefit. The downsides of the method are their limited representation power and poor updatability. QTA represents the measurement data as trends describing the process behaviour. The extraction of trends makes QTA robust against noise, but also computationally heavy when the number of variables is high. The quantitative data-based methods include artificial neural networks (ANN) and statistical methods. While the statistical methods are used only with linear processes, the ANNs can be applied to highly nonlinear processes. The fact that the quantitative data-based methods can be used with little prior knowledge, are fast to create and have a wide range of applicability, makes them an attractive option for fault detection applications. A large number of industrial applications of these data-based monitoring methods with successful results have been reported (e.g. Komulainen *et al.* 2004, Jämsä-Jounela *et al.* 2003 and Kämpjärvi *et al.*, 2007) and reviewed, e.g. by Isermann and Ballé (1997) and Meireles *et al.* (2003).

The information about fault detection and diagnosis has traditionally been used by operators, who have used their process knowledge to determine control actions that mitigate the effects of the faults on the process. Systems that automatically take into account the effects of the faults in process control have only recently been

introduced. These fault tolerant control (FTC) systems aim to maintain the performance of the process at a nominal level in the presence of faults. Passive and active approaches for FTC have been suggested in the literature, e.g. by Polycarpou and Helmicki (1995) , Rausch (1995), Ballé *et al.* (1998) and Noura *et al.* (2000). The passive approach utilises robust control techniques for ensuring that a closed loop system remains insensitive to certain faults. In the active approach, a new set of control parameters is determined for enabling the faulty system to reach the nominal system performance. An active FTC strategy formulated as a supervised control system has been presented in Aubrun *et al.* (2003) to improve the robustness of a sludge dewatering process. Several studies have been published about active FTC strategies utilising model predictive control (MPC) (Maciejowski (1999), Pranatyasto and Qin (2001) and Prakash *et al.* (2002)). More recently, Järvinen *et al.* (2006) showed that the inherent accommodation properties of MPC can readily be exploited to implement different types of FTC strategy, providing that the necessary FDI information is available. These concepts have been elaborated further in Sourander *et al.* (2006).

The efficiency of an FTC system depends e.g. on the type of fault, the robustness of the process and the availability of alternative control strategies, and the underlying assumption of FTC is that accurate fault detection and diagnosis information is readily available. FTC is thus a research topic that is closely related to the FDI problem.

1.2 Research problem and asserted hypothesis

The fundamental problem motivating this thesis work is to develop a methodology for creating a fault detection system for an industrial production plant by utilising data-based fault detection methods. With the help of efficient monitoring, processes can be controlled more accurately to run under the desired conditions. The main

benefit for running the process near an optimal state is that the product quality is constantly high which, in turn, may increase the market value of the product or at least reduce the quantity of off-spec production. The diagnostic information provided by the FDI also enables the use of fault tolerant control systems.

The recent tendency for using data-based techniques in FDI is well justified by the developments in modelling theory and in information technology because process history data are now easily accessible and the computational capacity of normal computers is high enough for solving the mathematical problems related to the modelling. However, using only process data for creating process models is not an optimal solution as prior knowledge could often be utilised to achieve better results. Integrating prior knowledge with the data-based FDI systems also makes them more attractive to the end-user because sometimes the difficult-to-interpret black-box models are not easily accepted by the operating personnel. The focus of this thesis work is to improve the accuracy and applicability of fault detection systems based on existing data-based modelling methods, and to expedite and systematize the development projects of the fault detection systems for industrial processes. The hypotheses of this thesis are:

(1) Modern data-based modelling methods can provide all the necessary diagnostic information needed to implement fault tolerant control in industrial environments.
(2) Estimation accuracy of linear data-based regression models (PLS and SMI) can, in certain situations, be improved by combining them with nonlinear models.
(3) Considering variables describing unmeasured process phenomena when selecting input variable sets for data-based models, in addition to the measured variables, may result in more accurate models. The values of the unmeasured variables are determined using the values of the measured variables.

To prove the claims of the hypothesis, the following four tasks are carried out during

the research:

Task 1. A methodology for creating a fault detection system based on data-based modelling and prior knowledge is created. The methodology is tested by applying it to a dearomatisation process.

Task 2. Different types of data-based process model are created and their suitability for detecting faults in online quality analysers and for providing diagnostic information for fault tolerant control of a dearomatisation process is analysed.

Task 3. The effects of using prior knowledge in data pre-treatment in connection with input variable selection and the creation of calculated variables on the fault detection results are studied with simulated data of a dearomatisation process. The task covers the utilization of both linear and non-linear data-based modelling methods.

Task 4. The usefulness of calculated variables describing unmeasured process phenomena in fault detection is proved on the basis of a comparative study using both linear and non-linear modelling methods and data of a simulated dearomatisation process.

1.3 Scope and outline of the thesis

This thesis covers the topic of using process history based quantitative modelling methods and prior process knowledge to achieve accurate methods for fault detection and process equipment state identification. The phases involved in developing a monitoring system are presented as a methodology that can be used to set up a monitoring method for an industrial process. The opportunities of utilising existing information about process operation in the individual phases of data-based process modelling are studied. The outputs of the developed monitoring system can be used

to achieve FTC, as shown e.g. in Sourander *et al.* (2006).

In the first chapter an introduction to the process monitoring is given, the research problem is described, and the hypotheses are asserted. In addition, the scope and significance of the thesis are also presented. A state-of-the-art review of data based fault detection methods is presented in chapter 2. The methodology for setting up a fault detection system is described in chapter 3. Chapter 4 is devoted to present the dearomatisation process and the industrial process environment of the Neste Oil Oyj Naantali refinery. Chapters 5 through 9 describe the setting up of fault detection systems for three testing experiments in accordance with the proposed methodology. The suitability of the FDI-related information for FTC of the dearomatisation process is evaluated in chapter 10. Analysis of the results in chapter 11 illustrates the positive effects of the most important improvements for data-based modelling suggested in the methodology. The thesis ends with a conclusion in chapter 12.

1.4 Contribution of the thesis and the author

The novelty of this work comes from improving the accuracy of the monitoring methods by combining linear data-based modelling methods, partial least squares (PLS) and subspace identified state-space model (SMI) with a nonlinear multi-layer perceptron network (MLP) and by integrating prior knowledge into the data-based modelling. The whole process of setting up an FDI system has been covered in the form of a systematic, step-by-step methodology. Several improvements for data-based modelling have been suggested: Guidelines for selecting the most suitable modelling methods are given and a novel input variable selection method is presented. A new measure for estimating the integrity of data sets is introduced. The measure can be used to define theoretical limits for modelling errors and to select an optimal input variable set for modelling. Furthermore, the process history data-based FDI systems are designed to provide information, most importantly the fault

indication, the reliability of the fault, and the estimated magnitude of the fault, in order to enable the use of fully automated fault-tolerant control strategies.

Author's contribution. This work has been carried out as a part of the Networked Control Systems Tolerant to Faults (NeCST, IST-004303) Project funded by the European Union. As a member of the project, the author has co-operated with other researchers from the Helsinki University of Technology (TKK), University Henri Poincare (France), University of Duisburg-Essen (Germany) and Neste Jacobs Oy. The methodology has been created by the author. The preliminary analysis of the research problem at the Naantali oil refinery was performed by a small team of researchers from TKK, including the author. In addition, the researchers of Neste Jacobs also contributed in defining the requirements for the monitoring system. The final testing of the implemented monitoring system was performed by the author and the researchers of Neste Jacobs. Other phases in the implementation of the methodology have been conducted by the author. The FTC strategies have been developed by the personnel of Neste Jacobs.

2 State of the art of data based process monitoring

A state-of-the-art review of fault detection methods is presented and discussed in this chapter. First, a classification of the FDI methods is presented and a general review is given for model-based and qualitative FDI methods. The quantitative data-based modelling methods are reviewed in more detail in section 2.2, and the use of prior knowledge in data-based modelling in section 2.3.

2.1 Classification and overview of fault detection methods

A number of different classifications of the FDI methods have been proposed: Chowdhury and Aravena (1998) classify them as model-based or model-free methods. Chiang *et al.* (2001) use categories of data-based, analytical and knowledge based methods. Venkatasubramanian *et al.* (2003a) first divide the FDI methods into either model-based or process history based. Both of these groups are further divided into quantitative and qualitative methods, as shown in Figure 2.1. This review is organised on the basis of the categories proposed by Venkatasubramanian *et al.* (2003a)

Traditionally, the FDI has been based on quantitative model-based methods; observers, filters, and parity relations. These methods are based on detecting deviations in residuals that are generated, using the models, from the measurement data. When the process parameters are unknown, parameter estimation methods are used, and when the parameters are known, observers can be constructed. Parameter estimation methods are suitable for detecting multiplicative faults, while observers are mainly used to detect additive faults (Stephanopoulos and Han, 1996).

The robustness of observers against plant-model mismatch has been improved by

introducing unknown input observers (Chen *et al.*, 1996; Patton and Chen, 1997). An optimal fault detection filter that maximises the isolation of faults for unknown input observers has been introduced by Chen *et al.* (2003). The applicability of observers for nonlinear cases has been improved by the introduction of sliding mode observers (Slotine *et al.*, 1986; Walcott *et al.*, 1987; Edwards and Spurgeon, 1994). Edwards *et al.* (2000) and Tan and Edwards (2002) further developed the method so that faults could be reconstructed, thus providing more information about the fault. Recently, Tan *et al.* (2008) suggested using a supplementary observer in cascade with a normal observer to reconstruct the fault signals and to estimate the system states. It has been shown that the dual observer systems are applicable to a wider range of processes than single observers because the conditions of existence for dual observer system are milder. Bhagwat *et al.* (2003) have used simultaneously multiple observers and filters to detect process faults during transitions. Their method breaks transitions into phases that are modelled separately using closed-loop observers for observable, and open-loop observers for unobservable variables.

The work on utilising (extended) Kalman filters (KF, EKF) for FDD was pioneered by Mehra and Peschon (1971). Chetouani (2004) utilised EKF for detecting faults in a nonlinear, ideal, continuously stirred tank reactor (CSTR) process. Recently, the area of applicability of the filter-based FDD methods has been expanded. Kano *et al.* (2008) proposed a two-stage subspace identification method for designing softsensors that can estimate unmeasured disturbances without making the normal KF assumption of Gaussian innovations. The usability of the method has been demonstrated with an application to industrial ethylene fractionator. Li *et al.* (2008) presented an application of KF in non-uniformly sampled multirate systems.

The parity space methods are based on analytical redundancy. Balance equations, called parity relations, are generated to relate input and output data and a model is used to check whether process data satisfy the equations or not. Using parity space methods for FDD has been researched e.g. by Gertler (1997). The parity relations

can be designed to use values determined by observers, and thus these two model-based techniques are very closely related, as shown in the work on the decoupling of faults in linear periodic systems by Zhang and Ding (2007).

Constructing exact quantitative mathematical models for modern processes is challenging due to their complexity and thus the qualitative model-based and data-based approaches have gained interest during the previous decades, signed digraphs (Iri *et al.*, 1979; Vedam and Venkatasubramanian, 1997; Cheng *et al.*, 2008), fault trees (Yang *et al.*, 1995; Kavcic and Juricic, 2000; Papadopoulos, 2003; Kim and Zuo, 2007) and qualitative trend analysis (QTA) (Maurya *et al.*, 2005 and 2007; Dash and Venkatasubramanian, 2000; Janusz and Venkatasubramanian, 1991) receiving most of the attention.

This thesis focuses on the statistical quantitative process history based and neural network methods. They are reviewed in more detail in the following section.

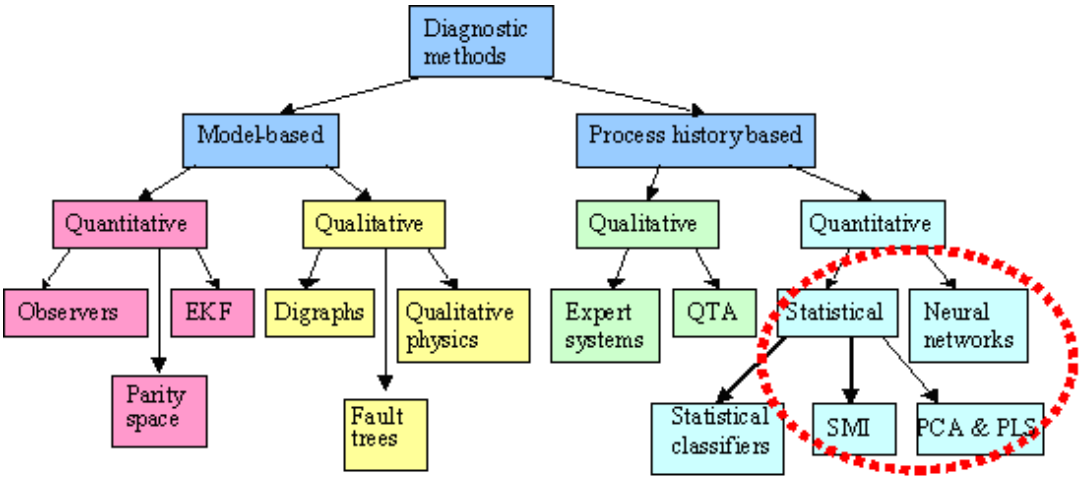

Figure 2.1: Classification of diagnostic algorithms (Venkatasubramanian *et al.*, 2003a)

2.2 Quantitative data based process modelling methods

The most commonly used statistical multivariate method is principal component analysis (PCA). The idea of reducing the dimension of a data set by using principal components was introduced by Pearson (1901). The significant initial developments of PCA were performed by Hotelling (1933) and Rao (1964). Further development on PCA started in the beginning of the 1990's with the publication of a comprehensive handbook about the method by Jackson (1991). Kosanovich *et al.* (1994) and Nomikos and MacGregor (1994) applied PCA to an industrial batch process by unfolding the three-dimensional batch process data into two dimensions, and then performing normal PCA using the 2-D data matrix. This technique is called Multi-way PCA (MWPCA). The basic PCA models are linear and describe static relationships between the variables. To extend the applicability of the methods to dynamic situations, Ku *et al.* (1995) proposed augmenting the training data set using time-lagged variables. This variant of the PCA that takes the autocorrelation of the variables into account is known as dynamic PCA (DPCA). The suitability of the DPCA for fault detection within the Tennessee Eastman process has been demonstrated by Russell *et al.* (2000). DPCA has also been used for monitoring the crystallisation process (Pöllänen *et al.*, 2006).

To make the method suitable for nonlinear processes, several modifications have been introduced. Gnanadesikan (1977) introduced the concept of generalised PCA (GPCA) by expanding the measurement data matrix with variables whose values are determined with nonlinear functions of the measured values, and then performing normal PCA on the expanded data. In nonlinear PCA (NLPCA) the lines, i.e. principal components describing the directions of the largest variations of the data, are replaced by curves. The first NLPCA method was suggested by Kramer in 1991. The method utilised a 5-layer autoassociative neural network with a 'bottleneck' layer to determine the curves. Tan and Mavrovouniotis (1995) proposed an Input Training NN (ITNN) based NLPCA method that Jia *et al.* (1998) used for

detecting faults in a complex nonlinear industrial reactor. An industrial case study of ITNN based NLPCA combined with wavelets has been presented in Shao *et al.* (1999). Dong and McAvoy (1996) used principal curves, together with neural networks, to determine the nonlinear principal components. Schölkopf *et al.* (1998) introduced the kernel PCA, in which the principal components are not determined for the original variables, but for calculated variables having nonlinear relations with original inputs. The problem with this type of NLPCA is that input data cannot be reconstructed from the principal components. A nonlinear dynamic PCA based on DPCA and an ANN classifying the process conditions in the principal component sub-space has been introduced in Lin *et al.* (2000). A kernel PCA based FDI system with an application to a nonlinear CSTR process was introduced by Choi *et al.* (2005). Recently, Hsieh (2007) researched the overfitting problem of the ANN based NLPCA methods, and presented a method for creating NLPCA models with noisy data. Hsieh succeeded in constraining the complexity of the NN part of the NLPCA and thus avoiding the overfitting problem, but the method is applicable only to data sets with one dominating signal.

Bakshi (1998) combined PCA with wavelet analysis (multiscale PCA, MSPCA) for more efficient use of information that is contained in different frequencies of the measurement signals. For a successful implementation of MSPCA for process monitoring and fault diagnosis see e.g. Misra *et al.* (2002). In cases where the process data are not normally distributed, the estimate for the data correlation or covariance matrix used in PCA will be inaccurate, resulting in insensitive PCA models. Wang and Romagnoli (2005) addressed this problem in the MSPCA framework by using robust M-estimators in each frequency band to replace multi-variate outliers with their robust estimates in order to rectify the measurement data. The resulting robust MSPCA (RMSPCA) achieved better results than MSPCA in a case study of a pilot scale process, the drawback of the method being that it requires heavy calculations during online use. The idea of monitoring different parts of a process separately was introduced by Wold *et al.* in 1987 (multi-block PCA, MBPCA).

Two different ways of combining information provided by the separate models of MBPCA have been proposed, the hierarchical PCA (Westerhuis *et al.*, 1998) and the consensus PCA (Qin *et al.*, 2001). Recently, Choi *et al.* (2008) studied the use of multiple scales with KPCA for FDI with a simulated CSTR process.

Li *et al.* (2000) introduced two recursive algorithms for calculating PCA (RPCA). RPCA can be used with processes that need an adaptive monitoring system due to gradually changing process conditions. For similar cases, Kano *et al.* (2001) proposed applying PCA to a moving time window, and changes in the process cause a detectable change in the principal components (moving PCA, MPCA). Recently, AlGhazzawi and Lennox (2008) introduced the idea of combining the RPCA with MBPCA to achieve recursive multi-block PCA (RMBPCA). The RMBPCA has proved to give better results than regular PCA or MBPCA with a condensate fractionation process, having multiple operating points and two logical sections.

Another solution for monitoring processes with changing operation conditions is to use multiple PCA models. The super PCA method utilising hierarchical clustering was introduced by Hwang and Han (1999). Multiple models have also been used to enhance the fault isolation properties of the PCA. Huang *et al.* (2000) introduced a method for detecting sensor and analyser faults by making several smaller GPCA and NLPCA models for the process. The structured partial models are designed so that a single model is sensitive only to a specific subset of faults, and thus the faults can be isolated using parity relations.

Another popular multivariate regression method is the partial least squares or projection to latent structures (PLS) introduced by H. Wold in the late 1970's. The suitability of the method for chemometry was first demonstrated by S. Wold *et al.* (1983) and, ever since, its applicability for different types of processes has been improved with a range of extensions and modifications. For time varying processes an adaptive version of PLS is needed, and a recursive algorithm for PLS (RPLS)

was introduced in Helland *et al.* (1992) and has since been enhanced by Dayal and McGregor (1997), Qin (1993, 1998) and Wang *et al.* (2003). For very large data sets, Lindgren *et al.* (1993) proposed transforming the large data matrix into a smaller kernel matrix and then analysing the matrix with PLS (KPLS). Walczak and Massart (1996) proposed a modification to the kernel PLS by using nonlinear radial basis functions for creating a nonlinear kernel matrix. Nomikos and MacGregor (1995) presented an idea of multi-way PLS (MPLS) to apply PLS to batch processes. Kourti *et al.* (1995) extended MPLS so that information about the initial conditions of the batch can also be used in the model. More recently, Marjanovic *et al.* (2006) introduced a variant of the MPLS that can be used to estimate the end-points of variable length batches. The MPLS methods unfold the 3-dimensional batch process data into two dimensions and then perform normal PLS. A true n-dimensional PLS algorithm (N-PLS) for analyzing multidimensional data was introduced by Bro (1996). The PLS has been used for solving the problem of product transfer between plants, which considers finding operating conditions in one production plant to have the same product qualities as the same product produced in another plant. The first study in this area was reported in Jaeckle and MacGregor (2000). More recently, Muñoz *et al.* (2005) formalized the so-called Joint-Y PLS (JYPLS) technique and applied it to a scale-up case for a batch pulp digester.

Over the years numerous extensions have been proposed to make PLS applicable for nonlinear processes. Wold *et al.* (1989) introduced quadratic PLS (QPLS) in which nonlinear (quadratic) functions are used to describe the inner relationships between the input score vectors and the latent variables. In 1992, Wold proposed using splines to achieve NLPLS. Nonlinear versions of PLS (NLPLS) based on artificial neural networks (ANNs) have been proposed by Qin and McAvoy (1992) and Malthouse *et al.* (1996). Bang *et al.* (2003) have taken yet another approach, and integrated fuzzy reasoning to PLS (FPLS) to handle nonlinear cases. By exploiting the Takagi-Sugeno-Kang (TSK) fuzzy model, human expertise can be integrated into the FPLS model.

In addition to the modifications to the PCA and PLS, new statistical monitoring methods have been proposed. The problem of blind source separation gave rise to independent component analysis (ICA), which is used to isolate independent signals from a mixture of signals. ICA has successfully been applied to a process monitoring problem (Lee *et al.* 2004) and for analyzing stock market trends (Back and Weigend, 1997). ICA can also be used for feature extraction and data compression in order to find a compact presentation of image, audio and other kinds of data. (Hyvärinen and Oja, 2000). A dynamic version of ICA based on lagged variables has been presented in Lee *et al.* (2004). Kano *et al.* (2002) proposed a process monitoring method called DISSIM. The method is based on monitoring similarity between groups of data and it can detect small continuous deviations from normal operating conditions.

A rather new idea in linear system identification is subspace model identification (SMI). The SMI is used to define the A, B, C, and D matrices and the covariance matrices of a state-space model directly from the measured process data. The most common SMI algorithms are canonical variate analysis (CVA; Larimore, 1990), numerical algorithms for subspace state space system identification (N4SID; van Overschee and de Moor, 1994) and multivariable output error state space (MOESP; Verhaegen, 1994). The similarities between the different algorithms are studied in van Overschee and de Moor (1995)), in which a unifying SMI theorem combining the three algorithms is also proposed. For a detailed description of the SMI methods see e.g. van Overschee and de Moor (1996). Chou and Verhaegen (1997) proposed using instrumental variables to make SMI more insensitive against noise in input and output data. Wang and Qin (2002) proposed using PCA with instrumental variables for SMI. A similar errors-in-variables (EIV) approach has been presented by Treasure *et al.* (2004). Their method allows the use of T^2 and squared prediction error (SPE) statistics in fault detection. Li and Qin (2001) demonstrated the close relationship between DPCA and SMI and proposed an EIV SMI-based method to derive consistent dynamic PCA including optimal determination of the model order. The traditional SMI methods (CVA, MOESP and N4SID) are biased under

closed-loop conditions. An unbiased SMI method for closed-loop systems has been presented in Chiuso and Picci (2005).

Another category of data-based modelling methods are the artificial neural networks (ANN). The idea of ANNs mimicking the functions of the human brain was first suggested around 1960, when Rosenblatt introduced the concept of perceptron (Rosenblatt, 1958). ANNs can be categorized by their training method or by their architecture, number of layers and the type of activation functions. The most well known type of ANN is the multilayer perceptron network (MLP), which is a multilayer feedforward network typically trained with the supervised back-propagation learning algorithm. Each neuron in MLP is connected to all neurons in the subsequent layer, while there is no connection between neurons in the same layer. According to a study carried out in 1995, 81,2 % of all ANN based applications utilized the MLP structure (Haykin, 1995). MLP has been utilised as such in detecting faults in a packet distillation column (Sharma *et al.*, 2004) and it has also been combined with fuzzy logic systems (Ruiz *et al.*, 2001). Yamamoto and Venkatasubramanian (1990) introduced an FDD system for a process consisting of a CSTR and a distillation column based on separate MLP models for continuous and discontinuous measurements.

The uncertainty of the ANN models have been studied by Mrugalski *et al.* (2007). They proposed a method for minimising the structural and parametric uncertainty of an MLP by the Outer Bounding Ellipsoid technique (Milanese *et al.*, 1996). Similarly, the uncertainty of a Group Method of Data Handling (GMDH) NN can be estimated with the Bounded Error Approach (Milanese *et al.*, 1996) to achieve robust FDI. In order to obtain robustness against the parameter uncertainty of GMDH NNs, Puig *et al.* (2007) proposed a method for determining adaptive fault detection thresholds.

The easier interpretability of the NNs has been studied by Tan *et al.* (2007). They

proposed a hybrid NN structure for a fault detection system providing automatic generation of rules to classify process conditions. The FDI system based on fuzzy ARTMAP (FAM) and rectangular basis function network (RecBFN) was tested with a circulating water system of a power generation station.

To take into account the process dynamics, the neurons or the structure of the networks are modified so as to utilise also the past values of the input variables. A general structure for locally recurrent, globally feedforward networks (LRGF) has been introduced by Tsoi and Back (1994). Patan and Parisini (2005) proposed using Akaike Information Criterion and Final Prediction Error for selecting the proper detailed LRGF structure. A fault detection application of LRGF to a catalytic cracking process has been reported by Patan and Korbicz (2007). Instead of local feedback structures, Zhou *et al.* (2003) proposed using polynomial features with pure feedforward NNs to utilise the the past inputs. Yang *et al.* (2000) proposed using an ANN and the dynamic features of process data, extracted with wavelets and qualitative interpretation, to detect faults during transient process conditions in a fluid catalytic cracking reactor.

Another popular type of ANN is the Self-Organizing Map (SOM), which is trained with unsupervised competitive methods. The original SOM algorithm developed by Kohonen (Kohonen, 1990) has been optimized and modified for different applications. Kangas *et al.* (1990) proposed algorithms for dynamic weighting and more efficient definition of the neighbourhood parameter in SOM training, Shah-Hosseini and Safabakhsh (2003) developed a time adaptive variant of SOM, TASOM. Hirose and Nagashima (2003) introduced a variant of SOM with predictive properties, P-SOM. The drawback of the original SOM is that the size of the map needs to be determined before the training of the map. A solution to this problem has been presented by Fritzke (1995). The Growing Grid algorithm introduced by Fritzke utilises an iterative training process in which the size of the map is increased when needed. The repeated training of the map is time consuming and a more efficient algorithm

for automatic determination of the size of SOMs, the Growing Hierarchical SOM (GHSOM), was proposed by Rauber *et al.* (2002). GHSOM produces a hierarchical structure of SOMs instead of only one map. A technique for merging the information presented by the separate maps has been recently published (Soriano-Asensi *et al.*, 2008). To achieve robustness against outliers, two variants, (Marginal Median SOM, MMSOM and Vector Median SOM, VMSOM) using median based weight update algorithms have been published by Pitas *et al.* (1996). MMSOM has been successfully applied to a document classification case by Georkakis *et al.* (2004) and the benefits of the robustness have recently been analysed in Moschou *et al.* (2007). Kohonen suggested using supervised learning vector quantization (LVQ) in training the map for more efficient separation of different types of sample, if the SOM is used only for solving a classification problem (Kohonen, 1990).

Radial Basis Function Networks (RBFN) are ANNs that utilise exponential Gaussian activation functions to perform input output mapping and classification tasks. The original ideas for RBFN were introduced by Moody and Darken (1989). RBFNs have been used for classifying process faults (Leonard and Kramer, 1991) and for detecting sensor faults in chemical processes (Yu *et al.*, 1999). RBFNs have also been used in combination with other methods to provide a classification of the faults (Kämpjärvi *et al.*, 2007; Zhou *et al.*, 2003).

2.3 Prior knowledge in data based modelling

In addition to the the individual modelling methods, several modelling related tasks common to all data based methods have also been researched. These tasks include input data preparation, input variable set selection, reduction of the dimension of the input variable set, and the use of combination models. The main ideas and the latest developments of these supporting tasks are presented in this section, and different ways to utilise prior knowledge in preparing input data sets and in model

construction are also discussed.

The task of preparing the data sets that are used to train the data-based models consists of three phases: (1) selection of input variables, (2) creation of additional variables, and (3) removal of the effects of known process phenomena from the data. The phases are discussed in the following.

In some applications, the process conditions change significantly during normal operation. The changes may be caused by e.g. variation in feed composition or changing specifications of the product. Although these alterations are sometimes made intentionally and may not be harmful as such, the variation they produce may conceal the effects of the simultaneously occurring faults. To solve this problem, Takane and Shibayama (1991) proposed a method to decompose the observed data matrix so that the effects of the known phenomena could be removed from the data before they are analysed with the monitoring methods. The data decomposition method has successfully been tested e.g. with data collected in a styrene monomer production plant (Yoon and MacGregor, 2001).

Processes that have clearly separable operating points or strategies may be easier to model using multiple models. For each operating point, which is usually determined by the current feed stock or the desired quality of the product, a separate model can be constructed. The benefit of this approach is that it allows monitoring nonlinear processes with multiple linear models (Zhang *et al.* 2003). Using a combination of separate models to describe the linear dynamic and nonlinear static characteristics of nonlinear processes has also been suggested. If the linear block precedes the nonlinear block, then the structure is called a Wiener model, while the reverse situation is referred to as a Hammerstein model. Several methods have been suggested for identifying Hammerstein models (e.g. Narendra and Gallman, 1966; Jia *et al.*, 2005; Pottmann *et al.*, 1993 and Laksminarayanan *et al.*, 1995) and Wiener models (Billings and Fakhouri, 1982). The third published approach suggests combining

data-based and first principles models. Existing process knowledge is used to create a simple mechanistic model and it is used together with a separate black-box model trained with the part of the training data that is not explained by the FP model. This method has been applied successfully to a fed-batch penicillin fermentation process by Thompson and Kramer (1994).

The role of prior knowledge in using statistical modelling methods is mostly restricted to the input variable selection (IVS) and the creation of calculated variables. With neural networks, prior knowledge can be utilised more extensively. It can be used to alleviate problems associated with noisy and/or sparse input data (Thompson and Kramer, 1994), and the outputs of an ANN model can be made to agree with the known facts about the modelled process. Joerding and Meador (1991) classified the constraining methods into Architecture Constraint (AC) methods and Weight Constraint (WC) methods. In AC, the architecture of the ANN is chosen so that it will always, regardless of the weights, fulfill the conditions set by prior knowledge. In WC, the weights of the ANN are constrained during the training so that the output does not violate the conditions. Later on, Chen and Chen (2001) realized the need for a third category, Data Constraint (DC) methods. The DC methods manipulate the data, which are used to train the ANNs in such a way that the models' outputs agree with the known values. These methods differ from the design and training hybrid model approaches introduced by Thomson and Kramer (1994) in that the modelling is accomplished with a single NN model. In addition to the earlier categorisation, Chen and Chen (2001) proposed dividing the methods into internal methods, that constrain the architecture or weights directly and external methods, which encode the knowledge into the ANNs indirectly.

3 Methodology for setting up a process monitoring system

In this section, a systematic methodology for developing monitoring systems for complex industrial processes is presented. The methodology consists of five major parts. The first part consists of drawing up the specifications for the system and defining the user requirements as presented in section 3.1. Guidelines for selecting the most suitable data-based modelling method are presented in section 3.2. The different phases of data analysis and preparation of the data sets for the modelling are illustrated in section 3.3. Constructing the models is described in section 3.4, and formulation of the change detection and fault identification blocks is described in section 3.5. Finally, the steps required in analysing the results are presented in section 3.6. The main phases of the methodology described in the following sections are presented in Figure 3.1.

3.1 Specifications of a monitoring system and definitions of the user requirements

The first phase of the preliminary analysis is to define the specifications of the target application and the user requirements for the monitoring system. The background information is acquired by interviewing experienced personnel at the production plant. In order to obtain as wide a perspective as possible to the problem, persons working in the different operational sectors are interviewed: operators, engineers, and management personnel. The interviewing method could, for instance, be the Delphi method (Linstone and Turof, 1975), which consists of number of interview rounds. After each round the results are analysed and presented to the interviewees, and the interviewing process is continued until the results have converged. The

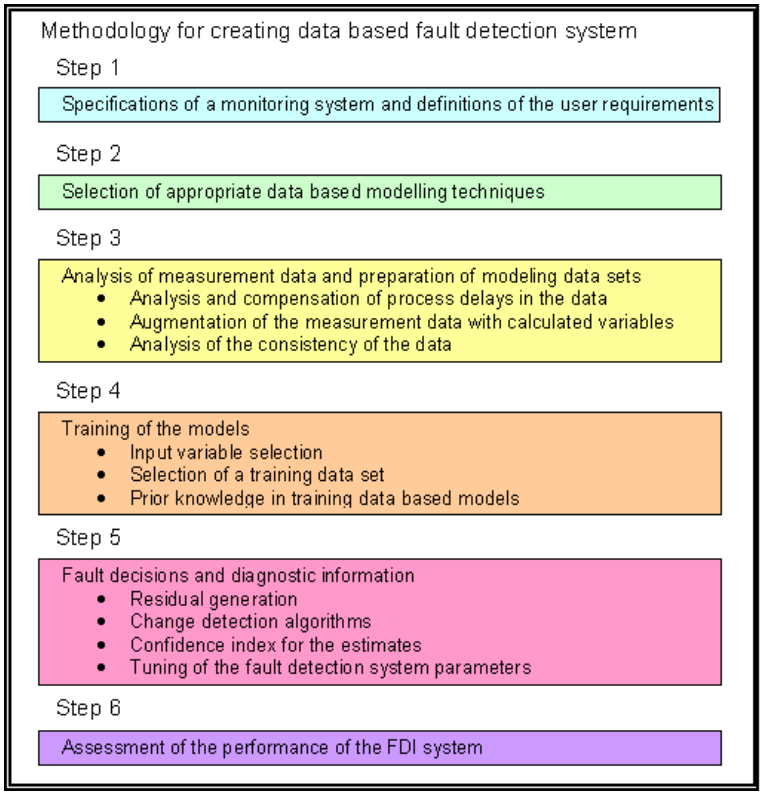

Figure 3.1: Major steps of the proposed methodology for creating an FDI system

following information is acquired through the interviews: (1) detailed description of the required functionality of the FDI system, (2) description of the process conditions under which the system will be used, (3) end-users' attitudes toward false alarms and missed detections, (4) specifications of the environment in which the system is implemented, and (5) specifications of the user-interface. The first two items are required in setting up the system, the third is related to the tuning of the system, and last two concern the external restrictions.

3.2 Selection of appropriate data-based modelling techniques

Selection of the most suitable modelling method for a specific monitoring problem depends on the process and its dynamics. A wide selection of modelling methods is available for linear processes, but only a few nonlinear methods can be used to model nonlinear processes. Processes indicating strong nonlinear behaviour are modelled with ANNs (e.g. MLP, RBFN, SOM) or nonlinear versions of the linear methods like NLPCA, NLPLS, while it is recommended that linear and weakly nonlinear processes are modelled with statistical modelling methods. A second important characteristic of processes which affects selection of the modelling method is the number of operating regions. Data representing multiple operating regions do not follow the normal distribution and are therefore unsuitable for most of the statistical modelling methods. Special cases in modelling are batch processes, because they require accurate modelling of transient process phases. It is recommended that batch processes are modelled with methods specially designed for handling process transients, such as Multi-way PCA and Multi-way PLS.

The intended use of the model needs to be taken into account when deciding which monitoring method to select. Three common aims of process monitoring models are: (1) to detect deviations between the current process state and a desired reference state, (2) to estimate the values of certain process variables, and (3) to classify the current process state. The first task consists of monitoring a group of variables and detecting deviations between a new data set and the original that was used for the training. This kind of similarity analysis provides information about whether the process is at the same operating point as when the training data were collected. The information is important in the case of processes that have only one or a restricted number of desired operating points available for detecting deviations from the previously optimised conditions. The second monitoring task consists of using values of measured variables to estimate the value of another variable. The estimates produced by regression models are compared with the corresponding measured or

analysed values in order to detect sensor or analyser faults. The third task is to classify the process states and faults. The classification is similar to the first problem but, in this case, the current state is not compared against a single reference state, but instead against multiple states that were present in the training data of the models. These three cases are the most widely used ways to utilise process monitoring systems. The tasks are clearly so different from each other that they require specific model types for each case. Comparing the current process state with a reference state can be performed by PCA, ICA and DISSIM, while the regression models utilised in the second type of problem can be constructed with PLS, SMI, MLP and RBFN. Commonly used classifying methods include SOM and ART.

In addition to the process characteristics and the intended use of the models, a number of other factors also need to be considered. For online monitoring systems, the limit for the complexity of the mathematical operations that are performed in each monitoring cycle is set by the specifications of the implementation environment. The quality of the training data also limits the use of certain methods. Statistical methods like PCA, PLS and ICA reduce the dimension of the data by identifying general factors that explain the variation within the data. With certain limitations, these factors are also valid for describing situations that have not been present in the training data. In contrast, ANNs that are universal approximators do not find general trends in the data, but rather approximate the values of the functions with a very high degree of accuracy. This ability is the origin of the two drawbacks of ANNs; overfitting, i.e. modelling the noise component in the training data, and the poor performance in situations that were not present in the training data. RBFNs are especially poor at extrapolating data due to the exponential nature of the Gaussian functions and, consequently, should not be used with sparse data sets (Leger *et al.*, 1996). In addition, if the classification of faults is to be an important feature of the monitoring system, then the data must also contain measurement information about the faults that are to be be classified. One way to present the suitability of the modelling methods described in the state-of-the-art section for different process

types is described in Table 3.1, in which the methods that share similar characteristics, e.g. all the NLPLS methods, are grouped together. The modelling accuracy is not given in the table as a model characteristic because it depends strongly on the case in question, and therefore ranking the methods on the basis of the results achieved in specific benchmark problems would be misleading. When all the user requirements, the requirements set by the implementation environment and the constrictions set by the data are taken into account, there might be several suitable candidates for the modelling method. In this case, the simplest method should be chosen.

The first monitoring task consists of analysis of the properties of one variable set during process operation. Any deviations from the nominal values are interpreted as faults. The PCA and its variants and the DISSIM are suitable methods for this kind of task. The DISSIM method is especially useful for detecting slowly developing faults (Kano *et al.*, 2002). If the results of these methods are not acceptable, then ICA can be used provided that the signals to be separated are statistically independent. PLS and its variants can be used for detecting changes in the relationship between two data sets, e.g. input and output. If a state-space model of the process is needed, then SMI can be utilised to determine the related matrices. Regression models constructed with PLS or SMI are also suitable for detecting sensor faults. If the accuracy of the linear models for nonlinear processes are not satisfactory, then ANNs should be used. ANNs can be used both for prediction and classification problems. The MLP is the most common ANN type, but RBFN has recently gained interest due to the number of advantages it has over MLP. SOM has proved to be a powerful tool for classification problems (e.g. Laine *et al.*, 2000).

Table 3.1: Model characteristics of some data based fault detection methods

	Data similarity										Regression							Classification	
Model characteristics	PCA	Multi-way PCA	Moving PCA	Multi scale PCA	Robust multi scale PCA	Recursive PCA	Dynamic PCA	Nonlinear PCA	ICA	DISSIM	PLS	Nonlinear PLS	Multi-way PLS	Recursive PLS	SMI	MLP	RBFN	SOM	ART
Suitability for linear processes	X	X	X	X	X	X	X	x	X	X	X	X	X	X	X	x	x	x	x
Suitability for nonlinear processes								x				x				X	X	X	X
Suitability for steady state processes	X	X	X	X	X	X	X	x	X	X	X	X	X	X	X	x	x	x	x
Suitability for non-steady state processes			X			X	X	x						X	X			X	X
Suitability for batch processes		X											X					X	X
Ability to identify faults																		X	X
Ability to detect new fault types	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	x	x
Ability to detect simultaneous faults																		x	x
Heavy calculations during online use			x	x	x	x		x						x					
Easy interpretation of results	x	x	x			x	x	x	x	x	X	X	X	X	X	X	X	X	X
Good updatability			x			x								x					
Strict requirements for training data								x								X	X	X	X
Robustness against noise in training data	x	x	x	x	X		x		x	x	x		x	x	x			x	

The small 'x' denotes that the property describes a method, capital 'X' denotes that the property describes a method very well.

Processes that exhibit significant nonlinear behaviour cannot be accurately modelled with linear models and nonlinear modelling methods should be used instead. However, linear models have the following strengths; they are easy to understand and their behaviour is predictable under new operating situations. In contrast, nonlinear ANNs are black-box models whose structure and parameter values have no direct connection to the process phenomena and their effects on the outputs are therefore difficult to understand. Moreover, the nonlinear nature of the models indicates that they can provide unexpected results when new operating conditions are encountered. For these reasons, the use of pure ANN models is not encouraged in real industrial implementations. It is recommended to use a combination of the two in order to obtain the benefits of both linear and nonlinear models. The type of models used in the hybrid models could be e.g. PLS, SMI and MLP. Several methods for combining the models exist: (1) completely independent models are used and the most suitable model for each situation is selected online according to a predetermined switching logic, (2) a linear model is constructed and the unexplained part of the data is used to train a MLP model, and (3) a dimension reduction method, e.g. PLS, is used to transform the input data into lower dimension space and the modified data is then used to train a MLP model. The second approach is recommended in the methodology because, in the first approach, the creation of the switching logic provides additional challenges for the modelling, and models constructed according to the third option may give unexpected outputs in new operating conditions, as is the case with pure ANNs.

In order to train the hybrid models, the linear part of the model is first created. Then an MLP model is trained with the same input data set as the linear model, while the output is the residual between the modelled variable and the corresponding values estimated by the linear model. Before training the MLP, the residual signal is bounded between $-a$ and a in order to train the ANN to compensate for deterministic modelling errors of the linear model and to minimise the effects of isolated instances with large absolute values on the training process. The value of

parameter a is determined on the basis of the case in question and on the accuracy of the linear model.

3.3 Analysis of measurement data and preparation of modelling data sets

3.3.1 Analysis and compensation of the process delays in the data

For non-stationary processes with significant delays or slow dynamics, there is no causal relationship between measurements that are made simultaneously in different parts of a process. For this reason, the process delays are commonly determined and compensated for by time-shifting the data. By using time-shifted data, the process delays can then also be taken into account with static modelling methods.

The outlier values in the data are removed prior to delay estimation. This can be done e.g. with a median filter (3.1)

$$u_k^f = median\{u_{k-(w-1)/2}, ..., u_{k+(w-1)/2}\} \tag{3.1}$$

where u_f is the filtered value, and w is an odd number representing the length of the filtering window. The value for w is determined by analysing the data. It is recommended that this is set to a value greater than twice the length of the longest sequence of outliers.

The delays can be determined by performing cross-correlation analysis between each input-output variable pair. The analysis can be performed for input variables that correlate with the outputs. The cross-correlation values of those variables that do not affect the outputs of are often low over the analysed interval, thus indicating the absence of a deterministic relationship between the variables. These variables are discarded from the data set and are not used in the modelling.

The estimates for the delays acquired with the cross-correlation analysis are only approximate, and the delays may change according to the operating conditions. For these reasons it is recommended to utilise several measured values in determining the values of the time-shifted values of input variables. Using an average of multiple measured values further suppresses the noise in the measurements. The delay compensated i.e. time-shifted values can be determined by (3.2) as the average of n samples measured before and after the estimated delay.

$$u_k^{dc} = \frac{\sum_{m=1}^{n} \left(u_{k-d+floor(n/2)+1-m} \right)}{n} \tag{3.2}$$

where $-b$ is the delay, u_k^{dc} is the delay compensated value of u at instant k, $floor$ is a function used to round down the ratio $n/2$, and n is the number of values used in the filtering. Taking the moving average of the original values attenuates high frequencies, but does not introduce a delay in the compensated value as long as the estimated delay is longer than $floor(n/2)$.

3.3.2 Augmenting the process data set with calculated variables

Calculated variables are quantities whose values are determined with mathematical functions using a subset of the measured variables as inputs. They are mainly used for three purposes: (1) to reduce the number of variables, (2) to form variables describing process phenomena, and (3) to linearise the data. Data reduction is the most common reason for utilising calculated variables, and several methods are readily available for the task. Some traditionally used methods include clustering methods (e.g. SOM, K-means), linear transformations of the input variables (PCA, linear discrete analysis; LDA), spectral transformations (Fourier, Hadamard), wavelet transformations or convolutions of kernels. The second motivation for using calculated variables is to create variables that capture the complex characteristics of processes (Laine *et al.*, 2000 and Yang *et al.*, 2000). The third reason for creating calculated variables is to convert the non-linear relationships of the process

data into linear ones, thus making the data usable for linear modelling methods (Kourti, 2002). Ramaker *et al.* (2002) suggests using logarithmic transformations for creating such features.

The role of prior knowledge in data-based modelling is of great importance in the creation of process phenomena related variables. They can be derived with mathematical models like the fundamental balance equations (Guyon and Elisseeff, 2003). This type of calculated variable is also referred to as a soft-sensor. According to Nomikos and MacGregor (1994), using soft-sensors increases the information content of a PCA model such that the variation related to the key phenomena will become dominant in the principal components and thus improve the accuracy of the model. Moreover, Yoon and MacGregor (2001) state that the use of soft-sensor estimated values in the data matrix makes it easier to interpret the contribution plots because only one principal component is highlighted. Adding calculated variables to the data set without removing the original variables is also known as the augmented data matrix method.

Two interesting modelling techniques for utilising the process phenomena type of calculated variables are Functional Link ANNs (FLANNs) (Pao *et al.*, 1992) and Genetic Programming (GP) (Koza, 1992). FLANNs include a separate functional expansion module in which the input signals are used to create calculated variables that are passed to the ANN structure. The functions that are used in FLANNs depend on the case in question, and need to be determined before the training of the models. Patra and van den Bos (2000) studied the use of polynomial functions, such as Chebyschev, Legendre and power series. As the input set is expanded to include strongly nonlinear components, the structure of FLANNs can be simple. The ANN part of the models has usually been a single layer network and, compared to MLP, the FLANNs have produced almost as good results with a lower computational burden. However, one problem associated with FLANNs is that the number of functions needs to be high in order to improve the probability of finding suitable

calculated variables. This leads to increased dimension of the input variables.

Koza (1992) introduced genetic programming (GP) for creating calculated variables by forming tree structures consisting of operator and parameter nodes. The best structure for presenting a modelled data set is searched using genetic algorithms. GP has been successfully used in many applications (e.g. Greeff and Aldrich, 1998 and Zhang *et al.*, 2005), but the main drawback of the method is the heavy calculation burden if the number of variables and operators is high. However, with more complex systems, the models i.e. the calculated variables are harder to find by means of evolutionary programming methods.

The calculated variables representing process phenomena make it easier to model the process with data-based methods. To promote and systematise the use of calculated variables that describe process specific phenomena and to overcome the restrictions of FLANN and GP, the methodology includes a set of calculated variables. The set consists of process primitives that are calculated variables representing unmeasured quantities containing important information about the process conditions. Two major benefits of using calculated variables describing nonlinear process phenomena are (1) that linear models can be used with nonlinear processes, and (2) that there is improved accuracy in interpolating and extrapolating the data. The interpolation and extrapolation characteristics are important in situations where the data sets do not cover the whole operating area for which the models should be valid. The primitives presented here are divided into two categories: those related to process equipment and those related to physico-chemical phenomena. The equipment-based primitives are applicable to a wide range of industrial processes that comprise the same types of process equipment. Furthermore, the primitives related to fundamental physical laws, e.g. mass transfer, are applicable to many processes. Primitives describing the characteristics of chemical reactions are considered less useful as their applicability is limited to specific reactions and processes.

The first primitive type is the difference between two measured temperatures. These primitives indicate, for instance, the part of the reactor in which an exo- or endothermic reaction takes place or the effectiveness of a heat exchanger. The differences give operating point independent information about the process, which is useful for detecting faults in multiple operating point processes. A general equation for creating the primitive is given in (3.3).

$$u_{\Delta Temp} = u_1 - u_2 \tag{3.3}$$

where u_1 and u_2 are the temperature measurement values.

The second primitive type is the total quantity of a specific compound or property. It can e.g. describe the total amount of substances that react in the same way in a process. The primitive is determined with (3.4)

$$u_{tq} = \sum_{k=1}^{n} u_k \tag{3.4}$$

where u_{tq} is the total quantity of the substance, n is the number of different sources of the substance and u_k are the quantities of the separate sources of the substance. When the fraction of a substance in the reactive material is more of a descriptive quantity than the total quantity, a more suitable primitive is expressed as

$$u_{fr} = \sum_{k=1}^{n} \frac{u_{fr}^k F^k}{F^k} \tag{3.5}$$

where u_{fr} is the total fraction of substance u after the flows, F^k, including different fractions of u, u_{fr}^k, have been combined. The n is the number of flows.

The third primitive type is the ratio of two process flows. This primitive describes quantities that are often controlled, e.g. the mixing of inert and reactive substances to control the strength of the reactions. The primitive is insensitive to the total feed flow rate and can thus be used in FDI that relies on classification. On the other hand, it is generally not usable as an input variable for regression models. The primitive is described by (3.6).

$$F_{ratio} = \frac{F_1}{F_2} \tag{3.6}$$

where F_{ratio} is the ratio of flows F_1 and F_2.

Balance equations, especially the heat and mass balances, are a common type of primitive. These primitives indicate changes, i.e. possible faults, within specified subject areas, leaks in the case of mass balances, and malfunctions in heating devices or insufficient insulation in the case of heat balances. Using balance primitives requires prior knowledge about the interesting areas over which the balances are created. The balance equation primitives are based on the conservation principle given in (3.7) (Ogunnaike and Ray, 1994).

$$\text{Accumulation} = \text{Input} - \text{Output} + \text{Internal Production} \tag{3.7}$$

Primitives that are connected closely to specific process equipment are also created. For instance, an important primitive indicating operating conditions within distillation columns is the pressure compensated temperature. A linear approximation of the pressure compensated temperature is given by (3.8).

$$u_{pcT} = u_T - b(u_P - a) \tag{3.8}$$

where u_{pcT} is the pressure compensated temperature, u_T and u_P are measured temperature and pressure, and a and b are tuning parameters.

All primitives are created when a monitoring system is being developed, and those that are useful in modelling are identified using the input variable selection methods described in section 3.4.1. The set of calculated variables implemented in the chemical engineering library is not exhaustive, and other primitives related to specific types of process equipment or processes can be constructed on the basis of process-specific knowledge. The usefulness of primitives in fault detection have been demonstrated e.g. in Laine *et al.* (2000), Komulainen *et al.* (2004) and Kämpjärvi *et al.* (2007).

After augmenting the data set with calculated variables, all the information about

the process in non-numerical form is presented in a numerical form and added to the data set. The non-numerical information is related to quantities of processes that are not measured, but are otherwise known, e.g. the composition of the feed might vary depending on the raw material supplier. Non-numerical information is presented as crisp class membership values, 1 if the value of a quantity belongs to a certain class, 0 otherwise. The next steps of data augmentation are data regularisation and interpolation, which are normally used with ANNs. In data regularisation the measured values are rectified so that they more accurately describe the situations in which they were measured. One typical requirement for rectified data is that it is smooth. Creating additional data points by interpolation is done in order to avoid the over fitting problems that follow if sparse training data is used with ANNs.

The last step described here is to augment the data matrix with the differences between consecutive measurements. These rate-of-change variables enable static modelling methods to model some of the dynamic characteristics of processes.

3.3.3 Analysis of the consistency of the data

Next, the consistency of the data and the suitability of the data for modelling are determined. The inconsistencies within the data are mainly caused by measurement noise and changes in the process and in the instrumentation during the data collection period. A method for determining the consistency and thus the validity of the data set for modelling and the limits for acceptable modelling errors is presented in the following.

For a single input single output (SISO) case in which only one variable is used to explain the other variable, the consistency analysis is performed as follows. First, the amplitude of the noise of the explanatory variable is determined. The noise is assumed to be high frequency white noise and its amplitude is estimated from the

residual of a low-pass filter. Second, the two variables are plotted against each other. The minimum and maximum values of the explained variable and their difference are determined for all values of the explanatory variable, taking into account the uncertainty of the input variable. A graphical example for determining the maximum and minimum values corresponding to a single measured value, 50, with uncertainty of ± 10 is given in Figure 3.2. Three key figures for the maximum-minimum differences are determined: the medium difference, the difference that is larger than that in 95 % of the cases and the maximum difference. For an optimal model trained with the data set, the medium value divided by two is an expected value for the modelling error, and the maximum value describes the largest expected modelling error. As the largest errors are due to outliers, a more deterministic maximum value for the expected systematic modelling error is the 95 % value.

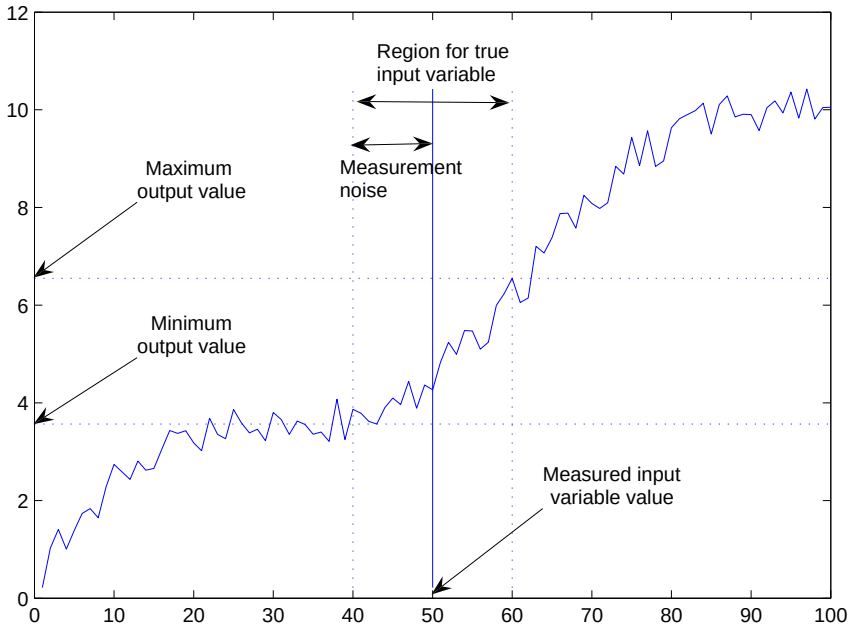

Figure 3.2: Analysing the quality of the modelling data, SISO case

In multivariable cases, in order to determine the data inconsistency a clustering tool is required to find data samples with similar values in the input variables. The

methodology recommends using SOM for this task. When the number of neurons in a SOM is high compared to the number of samples in a data set, the data samples classified in a single neuron have almost the same values for the input variables. The minimum and maximum output values related to these data samples are readily available and the analysis can be carried out. An important feature in the analysis is related to finding the proper number of neurons in the SOM. An equation (3.9) is given for determining the suitable number of neurons.

$$N_{neurons} = \frac{r_1}{n_1} \prod_{k=2}^{m} b \frac{r_k}{n_k} \tag{3.9}$$

where $N_{neurons}$ is the number of neurons, k is the number of variables in the data set, b is a tuning parameter, r is the range of a variable, and n is the amplitude of the noise in the same variable. If the input variables are independent of each other then the value for b is one. The stronger the correlation between the variables, the closer the value is to zero. The inconsistency measure of a data set depends on the selection of the explanatory variables, and thus the measure can be used in input variable selection.

3.4 Construction of the data-based models

The data-based FDI models are then constructed. This starts with selection of the input variables and determination of the training data set, and continues with construction of the models. During the selection of input variables, the goodness of different models needs to be evaluated. Generally used indicators for the goodness of models include accuracy, understandability, applicability over a wide range of operating conditions and reliability under new operation situations. In this study the goodness of the models is determined on the basis of the root mean squared error of the estimation (RMSE) index.

3.4.1 Selection of the input variables

In order to deal with the large processes that are nowadays frequently encountered, the modelling task should be divided into smaller, easier tasks whenever possible so as to avoid unnecessary complexity. MacGregor *et al.* (1994) state that statistical monitoring methods like PCA and PLS can be applied to large processes that consist of partly independent process sections, by creating a model for each section. Instead of dividing large monitoring problems into spatially or logically separated blocks, they can be decomposed into smaller sections by modelling the different frequency ranges of the process separately (Bakshi, 1998). The main advantage of dividing up the monitoring problem, in addition to the reduction in the number of input variables, is the easier interpretation of the monitoring results. The methodology recommends dividing large monitoring problems into smaller sections if a suitable division can be identified.

Even after the monitoring problem is divided into smaller parts, the number of available variables is often impractically high to be used with the data-based methods. The dimensions of the input variable sets corresponding to the different process parts are further reduced in connection with input variable selection (IVS). Langley (1994) defined the IVS problem as selecting the best subset of variables in a search space consisting of all possible combinations of variable subsets. Langley also formulates the four main issues of the search: (1) determination of the starting point of the search, (2) organisation of the search, i.e. rules on how to proceed from one variable subset to another, (3) evaluation of the subsets of variables, and (4) determination of the criterion used to stop the search.

Mathematical algorithms have been developed for the IVS task. John *et al.* (1994) divide the algorithms into two categories, wrapper methods and filter methods. The fundamental difference between the two categories is that the wrapper methods are used to select variables for a specific monitoring method and the filtering methods

try to select an optimal set that can be used with any method. Filter methods are based on performing a statistical test between subsets of the input variables and the desired outputs of the model. A commonly used statistic is correlation. Bonnlander and Weigend (1994) proposed using a more complex measure, mutual information, for analysing the strength of the relationship between two variables.

Wrapper methods consist of the following four steps: (1) selection of a process model, (2) determination of an input variable set, (3) training of a model with the selected variables, and (4) assessment of the performance of the model using a cost function. Steps 2 - 4 are repeated until the stopping criterion is met. Method development related to wrapper methods mainly focuses on choosing an input variable set. In forward selection (FS), the starting subset is empty and variables with the strongest correlations with the modelled output are included in the model, one variable at a time. Backward elimination (BE) starts by using all the available variables and then, at each iteration round, the variable that correlates the least with the desired output is removed. Both FS and BE have more refined step-wise versions in which variables can be discarded (FS) or re-selected (BE), or the variables can be selected or eliminated in groups. A completely different approach for search organisation is to select the candidate subsets randomly with a genetic algorithm (GA).

The performance of the IVS methods depends on the data that is used for evaluating the performance of the set. The prediction results obtained with the leave-one-out cross validation (LOOCV) method do not agree with the results obtained using an independent testing data set. Using LOOCV, sequential forward floating selection (SFFS, Pudil *et al.*, 1994) gives better results than sequential forward selection (SFS), as shown in Kudo and Sklansky (2000). When a completely new data set is used in testing, the simpler SFS is equally good to or even outperforms the SFFS (Reunanen, 2003). The same phenomenon has been reported by Kohavi and Sommerfield (1995), Kohavi and John, (1997), and Scheffer and Herbrich, (1997). The methodology recommends using an independent data set in evaluating the perfor-

mance of input variable sets.

In this methodology, an IVS method consisting of three main steps is proposed. First, prior knowledge is used to remove irrelevant and otherwise unsuitable variables. Second, mathematical filter and wrapper methods are utilised, and a GA is then used to find an input variable set. Third, the number of variables of the set is reduced with PCA or similar data compression methods. While all of the suggested IVS steps should always be carried out, the importance of the steps varies depending on the case in question. If the target process is poorly instrumented, i.e. only a relatively small number of variables are available for modelling, then the emphasis in the IVS should be placed on using the prior knowledge. As the number of variables increases, manually selecting the variables gradually becomes more challenging. Thus, for complex and well-instrumented processes, the IVS is mainly done with mathematical methods. The five main steps of the proposed IVS method are described in the following.

Step 1. Irrelevant variables are discarded from the training data set. The relevance of the variables is determined with prior knowledge. Typically, irrelevant variables are those that are known not to have a causal relationship with the process of interest. The averages of sensors that measure the same quantity, such as multiple temperature sensors at the same location, are determined. During the monitoring, the redundant measurements can first be used to check for faults in the measuring devices, and an average of the non-faulty measurements can then be calculated and used in monitoring.

Step 2. A preliminary selection of input variables is performed using the following method. First, the variables are ranked according to the correlation between them and the output variable. Input variables correlating weakly with the output variable are discarded. The IVS starts with an empty variable subset, and all the variables are included in the input variable set one at a time in the descending order of the

correlation. After each addition, the resulting output of the model is compared to the output of the previous model. If the performance measured with J_{MSE} (3.10) is improved, then the variable is included in the subset.

$$J_{MSE} = \frac{1}{n}\sum_{k=1}^{n}(\hat{y}(k) - y(k))^2 \tag{3.10}$$

After a variable has been included, the validities of all the previously included variables are tested in leave-one-out manner, and the corresponding J_{MSE} values are determined. If a lower J_{MSE} value is achieved by leaving out a variable, then that variable is removed from the rest of the input variable selection procedure. This pruning of the input variable set is repeated for as long as the J_{MSE} is improved by removing variables. The search continues until all the variables have been added to the subset. The calculation time of the method is considered to be moderate because the J_{MSE} is determined maximum of $m^2 + m - 2$ times, where m is the number of variables. Next, better input variable sets are searched for with a genetic algorithm. The GA is given the input variable set found in the previous phase as a starting point in order to speed up the search and to ensure that the resulting variable set is at least as good as the reference set. The search is also made more efficient by applying elitism, i.e. by preserving the best performing variable set as such throughout the generations and also by automatically including it in the mutation phase of GA. The genetic algorithm used in the methodology is as follows:

(1) m random chromosomes are created, where m is the number of available input variables. The number of variables in the chromosomes is set such that there is one chromosome with one variable, one with two, and so forth. The last chromosome thus contains all the variables.

(2) Models are constructed using input variable sets determined by the chromosomes, and the models' performances are evaluated.

(3) $\frac{m}{2}$ chromosome pairs are chosen to create the next generation of chromosomes.

The selection is done randomly, but the chromosomes producing models with good performances have higher probabilities to be chosen.

(4) A new generation of chromosomes is created by mating the selected chromosomes of the pairs. The chromosomes are cut into two pieces and then two new chromosomes are formed using the pieces. The cutting point is determined randomly and the two chromosomes are cut using the same point in order to have pieces of suitable length to form new chromosomes with exactly m genes. The chromosome pairs also have a low probability of not becoming mated, and the new chromosomes are then clones of the original ones. (5) The genes of the new chromosomes are mutated with a low probability.

Repeat (2) to (5) until a stopping condition is met. The two stopping conditions are: (1) a predetermined maximum number of generations is reached, and (2) the performance of a model exceeds a predefined limit.

Step 3. The dimension of the input variable set is reduced. The aforementioned mathematical methods search for input variable sets that minimise the prediction errors of the models. If several input variable sets produce similar prediction errors, then the one with the least variables is selected. In addition, the previously determined input variable sets are also pruned with the following algorithm.

(1) Leave one variable out of the input variable set and determine the model's performance. Calculate the leave-one-out performance for every input variable.

(2) Discard the variable corresponding to the best leave-one-out performance.

Repeat (1) and (2) as long as the performance of the reduced model remains acceptable. In addition to pruning, the dimension of the set can also be further reduced using dimension reduction algorithms like PCA and ICA. In this manner, techniques

that can also be used independently to model the processes, can be used as a part of other methods.

3.4.2 Selection of a training data set

Next, the available process data must be divided into the training data set and the testing data set that is used to evaluate the constructed model. Both data sets should contain information about all possible process conditions. In this methodology it is recommended that the training and testing data sets be constructed with selected short periods of data covering the different parts of the data collection period and all major process conditions. Following this recommendation ensures that both data sets contain information about slowly occurring changes in the prevailing process conditions, e.g. due to fouling of the process equipment.

3.5 Diagnostic information and fault decisions

Next, the diagnostic information provided by the FDI systems and the utilised fault decision algorithm of the methodology are described. First, the individual residuals provided by the models are presented in section 3.5.1. Then the proposed change detection algorithms, utilising the residuals, are introduced in section 3.5.2. The information provided by the change detection algorithms can be used in fault tolerant control. The derivation of the confidence indices for the estimates are described in section 3.5.3 and, finally, the tuning of the FDI system is presented in section 3.5.4.

3.5.1 Residual generation

In the methodology, the difference between the estimated and measured values is the most important residual used in detecting faults. When using multivariate statistical models, the separate residuals of all the estimated variables are squared and summed together. This residual index is the squared prediction error (SPE), also known as the Q indicator. Another multivariate index used in the methodology is the Hotelling T^2 indicator. It describes the model's validity for creating the estimate under current operating conditions. The Hotelling T^2 index is determined from the principal components or latent variables that have been included in the model because the PCs or LVs that explain only a small fraction of the variance of the input data are sensitive to noise and typically degrade the reliability of the Hotelling T^2 index if they are included in the calculations (Chiang et al., 2001). The Hotelling T^2 and SPE are used in determining the confidences of the estimates as described in section 3.5.3.

3.5.2 Change detection algorithms

Making the decision about whether a monitored system is in a faulty or normal state is a logical operation with a binary result. The decision is made by comparing the residuals produced by the models to the values corresponding to normal process operation. The limits for normal behaviour are commonly set within three times the standard deviation of the normal data set from the average value. For data following the normal distribution, these so called 3-sigma alarm levels correspond to a 0.27 % alpha risk, meaning that 99.73 % of the normal situations are classified correctly and for 0.27 % of the cases a false alarm is given (e.g. Chiang and Colegrove, 2007).

An alarm limit for the Hotelling T^2 indicator (3.11) has been introduced by Jackson

(1959), and for the SPE (3.12) by Jackson and Mudholkar (1979).

$$T_\alpha^2 = \frac{m(n-1)(n+1)}{n(n-m)} F_\alpha(m, n-m) \tag{3.11}$$

where m is the number of principal components or latent variables included, n is the number of data samples, and $F_\alpha(m, n-m)$ is the critical value for an F distribution with m and $n-m$ degrees of freedom.

$$Q_\alpha = \Theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\Theta_2}}{\Theta_1} + 1 + \frac{\Theta_2 h_0 (h_0 - 1)}{\Theta_1^2} \right]^{\frac{1}{h_0}} \tag{3.12}$$

where $\Theta_i = \sum_{j=a+1}^{n} \sigma_j^{2i}$, σ is a singular value, $h_0 = 1 - \frac{2\Theta_1\Theta_3}{3\Theta_2^2}$ and c_α is the normal deviate corresponding to the $1-\alpha$ percentile.

Analysing the residual values separately is used to detect large abrupt faults, but the method is vulnerable to outlier values of measurements that can cause false alarms. The limit-checking methods are also insensitive for small and incipient faults. As a consequence, more sophisticated tests are also used. For instance, alarms should only be given after a few consecutive limit violations have been detected. The same conservative approach should be adopted when deciding when the process has recovered from a fault and is given a normal operation status. Waiting for a few consecutive alarm limit violations will induce a delay in the fault detection, but it will also greatly reduce the number of false alarms. Another option to avoid the effects of noise and outliers is to use cumulative sums of the residuals in alarm limit checking instead of single values. The use of CUSUMs was first suggested by Page (1954) and has been utilised in an ANN based FDD system (Leger *et al.*, 1996).

In the methodology, the main criteria for the fault decision making are the correctness of the decisions and the fault detection speed. An acceptable compromise between these conflicting objectives is sought after using a modified Page-Hinkley cumulative sums method. The algorithm has two branches, one for detecting upward faults and another one to detect downward faults, and it performs both change detection and fault diagnosis. The original Page-Hinkley algorithm equations for de-

tecting positive jumps, i.e. changes in the signal, are presented below.

$$U_0 = 0 \tag{3.13}$$

$$U_n = \sum_{k=1}^{n} (y_d(k) - \nu) \tag{3.14}$$

$$m_n = \min_{0 \leq k \leq n} U_k \tag{3.15}$$

$$\hat{f} = U_n - m_n \tag{3.16}$$

where U_n denotes the cumulative sum, y_d is the residual, ν is the detection limit, i.e. the magnitude of the smallest fault that will be detected, m_n is the minimum value of the cumulative sum, and $\hat{f}$ is the estimated fault. After the change detection, the fault diagnosis is made using a simple threshold method: If $\hat{f} \geq \lambda_f$, where λ_f is the predefined limit for fault, then a fault is detected. Similar sets of equations are used to detect negative jumps.

The original Page-Hinkley algorithm has a flaw that may cause problems when the algorithm has been running for a long time on a computer; the *min* and *max* values approach minus and plus infinity when the signal represents normal behaviour. To avoid this problematic situation, the algorithm is modified to keep the *min* and *max* values at zero (3.17).

$$m_n^{'} = \max(0, m_n) \tag{3.17}$$

where $m_n^{'}$ is the modified minimum value, which is used instead of m_n in (3.16). A similar modification is made in the algorithm for detecting negative jumps.

Another weakness of the original algorithm is the wind-up: during a long-lasting fault, the cumulative sum increases continuously and, after the fault has been fixed, the algorithm still indicates the fault, because the cumulative sum remains above or below the alarm limit. The algorithm is modified further to speed up the return to a normal state. After three consecutive instants the residual is within the limits for the normal state (i.e. $y_d(\kappa) - \nu < 0$), the cumulative sums are reset to zero and the fault indication is removed.

The reliability of a fault is determined on the basis of the ratio of the cumulative sum and the threshold limit according to (3.18). (3.18).

$$F_{rel} = \min\left((\frac{U_n}{\lambda_f} - 1)/a, 1\right) \tag{3.18}$$

where F_{rel} is the reliability of the fault and a is a tuning parameter.

The magnitude and direction of detected faults are determined by comparing the estimated values with the real outputs. The residual is the estimated magnitude of the fault.

$$F_{est} = \hat{y} - y \tag{3.19}$$

where F_{est} is the estimated magnitude of the fault, $\hat{y}$ is the estimated value, and y is the corresponding output.

3.5.3 Confidence index for the estimates.

Data-based models are valid only under process conditions that are described by the training data. As a consequence, the estimates made under new conditions are unreliable. To determine the validity of the models and the confidence of the estimates an index is developed. For PCA and PLS models, the measures suitable for the validity of the model are the Hotelling T^2 and SPE indices. These indices give information about abnormal process measurement and systematic changes in the structure of the data. It is assumed that, under these conditions, the models are no longer valid.

The reliabilities of the estimates of the PCA and PLS models are based on the Hotelling T^2 and SPE indices, as indicated in Equations (3.20) and (3.21).

$$HT^2_{sig} = \frac{1}{1 - e^{-c(HT^2_0/a - d)}} \tag{3.20}$$

$$SPE_{sig} = \frac{1}{1 - e^{-f(SPE_0/b - g)}} \tag{3.21}$$

where HT^2_{sig} and SPE_{sig} are the scaled values of fault indices, HT^2_0 and SPE_0 the original values, and a, b, c, d, f, and g are tuning parameters. The final value for the reliability when combining the two indices is determined by (3.22).

$$\hat{y}_{rel} = 1 - \frac{HT^2_{sig} + SPE_{sig}}{2} \tag{3.22}$$

where $\hat{y}_{rel}$ is the final reliability of the estimated value.

3.5.4 Tuning of the fault detection parameters and optimisation criteria for fault detection decisions

The main objective of a FDI system is to classify the normal and faulty operation conditions correctly. This is presented as an objective function (3.23).

$$\text{Min } J_{tot} = \sum_{i=1}^{n} (|\hat{f}^0(i) - f^0(i)|) \tag{3.23}$$

where $\hat{f}^0(i) \in 0, 1$ and $f^0(i) \in 0, 1$ are the estimated and real status of the process, i is the time instant, and n the length of the testing period. The parameters that are optimised are the λ and ν of the fault detection equations (see section 3.5.2). Since no general analytical solution is available for solving the minimisation problem, the optimum tuning parameter values are determined by means of simulations.

3.6 Assessment of the performance of the FDI system

The last phase in the setting up of an FDI system is to evaluate its performance. In the methodology, the performance is determined as the percentage of correctly indicated states of the process. To obtain a realistic assessment of the performance, the data used for validation should be rich process data containing different operation regions and transitions between the regions. The data should represent the

process in the normal operation state and faults, similar to those occurring in the process, are artificially added to the data. This arrangement is recommended because the exact starting times and magnitudes of the faults are then known, enabling an accurate performance analysis.

In the evaluation of the FDI system consisting of multiple models, it is recommended that the performances of the models be considered separately. This is important, because the different types of model provide different kinds of information about the process. Methods that monitor only process conditions, like DISSIM and PCA which do not emphasise the importance of the product information, produce results that can be used as the first indicators that something is happening in the process. These changes in the process may not have a direct effect on the product quality but, if left unsupervised, the deviation from normal state might grow and become a more serious problem. PLS and other regression models monitor the process and give more weight to those variables that affect product quality. Therefore alarms given by these models should instigate an immediate response. It is recommended to give more weight to the models using calculated variables, because they detect deviations from the important process features represented by the calculated variables. The rules governing the determination of the total outcome of the fault detection and different roles of the models are determined using prior engineering knowledge.

4 General description of the Naantali oil refinery dearomatisation process and its automation and control systems

The methodology presented in Chapter 3 is validated by building an FDI system for the Naantali oil refinery dearomatisation process. Next, the process, including the automation and control systems, is described.

4.1 Dearomatisation process

Dearomatisation processes are widely used in the oil refining industry. These processes are used to remove aromatic compounds in the feedstock by hydrogenation. The Naantali process consists of two trickle-bed reactors with packed beds of catalyst, a distillation column, a filling plate stripper, several heat exchangers and separation drums, and other unit operations. The process diagram of the dearomatisation process at the Naantali refinery is presented in Figure 4.1.

The cold, liquid feedstock fed to the unit is heated with streams from the two reactors in the heat exchangers EA1 and EA2 and with hot oil in EA3 and then fed to the reactor DC1 together with hydrogen and recycle liquid. Exothermic saturation reactions in the first reactor remove most of the aromatic compounds when the catalyst is new, while most of the reactions occur in the second reactor when the catalyst is older and has been partly deactivated. After dearomatisation in the reactor DC1, the reaction product is cooled in the heat exchanger EA1 and then fed to the gas separation drum FA1. Gaseous and liquid reaction products are separated in the drum. Part of the liquid is circulated back to the reactor DC1. The rest of the liquid, together with separated gas and fresh hydrogen, are fed to

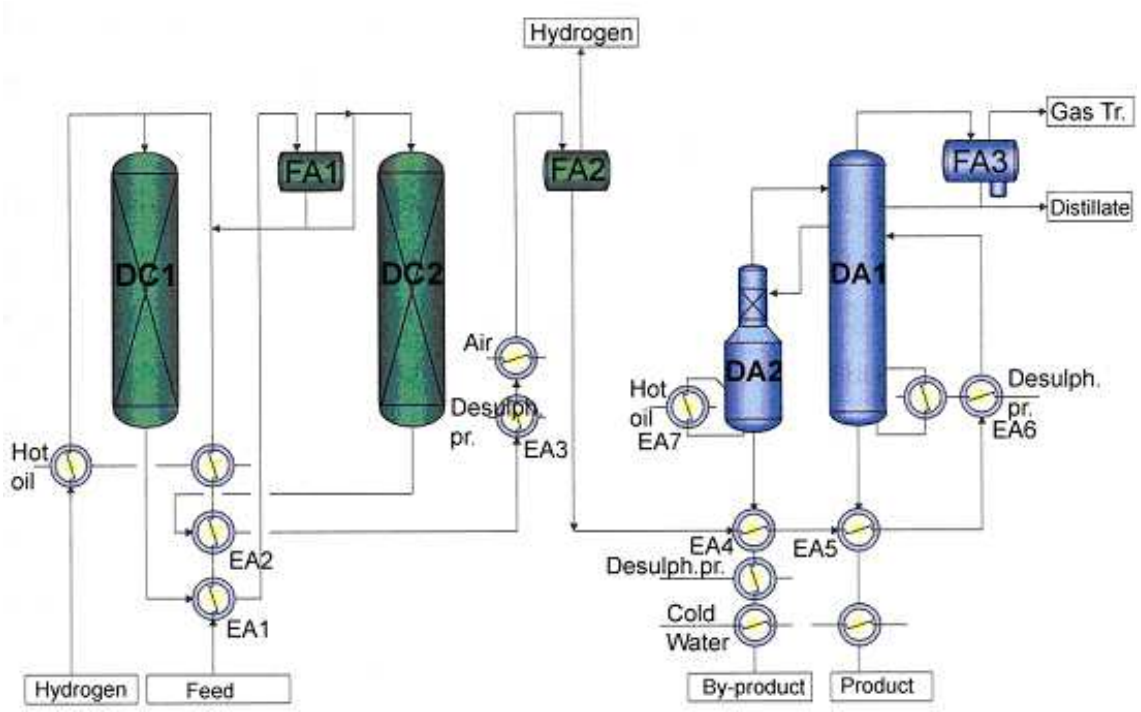

Figure 4.1: Dearomatisation process at the Naantali refinery. (Kinnunen, 2004)

the second reactor DC2, where the aromatics level of the product drops to near zero. After the second reactor, the reaction product is cooled in the heat exchangers EA2 and EA4 and fed to the second gas separation drum FA2. Gas separated from the liquid mainly consists of unreacted hydrogen, which is recycled back to the first reactor, and the rest of the gas is removed. The separated liquid is heated with by-product and product streams in the heat exchangers EA5 and EA6. Part of the liquid is further heated in the heat exchanger EA7 in order to achieve the final temperature before the stream is fed to the distillation column DA1. The overhead of the column is cooled in a cooler and then fed to the separation drum FA3, where the gaseous part is removed and the liquid is divided into reflux and distillate. The distillate consists of the lightest compounds of the reaction product. The heat exchanger EA6 produces heat for reboiling the bottom stream. With certain feed types a side stream is conducted to the stripper and heated up with the heat exchanger EA8. A by-product stream is drawn off from the bottom of the

stripper (DA2). The non-aromatic main product is drawn off as the bottom product of the distillation column DA1 and cooled in the heat exchanger EA6. The quality of the cooled product is determined online with the flash point temperature and distillation curve analysers. Laboratory assays of the product quality are performed twice a day. The dearomatisation process has no noticeable effect on the heaviest parts of the distillation curve of the feedstock, but the properties of the lightest cuts are strongly affected by the distillation. Six petroleum cuts with different properties are used as feedstocks to the process and a change of feed type is made, on the average, once every 4 days. In addition to having feed types with clearly distinct properties, the composition of the feed may also vary within the feed types.

4.2 Online quality analysers

The product quality of the dearomatisation process is monitored by three online quality analysers; distillation temperature analyser of the bottom product, flash-point temperature analyser of the bottom product and flashpoint temperature analyser of the side product. The higher level quality control uses the analyser results as feedback. Faults in the analysers lead to situations in which the quality of the product is higher than required or below the specifications. Both of these situations cause losses in profitability. Because of the highly nonlinear blending characteristics of the initial boiling point (IBP) and flashpoint (FP), even short periods of off-spec production can contaminate large quantities of solvent in product tanks. Contaminated products are reprocessed or downgraded to less valuable diesel oil. The distillation analyser is used only with the dearomatisation process while the bottom product FP analyser is also used for analysing product flow of another unit process. As a consequence, the flashpoint temperature analysis results for the bottom product may be unavailable for long periods of time. The side product flash point temperature analyser is only used with certain feed stocks that require the use of the second column, DA2.

4.2.1 Bottom product distillation analyser

The distillation curve analyser operates on the principle that a sample is heated up in a vessel and the vapours are conducted in another vessel. The volume of the condensed fraction of the sample is measured. Temperatures, in which the volume starts to increase and when 5, 10, 50, 90, 95 and 100 % increase of the volume has been registered, are given as the outputs of the analyser. The values become available in the automation system one after another with several minutes intervals, as the evaporation and condensation progress. The most significant characteristic of the distillation analyser is long cycle time between the assays. Handling of the sample, cooling of the flask, careful heating of that sample and the cleaning of the equipment between samples takes about 40 minutes. An additional feature of the analyser is that the delays between the analysis cycles are not constant, but vary between 40 and 43 minutes according to the quality of the sample.

4.2.2 Flashpoint analyser

The flashpoint temperature analysers operate by heating a liquid sample in an open-air container. An electric spark is generated periodically over the container and the temperature in which the mixture of air and evaporated sample ignites is given as the flashpoint temperature. Opposed to the distillation analyser, the analysis cycle of the flashpoint analyser is quite short, only about 2 minutes. The delay between the analysis results is, however, variable and changes normally between 1 and 3 minutes.

4.3 Automation system at the Naantali refinery

The automation system at the Naantali refinery follows the three layer structure commonly encountered in process industry: process layer, basic control layer, and supervisory control layer. The process layer consists of analog point-to-point connections between the field devices and process stations. The basic control layer consists of a Damatic XD DCS system. The DCS system connects various stations (process stations (PCS), operation stations etc) in the same network and executes the basic controls. The communication in the network is implemented with proprietary Damatic XD protocols. Due to the refinery's high safety requirements, the DCS system has been designed to be very reliable. The system is totally hot standby doubled with explosion-proof instrumentation. In addition, the DCS system is built up such that there are no significant network delays in process control. An outline of the Naantali refinery automation system is presented in Figure 4.2.

On the supervisory control layer advanced controls are performed. One essential part of the layer is a process control computer system. Advanced control applications, such as the Neste Jacobs NAPCON advanced controller and real time database, are implemented in the process computer. Connections to the DCS and other systems, such as Oil Movement and Storage (OMS) are established through the process control computer. The operation system of the process control computer is OpenVMS for AXP Alpha, and the programming language that is most widely used in the advanced applications of the computer is DEC Fortran. The network in the supervisory control layer is implemented with Ethernet technology. The supervisory control layer is connected to the basic control layer through the process computer interface gateway station (GTW). The gateway forwards commands, control outputs and data sent by the process computer to the DCS system, and vice versa. Proprietary application protocols are used in the communication.

The process data acquired through the DCS system are stored in the real time

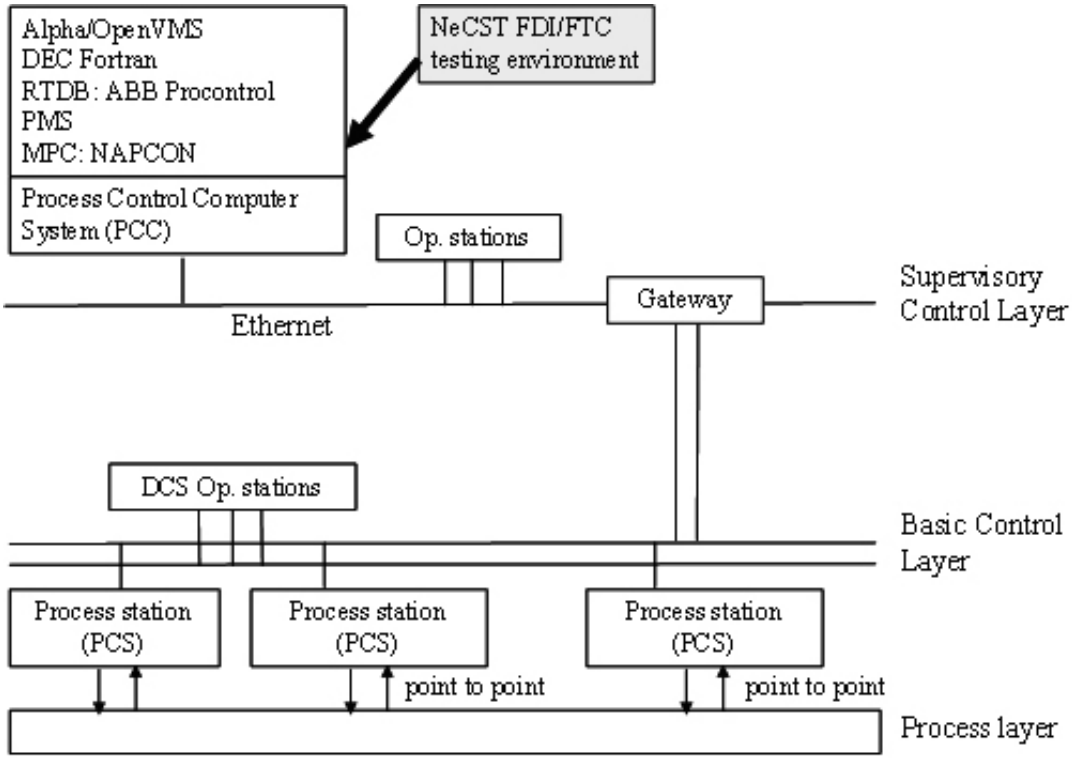

Figure 4.2: Simplified view of the Naantali refinery automation system hierarchy. (Vatanski *et al.*, 2005)

database. The raw, high frequency, measurement data is first compressed to 1-minute averages which are stored for several weeks before further compressing them into 1-hour averages. As the process dynamics are quite slow, the 1-minute average data is suitable to be used for fault detection.

4.4 Control of the dearomatisation process

The dearomatisation process is a highly instrumented process environment with a large number basic and upper level controls. The overall control strategy is presented in the following. The volume of the liquid feedstock fed to the first reactor is set by the product flow coming from a desulphuration process preceding the dearo-

matisation process. The feed flow rates into the second reactor and the distillation column are determined by the level controls of the separation drums. The bottom levels of the distillation column and side stripper are controlled by manipulating the product flow rates. Pressure in the first reactor is controlled by manipulating the hydrogen recycle feed flow rate from the separator drum, and by controlling the pressure of the hydrogen feed. The pressure of the hydrogen feed can be lowered by directing a fraction of the flow to other sub-processes using the same hydrogen line. The pressure level and pressure difference over the reactors are measured with four pressure sensors. The inflow and outflow temperatures of both reactors are measured. Temperature profiles of the catalyst beds inside the reactors are monitored using four equally spaced temperature sensors in each reactor. Only the operation temperature and the temperature of the inflow of the first reactor are controlled. The desired inflow temperature is achieved by bypassing heat exchangers or by enabling the hot oil heat exchanger. Since the operation of the first two heat exchangers is strongly connected to the process, control of the inflow temperature is usually done by manipulating the hot oil heat exchanger flow rate. The rate of the exothermic hydrogenation reaction is controlled with the hydrogen recycle stream with low aromatic content from the separator drum. The control is based on keeping the temperature difference between the reactor and the hydrogen feed flow below the maximum value. In the distillation part of the process, the temperature of the feed flow rate is controlled with the flow from the desulphuration process to heat exchanger EA6. The reflux ratio of the distillation column is maintained constant by controlling the flow rate of the recycle stream. The pressure of the column is kept stable with gas treatment flow controls. The operating condition of the distillation column is controlled by adjusting the amount of flow into the bottom boiler and by changing the boiler's power setting. (Komulainen, 2003)

5 Preliminary analysis and conceptual structure design of the fault detection system for a dearomatisation process

The first phase of the testing of the proposed methodology with the dearomatisation process is the preliminary analysis of the process and the conceptual structure design of the fault detection system. The reliable functioning of the multivariate quality control of the dearomatisation process is an essential factor in maintaining the profitability of Neste Oil Oyj Naantali refinery. Thus, the importance of the multivariate control and the related instrumentation is emphasised in defining the aims and user requirements for the FDI and system as presented in section 5.1. The quality control relies on the feedback provided by the online analysers. Thus the reliable operation of the analysers is important and the most common faults within the process are presented in section 5.2. The fault detection problem is defined and the structure of the system, including the FTC part, is presented in section 5.3.

5.1 General fault detection and isolation requirements for a process monitoring system

The online product quality analysers are essential parts of the Naantali dearomatisation process enabling the higher level quality control of the dearomatisation process. The determination of the user requirements for the system was done by interviewing refinery's personnel representing different user groups from operation and maintenance personnel — experts, engineers, technicians, operators, maintenance personnel, and managers — about the requirements of the monitoring system. The interviewing method was based on the Delphi method (Linstone and

Turof, 1975) consisting of two interviewing rounds. The interviews consisted of four main topics. First, the properties and features of the FDI system related to the user interface, interfaces to other software, and installing, updating and upkeeping of the FDI system were enquired after. Second, typical faults related to the analysers were analysed in more detail. The emphasis was on faults that can be predicted and prevented and on faults that are difficult to foresee but whose effects on the process can be mitigated. Third, tools and information currently available for the operating personnel for detecting faults were discussed to determine the possibilities for improving the situation either by detecting previously undetectable faults or by more rapid detection of other faults. The fourth and last topic was related to the actions that are taken after a fault has been detected. The current situation was enquired and the need for new partly or completely automated recovery actions was discussed. After the first round of interviews the answers were analysed and a set of more detailed questions about the most important analyser faults was created and used in the second round of the interviews. The interviews were conducted at the Naantali refinery on October 27^{th} and November 9^{th} 2004. The requirements relating to the monitoring and fault detection are presented in the following sections. The complete set of the requirements, are given in Vatanski *et al.* (2005).

The faults occurring in industrial production sites have been classified in four distinctive categories to focus the efforts in developing specific fault detection and fault tolerant control (FTC) algorithms. Two main categories are predictable and unpredictable faults. Predictable faults can be foreseen if the variables involved can be measured to an adequate accuracy. Unpredictable faults, on the other hand, occur without any measurable warning. Both predictable and unpredictable faults can be further divided into two categories, faults whose effects can be mitigated or restricted to a specific area, and faults whose effects cannot be mitigated. The fault detection methods implemented in the FDI system are aimed at detecting those predictable faults whose effects can be mitigated or restricted. In the interviews, special attention was paid to reducible predictable faults as they are the most suitable targets for

fault detection and fault tolerant control. The general FDI requirements are listed in Table 5.1.

Because the two online analysers operate independently of each other, the monitoring problem was decided to be divided into two parts, i.e. to construct separate FDI models for the flashpoint analyser and for the distillation analyser. The distillation analyser gives multiple outputs, but only the initial boiling point temperature (IBP) used in quality control, was decided to be considered in monitoring. Thus, the two variables in which faults are detected are the flash point temperature (FP) and the IBP. A dynamic simulator with an existing model of the dearomatisation process was decided to be utilised in the creation of the FDI system for two reasons. First, data generated by the simulator could be used in creating the FDI models for the real process as shown in Brydon *et al* (1997) and second, using the simulator allowed more rigorous testing of the application before the it was validated at the refinery.

Table 5.1: General requirements for the fault detection system of the dearomatisation process

Requirement topic	Requirement
Detected types of fault (REQ-1)	The FDI system shall detect incipient faults that do not cause variables to violate their alarm limits. (REQ-1.1)
	The FDI system should detect faults, especially drifting of the measuring devices. (REQ-1.2)
Time instant of fault detection (REQ-2)	The system should be able to inform the operator about faulty conditions as early as possible. When the faults are detected in time, their effects are easier to mitigate and have a smaller impact on the overall process. (REQ-2.1)
Background information about the FDI methods (REQ-3)	The system shall provide background information about the fault detection and isolation methods used, as well as the assumptions used in diagnosis.
FDI taking into account external factors (REQ-4)	The FDI system shall identify and be aware of the current operating point in order to able to detect smaller deviations from nominal operation conditions. (REQ-4.1)
	The change in operation point shall also be detected and must not be categorised as a fault. (REQ-4.2)
Being aware of the calibration of measuring devices (REQ-5)	The effects of calibrating measurement devices shall be stored and taken into account in fault diagnosis because calibration produces sudden changes in measurement values, and these might be detected as faults. (REQ-5.1)
	The date of calibration shall be used as one reliability measure of the measurement. (REQ-5.2)

5.2 Most common faults in the dearomatisation process

Next phase of testing the methodology is the analysis of the faults occurring in the dearomatisation process. In the study (Liikala, 2005) the refinery's logbook and process history were examined in order to gather specific information of the faults and the abnormal situations in the dearomatisation process. The chart in Figure 5.1 illustrates the different types of faults and their proportion out of all the faults recorded during one year of operation. The majority of the faults were related to analyser malfunctions. This result was confirmed by the interviews. Based on the interviews and the Liikala study, it was anticipated that the most significant potential for improving the operation of the process lies in the earlier detection of analyser faults. Early detection of faults enable timely execution of corrective control actions and the quality of the end product can be kept within the production limits, thus improving the plants economical performance.

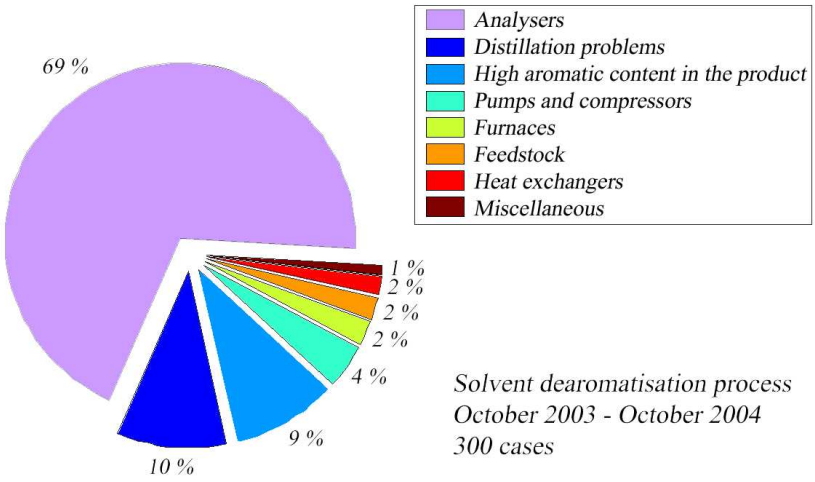

Figure 5.1: Most common faults in the dearomatisation process during one year of operation. (Liikala, 2005)

The distillation and flashpoint analysers suffer mainly from contamination of the sample with water and in the case of distillation analyser, from the carbonisation

of the flask. The contamination of the sample causes the analyser outputs drop abruptly while the fouling of the flask introduces slowly progressing deviation from the correct value. Currently, the analyser faults are detected and handled in the following manner: First, the analyser faults are mainly detected heuristically by comparing the analyser measurement values with results expected on the basis of process measurements. In some cases, the first indication of a fault is an abnormal set point value given by the advanced control system. If a fault is suspected, an additional laboratory analysis is made and the suspected analyser fault is verified by comparing the analyser result to the result of a laboratory analysis. If the presence of a fault is verified, the control of the corresponding quality variable is disabled in the advanced control system and an analyser maintenance team responsible for repairing the analysers is informed. The process is then operated on the basis of the last laboratory analysis. During these situations when the analyser if off, the laboratory carries out analyses more frequently than during normal operation. In some cases, changes are made to the product flows in order to mitigate the effects of the fault. After the analyser has been repaired, it is taken back into use and the previously disabled quality control is switched on.

5.3 Structure of the monitoring system for the dearomatisation process

Next, a preliminary design of the structure of the FDI system was created. The monitoring system was decided to be implemented as a part of a more complex operator support system developed within the Networked Control Systems Tolerant to Faults (NeCST) project. This system consists of two main parts: a fault detection (FDI) system and a fault tolerant control (FTC) system, acting on a model predictive controller (MPC). The FDI system is responsible for detecting faults in the online quality analysers of the dearomatisation process. Information provided

by the FDI is given to the FTC, which executes fault tolerant actions according to a predefined triggering logic. The FTC actions affect the tuning parameters of the MPC and are thus realised with the MPC and lower level controllers. The NeCST software prototype was designed to have an access to the Naantali refinery automation system through the process control computer. The structure is presented as a simplified block diagram in Figure 5.2.

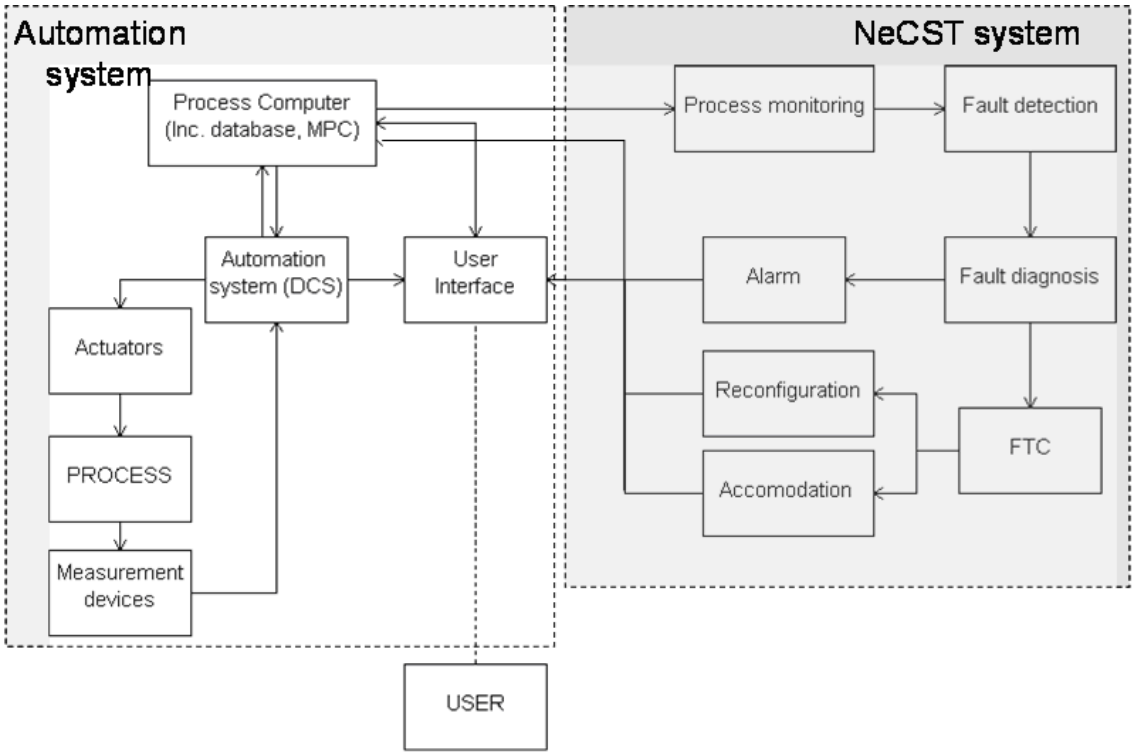

Figure 5.2: Conceptual structure of the existing Naantali refinery automation system and the NeCST system. (Vatanski *et al.* 2005)

In addition to the structure, also a functional model, illustrated in Figure 5.3, of the system was developed to ensure that the functionality of the NeCST platform will fulfill the requirements on FDI and FTC. The model was designed using partial CIM-OSA models (Kosanke *et al.*, 1999) with IDEF0 modelling language. The importance of FDI/FTC of the process related faults was emphasised over the faults of the communication network. This was due to safety and economical reasons, but also

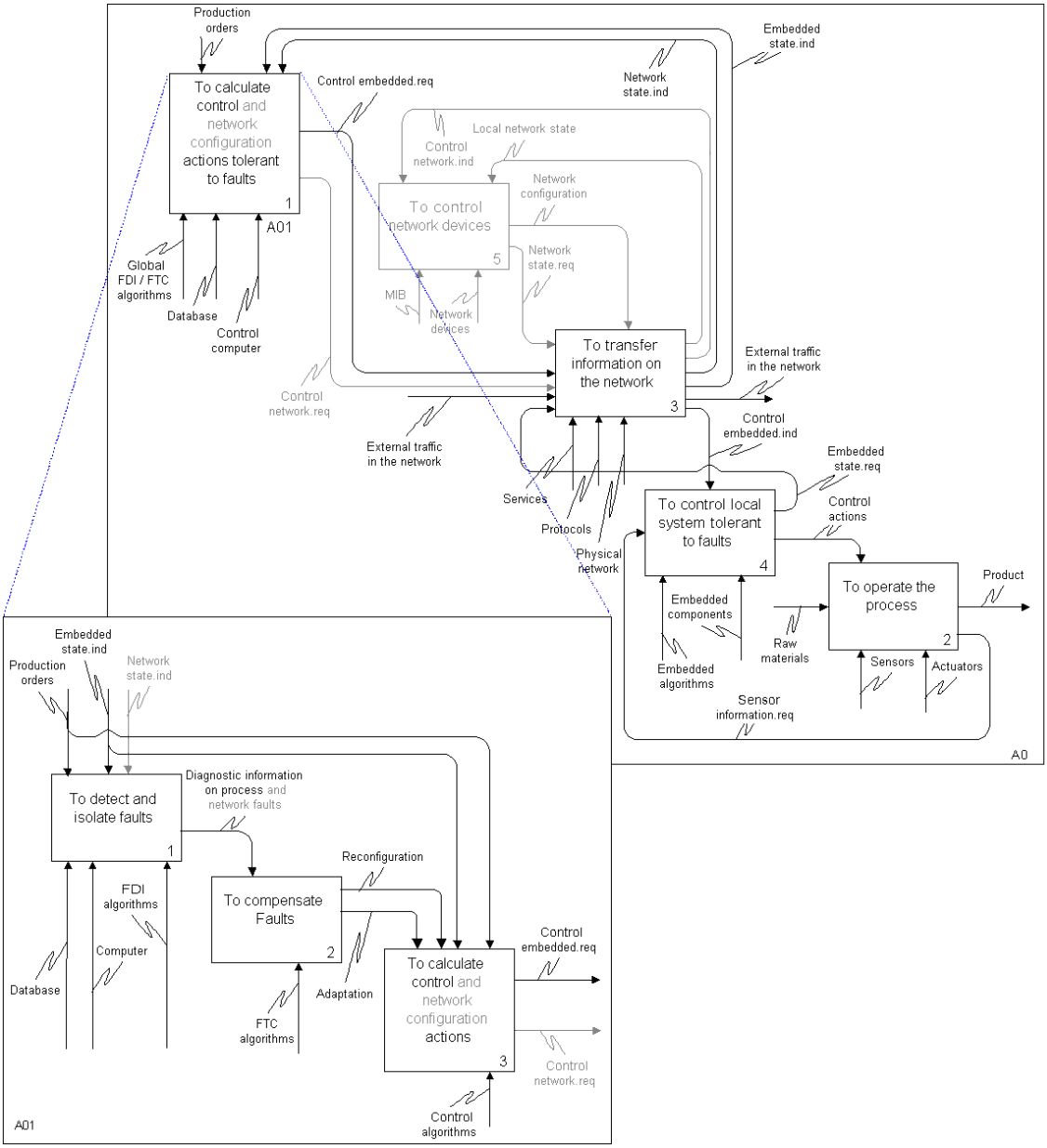

Figure 5.3: Functional model of the NeCST system at the Naantali refinery. (Vatanski *et al.* 2005)

due to the fact that at the refinery, the communication network is duplicated and has not experienced any problems in the past. Because the faults in the communication network were considered less important, the blocks related to those issues were greyed out and were not implemented.

The FDI system was decided to be implemented into two separate environments with slightly different characteristics. The first environment was used to test the system offline, with both simulated data and with real measurement data transferred over a remote connection to the testing facilities. The second application was implemented at the Naantali refinery environment for online testing.

The FDI system is a part of a larger process monitoring and control system including FDI and FTC functionalities. To test the systems functionality offline a database dump from the process control computer of the Naantali oil-refinery was transferred every 30 minutes to a local computer. A so-called local application gave new process data sample every minute as an XML-based message to so-called global application. The global application consisted of several functionalities of the CASIP (property of PREDICT) system. The FDI routines were attached to the CASIP as C++ .dll and .lib files compiled from the Matlab m-files with the Matlab v6.5 compiler. Based on the information produced by the FDI, CASIP orchestrated the FTC actions that had no effect on the process control in this remote setting. An alternative off-line set-up was to connect the global application to a dynamic simulator (PROSimulator). In this setting, the simulated process measurements were used with the FDI and the FTC actions were given to NAPCON controlling the simulated process. The described offline system was mostly used for testing the systems FTC functions, more through testing of the FDI functions was carried out using only Matlab. The structure of the online system was similar to the offline version. The major difference being that in this setting the local application ran within the process control computer and the process measurement values given to the global system were the current ones. In addition, during the online validation the quality control (NAPCON MPC) was affected by the FTC actions and thus FDI/FTC affected the operation of the real process. The structures of the different implementations are presented in Figure 5.4 below.

During the online evaluation, the FDI system acquired all the process values through

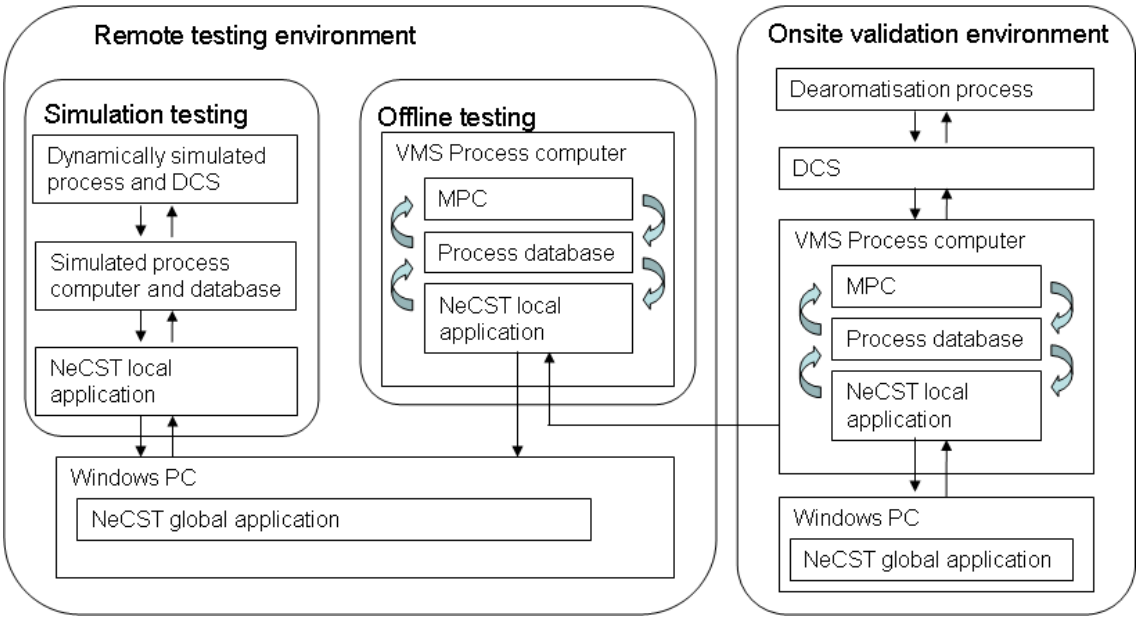

Figure 5.4: Structures of the offline and the online implementations of the NeCST system

the process computer, which provided means to introduce the faults. To introduce a bias to the original analyser outputs a separate function block was implemented at the process computer and thus the analyser values received by the FDI system could be modified for testing purposes. Also the higher level quality control used the modified analyser output in calculating the control actions.

The information provided by the system is shown to the operators with a display, shown in Figure 5.5. The display also enables operators to access the most important functions necessary to use the system. The display is divided into two parts: on the left, operator has buttons to enable and disable the FDI and FTC, to initialise the bias removal, and to acknowledge the detected faults. On right, the two curves display information about the IBP and FP analyser outputs and the corresponding estimates, residuals and reliabilities of detected faults.

In the multiple operating region simulation study, the data was generated with the

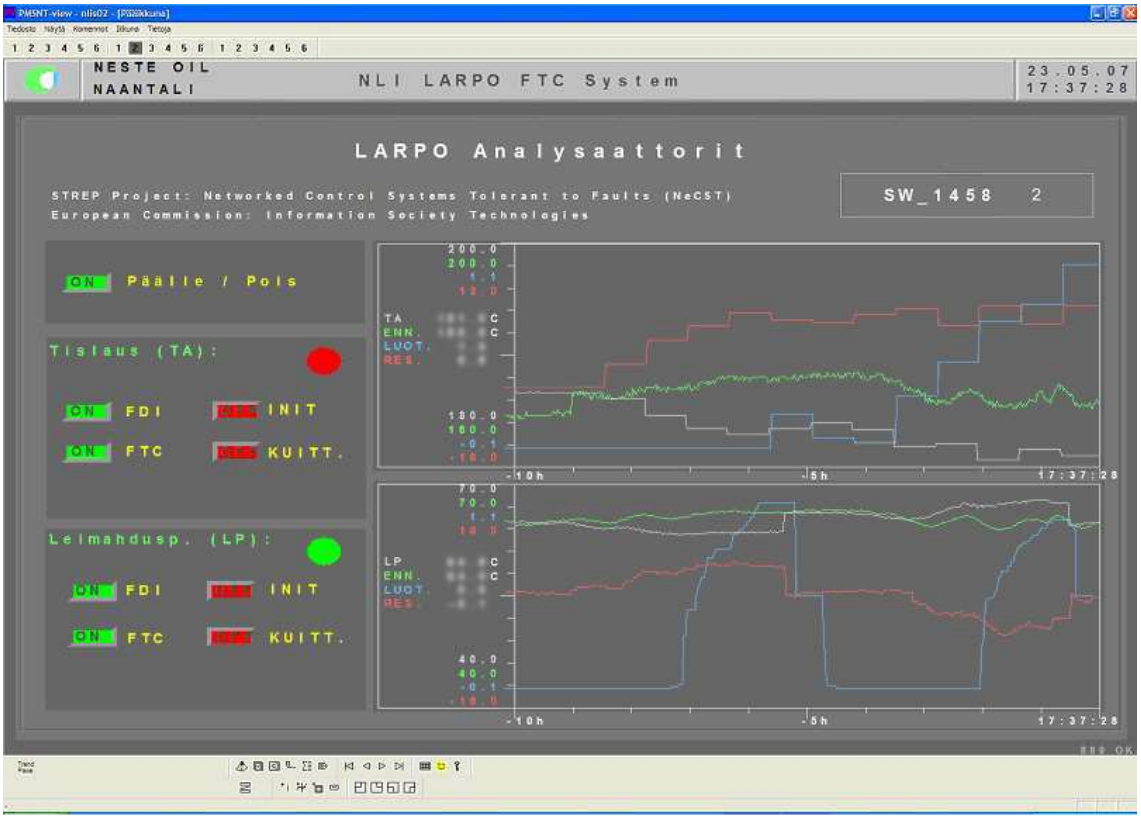

Figure 5.5: Operator display of the NeCST FTC-system.

PROSimulator that was connected to a database. The Neste Advanced Process Controller (NAPCON) was used in creating evaluation data sets with higher level control enabled. The fourth essential part of the simulation environment was Matlab, that was used to create changes in the relevant process variables during the simulations and to every minute store the history data of the process variables. The dynamic PROSimulator simulated the operation of the process and calculated new values of the process variables in 5 second intervals. Once every 60 seconds the values are stored into the database. At the same frequency the NAPCON read the values of the process variables and made changes to the values of the manipulated variables (Table 6.2) to control the initial boiling point temperature of the bottom product. Matlab made changes to process variables that are known to be relevant to modelling. The changes were performed at different frequencies so that

the variables' values were first decreased in 20 steps to a minimum value and then kept constant for 240 minutes. After that the value was increased in 40 steps to it maximum value, where it remained for another 240 minutes. Finally the value was decreased to its original value in 20 steps. In the ramp like changes the delay between two successive steps was 9 minutes. The variables' values were changed one at a time and the process was let to stabilise for 240 minutes before starting the change sequence on another variable. The PROSimulator, database (TMLDB) and NAPCON are all developed by Neste Jacobs Oy, the Matlab is property of MathWorks Inc. The application specific Matlab scripts and functions are written by the author. The simulation environment is illustrated in Figure 5.6.

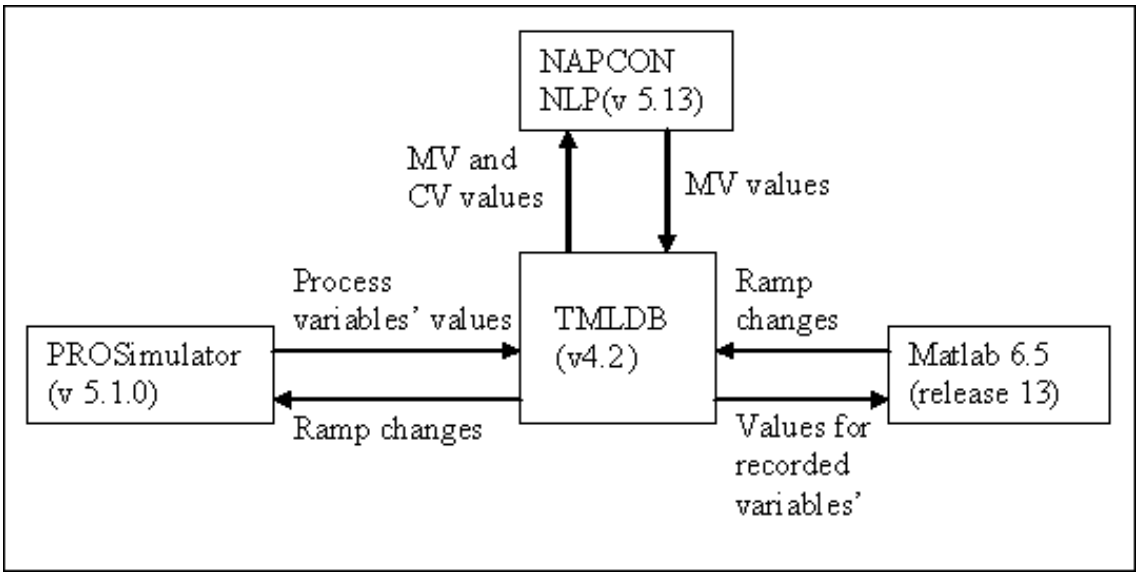

Figure 5.6: Environment for making the dearomatisation process simulations

6 Selection of the data-based monitoring methods and preparation of modelling data for training the models

Next, the appropriate monitoring methods for the dearomatisation process are determined in section 6.1 according to the guidelines of the methodology presented in Chapter 3. In section 6.2 the measurement data are prepared for training the models.

6.1 Selection of monitoring methods for the dearomatisation process

The fault detection system of the dearomatisation process has two main objectives: first, to detect faults in the product quality sensors i.e. the online analysers and second, to provide information of the reliabilities and magnitudes of the detected faults to the FTC. The first objective can be achieved with all the FDI methods mentioned earlier in Chapter 2, but estimating the magnitudes of the faults requires using regression type of models. The methodology recommends using the following regression methods, different versions of PLS, SMI, MLP and RBFN (Table 3.1). This group can be expanded further with PCA and SOM, as they can be modified to give estimates for single variables, in this case the analyser outputs.

The second major factor for selection of monitoring methods is the nature of the target process. The dearomatisation process is known to be nonlinear, but the nonlinearities are not so strong that they would prevent the use of linear FDI models (Komulainen *et al.*, 2004). Instead of strong nonlinearities, the most challenging

characteristic of the process is the changing operating conditions. Recommended methods for monitoring non-stationary processes are Moving PCA, Recursive PCA, Recursive PLS, and the ANNs: MLP, RBFN, SOM and ART.

The third factor that needs to be considered in the selection of the monitoring methods is the restrictions set by the implementation environment. The FDI system for the dearomatisation process was planned to be implemented on the real time NeCST software platform. The real time operation restricts the use of models which rely on heavy mathematical operations. Mathematically light methods include the PCA, Dynamic PCA, PLS, Nonlinear PLS, Multi-way PLS, SMI, and the ANNs. The second restriction imposed by the NeCST platform application is the maximum number of FDI models and thus only six models could be included in the implementation.

The final limiting factors in the model selection are the quantity and quality of the available process history data. The acquired data covered the operation of the process for a period of four months and one week and thus the number of the data samples was sufficiently high for all of the methods listed above. The quality, however, was not optimal for the data driven modelling. In spite of the length of the data collection period, the data did not cover the whole operating range. The reason for this was the numerous feed stock types, some of which are used only rarely. Because not all possible operation conditions were presented in the data set, the RBFN and ART relying on local approximation functions were decided to be unsuitable for the modelling. Furthermore, the multivariate outlier values degrade the performance of the methods based on minimising a least square cost function, PCA and PLS (Daszykowski *et al.*, 2007) and the noise included in the data degrades the performance of the ANNs.

In conclusion, no single method met all of the requirements set by the users, the implementation environment and the process characteristics. Considering these requirements, the most suitable method based on its characteristics was found to be

MLP. The drawbacks of MLP were however its sensibility to noise, possibly leading to over-fitting problems and its unpredictable behaviour in new process conditions. PLS on the other hand can handle the noisy input data, but the nonlinear characteristics of the process are only approximated (Wold *et al.*, 2001). The same is true for SMI, which met all the other criteria. Fourth possible method, SOM, met all the presented requirements, but the downside of using SOM as a regression method is the limited resolution of the model; the number of different output values is equal to the number of neurons of the SOM. PLS was selected as one modelling method as the nonlinear properties of the process were considered to be rather mild. PLS has also been shown to be suitable method for the process (Komulainen *et al.*, 2004). To address the possible problems in modelling the nonlinear process characteristics, the PLS model was decided to be augmented with an MLP model. Using the MLP together with a linear method, also reduced the significance of the weaknesses of noise handling and unpredictable behaviour of the MLP. This PLS-MLP combination model was the second selected model type. The third one was SMI, and for the same reasons as with PLS, it was also augmented with an additional MLP model part. Total maximum number of the implemented models was limited to six, and thus only these methods, PLS, SMI, PLS-MLP and SMI-MLP were chosen for modelling the two analyser outputs of the dearomatisation process FDI application.

To evaluate the applicability of the methodology for different types of cases two simulation cases were studied. The first testing experiment considered the detection of faults in online analysers of a simulated dearomatisation process within a single operating region. In this case, data covered the whole operating range, the process was stationary and could be assumed to be linear. Thus, following the reasoning presented earlier in this section and using Table 3.1, the selected methods for this case were: PCA, PLS, SMI and SOM. In the second testing experiment, the dearomatisation process operation was simulated in multiple operating regions and the data corresponded thus to a non-stationary process. The nonlinearities of the simulated process were assumed to be less complex than those of the real process. The

same monitoring methods were selected than with the real process case, but as the maximum number of the models was not limited in this case, SOM and a pure MLP model were also included.

6.2 Preparation of the data for training the monitoring models

6.2.1 Analysis and preparation of simulated process data

The data used in the single operating region study were created with the PROSimulator software developed by Neste Jacobs Oy. A data set consisting of 2395 data points, covering a period of almost 40 hours of operation, was used for modelling. Every hour of simulated operation the values of 1-3 variables were manipulated in order to create variance in the data. These manipulations initiated changes to process measurements similar to changes in real process data caused by normal operation actions. The feed type was not changed during the simulation and thus only one of the many possible operating regions was presented. The variables used to excite the process are presented in Table 6.1. The basic level control loops of the simulated process were closed, but the higher-level quality control was not active.

For the simulated multiple operation region case, 7 data sets were generated; one for creating the models and 6 for evaluating the performances of the models under different operation conditions. In the real dearomatisation process, the properties of the feed stock vary depending on the quality of the original feed material and the operation of previous processes. During the simulations two clearly different qualities are used; feed type 1 consisting of the lighter fractions of processed solvent and feed type 4 consisting of the heavier fractions. The initial boiling point temperature for feed type 1 is typically around 160 °C and for the feed type 4 it is around 210 °C.

Table 6.1: Manipulated variables of the first testing experiment simulation

Composition of the solvent feed
Flow rate of the solvent feed
Composition of the hydrogen feed
Temperature of the hot oil to heat the boiler of the column 1
Flow of reflux from the separation tank 1 to the reactor 1
Temperature of hot oil to heat up the feed to the reactor 1
Fan speed of air cooling of solvent before the separation tank 2
Flow of reflux to column 1
Temperature of hot oil to heat up the stripper
Setpoint of pressure controller of the column 1
Setpoint of temperature controller of the flow from the stripper to the column 1

The first simulation was done to generate a data set suitable for constructing the models, thus the higher level control (NAPCON) was disabled. The simulation started with the feed type 4. Ramp type changes were made to 19 variables to excite the process. The manipulated variables are listed in Table 6.2. In the table, the typical values of the variables associated with the two feed types are listed as well as the magnitudes of the changes. The ramp changes were performed both downwards and upwards with the feed type 4, and then the feed stock was changed to feed type 1. During the transition phase, the variables whose values differ between the feed types were gradually changed, in 20 steps, to the new values related to the feed type 1 operation. Then the experiment sequence was performed again in the new operation region. The data set consist of 35500 data samples, each corresponding to 1 min of process time i.e. the data covers 24 days, 15 hours and 40 minutes of operation.

The two operation regions are clearly distinct as illustrated by IBP having 50 °C

Table 6.2: Manipulated variables of the second testing experiment simulation, quality control off

Description	Feed 1	Feed 4	Max. change
Temperature setpoint for the solvent entering the reactor 1 [°C]	140	170	30
Temperature difference across the reactor 1 [°C]	21	21	5
Temperature of solvent feed [°C]	50	2	20
Temperature of the H2 feed [°C]	79	79	20
Flow rate of the cooling water for the distillate of column 1 [t/h]	75	75	5
Temperature of the cooling water for the distillate of column 1 [°C]	25	25	5
Hot stream 2 temperature [°C]	288	333	8
Column 1 reflux flow rate, setpoint [t/h]	7.9	7.9	0.1
Column 1 pressure, setpoint [kPa]	230	265	5
C-2 boiler heating oil flow rate, setpoint [t/h]	0	225	5
Feed flow rate [t/h]	0.9	0.9	0.2
Flow rate to H-4 (cooling after R-2) [t/h]	36	35	10
Hot stream 2 flow rate [t/h]	60	60	10
Feed entering column 1 temperature [°C]	186	235	5
Feed flow rate to column 2 [t/h]	0	1.8	1
Feed 1 flow rate to the process [t/h]	29	0	4
Feed 4 flow rate to the process [t/h]	0	27	4

higher temperature with feed type 4 than with feed type 1. The histories of the IBP and FP during the simulation are shown in Figure 6.1 where the change between the operating regions starts at simulation step 17000.

The first evaluation data set consists of experiments similar to those made in the first simulation with the variables listed in Table 6.3 using feed type 4 and a feed type change to feed type 1. The higher level controller, NAPCON, was active controlling

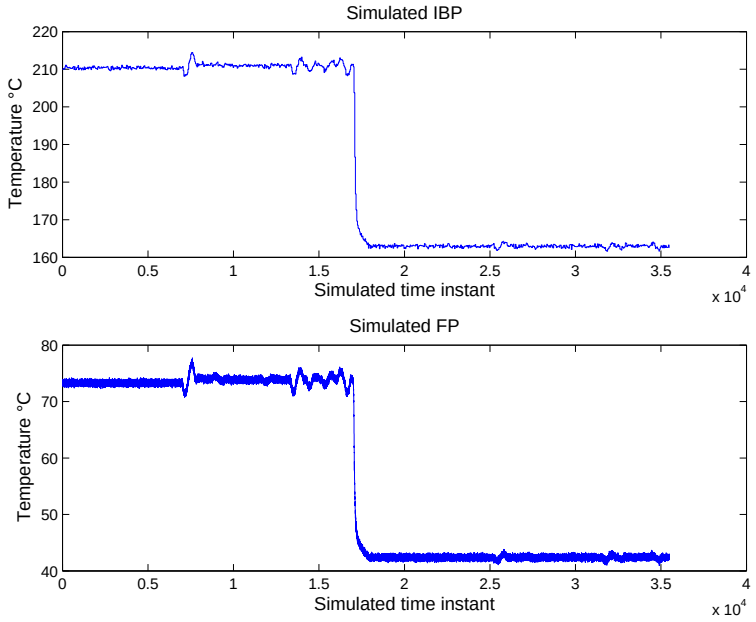

Figure 6.1: IBP and FP of the simulated data set used for training the FDI models

the IBP of the bottom product. Two types of faults were simulated to lower the analysis result for the bottom product IBP value. The first fault type consists of gradually increasing faults and the second of abrupt faults. During the simulation, both fault types were introduced in 3 different magnitudes. The data set can be used to evaluate the performance of the models with feed type 4 and during a transition between the feed types. The set contains 10850 data samples, corresponding to 7 days, 12 hours and 50 minutes of operating time. The effect of the NAPCON controller on IBP in presence of downward analyser faults is illustrated in Figure 6.2. As the IBP decreases, the controller compensates for this and as a consequence, the real IBP is raised over the desired level meaning give-away in the product quality and decreased profit for the plant.

The second evaluation data set includes experiments in variables listed in Table 6.3 during feed type 1 operation. The NAPCON was on, and the same sequence of faults as in the first evaluation data set were introduced. This data set was used

Table 6.3: Manipulated variables of the second testing experiment simulation, quality control on

Description	Feed 1	Feed 4	Maximum change
Temperature setpoint for the solvent entering the reactor 1 [°C]	140	170	30
Temperature difference across the reactor 1 [°C]	21	21	5
Temperature of solvent feed [°C]	50	2	20
Temperature of the H2 feed [°C]	79	79	20
Flow rate of the cooling water for the distillate of column 1 [t/h]	75	75	5
Temperature of the cooling water for the distillate of column 1 [°C]	25	25	5
Hot stream 2 temperature [°C]	288	333	8
H2 feed flow rate [t/h]	0.9	0.9	0.2
Feed 1 flow rate to the process [t/h]	29	0	4
Feed 4 flow rate to the process [t/h]	0	27	4

for evaluating the models performance within the operating region associated with the feed type 1. The set contains 11000 data samples, corresponding to 7 days, 15 hours and 20 minutes of operating time.

The third evaluation set is otherwise similar to the first evaluation set, except in this set the direction of the faults was different, raising the analyser outputs. The set contains 6360 data samples, corresponding to 4 days and 10 hours of operating time.

Evaluation data set 4 consists of experiments within the operating region associated with feed type 4. The NAPCON was off and no faults were introduced in the analyser outputs during the simulation, as there was no feedback to the process. Instead, the faults in IBP were introduced after the simulation. Variables that were

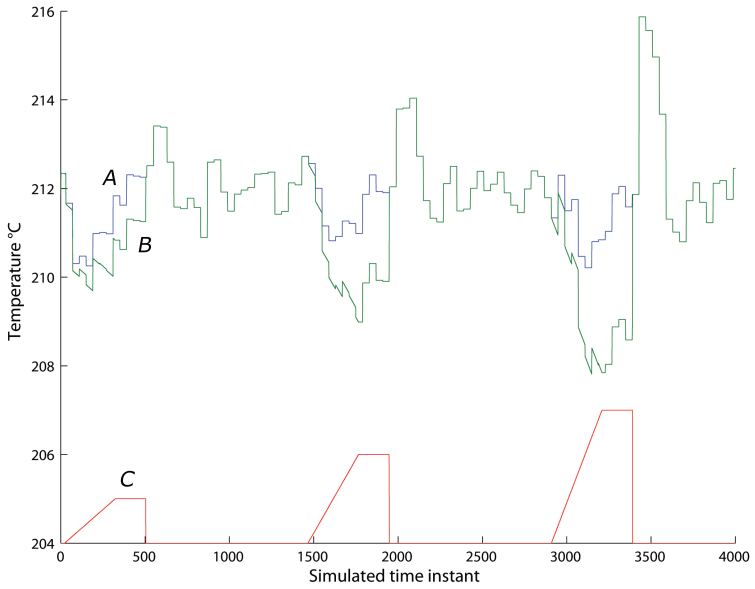

Figure 6.2: Effects of the NAPCON quality control on IBP in the presence of downward faults. True IBP (A), faulty IBP (B) and magnitudes of faults (C)

used to excite the process are listed in Table 6.2. In addition to manipulating these variables, the composition of the feed was altered. The changes introduced in the feed composition were similar to the differences between feed types 1 and 4, only the magnitudes were smaller. This data set is used to evaluate the performance of the models when the quality of the feed is changed. The set contains 2469 data samples, corresponding to 1 day, 17 hours and 9 minutes of operating time.

The fifth evaluation data set was generated while the process was operated with the feed type 4. The NAPCON was used to control the IBP, and both drift and abrupt types of faults were introduced to decrease the analyser outputs. Experiments were made in a random sequence manipulating the variables listed in Table 6.3. The magnitudes of the changes and also the delays between experiments were random. The data set is used to evaluate the performance of the models in a more realistic operation situation, when changes are made more frequently than in the evaluation data sets 1 to 4. The set contains 8553 data samples, corresponding to 5 days, 22

hours, and 33 minutes of operating time.

The sixth evaluation data set was simulated with feed type 1. The NAPCON was inactive and no faults were introduced. The variables that were changed are listed in the Table 5. The set contains 17079 data samples, corresponding to 11 days, 20 hours, and 39 minutes of operating time.

The main characteristics of the data sets are summarized in the Table 6.4. For the multiple operating region case the described training data set was used as such for creating the models and the seven data sets describing different operating conditions were used for model evaluation.

Table 6.4: Characteristics of the simulated data sets

Data set	NAPCON	Faults	Feed type	Feed type change	Changes to variables
Training	Off	NO	Feed 4, Feed 1	YES	Table 6.2
Evaluation 1	On	YES, down	Feed 4	YES	Table 6.3
Evaluation 2	On	YES, down	Feed 1	YES	Table 6.3
Evaluation 3	On	YES, up	Feed 4	NO	Table 6.3
Evaluation 4	Off	NO	Feed 4	NO	Table 6.2 and feed composition
Evaluation 5	On	YES, down	Feed 4	NO	Table 6.3, random order
Evaluation 6	Off	NO	Feed 1	NO	Table 6.2

6.2.2 Analysis and preparation of the real process history data

The data for training the models was acquired from the Naantali oil refinery and it covered the operation of the dearomatisation process during a period of 4 months

and 1 week (1.12.2006 - 5.4.2007). The measurement values were 1 minute samples except the analyser results, which had a sampling frequency of about 2 minutes in the case of FP and about 40 minutes in the case of IBP. During the data collection period, 7 types of feed were used. The operation conditions for the different feed types were significantly different, as indicated by the significantly high variation in IBP (77 °C) and FP (62 °C) between the lightest and heaviest feed stocks. Some key figures of the data are summarized in Table 6.5. It is noted that a bias has been added to the presented values of IBP and FP as it is a policy of the Naantali refinery that no real production data is published. Nonetheless, the differences between the modified values corresponding to the different feed stocks are real. The frequent changes in the feed stock are noted and special care needs to be taken to handle these situations within the FDI system. The values of FP and IBP during the data collection period set are shown in Figure 6.3.

Table 6.5: Key figures of the real process data used for training the FDI models

Feed type	Mean IBP (°C)	Mean FP (°C)	Number of runs	Average length of a run (min)	Percentage of total period (%)
1	-38.16	-20.89	5	7137	18.5
2	-10.24	0.82	8	4899	20.3
3	5.08	16.65	2	7362	7.6
4	37.52	39.78	4	7549	15.7
5	34.25	1.03	5	9540	24.7
9	38.82	40.72	2	8526	8.8
11	-5.50	10.73	2	4191	4.3

The raw measurement data set contains sections that represent abnormal operation of the process or the analysers. These sections must not be used in the modelling and have been removed. The removal of the erroneous samples was feasible as the large

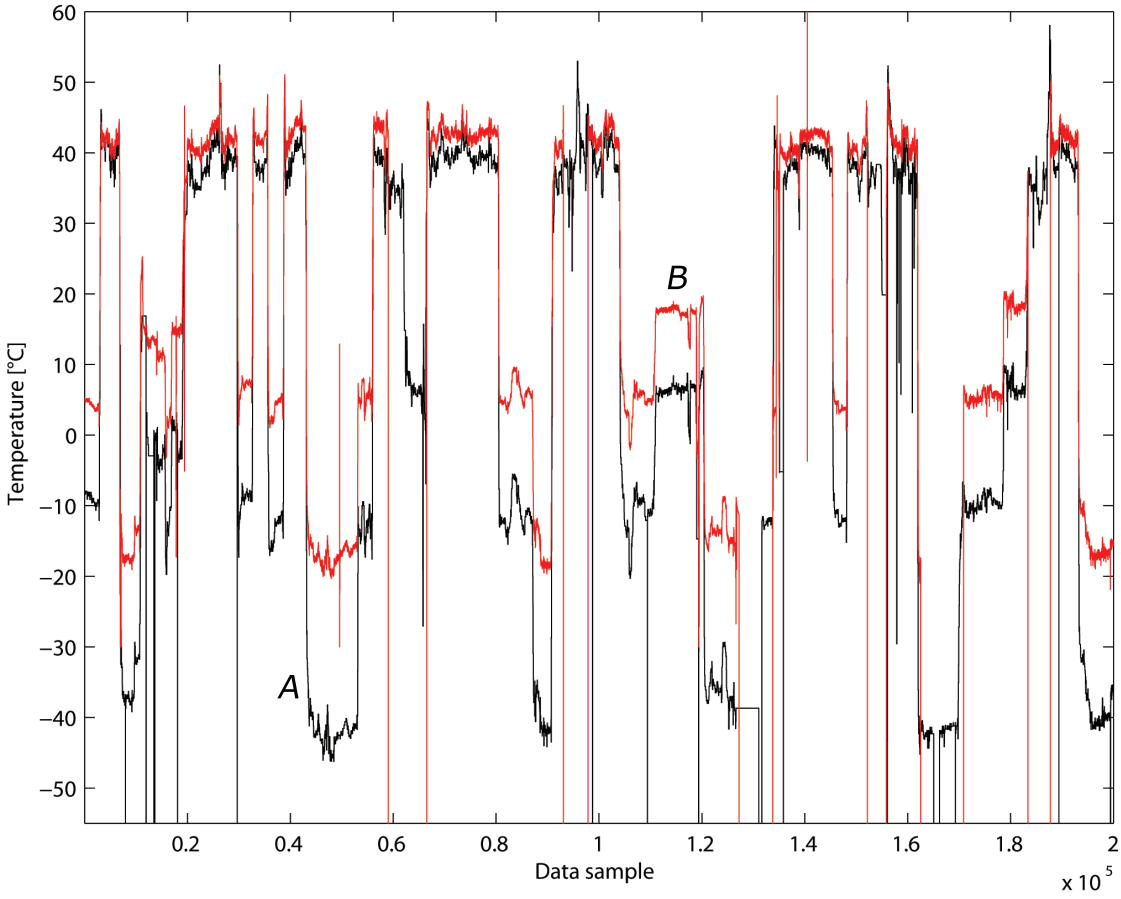

Figure 6.3: Modified IBP (A) and FP (B) temperatures during the data collection period

number of data samples allowed the removal of unsuitable periods of data without introducing problems that come from using too small data sets in modelling. The majority of removed data corresponded to periods when the distillation analyser had been switched off. For these long periods, the median filter (3.1) is unsuitable because the length of the filtering window would have to bee too long. Instead, the outliers were removed manually. The minimum value for the IBP was set in the DCS to be well below the limits for normal operation and thus the low valued outliers were easy to recognise and remove. In addition to the sections during which the analysers had been off, it was noted that during certain periods, e.g. around data sample 160000 the analyser values fluctuated vigorously. These periods of abnormal

process behaviour were examined in detail and certain periods listed in Table 6.6 were discarded from the modelling data.

Table 6.6: Data sections of the real process training data corresponding to abnormal operation of the distillation analyser

Removed data samples	Reason for removal
7960-8010	Calibration of the analyser
11250-13700	Analyser not in use
29550-29750	Fault in the analyser
65740-68560	Fault in the analyser
72000-79000	Fluctuation in analyser results
92500-99000	Fluctuation in analyser results
109000-110000	Analyser off-line and a possible calibration
118900-119500	Analyser off-line and a possible calibration
126500-136500	Analyser off-line and a possible calibration
153500-156500	Analyser off-line and a possible calibration
157500-159000	Outlier values in analyser outputs
160500-161500	Outlier values in analyser outputs
164000-166500	Fluctuation in analyser results
169000-169500	Possible calibration of the analyser
189350-189500	Possible calibration of the analyser
193000-195500	Unsteady process state after feed stock change
223000-224000	Fault in the analyser
228500-229000	Fault in the analyser

Also the flashpoint analyser had some clear outlier values, e.g. around data samples 50000 and 140000. The periods during which the analyser had been used with another process were also clear, as then the analyser values were equal to zero. The data sections with outlier values have been removed.

Next, the data was analysed to find possible periods of faulty analyser operation. As described in section 5.2, the analysers are mainly troubled with two different types of faults, abruptly occurring contamination of the sample and slowly occurring fouling of the flask. The water contamination of the sample typically causes the initial boiling point temperature to drop abruptly while the other distillation temperatures remain at the normal values. The phenomenon is illustrated in Figure 6.4 showing the abnormal behaviour of the IBP marked with tan colour (the line with lowest values). Several instances of this fault type were discovered and the corresponding data sections were removed. The second fault type occurs more infrequently in the analysers of the dearomatisation process and no clear indication of the carbonisation of the flask could be found when the data was analysed.

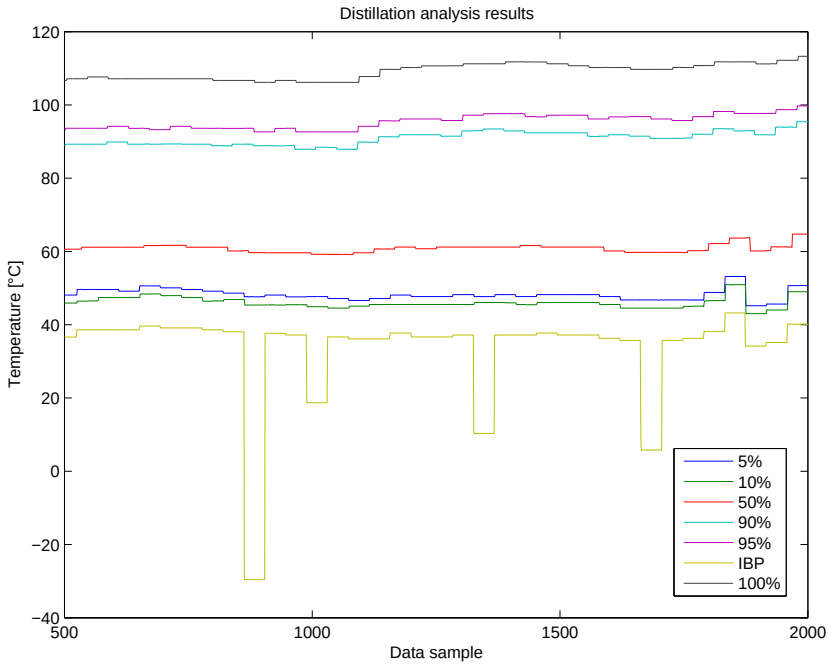

Figure 6.4: Examples of distillation analysis results when samples have been contaminated with water

A visual inspection of the analyser values revealed occasional arbitrary changes of few degrees Celcius in the analyser values. These changes were caused by calibration of the analysers. The calibrations are done after the weekly laboratory analysis if

there is a difference of over 4 °C between the distillation analyser and laboratory results or over 2 °C deviation with the flash point analyser. The frequent calibration of the distillation analyser is required since the outputs are biased and the values of the biases have nonlinear dependence with the distillation temperatures. Thus, the analyser outputs are unbiased only for one operating region at a time. The calibration is done separately for each boiling point temperature and in most cases only one or two distillation points have been calibrated simultaneously. It was noted that the laboratory analysis results that were used for calibrating the online analysers, were also subject to errors. In consequence, the biases in the analyser outputs and the calibrations need to be considered factors that limit the best achievable modelling accuracy.

Industrial data always includes noise and other components that are not explained by the available measurements. In the case of dearomatisation process, calibrations are one type of unmodellable events that affect the outputs of the analysers. Other typical factors reducing the accuracy of the models include the variation i.e. noise of the analyser results, changes in the composition of the feed and the events taking place in other unit processes that affect the dearomatisation process e.g. through energy integration. The effects of these anomalies on modelling, the integrity of the data set was analysed as described in section 3.3.3. First, the noise amplitudes were estimated. For the analysers, the noise estimation was challenging due to two reasons. First, during the long delays between the IBP results the product quality may have changed significantly and thus the differences between the successive results are not caused only by noise. Second, the amplitude of noise in the flashpoint analyser is not constant, but depends on the operating region. Typical sequences of IBP and FP results and the corresponding noise components have been illustrated in Figure 6.5. For FP, the minimum noise amplitude was found to be about 0.5 °C and the maximum value was about 1.5 °C. For convenience, a constant value of 1.0 °C was used in data quality analysis. For IBP, the amplitude of noise could only be approximated during longer periods when the process was almost at steady state.

The value for IBP noise amplitude in the data quality analysis was estimated to be 1.0 °C.

The consistency of the data set, using all input variables, was analysed with SOM as described in the methodology. The consistency values for the real process data are given in Table 6.7. For comparison, the corresponding values for the more consistent simulated training data are also shown.

Table 6.7: Data integrity index values of real and simulated data sets

	Real plant data		Simulated data	
	IBP	FP	IBP	FP
max inconsistency	89.712	24.650	11.434	1.871
95% inconsistency	1.8514	0.8659	0.6152	0.5936
average inconsistency	0.6581	0.2672	0.2303	0.2440

Next, the data was time-shifted to compensate for the process delays as described in the methodology (section 3.3.1). The data set used in the delay compensation was chosen to present normal operation conditions and to include two feed stock changes. The set consisted of 20 000 data samples covering a period of almost two weeks. The determined delays of all variables relevant to modelling are presented in Table 6.8. The estimated delays were used to time-shift the measurement data.

Next, process knowledge was used to generate calculated variables describing process phenomena as described in section 3.3.2. The dominating unit of the process is the main distillation column and thus the efforts for creating calculated variables were directed to variables describing the state of the distillation column. Under normal operating conditions, the temperatures at different plates of the column are mostly determined by the characteristics of the feed type. This dependence of the unmeasured feed quality makes the temperature measurements less useful

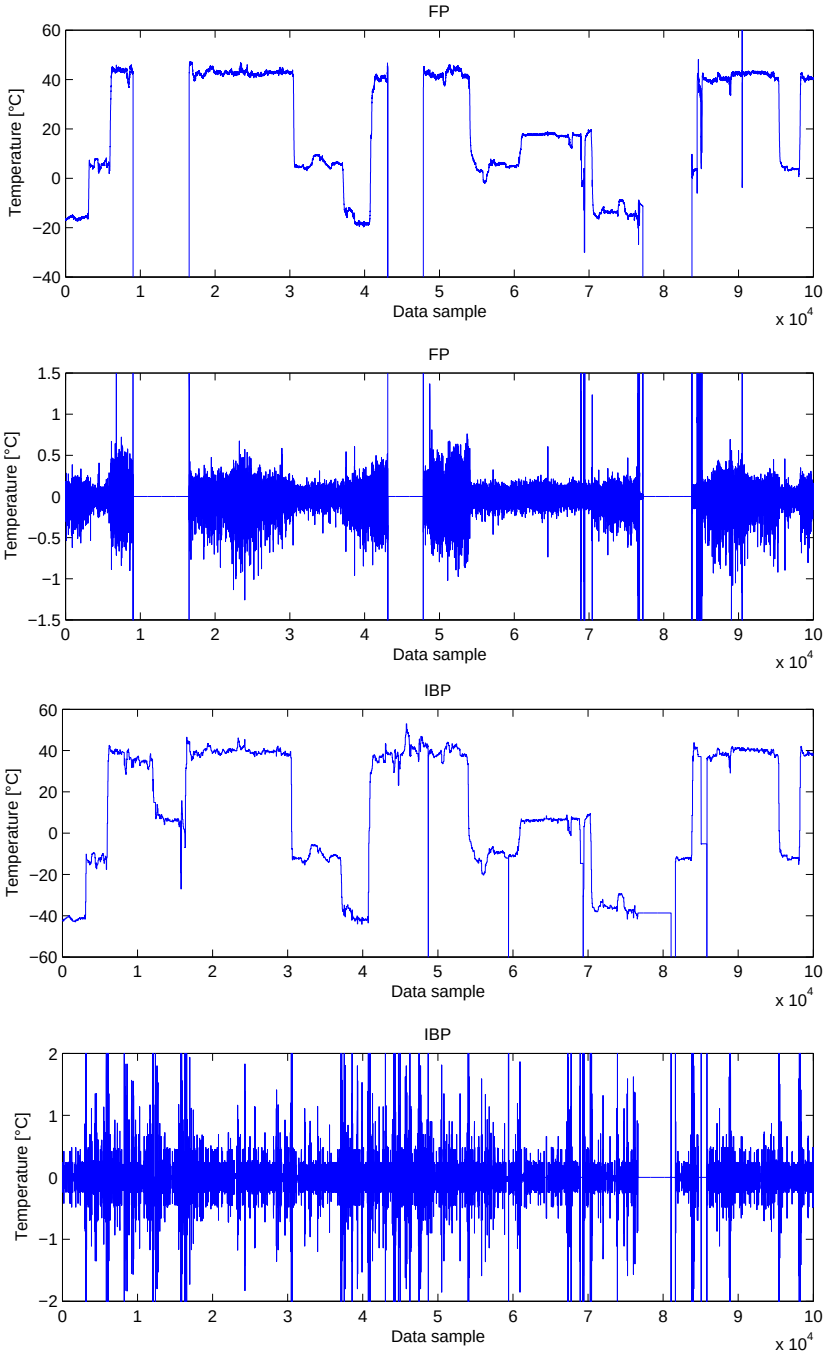

Figure 6.5: FP (top 2 figures) and IBP (bottom 2 figures) analyser outputs for estimating noise amplitude and the corresponding sequences with moving average of 11 samples removed

Table 6.8: Estimated delays between process measurements and analyser outputs of the real dearomatisation process

Variable	Delay to analyser outputs (min)	
	IBP	FP
Side product flow rate	80	74
Column 2 reboiler oil flow rate	30	35
Column 1 top pressure	63	59
Column 2 top pressure	41	42
Column 2 bottom pressure	41	38
Column 1 feed temperature	68	54
Column 1 lowest middle temperature	52	46
Column 1 lower top temperature	2	0
Column 1 upper middle temperature	56	44
Column 1 lower middle temperature	65	57
Column 1 upper bottom temperature	52	45
Column 1 reboiler temperature	46	42
Column 1 recycle flow rate	85	52
Column 1 reboiler hot oil flow rate	60	43
Column 1 lowest middle temperature, pressure compensated	53	47

in the fault detection. The differences between the temperatures at different plates are, however, less affected by the feed composition, and offer thus useful information about the status of the process in all operation conditions. These calculated variables are shown in Table 6.9 as Group 1. The second major group of calculated variables are ratios of process flows around the distillation column and the fractions of the feed they represent. The absolute flow rates depend on the feed rate and thus correlate strongly with the operating region. The ratios between the flows depend less on the operating region and are thus more useful quantities in abnormal situation detection. The constructed calculated variables describing the ratios of flow rates are also presented in Table 6.9. This kind of flow rate ratio calculated variable is also used as a controlled variable in NAPCON quality control.

Table 6.9: Calculated variables of the dearomatisation process

Group 1: Temperature differences
Over distillation column 1 (DC1)
In the bottom part of DC1
Bottom part temperature - pressure corrected middle temp of DC1
Lower middle part temperature - pressure corrected middle temp of DC1
Over middle part of DC1
Middle - top part of DC1
In the top part of DC1
Group 2: Ratios of flows
AUX Distillate / Bottom prod flow
AUX Distillate / Reflux
AUX Distillate / Column feed
AUX Reflux / Column feed
AUX Reflux / Bottom prod flow
AUX Bottom prod flow / Column feed

The final phase in the preparation of the data was the determination of the training and evaluation data sets. During the data collection period there were no major changes in the instrumentation of the target process. For this reason, it was not necessary to construct the data sets using randomly chosen sequences as stated in the methodology, section 3.4.2. Training data of the real process was selected to include 150000 data samples from the beginning of the data collection period of which the bad sections were then removed (Table 6.6). The dimension of the data set was further reduced by removing all data samples for which there was no new analyser result available. In the case of distillation analyser, new values are received about once in every 40 minute. Thus the final training data set for the distillation analyser was 2586 samples and the data set used for model evaluation consisted of

873 samples. In the flash point analyser case, the dimension of the data was not reduced and the training data set consisted of 119360 and the evaluation data set of 41470 samples. The values of FP and IBP for the training and the evaluation sets are shown in Figure 6.6. The training data contained all 7 feed types while the data set used for evaluating the models included 5 most common feed types while two of the rarer feed stock types were not represented.

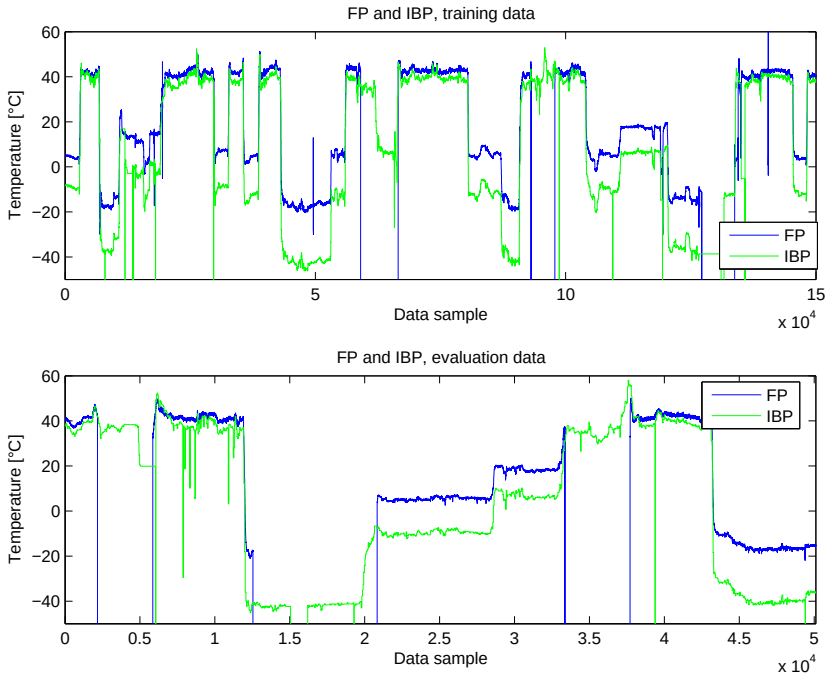

Figure 6.6: FP and IBP of real process data used for training (top) and evaluating (bottom) the models

7 Construction of the data-based monitoring models for detecting analyser faults in the dearomatisation process

Next the models for detecting faults in the online analysers of the dearomatisation process are created. To test the methodology, several testing experiments are made. The first experiment is performed with simulated data covering process operation in a single region. The second experiment utilises simulated data covering multiple operating regions. The third experiment is performed with real process measurement data. Separate models are created for each experiment. The estimation accuracies of the built data based models are evaluated and their suitability for detecting the analyser faults is assessed. Mathematical descriptions of the utilised modelling methods are presented in Appendix 1.

7.1 FDI models for the first testing experiment

The first simulation case consisted of training of four different monitoring methods and studying their suitability for detecting the analyser faults within one operating region of the dearomatisation process (Vermasvuori *et al.*, 2005). The tested modelling techniques were PCA, PLS, SMI and SOM. Two PCA models, one for each analyser, were constructed using process measurements and the corresponding analyser output as input variables. Another two models were created including also calculated variables. All PCA models were constructed of four PCs explaining about 97 % of the total variance of the data.

Two PLS models were constructed for both analysers, one using the direct measurements as inputs and the other also using the calculated variables. The outputs were

the corresponding analyser results. The latent variables of the PLS were calculated with the NIPALS algorithm (Wold, 1975). The PLS models were created with the five latent variables that captured about 95 % of the variance of the input data and 82 % of the output data.

Subspace identified state-space models were created so that the maximum dimension of the system matrices was set to five, and in the PCA data reduction phase three PCs were used. Again, two models were created for each analyser, one with the calculated variables and one without.

In the study, two separate types of SOM were created. Both sets consisted of four models; one without and the other one with the calculated variables for both analysers. The models in the first set were trained with the original autoscaled data including the analyser output. The second set of SOMs was trained with analyser results and PCA transformed data. After autoscaling, PCA was performed and four PCs explaining about 97 % of the variation were selected and the data were then projected into the PC space and used to train the second set of SOMs. During the testing, the analyser output was not used in determining the best matching neuron. Instead the analyser output value related with the neuron was the output of the model.

7.2 FDI models for the second testing experiment

The modelling methods selected in section 6.1 for the multiple operating region case study were: PLS, MLP, SMI, SOM, PLS-MLP, and SMI-MLP were chosen to be implemented. Each of these model types has at least one tuning parameter (listed in Table 7.1) that needs to be set before training the models. The optimal values of the tuning parameters were determined by minimising the RMSE of the models

(7.1).

$$J_{RMSE} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(\hat{y}_i - y_i)^2} \tag{7.1}$$

where $\hat{y}$ is the estimated value for the output, y is the real output and n is the number of analysed samples.

Table 7.1: Tuning parameters of the models of the second testing experiment.

Modelling method	Tuning parameter
PLS	Number of latent variables
MLP	Number of neurons in the hidden layer
SMI	Number of states
SOM	Number of neurons

The effect of the number of LVs on the performance of a PLS model is illustrated in Figure 7.1. It was noted that for the FP model, the lowest average RMSE values for the evaluation sets one to six were achieved with models having two, four, five and six LVs. Several models had almost equal performance and the one with fewest LVs should have been chosen. However, the model with four LVs had almost as good RMSE and significantly lower maximum estimation error than the model with two LVs. Thus, instead of choosing the best PLS model purely based on the optimisation criterion (RMSE), the one with four LVs was decided to be the best one. Also for IBP, the model trained with four LVs was the best.

The MLP models had one hidden layer and tansig activation functions. The training was performed with the Levenberg-Marquardt training algorithm (A.27) with 50 epochs. The performance values of the MLP models with different number of neurons in the hidden layer are presented in Figure 7.2. The results reveal the characteristic property of ANNs, i.e. the overfitting of the training data and poor prediction capabilities of evaluation data sets when the number of hidden neurons is too high. For both the FP and IBP the best models for estimating the analyser

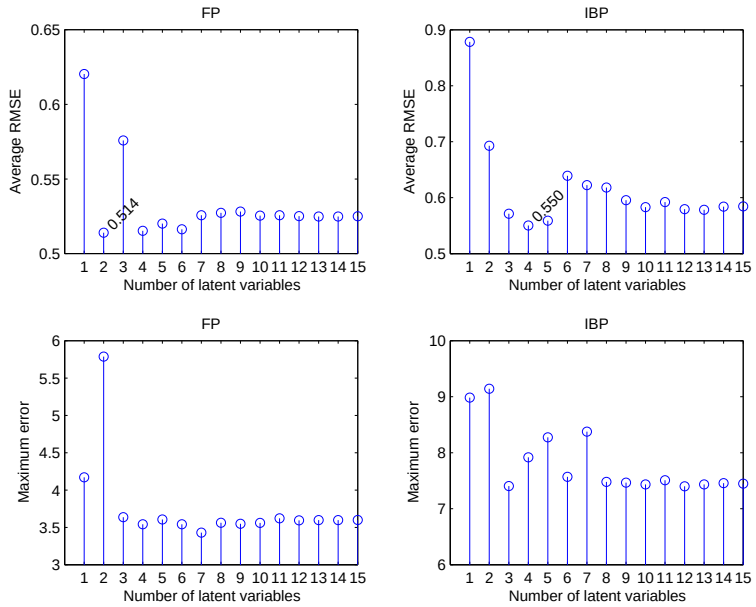

Figure 7.1: Average RMSEs using the evaluation data sets 1-6 (top) and the maximum estimation error of PLS models for FP (bottom left) and IBP (bottom right) with different number of latent variables

results with the evaluation data sets were the models trained with only one neuron in the hidden layer.

With SMI models, the number of available states had only minor effect on the estimation capabilities of the models. The best model for FP was found to be the one with three states, having the lowest RMSE and only moderately higher maximum error than the model with two states. For IBP the best results were achieved with a model with eight states. The performance values of SMI models with different number of states are shown in Figure 7.3.

The tuning parameter of SOMs is the number of neurons i.e. the size of the map. Another factor affecting the performance of the models is how the neurons are organised. In this study the neurons of all SOMs were organised in hexagonal grid with equal number of rows and columns. The best performing SOM for estimating

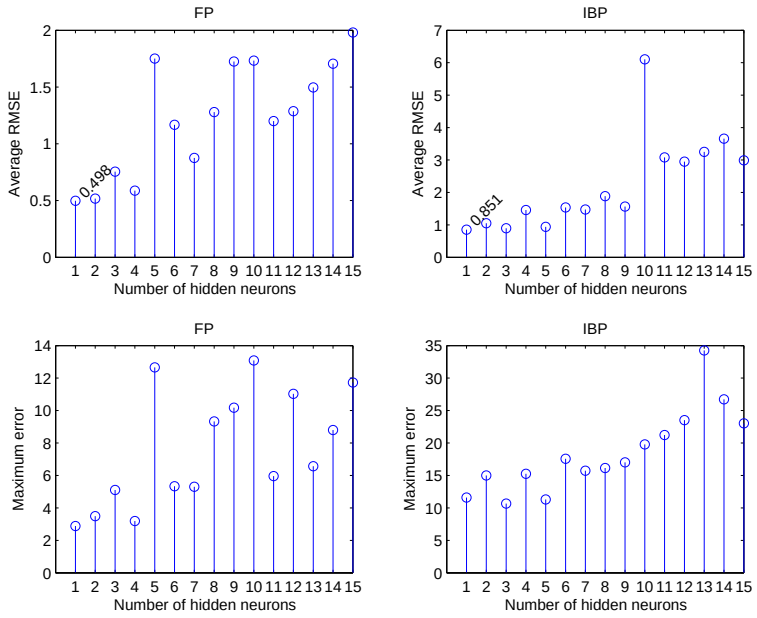

Figure 7.2: Average RMSEs using the evaluation data sets 1-6 (top) and the maximum estimation error of MLP models for FP (bottom left) and IBP (bottom right) with different number of hidden neurons

FP was the model with 100 neurons. It had the lowest average RMSE and almost as small a maximum error as the model with 64 neurons. Most accurate estimations for IBP were given by the model with 121 neurons. The performance values of the different sized SOMs are given in Figure 7.4.

The combination models PLS-MLP and SMI-MLP were shown to have optimal performance when the MLP part had only a few neurons in the hidden layer. The PLS-MLP models with the lowest RMSEs were obtained by combining the best PLS model for FP with an MLP having 2 hidden neurons and the best PLS model for IBP with an MLP having 1 hidden neuron. The average RMSEs of the combination models for the evaluation data sets were lower than those of the pure PLS models. In contrast, augmenting the best SMI models with an MLP part always resulted in worse estimation accuracy than what was achieved with the pure SMI models alone. An MLP with one hidden neuron had smallest impact on the performance of the

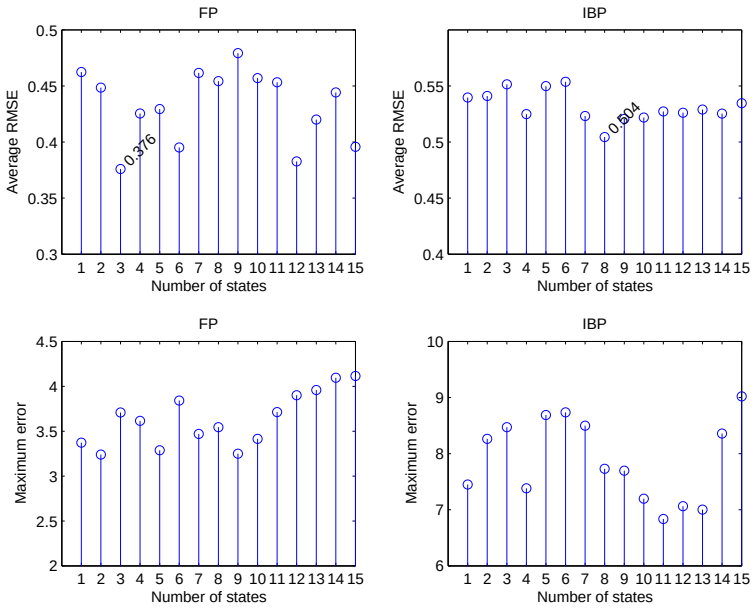

Figure 7.3: Average RMSEs using the evaluation data sets 1-6 (top) and the maximum estimation error of SMI models for FP (bottom left) and IBP (bottom right) with different number of states

SMI models for FP and IBP and the resulting combination models were almost as good as the pure SMI models. The results of the combination models with the MLP models having different number of hidden neurons are shown in Figures 7.5 and 7.6.

During the search for the best values of the tuning parameters in model creation, the input variable sets were optimised separately for each of the models. First, the wrapper type IVS algorithm based on the Forward Selection (FS) method (section 3.4.1) was utilised after which the variable sets are further optimised with GA. The GA was set to use 50 generations, as it was noted that no sets with significant improvements in the modelling performance were found with longer searches. In GA, the mating probability was set to 0.7 and the mutation probability to 0.1. The input variables selected for different model types by the modified FS and GA are listed in Table 7.2. It is noted that the different methods utilised different number of

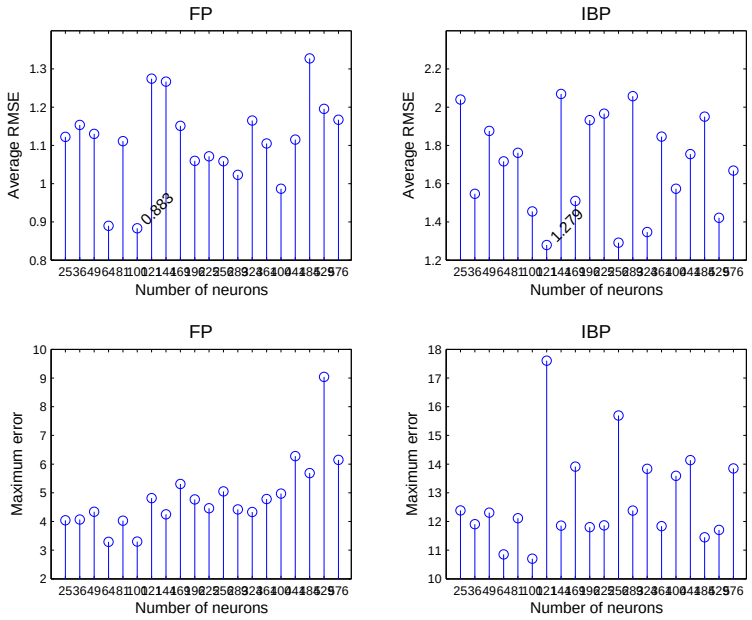

Figure 7.4: Average RMSEs using the evaluation data sets 1-6 (top) and the maximum estimation error of SOM models for FP (bottom left) and IBP (bottom right) with different number of neurons

variables. PLS and SOM models used only 5 to 6 variables while the other extreme, the MLP models, used 15 to 20 variables. The SMI model for FP with three states used only 6 variables while the IBP model with eight states needed 19 variables. The combination models used all the variables of the PLS and SMI models and additional variables for the MLP parts. The SMI-MLP for IBP was the only model utilising all the 22 variables that were found useful with the correlation analysis.

The RMSE values i.e. the values of the minimised objective function corresponding to the best models of each model type are summarised in Table 7.3. It is noted that with all methods the RMSE for the training data set of the FP models were higher than of the IBP models. This is mainly due to the higher number of samples in the FP data set compared to the IBP data set. With the evaluation data sets the opposite is true, the estimates for the FP were more accurate that the estimates for the IBP, although the magnitudes of the RMSEs were similar for both modelled out-

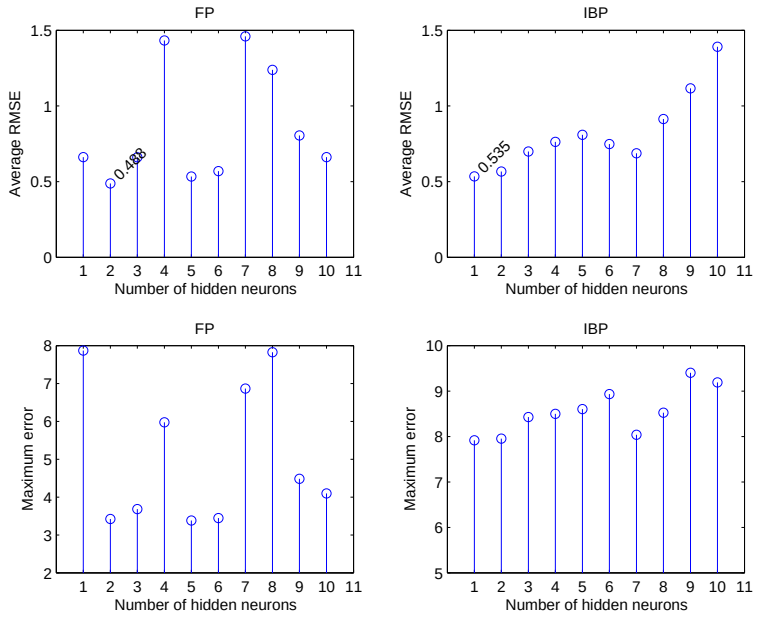

Figure 7.5: Average RMSEs using the evaluation data sets 1-6 (top) and the maximum estimation error of PLS-MLP models for FP (bottom left) and IBP (bottom right) with 4 latent variables (PLS) and different number of neurons (MLP)

puts. The most limiting factor in modelling the IBP was the signal's discontinuous nature. Between the tested modelling methods, the SMI produced most accurate models both for the FP and IBP and almost as good was the SMI-MLP combination. The PLS produced the third best models, the FP and IBP models having very similar accuracy. The MLP model for the FP was better than the corresponding PLS model, but in the IBP case the MLP model was significantly worse. The logical explanation for this is the lower number of samples in the IBP training set allowing the ANN model to become overfitted even with one neuron in the hidden layer. The least accurate of the models were the SOMs. The method was able to fit the training data, but the estimation accuracy with the evaluation data sets was much worse. Possible explanation for this is the lack of extrapolating ability of the SOM; all situations that will be encountered when using the model, should be included in the training data set. Another weakness of the SOMs is the limited number of possible outputs, a SOM with 100 neurons has only 100 possible output values.

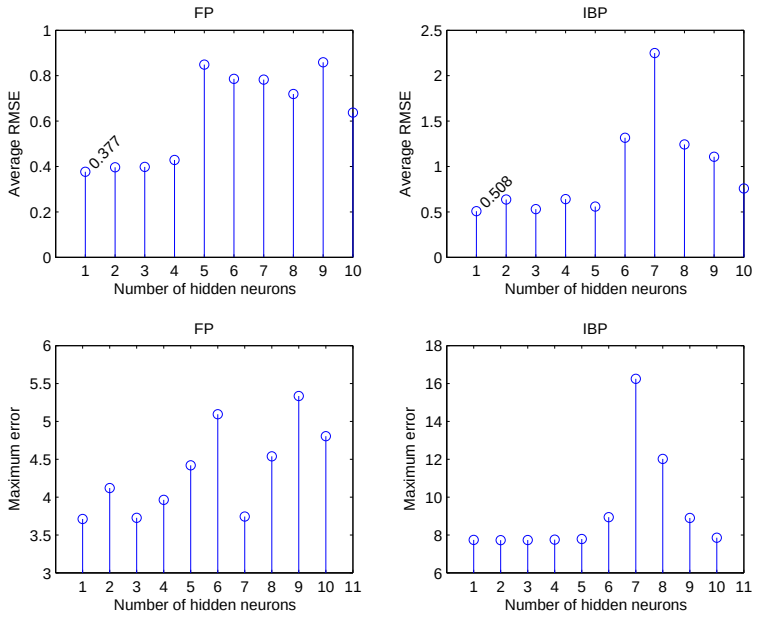

Figure 7.6: Average RMSEs using the evaluation data sets 1-6 (top) and the maximum estimation error of SMI-MLP models for FP (bottom left) and IBP (bottom right) with 3 states (SMI) and different number of neurons (MLP)

The estimation accuracies and the distributions of the errors of the best models of each model type for the training data set and all evaluation data sets were analysed. As indicated by the average RMSE values, the most accurate estimations for FP were provided by the SMI model, only with evaluation data set 3 the MLP produced more accurate estimations. The MLP and PLS models offered acceptable performance, but in many cases the SOM was unable to model the behaviour of the FP. The SOMs estimates were limited to the maximum value that was presented in the training data and with evaluation data set 1, the model was unable to give accurate estimates for the high values of FP.

For IBP, the most accurate estimates were given by the PLS-MLP model for the training data and evaluation data set 2, 3, and 5. Normal PLS model performed best with evaluation data sets 1 and 4 and the SMI model with evaluation data set 6. However, the differences between the model types were small and SMI, SMI-MLP,

Table 7.2: Input variable sets for FP and IBP models trained with simulated data of the second testing experiment

	PLS FP (2 LVs)	PLS IBP (4 LVs)	MLP FP (1 neuron)	MLP IBP (1 neuron)	SMI FP (3 states)	SMI IBP (8 states)	SOM FP (10x10 size)	SOM IBP (11x11 size)	PLS-MLP FP (2 LVs - 2 neurons)	PLS-MLP IBP (4 LVs - 1 neuron)	SMI-MLP FP (3 states - 1 neuron)	SMI-MLP IBP (8 states - 1 neuron)
Solvent feed flow rate [t/h]			X	X	X	X			X		X	X
C1 feed flow rate [t/h]			X	X		X			X	X	X	X
C2 feed flow rate [t/h]				X		X				X		X
C1 upper middle temp. [°C]	X		X	X		X	X	X	X	X		X
C1 lower middle temp. [°C]		X	X	X	X	X	X		X	X	X	X
C1 upper bottom temp. [°C]	X		X			X	X	X	X	X		X
C1 lower bottom temp. [°C]	X	X		X	X	X	X		X	X	X	X
C2 top pressure [kPa]		X		X						X		X
C2 bottom pressure [kPa]				X		X				X		X
C1 topmost bottom temp., pressure compensated [°C]	X					X	X	X	X			X
TCA_2			X	X		X		X	X	X	X	X
C2 vapour distillate temp. [°C]			X	X	X	X			X	X	X	X
C1 topmost bottom temp. [°C]		X	X	X				X	X	X	X	X
C1 product after 2 heat exchangers temp. [°C]			X	X		X			X		X	X
C2 product after heat exchanger temp. [°C]	X		X	X		X			X	X	X	X
C1 side product temp. [°C]			X	X		X			X	X		X
C1 feed temp. [°C]			X	X		X			X	X		X
C1 feed before heat exchanger temp. [°C]			X	X					X	X	X	X
C1, lower bottom - upper bottom temp. [°C]			X	X		X		X	X	X	X	X
C1, upper bottom - pressure compensated topmost bottom temp. [°C]			X	X		X			X	X	X	X
C1, pressure compensated topmost bottom - lower middle temp. [°C]		X		X	X	X			X	X	X	X
C1, lower middle - upper middle temp. [°C]	X			X	X	X			X	X	X	X

C1 stands for 'Column 1' and C2 for 'Column 2'

PLS-MLP and PLS had very similar average accuracy. Only MLP and SOM models were significantly more inaccurate.

Table 7.3: RMSE values for training and evaluation data sets one to six of all models in the second testing experiment

	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average
PLS FP	0.370	0.352	1.278	0.066	0.112	0.186	1.099	0.515
MLP FP	0.317	0.421	0.403	1.281	0.051	0.329	0.142	0.498
SMI FP	0.343	0.203	0.852	0.061	0.078	0.083	0.980	0.376
SOM FP	0.498	0.748	2.047	0.169	0.704	0.870	0.762	0.883
PLS-MLP FP	0.365	0.354	1.195	0.070	0.117	0.192	0.998	0.488
SMI-MLP FP	0.344	0.198	0.851	0.065	0.081	0.085	0.980	0.377
PLS IBP	0.273	0.294	0.826	0.086	0.081	0.127	1.888	0.550
MLP IBP	0.269	0.560	1.030	0.087	0.560	0.262	2.606	0.851
SMI IBP	0.269	0.306	0.785	0.092	0.095	0.134	1.614	0.504
SOM IBP	0.371	1.072	2.176	0.259	0.962	1.032	2.172	1.279
PLS-MLP IBP	0.268	0.298	0.751	0.084	0.085	0.124	1.865	0.535
SMI-MLP IBP	0.268	0.322	0.780	0.090	0.102	0.129	1.625	0.508

The data corresponding to the the training data set and the evaluation data set 1 are presented in Figures 7.7 to 7.10. Corresponding figures for other data sets are presented in Appendix B.

In addition to determining the performances of the models with RMSE values, the compositions of the input variable sets were assessed by analysing the dependencies between the estimation errors and the values of estimation and separate input variables. The estimation errors were plotted against the estimated values and the values of each input variable. E.g. the values corresponding to the PLS models for FP and IBP are shown in Figure 7.11. For other models, see Figures in Appendix B. In the figures, the two main operating regions are clearly separated and the connection between the two clusters represents the transition from feed type 4 operation to feed type 1 operation. The estimation errors within the two operating regions had

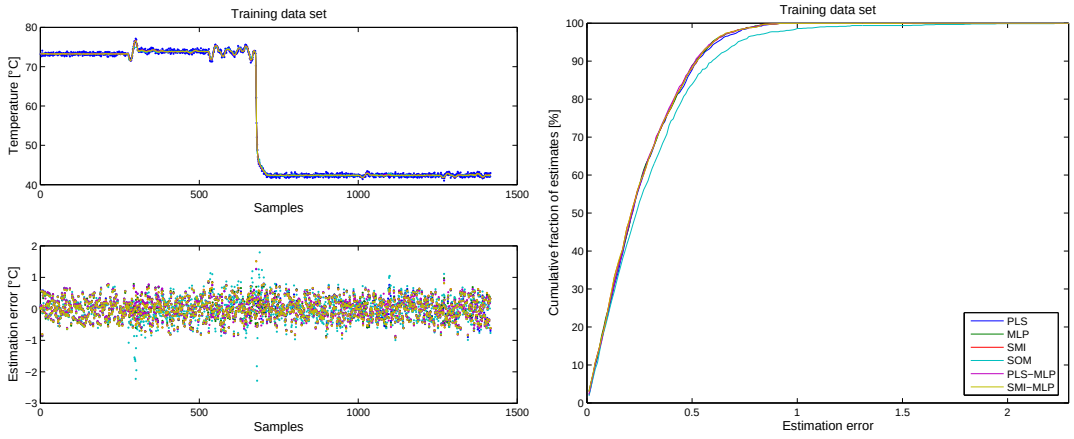

Figure 7.7: Simulated and estimated values of FP for the training data set (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

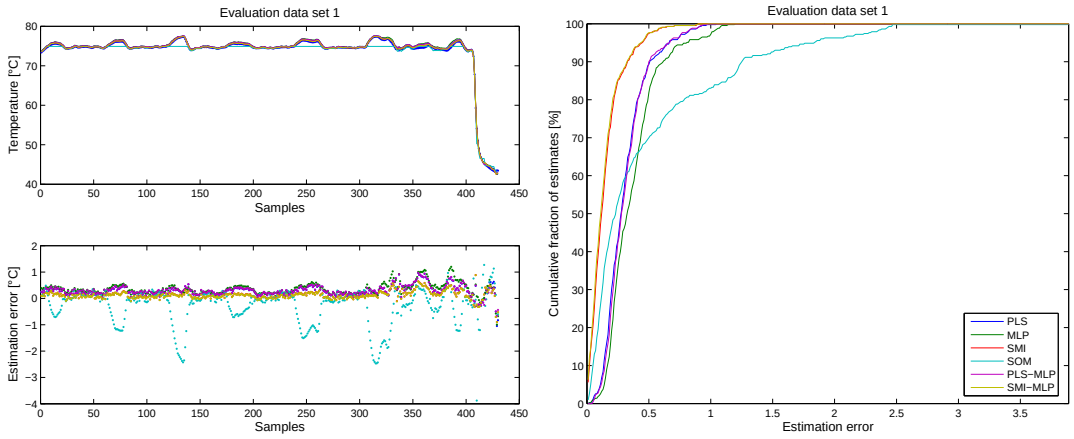

Figure 7.8: Simulated and estimated values of FP for the evaluation data sets 1 (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

near zero averages and were evenly distributed around zero indicating that there was no systematic modelling errors. On the contrary, the estimation error was systematically positive when the estimated values and temperatures of the lower parts of the column were between the typical values for feed types 4 and 1. For other input variables the estimation error was always zero mean. The fact that the estimation

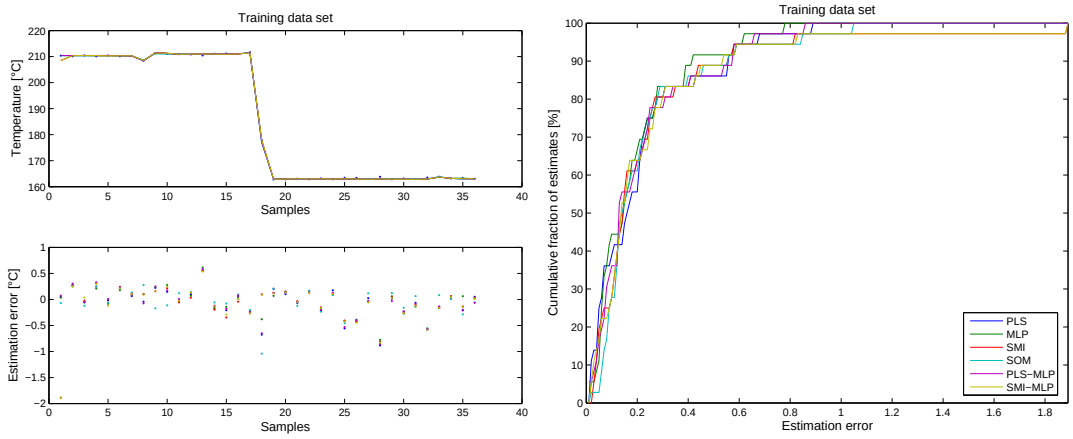

Figure 7.9: Simulated and estimated values of IBP for the training data set (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

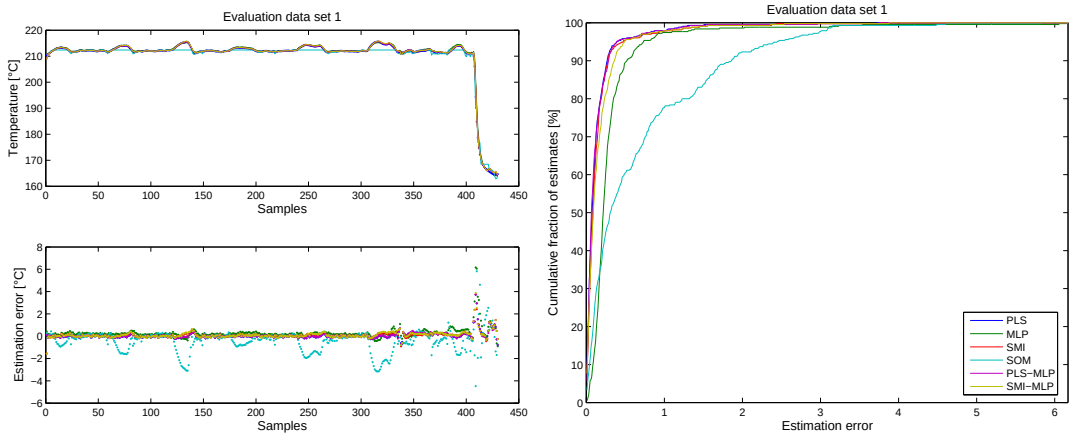

Figure 7.10: Simulated and estimated values of IBP for the evaluation data sets 1 (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

error had similar dependency on the temperatures of the lower parts of the column with both linear and non-linear model types suggests that the phenomenon was not caused by the non-linearities of the process. As the training data only contained a transition from a higher temperature operating region to a lower one, the systematic errors in estimations are assumed to be caused by errors in delay compensation, i.e.

the estimated values for delays between the input variables and the output were too high.

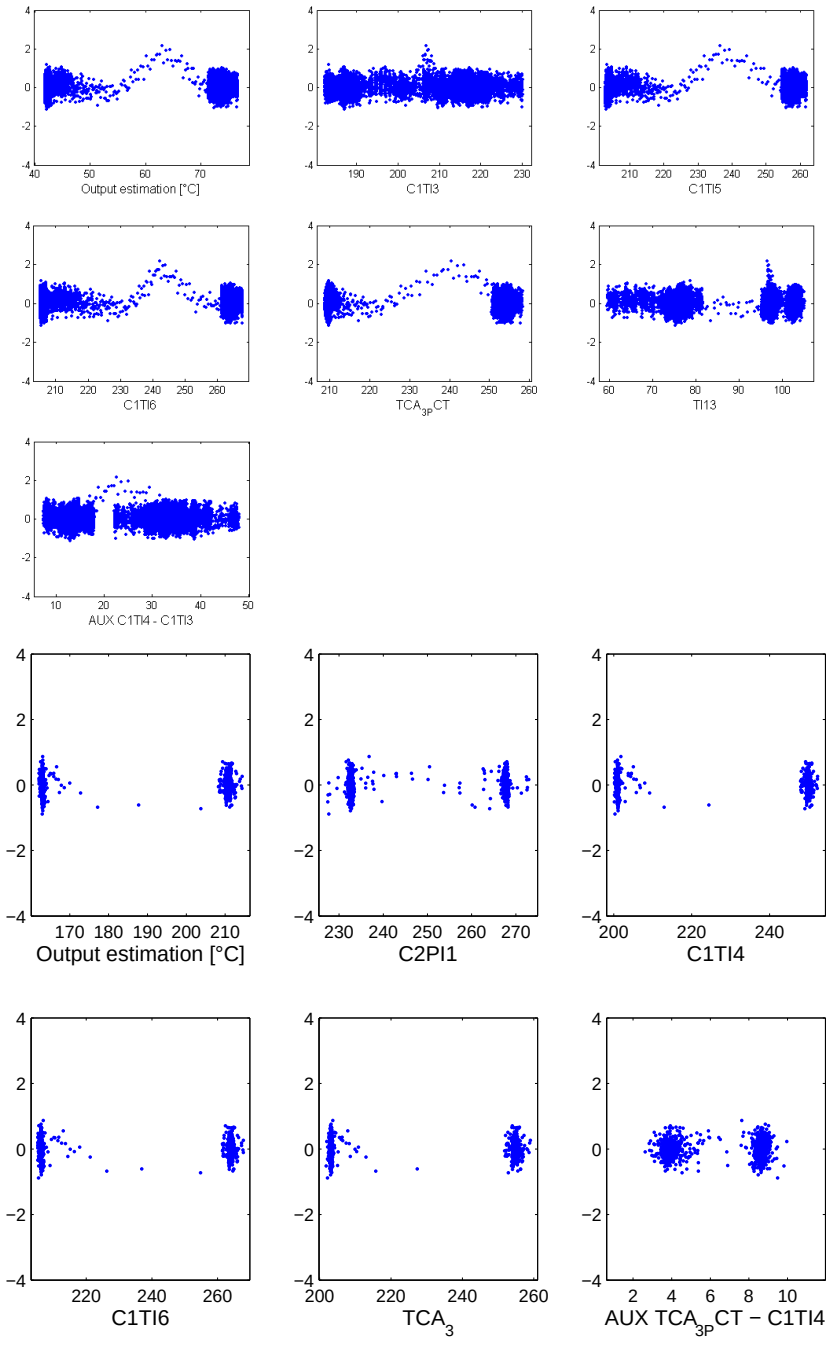

Figure 7.11: Dependence of the estimation errors on the estimation and the input variables, training data, PLS FP model (top) and IBP model (bottom)

A final analysis of the models' validity was carried out by calculating the autocorrelations of the estimation errors. For the training data set all models produced an estimation error signal that had low autocorrelation. Thus the errors at any given time were not related to the previous errors meaning that there was no major systematic errors in the estimates. The systematic errors occurring during the transition phase between the two operation regions presented only a very minor proportion of the whole data and the effect was thus not seen in the autocorrelation analysis.

In general, all the tested modelling methods except SOM, could be used to construct models that provided accurate estimates for the FP and IBP values of the simulated process. The performances of the other methods were on the same level, but the SOM models had significantly worse characteristics with an average RMSE of 0.883 and a maximum evaluation error of 3.3 °C. The small size of the best SOM indicates that the data used for training the models was not rich enough for that modelling method. The average RMSEs for the other model types for the evaluation data sets were 0.515 °C or less and the error signals had near zero means and low autocorrelation values. The analysis of the estimation capabilities of the models showed that it was feasible to use CUSUMs based fault detection methods with tight detection limits.

7.3 FDI models for the third testing experiment with real industrial dearomatisation process data

For the real industrial process, six models selected in section 6.1 were created: PLS models for IBP and FP temperatures, SMI model for IBP, a SMI-MLP model for FP and PLS-MLP models for IBP and FP. The models were trained with the training data set described in Section 6.2.2. The tuning parameters of the models were determined according to the methodology in the same way as in the multiple operation region simulation case study. The input variable selection was performed according to the steps of the methodology. The optimal tuning parameter values and the corresponding RMSE values for the models are given in Table 7.4 and the input variable sets are given in Table 7.5. It is noted that the estimates of the FP models were more accurate than those of the IBP models both for the training and the evaluation data sets. In case of FP, the average estimation error for the evaluation data was 1.8-1.9 °C, about two times as large as the 95 % inconsistency value for the FP data (Table 6.7). For IBP the average estimation error was 3.9-4.1 °C, also about two times as large as the 95 % inconsistency value for the IBP data. Most accurate models for both FP and IBP were the PLS and PLS-MLP, but the results of the SMI and SMI-MLP models were almost as good. The estimations of all models, the estimation errors and the cumulative distribution of the errors are presented for the training and evaluation data sets in Figures 7.12 to 7.15.

Next, the autocorrelation of the estimation errors were analysed. For the training data, the estimations of IBP were not correlated (Figure 7.16). However, the FP estimation errors were not independent and especially in case of the SMI-MLP model, the errors showed some correlation with previous errors. With evaluation data set, the autocorrelations of the estimation errors were more prominent. In case of IBP, all the models showed correlation peaks around samples 280 and 1320. The estimation errors that correlate were produced during similar transition periods

shown in Figure 7.14 (upper left hand corner) at around steps 170 and 690, when the estimation errors were very high for all models (lower left hand corner). In case of FP, the errors were even more correlated. The correlations were caused by the too low estimations of the SMI-MLP model in the middle of the evaluation data set and at the end of the testing period for all models as shown in Figure 7.15.

Table 7.4: Optimal tuning parameter values and corresponding RMSE values of the FP and IBP models trained with real process data

Model	Tuning parameter	Parameter values	RMSE	
			Training data	Evaluation data
PLS FP	No. of LVs	5	2.125	1.833
SMI-MLP FP	No. of states and neurons	2 and 3	2.102	1.920
PLS-MLP FP	No. of LVs and neurons	5 and 4	1.532	1.796
PLS IBP	No. of LVs	3	2.811	3.853
SMI IBP	No. of states	3	3.033	4.123
PLS-MLP IBP	No. of LVs and neurons	5 and 5	2.513	3.885

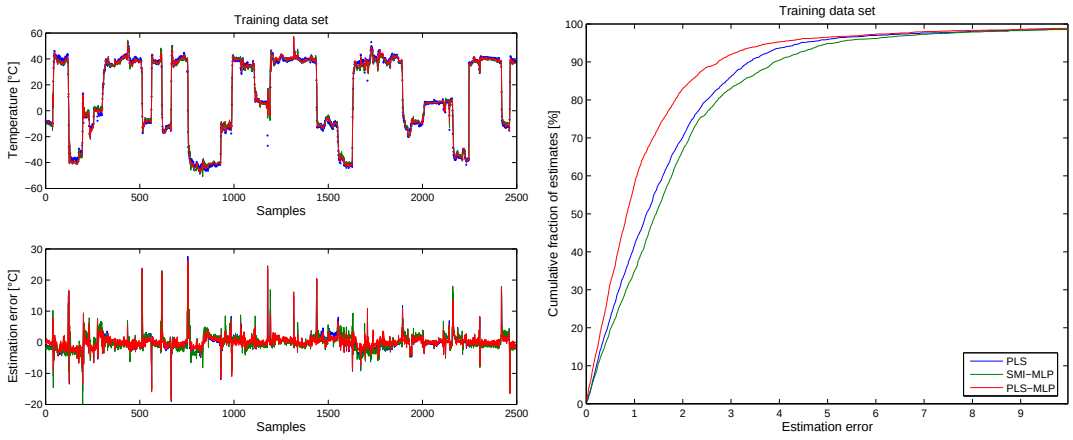

Figure 7.12: Measured and estimated values of IBP for the real process training data set (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

To improve the insensitivity of the models to unexpected disturbances in the measurement data, the estimates were checked against known process behaviour and

Table 7.5: Input variable sets for FP and IBP models trained with real plant data

	PLS FP	PLS IBP	SMI-MLP FP	SMI IBP	PLS-MLP FP	PLS-MLP IBP
Side product flow rate [t/h]			X			
Column 1 reboiler hot oil flow rate [t/h]			X			
Column 1 upper middle temperature [°C]		X				X
Column 1 lower middle temperature [°C]		X		X		X
Column 1 top pressure [kPa]	X	X	X	X	X	X
Column 1 topmost bottom temp., pressure compensated [°C]		X		X		X
Column 1 topmost bottom temperature [°C]	X	X			X	X
Column 1 reboiling temperature [°C]		X		X		X
Product grade	X				X	
Column 1, upper bottom temp. - topmost bottom temp. [°C]			X			
Column 1, topmost bottom temp. - lower middle temp. [°C]	X				X	
Column 1, upper middle temp. - lower top temp. [°C]	X				X	
Column 2 top pressure [kPa]				X		

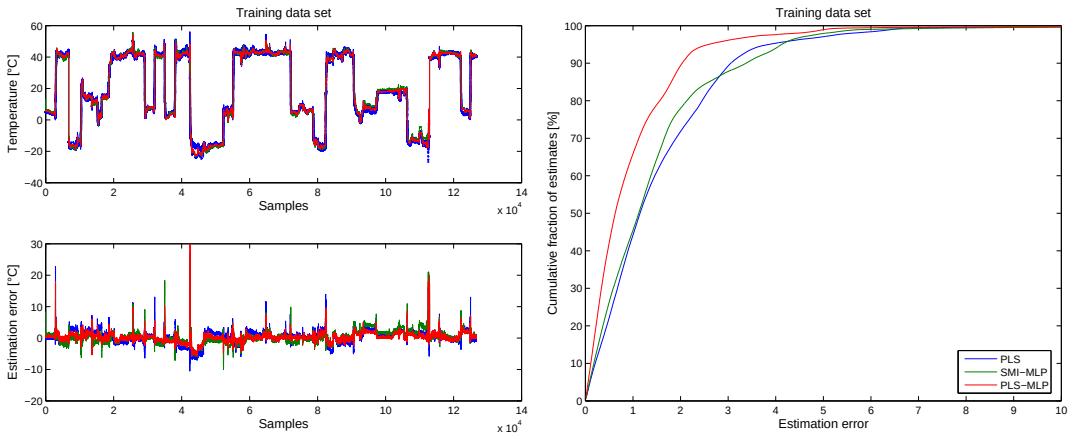

Figure 7.13: Measured and estimated values of FP for the real process training data set (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

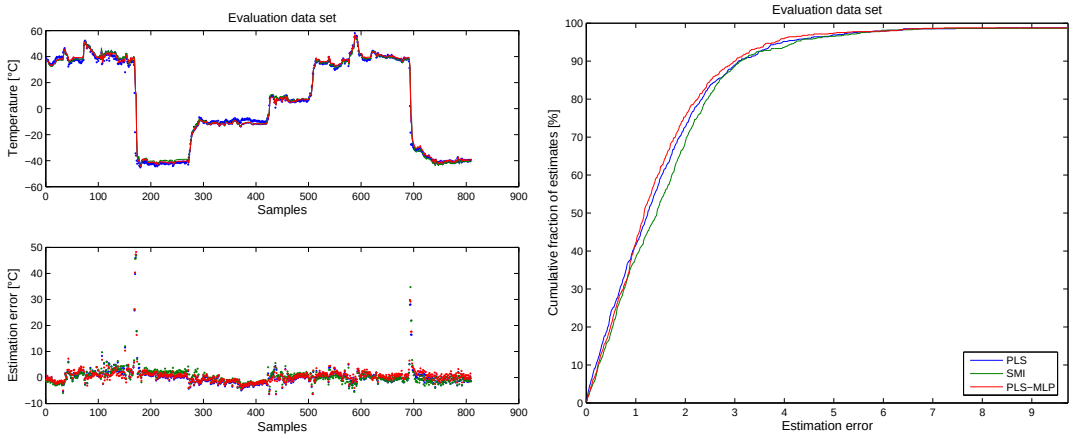

Figure 7.14: Measured and estimated values of IBP for the real process evaluation data set (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

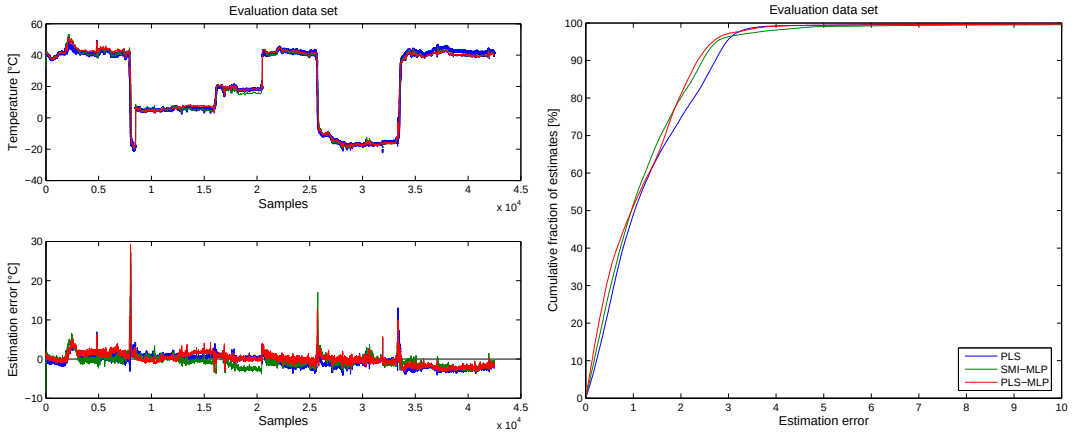

Figure 7.15: Measured and estimated values of FP for the real process evaluation data set (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

modified when needed. The distillation column of the dearomatisation process contains large volume of solvent during normal operation. This storage of solvent acts as an integrator filtering all high frequency variations in the bottom product quality. To take this into account, the estimates for FP are constrained to have a maximum

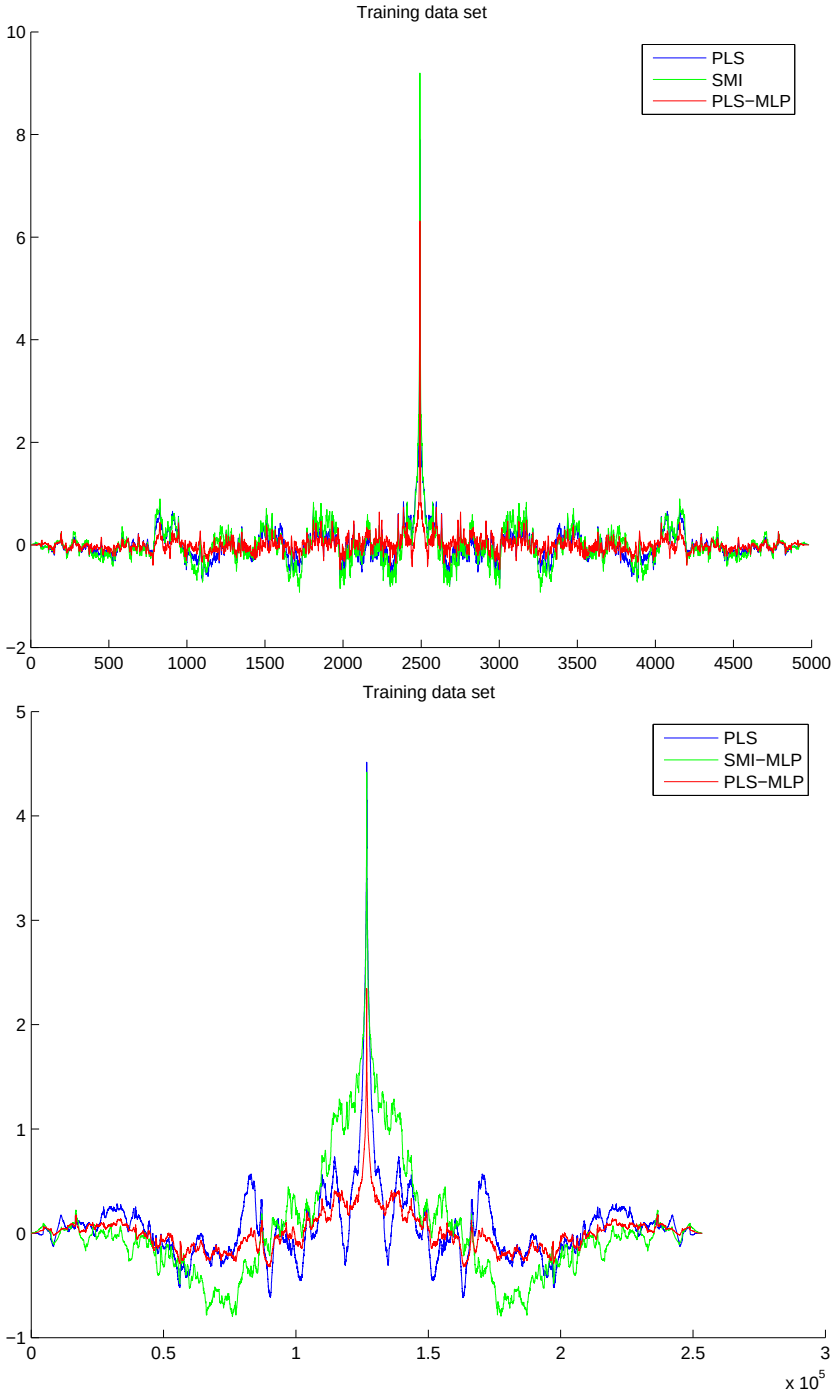

Figure 7.16: Autocorrelation of the estimation error signals for training data of all the IBP models (top) and FP models (bottom)

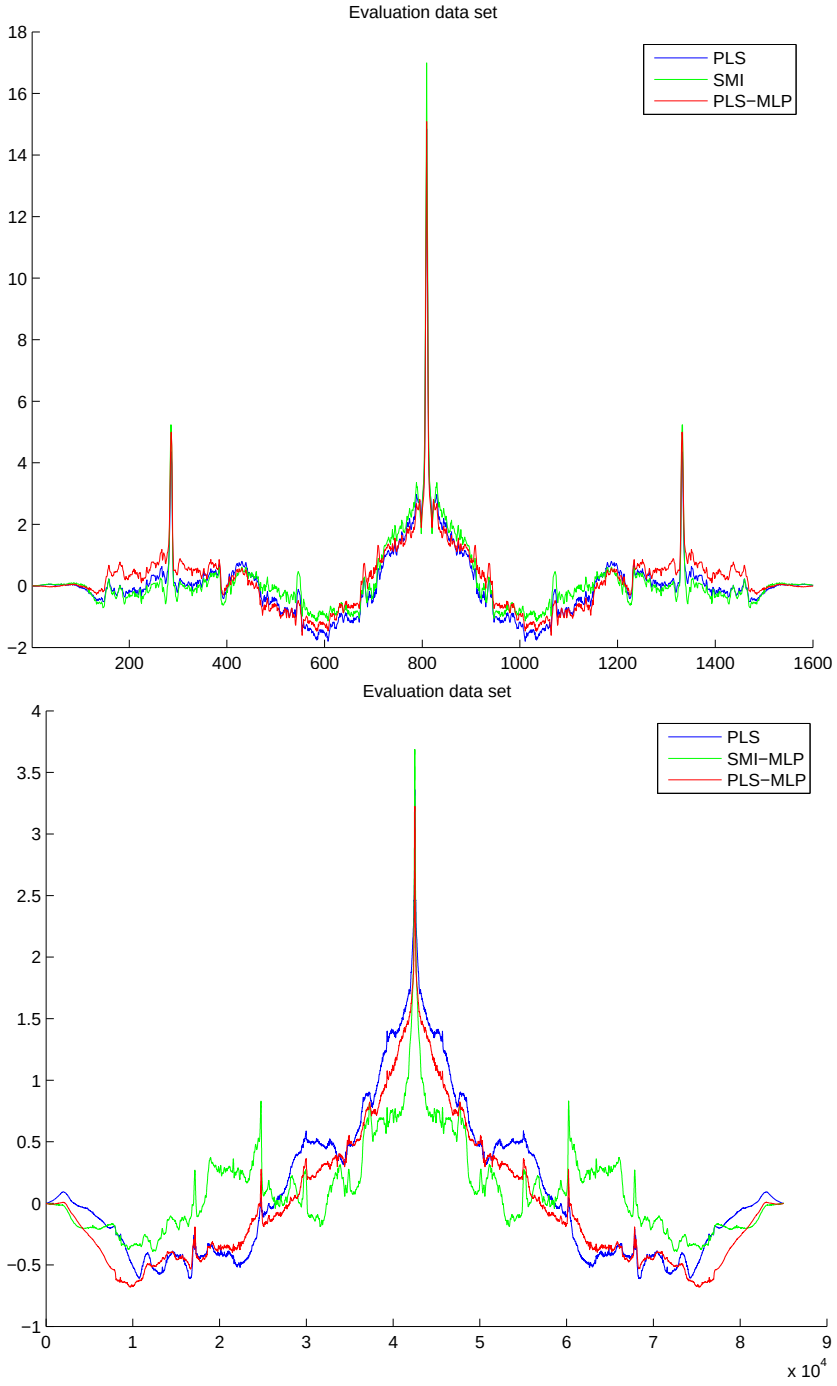

Figure 7.17: Autocorrelation of the estimation error signals for evaluation data of all the IBP models (top) and FP models (bottom)

rate of change and a maximum change in the rate of change (7.2 - 7.5).

$$L_u = 2 * est_{k-1} - est_{k-2} + \Delta roc \tag{7.2}$$

$$L_l = 2 * est_{k-1} - est_{k-2} - \Delta roc \tag{7.3}$$

$$\text{If} \quad est_k^0 > L_u \quad \text{then} \quad est_k^{mod} = min\,(L_u, lroc) \quad \text{and} \tag{7.4}$$

$$\text{If} \quad est_k^0 < L_l \quad \text{then} \quad est_k^{mod} = max\,(L_u, lroc) \tag{7.5}$$

where L_u is the upper limit, L_l is the lower limit, est_{k-n} is the modified estimate n instants ago, Δroc is the change in the rate of change, $lroc$ is the maximum rate of change, est_k^0 is the original estimate value, and est_k^{mod} is the modified current estimate at instant k. Suitable values for the parameters Δroc and $lroc$ were identified from the data to be 0.1°C/min and 0.4°C/min. The estimated values for IBP are not constrained as during the long delays between consecutive measurements the real temperatures may change dramatically.

8 Diagnostic information and fault detection decisions of the FDI system for the dearomatisation process

Next, in testing the methodology the change detection and fault diagnosis methods for the dearomatisation process were defined and the values of the FDI tuning parameters for the different testing experiments were determined.

The selected change detection method for the testing experiments was the Page-Hinkley method as suggested by the methodology. The algorithm also performed the fault diagnosis and thus no separate method is needed. The Page-Hinkley algorithm has two tuning parameters, the smallest significant error and the threshold for the cumulative sums (section 3.5.2), whose values were determined by using separate evaluation data sets. In the second testing experiment the evaluation data were the evaluation data sets 1 and 2. For the third testing experiment with real dearomatisation process data, the evaluation data set was the one described in section 6.2.2. The heuristic classification of the analyser statuses for the reference period of the real industrial data possibly included errors and introduced errors to the parameter optimisation of the FDI system. In both cases the fault free analyser output values were modified to represent abrupt and drifting type of faults with different magnitudes. E.g. the real and faulty FP and IBP values of the simulated evaluation data sets are illustrated in Figure 8.1.

The performances of the FDI systems were determined on the basis of the classification errors, i.e. the false alarms and missed detections. The objective functions for the FDI systems are given in (8.1) and (8.2)

$$Min\ J_{BPA} = \sum_{i=1}^{n} (|\hat{f}_{BPA}(i) - f^{0}_{BPA}(i)|) \tag{8.1}$$

$$Min\ J_{FPA} = \sum_{i=1}^{n}(|\hat{f}_{FPA}(i) - f^0_{FPA}(i)|) \tag{8.2}$$

where $\hat{f}_{BPA}(i) \in 0, 1$ and $\hat{f}_{FPA}(i) \in 0, 1$ are the estimated analyser states, $f^0_{BPA}(i) \in 0, 1$ and $f^0_{FPA}(i) \in 0, 1$ are the real states, i is the time instant and n is the length of the evaluation period.

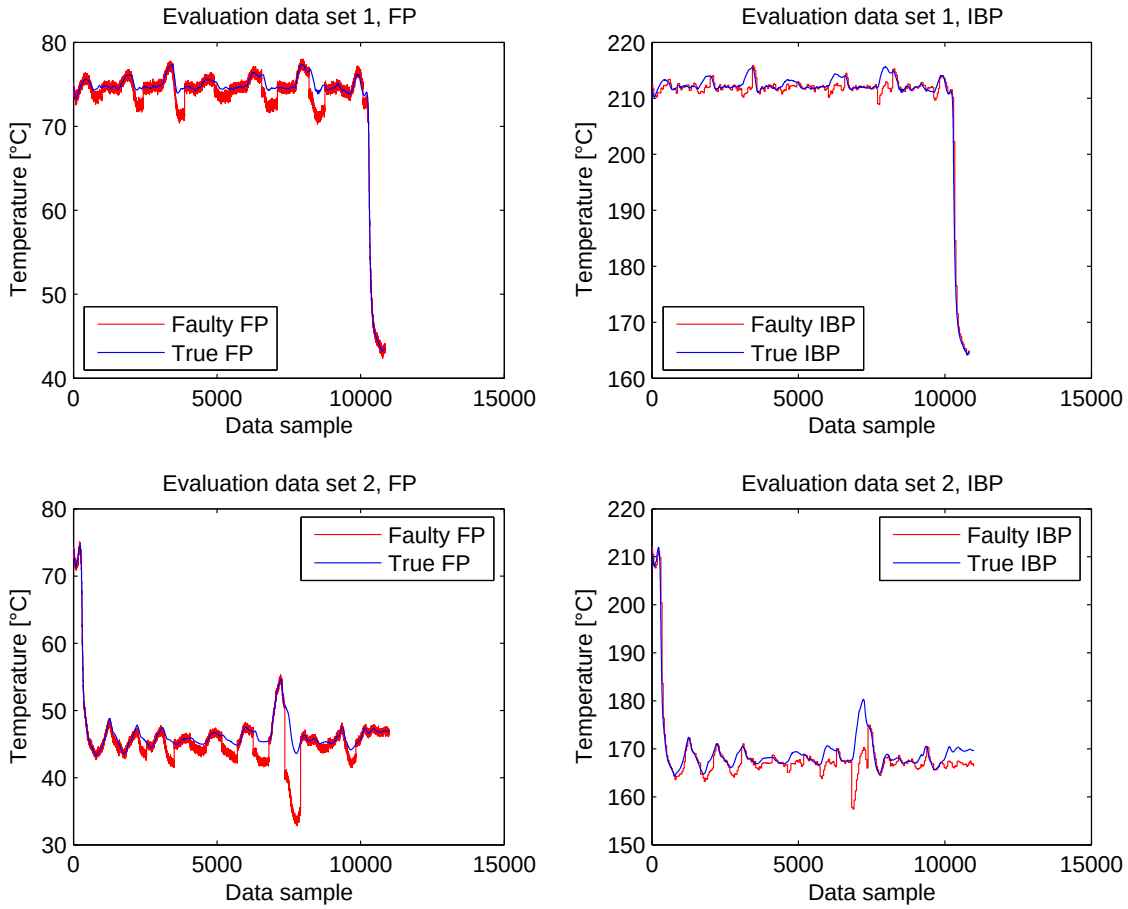

Figure 8.1: FP and IBP of evaluation data sets 1 and 2, simulated and faulty values

To minimise the values of the objective functions the several models were trained with different combinations of the tuning parameters. The minimised values of cost functions and the corresponding optimal FDI tuning parameter values are presented in Table 8.1. The error values represent the total number of minutes when the FDI systems' assessments of the analyser states was incorrect. The total length of

the testing signal was 21680 samples (minutes), containing 21680 FP and 542 IBP analysis results.

Table 8.1: Fault detection performances of the FDI systems based on different models and the corresponding optimal tuning parameters for the CUSUM.

	Mis-classifications in FP monitoring [min]	Mis-classifications in FP monitoring [%]	ν_{FP} [°C]	λ_{FP} [°C]	Mis-classifications in IBP monitoring [min]	Mis-classifications in IBP monitoring [%]	ν_{IBP} [°C]	λ_{IBP} [°C]
PLS	3909	18.0	1.9	3.8	134	24.7	3.1	0.1
MLP	3570	16.5	2.2	3.4	113	20.9	1.5	0.1
SMI	2714	12.5	2.5	3.0	139	25.7	2.1	0.1
SOM	6077	28.0	2.5	14.8	171	31.6	2.4	1.7
PLS-MLP	3521	16.2	2.1	3.2	134	24.7	3.2	0.1
SMI-MLP	2706	12.5	2.4	3.2	139	25.7	1.9	0.1

The optimisation criterion for the fault detection considered only the number of samples when the state of the analysers was estimated incorrectly. Instead, prior knowledge could have been used to set weighting coefficients to the two types of misclassification in the objective functions so that the system would avoid either false alarms or missed detections. In the same way, the system could have been made to be more sensitive to faults that either increase or decrease the analyser results. E.g. in the dearomatisation process analysers showing too high values is more harmful than faults that cause the analyser to indicate too low values.

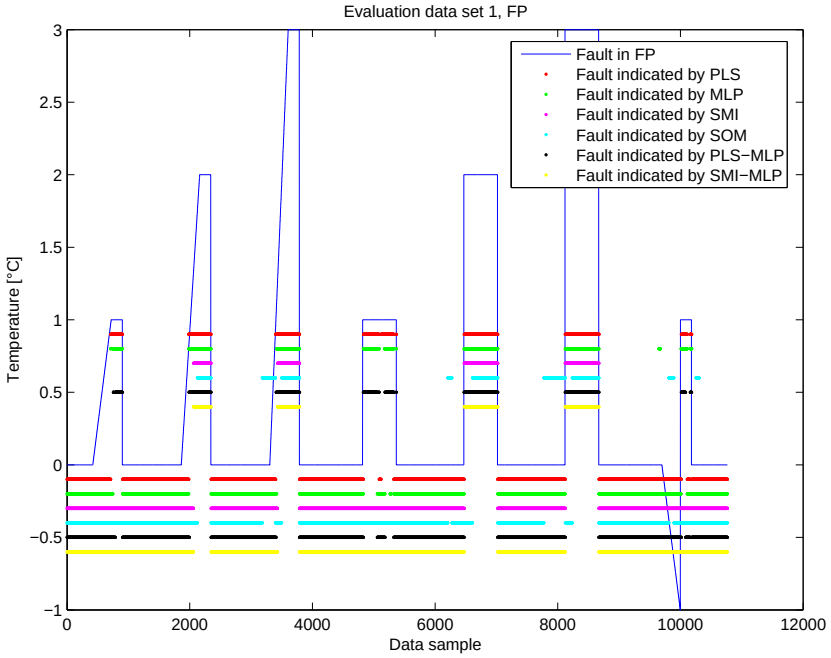

Figure 8.2: Faults in FP of evaluation data set 1 and corresponding fault indications of the different FDI models

It is noted that the faults used in the FDI tuning were small: The maximum magnitudes of the faults were only 1-3 °C. Furthermore, during the drift type faults and the smallest abrupt fault, the fault magnitudes were less than or equal to 1 °C for total of 4467 minutes of the total length of faults, 10033 minutes. The small magnitudes of the faults together with the objective functions with equal penalties for false alarms and missed detections lead to parameter values that made the FDI systems very sensitive to deviations in the analyser outputs. The high sensitivity to faults was preferred as the results of the FDI were given to the fault tolerant control. In the case of a dearomatisation process when the analyser results are infrequently available, the FTC strategies need to react to the earliest indications of suspected faults to maintain the process at the desired state. To support the early FTC-actions, the FDI was tuned to indicate even minor faults that could not be detected with certainty. Consequently, the FDI system also provide the FTC system with information about the reliability of the fault decisions. Also the reliabilities

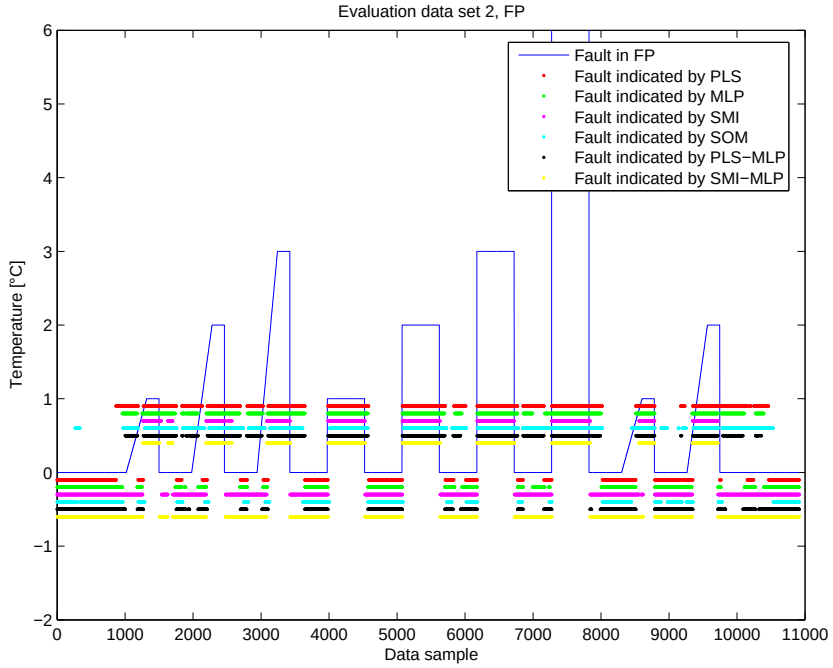

Figure 8.3: Faults in FP of evaluation data set 2 and corresponding fault indications of the different FDI models

of the estimated distillation and flashpoint temperatures were given as an output and help in determining whether the models were functioning well under the current operating conditions. The final piece of diagnostic information provided by FDI was the estimated size of a detected fault including the direction of the fault. In the following sections the derivation of these pieces of information are presented.

The reliability for the fault decision is given by (3.18) with a having a value of 4, i.e.:

$$F_{rel} = \min\left((\frac{U_n}{\lambda} - 1)/4, 1\right) \tag{8.3}$$

where F_{rel} is the reliability of the fault indication. The magnitudes of faults is estimated with (3.19) and the reliabilities of the estimated analyser outputs are determined with (3.22).

The suggested fault detection method based on cumulative sums is suitable for

issuing alarms under normal operation conditions. However, when the feed type is changed, the resulting rapid changes of significant magnitude in the operating conditions may cause the algorithm to trigger false alarms. This is mainly due to the fact that the delays in IBP and FP analysis are not constant. The estimation errors caused by comparing the analyser results few minutes too early or too late with the current estimated values may trigger the false alarms. It should be noted that under normal operation conditions the product quality does not change remarkably in few minutes, but during a feed type change the IBP and FP may increase or decrease notably. As a consequence, the varying delay in analysis becomes problematic. To prevent this problem, all fault indications were suppressed for an adjustable period of time after a feed type change was detected. This feature corresponds to the user requirement for gracefully handling the feed type changes (REQ-4.2 in Table 5.1) and the momentary loss of fault detection capability was seen as a better alternative to false alarms by the operating personnel.

9 Assessment of the performance of the FDI system for the dearomatisation process

Next phase in the validation of the methodology is to assess the performance of the three FDI systems for the dearomatisation process developed in the testing experiments.

9.1 FDI results of the first testing experiment

The fault-detecting abilities of the FDI systems based on the different models were evaluated using the simulated testing data set consisting of 1440 data points representing a period of 24 hours. Four faults to be detected were created on the analyser outputs: a large abrupt error (-50 % drop in the temperature), a smaller abrupt error (-20 % drop in the temperature), rapidly progressing drift (-10 °C per hour) and slowly progressing drift (-5 °C per hour). The abrupt faults correspond to the water contamination of the analysed sample fault of the real process and the drifting faults correspond to the carbonisation of the flask of the distillation analyser. In the first experiment the PCA models did not estimate the analyser outputs and consequently the fault decisions were not performed with the CUSUM method.

First, the PCA-based FDI system was evaluated. The Hotelling T^2 index was calculated and used to detect the faults. When the index alarm limit was set to 14, the PCA-based monitoring method was able to detect the abrupt faults correctly without a delay. The first drifting type fault in IBP and FP were detected after a delay of 74 minutes. The models were not, however, able to detect the slowly progressing analyser faults. In addition, the models gave false alarms after the first and before the fourth fault. Figure 9.1 (A) shows the faults in the evaluation data and

the alarms given by the six PCA models. The x-axis of the figure represents time and the y-axis the monitoring results; values close to zero mean a normal process situation, and values near to one represent given alarms.

The residuals between the PLS model outputs and analyser results were calculated and, when they exceeded a limit value of 1.4, an alarm was triggered. The PLS-based fault detection system performed well and it was able to detect the abrupt faults with no delay, and the incipient faults with delays of 9 and 23 minutes. The first drifting fault was detected when the analyser error was 2-3 °C and the second one before it was 2 °C. The PLS based monitoring system gave no false alarms. Figure 9.1 (B) shows the alarms given by the method against the actual fault.

Next, the SMI models were tested. The residuals between the estimated and simulated analyser outputs were calculated, and an alarm was given by the system when the index exceeded an alarm limit of 2.8. The system was able to detect the abrupt faults immediately and the incipient faults with delays of 8 and 35 minutes, when the deviation from the nominal was 1.5-3 °C. The SMI model for IBP gave few false alarms before the fourth fault. Figure 9.1 (C) shows the alarms given by the method and the actual faults.

Finally, the FDI systems based on the SOM models were tested. Both the SOMs trained with normal measurements and the ones trained with PCA transformed data were able to detect the abrupt faults immediately. In detecting the drifting type faults PCA pre-treated SOM was superior to the normal SOM. PCA-SOMs average detection delay of the slowly developing fault was 22 minutes compared to the normal SOMs average delay of 45 minutes. The difference in performance derives from the fact that the residuals produced by the PCA-SOMs had smaller variation than those of the normal SOMs allowing the use of a lower alarm limit of 4.5 instead of 5. Using the lower alarm limit with the normal SOMs would have resulted in a number of false alarms. The alarms given by the SOMs are shown in

Figures 9.1 (D) and (E).

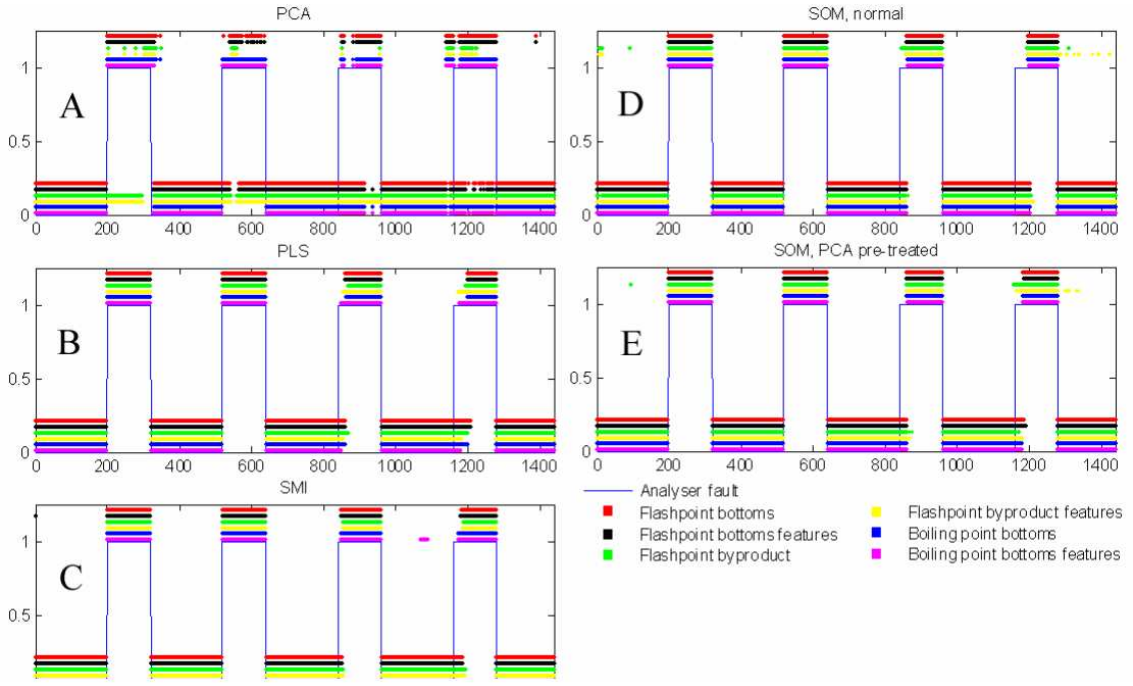

Figure 9.1: True states of the analysers and fault indications of the PCA (A), PLS (B), SMI (C), Normal SOM (D) and PCA-SOM (E) models

All the tested methods except PCA were able to detect the abrupt fault with greater magnitude without delays, as shown in Table 9.1. The alarm signals were given as soon as the fault occurred and they lasted as long as the fault persisted. The subtle differences in the performance of the methods became visible only in detecting the slowly developing faults: On the average the fastest method to detect faults was PLS followed by slightly slower SMI. The PCA pre-treated SOM performed better than the normal SOM. The performance of the PCA was worst of the tested methods; its detection delays were the longest and it was unable to detect all of the drifting faults. The poor performance of the PCA could be explained by the fact that the detected faults affected only one variable of the input variable set. Had the higher-level quality control been in use, the faults would have propagated into other variables and the fault would have been easier to detect. PLS models on the contrary considered

the input and output separately and performed well as the faults affected the whole set of outputs i.e. the analyser output. The SOMs were used here as nonlinear regression tools and the FDI results were good. The PCA treatment of the data done prior to the training of the maps proved to be beneficial as the PCA-SOMs detected the faults faster than normal SOMs. The training of the maps was also less time consuming as only five variables were used. Using the process phenomena describing calculated variables together with the direct process measurement data in monitoring improved the results of the SOM models in some cases, but on few occasions it made the delays longer. The fault detection performances of the different models are summarised in Table 9.1.

Table 9.1: Delays of fault detection, first testing experiment

	Fault 1		Fault 2		Fault 3		Fault 4	
	Dir	CV	Dir	CV	Dir	CV	Dir	CV
FP								
PCA	0	0	23	23	74	74	—	—
PLS	0	0	0	0	9	9	22	22
SMI	0	0	0	0	11	11	27	27
SOM norm	0	0	0	0	20	20	43	43
SOM PCA	0	0	0	0	20	20	27	31
IBP								
PCA	0	0	0	0	74	74	—	—
PLS	0	0	0	0	9	9	21	21
SMI	0	0	0	0	9	8	15	13
SOM norm	0	0	0	0	21	22	44	43
SOM PCA	0	0	0	0	20	21	22	22

'Dir' stands for models using only measured variables,
'CV' are the models created with measurements and calculated variables

The first simulation study showed that the PLS was the most suitable of the tested fault detection methods in this case covering operation of the dearomatisation with a single feed stock type. Deviations of about 2-3°C between the faulty and the nominal analyser output values were detected. This can be considered as a good result, as the nominal analyser measurements were not ideal, but included noise with amplitude of 1°C. SMI and SOM also proved to be very useful tools in FDI. The performance of the PCA was poor; either the faults were recognised only after their presence was obvious and/or a large number of false alarms were generated.

9.2 FDI results of the second testing experiment

Next, the fault detection performance of the FDI systems based on different model types were evaluated with the evaluation data sets 3 and 4. The testing data set consisted of 8641 samples (minutes) consisting 8641 FP and 217 IBP analysis results. The FDI system based on SOM model was the most accurate in detecting faults in the FP analyser. It was able to correctly classify 87.2 % of the analysis results, while the least accurate system, based on PLS model, correctly classified 81.4 % of the analysis results. For IBP, the system based on PLS-MLP model was most accurate with 88.0 % of the analyser results correctly classified, followed closely by the system based on PLS model (87.6 % correct classification). The least accurate results, 75.1 %, were achieved with the system based on MLP model. The systems based on the most accurate models SMI and SMI-MLP (see Chapter 7), performed well for both analysers. They were the second and third most accurate systems in detecting faults in the FP analyser and third and fourth in detecting faults in the IBP analyser. Considering both analysers, the most accurate FDI system was the one based on PLS-MLP model (85.4 % correct classifications) followed by systems based on SMI, SMI-MLP, and SOM, all three having the same classification accuracy of 84.6 %.

The testing results are comparable with the results acquired during the fault detection parameter optimisation described in Chapter 8 (Table 8.1) with two exceptions. During the optimisation, the system based on SOM was the least accurate in classifying the FP analyser results. The inaccurate estimations caused the optimal value of λ to be signicantly higher compared to the corresponding value of other systems. In consequence, during the testing, the system gave only few false alarms resulting in an accurate detection of faults. The other exception is the performance of the system based on MLP model. During the parameter optimisation, the system gave the most accurate estimations, and in consequence the optimal fault detection parameter values were smaller than with the other systems. In contrast, for the testing data, the estimates based on the MLP model were less accurate and because of the tight detection limits, several false alarms were given lowering the overall detection accuracy.

The results of the testing and the fault detection parameters of the systems based on the different model types are given in Table 9.2.

9.3 FDI results of the third testing experiment with real dearomatisation process data

Next, the online validation of the FDI system with the third testing experiment was performed during 29 days from May 3rd 2007 - May 31st 2007 onsite at the Naantali oil refinery, Finland. During the period, 4 different feed types were used in 6 runs. Most significant changes in operation were related to the feed type changes. For the last 12 days of the evaluation period, exceptionally heavy feed stock was used that provided another challenge for the FDI system.

Table 9.2: Fault detection performances for evaluation data sets 3 and 4 of the FDI systems based on different model types, second testing experiment

	Mis-classifications in FP monitoring [min]	Mis-classifications in FP monitoring [%]	ν_{FP} [°C]	λ_{FP} [°C]	Mis-classifications in IBP monitoring [min]	Mis-classifications in IBP monitoring [%]	ν_{IBP} [°C]	λ_{IBP} [°C]	Average mis-classifications [%] [°C]
PLS	1607	18.6	1.9	3.8	27	12.4	3.1	0.1	15.5
MLP	1551	17.4	2.2	3.4	54	24.9	1.5	0.1	21.2
SMI	1423	16.5	2.5	3.0	31	14.3	2.1	0.1	15.4
SOM	1110	12.8	2.5	14.8	39	18.0	2.4	1.7	15.4
PLS-MLP	1477	17.1	2.1	3.2	26	12.0	3.2	0.1	14.6
SMI-MLP	1429	16.5	2.4	3.2	31	14.3	1.9	0.1	15.4

9.3.1 Fault detection results during the onsite validation period

To evaluate the performance of the implemented FDI system, abrupt and drifting types of faults were introduced to the two online analysers. The emphasis was on faults lowering the analyser reading, as most of the real analyser faults are of that type. In order to minimise the disturbances caused by the introduced faults, the fault detection limits were tuned to be more sensitive than normally for the duration of the fault tests. In the following sections, typical examples of the fault scenarios are presented and analysed for both fault types and both analysers.

First, abrupt faults were introduced on the distillation analyser. The first fault scenario consisted of a single drop of 20 °C in the IBP. As a consequence, the

estimation errors increased and the first faulty analyser result caused the cumulative error sums to exceed the threshold limits. The fault was thus immediately detected by all tested methods. Because of the considerably large magnitude of the fault, the cumulative sums were well above the threshold limits after the fault had been removed and the system continued to indicate the fault for more than 200 minutes. The normal analyser operation was detected after three consecutive estimations agreed with the analyser results. The real, faulty and estimated IBPs are shown in Figure 9.2. In the figure, also the estimation errors, cumulative error sums and fault indications are presented for each model type.

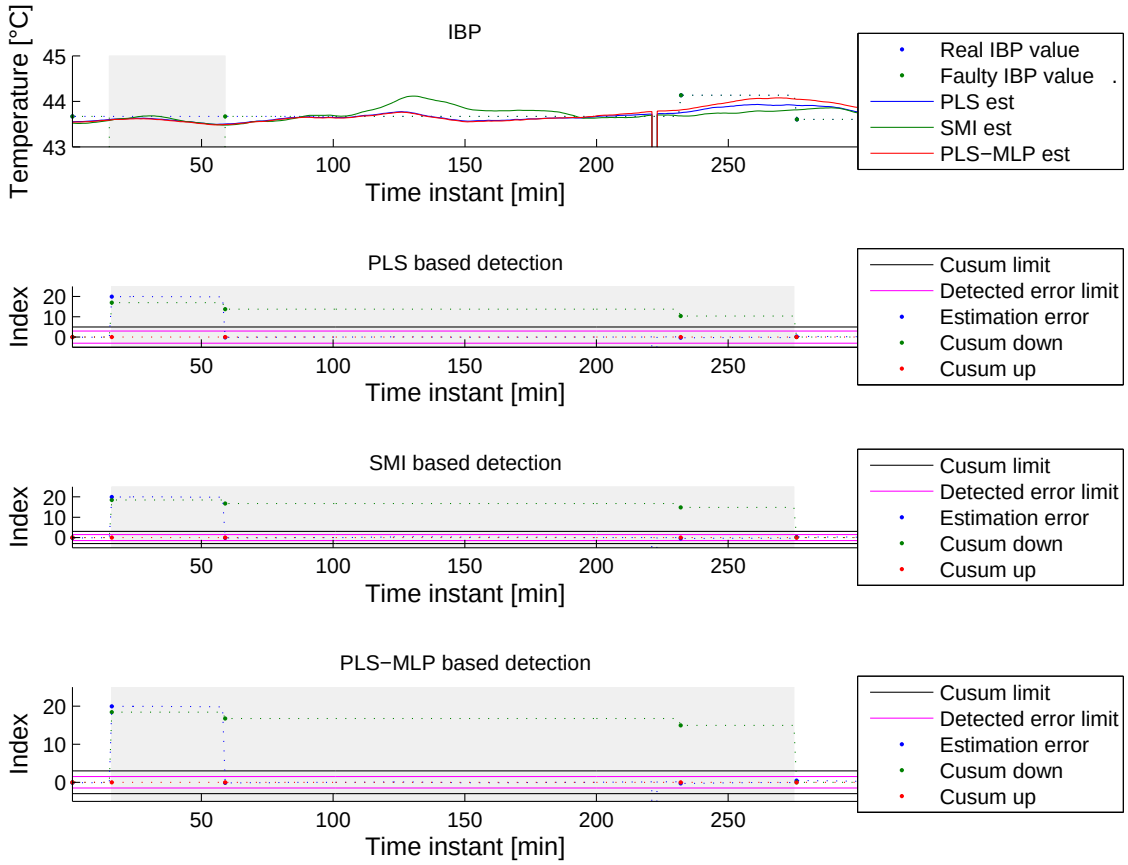

Figure 9.2: Abrupt fault in the distillation analyser. Periods of the introduced fault and the faults indicated by the models are highlighted with light grey background.

Next, drifting faults were introduced to the distillation analyser.

The IBP was first dropped by 2 °C and then by 1 °C and 0.4 °C, after which it was raised by 0.2 °C and finally, 340 minutes after the start of the experiment it was returned to its real value. The fault caused deviations between the measured and estimated analyser outputs, which in turn, increased the values of the cumulative sums. The Page-Hinkley parameter for smallest detected fault was 3 °C and the threshold for the cumulative error sum 3. 215 minutes after the first modification to the analyser output, all the models detected a fault. The larger fault of -3 °C had been present for 41 minutes. The bias was removed 126 minutes later and after 51 minutes, the FDI system returned to indicate normal operation of the analyser. During the fault, the confidence indices for the fault were raised to about 0.3 with the SMI and PLS-MLP models and to about 0.4 with the PLS model. The confidence values started to go down immediately after the fault was removed. In this case of a small fault with a short duration, the return to the normal state was caused by the cusums dropping below the thresholds.

Next, abrupt faults were introduced on the FP analyser. The output value was first raised abruptly by 1.5 °C and then it was further raised in 0.1 °C steps to 2.2 °C. After 80 minutes, it was raised to 2.4 °C. The fault was first detected by PLS based system 67 minutes after the introduction of the fault when the fault magnitude was 2.2 °C. The second fault indication was given by the SMI based system when the magnitude of the fault was 2.4 °C. The detection delay with SMI model was 133 minutes. The PLS-MLP based system missed the fault completely. The fault was removed after 158 minutes, and the PLS system returned to indicate normal analyser operation 21 minutes later and the SMI system 28 minutes later. The progress of the fault detection during the fault scenario is presented in Figure 9.4.

The last fault scenario consisted of drifting type faults in the FP analyser. In the example fault scenario, a fault was introduced by lowering the analyser output by 0.1 °C steps in 5 minutes intervals. The final magnitude of the fault was 2.4 °C. In this case, the system based on PLS-MLP was the fastest to detect the fault, 94

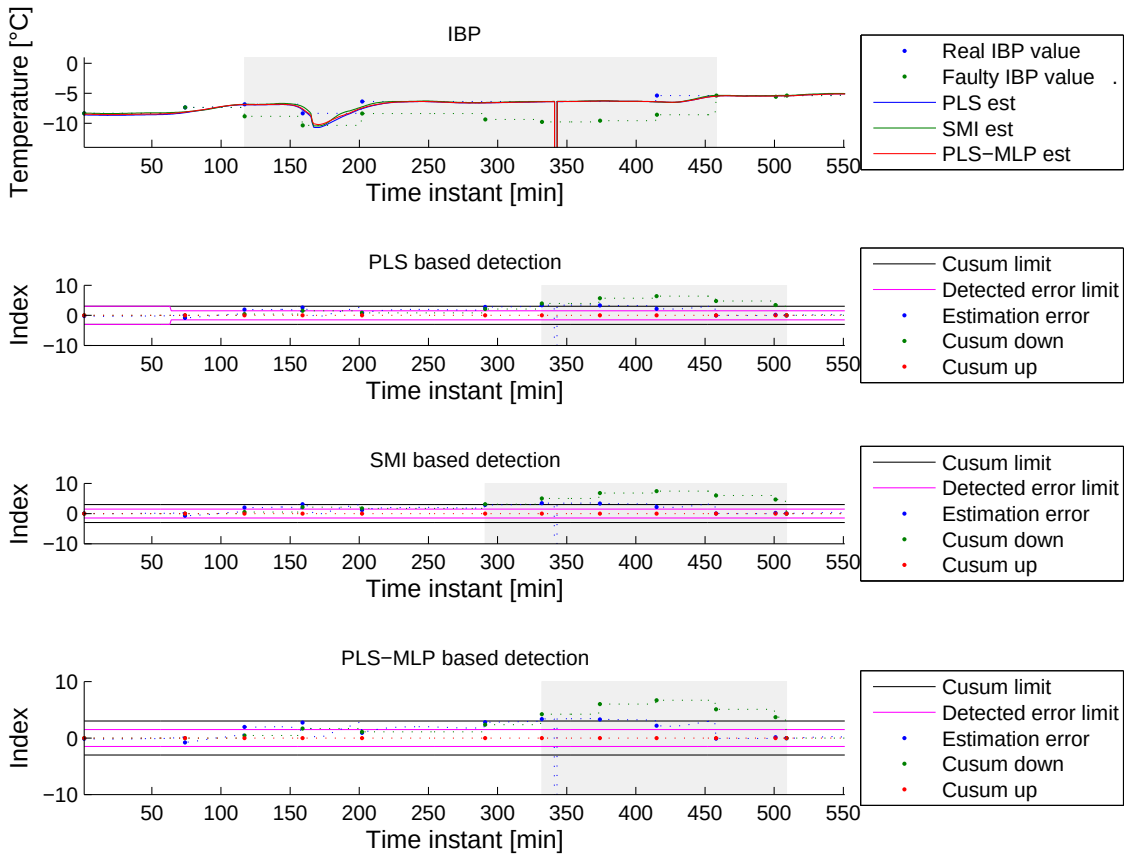

Figure 9.3: Drifting fault in the distillation analyser. Periods of the introduced fault and the faults indicated by the models are highlighted with light grey background.

minutes after the start of the fault. Two minutes later the PLS system detected the fault and followed by the SMI system one minute later. During the times of detection, the magnitude of the fault was 2.0 °C. The duration of the fault was 214 minutes, and after it was removed all the FDI systems returned to indicate normal analyser operation 3 minutes later. The relevant FDI information during the scenario is presented in Figure 9.5.

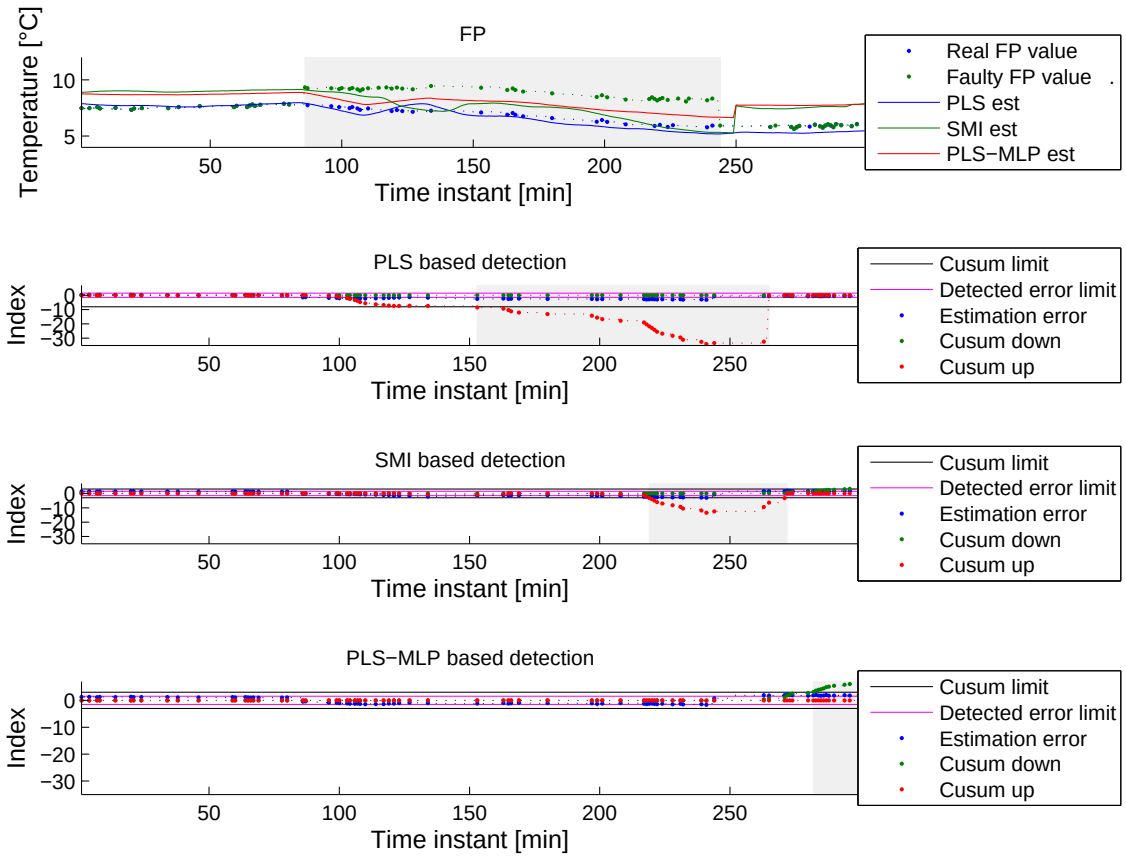

Figure 9.4: Abrupt fault in the flashpoint analyser. Periods of the introduced fault and the faults indicated by the models are highlighted with light grey background.

9.3.2 Identification of the feed stock changes and suppression of faults

One of the user requirements (REQ-4.2 in Table 5.1) was the identification of the feed stock changes. To suppress all fault indications during the transients in the process, the method described in section 8 was used. A separate function was implemented in the NeCST platform to detect the feed stock changes based on a variable describing the feed type. After a change in the variable value was detected, all faults detected by the system were suppressed for a period of 500 minutes. During that period, the process had stabilised to its new operating point and the normal operation of the FDI system was resumed.

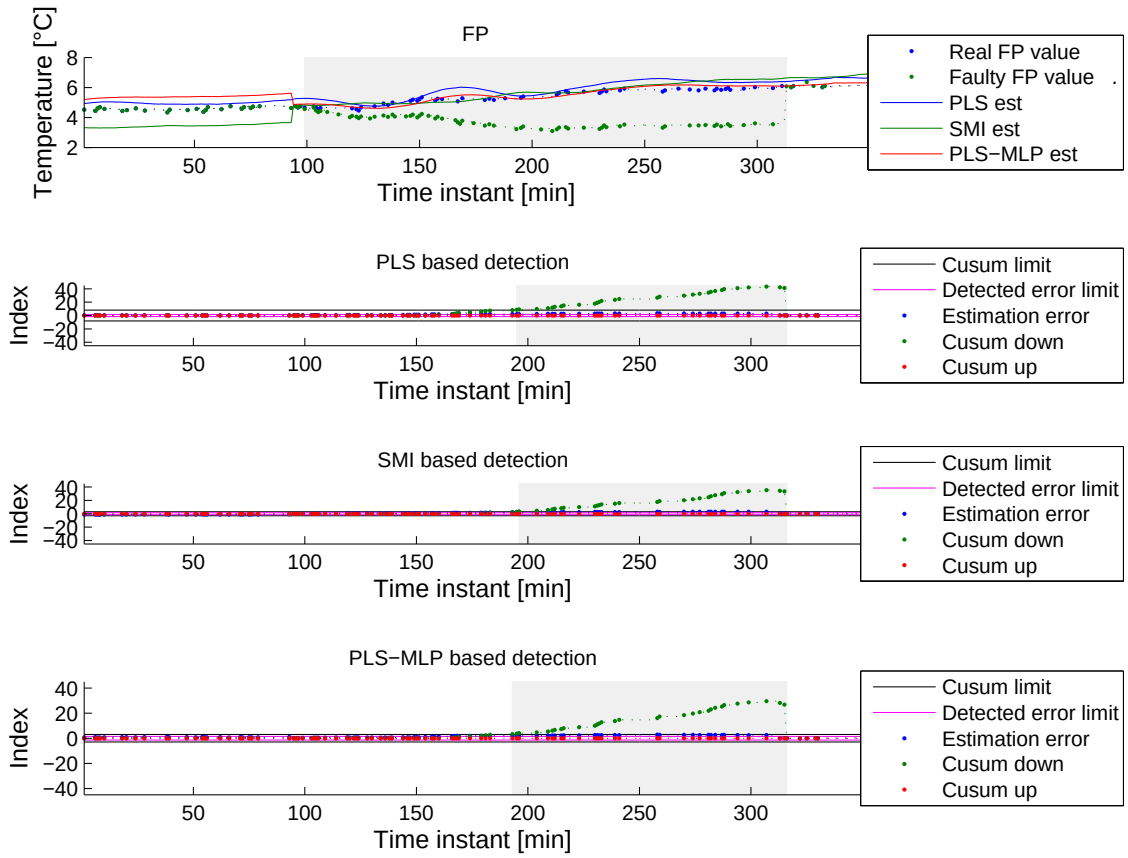

Figure 9.5: Drifting fault in the flashpoint analyser. Periods of the introduced fault and the faults indicated by the models are highlighted with light grey background.

10 Utilising the FDI information in fault tolerant control of the dearomatisation process

As described in Chapter 1 FTC systems are being developed to automatically utilise the information provided by the FDI systems. In the methodology, the structure of the FDI implementation was designed to support FTC actions (Chapter 5). In this section the implemented fault tolerant control (FTC) of the dearomatisation process is shortly described. The effects of the FTC on the process operation were evaluated onsite at the Naantali refinery and are illustrated through an example fault scenario.

10.1 Fault tolerant control of dearomatisation process

First, the minimum necessary FTC functions for the dearomatisation process were defined (Järvinen *et al.*, 2006). The developed FTC scheme consists of two types of strategies: proactive and reactive making intensive use of the fault detection reliability information produced by the FDI. The reactive FTC strategies are triggered when a fault has been detected with a high reliability. They are designed to cancel the further effects of the fault on the process. The first reactive FTC-action is to temporarily deactivate the feedback from the analyser to the model predictive controller (MPC) in order to prevent the faulty measurement from affecting control. Once the MPC feedback is deactivated, the MPC uses exclusively its internal models to estimate the product quality. If the detected analyser fault produces higher than true values, there is a risk of off-specification product due to the feedback action. In that case, a target manipulation strategy, is applied. In the target manipulation strategy, the target value of the controlled variable is modified using the information of the estimated size of the fault provided by the FDI system. The target manip-

ulation causes the control to reduce the effects of the previous control actions that have been made using the faulty feedback before the fault was detected.

Proactive FTC strategies are used to restrict any control actions when the analyser feedback is suspected to be faulty i.e. a fault is detected with a low reliability. At the same time, these FTC strategies aim at the same time to minimise the loss of control performance in cases the fault detection later turns out to be false. The proactive strategies automatically set new values for MPC parameters so that the control relies less on the analyser measurements. With the distillation analyser, the tuning parameter is a feedback filter factor and for the flashpoint analyser it is the deadband. Although the retuning can be a continuous function of the reliability index, in this application three threshold levels; I, II and III, with III being the most severe, were used for easier interpretability of the alarms.

The FTC is performed each minute and the actions are automatically activated or deactivated as required. Retuning is automatically cancelled if the fault detection turns out to be false. If a reactive FTC strategy is triggered, the proactive strategies become ineffective with the deactivation of the analyser feedback. In that case the retuning actions are automatically cancelled and cannot be reactivated before the fault has been corrected and the reactive actions cancelled.

10.2 Online testing of the FTC at the Naantali refinery

The typical behaviour of the FTC system in the real industrial environment is illustrated with a case similar to the previously described fault scenarios in section 9.3. In this case the analyser result for IBP is artificially cumulatively increased by 4.5 °C in 4 steps within a period of three hours. The elevation in the faulty IBP remained relatively small as the MPC controlled the process keeping the IBP close to the desired level. The controller used the distillation column temperature

to control the IBP and under closed loop operation, the fault in the IBP result was most prominently reflected in the value of this manipulated variable (MV). 72 min after the fault is introduced, the FDI gave a fault indication with moderate reliability and a proactive FTC strategy of MPC retuning I was triggered. After the first retuning action, the reliability of the fault increased rapidly and the next distillation analyser result for IBP became available, the reactive FTC strategies of MPC feedback deactivation and CV target manipulation were triggered. As a result of these FTC actions, the drop in the value of the column temperature caused by the faulty analyser feedback is almost completely cancelled as in the simulated case. The progress of the relevant process measurements and the outputs of the FDI system during the upward incipient fault are illustrated in Figure 10.1. In the figure, the IBP and column temperature (MV) are shown in addition to the information provided by the FDI: the estimated IBP, the fault indication, the reliability of the fault indication and the estimated magnitude of the fault. The pressure compensated temperature of the column with the corresponding MPC set point values are shown on the bottom of the figure.

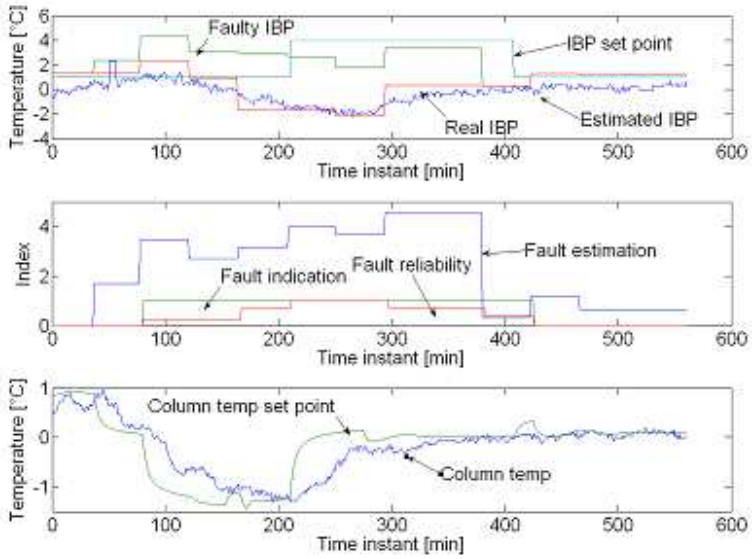

Figure 10.1: Incipient fault in the distillation analyser during onsite validation. IBP temperatures (top), FDI information (middle) and the pressure compensated temperature in column with corresponding MPC set point value (bottom)

11 Analysis and summary of the FDI results of the applying the methodology for a dearomatisation process case

The effects of the proposed improvements to the previously established FDI methods are illustrated with several comparative studies. First, the effects of the refined data preprocessing techniques of IVS and delay compensation are studied. Second, the benefits of creating and using process primitives have been studied by comparing the results of models exploiting the primitives to those that are trained only with direct process measurements. These studies correspond to the Tasks 3 and 4 described in section 1 for proving the claims of the hypothesis.

The effects of the delay compensation and the more sophisticated input variable methods on the modelling accuracy of the most accurate model for FP, the SMI, are illustrated within Table 11.1. In the table, the RMSE values are given for the SMI-MLP model created according to the methodology and for SMI-MLP models created using different combination of the steps recommended by the methodology. The input variable set for the reference model trained without methodology steps has been determined with the standard forward selection method (section 3.4.1). The input variable selection (IVS) step refers to the modified forward selection method. The effects of the GA in IVS were studied separately. The methodology steps of delay compensation and IVS, when used without others, resulted in more accurate models than the reference model. The corresponding average RMSE values for all evaluation data sets were 48 % and 56 % lower than with the reference. When calculated variables step was used without other steps, the results were the same than those of the reference model. However, when calculated variables were used together with delay estimation or IVS, additional drops of 4 % and 1 % in the average RMSE values were achieved. Using all three steps of delay compensation,

IVS and calculated variables, the average RMSE was 55 % lower than with the reference model. The most accurate model was achieved by also including GA in the IVS. The average estimation error of the model was 0.38 °C, 60 % lower than with the reference model. It was noted that by using the methodology, the number of states within the model was lower compared to the reference. The downside of the methodology is the significant increase of the approximate execution time of the training script.

Table 11.1: RMSEs of SMI models for FP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of states
Meth.			0.203	0.852	0.061	0.078	0.083	0.980	0.376		3
Ref.			0.569	2.614	0.282	0.262	0.455	1.487	0.945	116	6
X			0.301	1.268	0.061	0.147	0.126	1.043	0.491	128	1
	X		0.244	0.442	0.132	0.123	0.163	1.410	0.419	160	1
		X	0.569	2.614	0.282	0.262	0.455	1.487	0.945	118	6
X	X		0.262	1.045	0.071	0.116	0.140	1.099	0.455	329	8
X		X	0.170	0.762	0.097	0.093	0.118	1.487	0.454	136	1
	X	X	0.229	0.494	0.167	0.171	0.221	1.195	0.413	209	2
X	X	X	0.258	1.004	0.062	0.139	0.104	1.013	0.430	455	1

In case of the IBP the most accurate estimations were provided by the SMI-MLP model. The effects on the estimation accuracy of the methodology steps are similar to those in the FP case (Table 11.2). All steps improved the accuracy when used separately most significantly using IVS reduced the RMSE by 53 % compared to the reference. The best results for using two steps together were given by the model trained with IVS and calculated variables. Most accurate results were again

obtained by utilising all steps of the methodology including GA in IVS. The average estimation error of the model was 0.38 °C, 56 % lower than with the reference model.

Table 11.2: RMSEs of SMI-MLP models for IBP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of neurons
Meth.			0.198	0.851	0.065	0.081	0.085	0.980	0.377		1
Ref.			0.503	2.459	0.178	0.293	0.217	1.512	0.861	129	2
X			0.309	1.271	0.065	0.155	0.121	1.040	0.494	132	1
	X		0.246	0.440	0.092	0.096	0.123	1.419	0.403	205	1
		X	0.566	2.476	0.066	0.184	0.169	1.435	0.816	108	1
X	X		0.329	0.957	0.059	0.175	0.136	1.081	0.456	210	2
X		X	0.174	0.776	0.091	0.083	0.135	1.485	0.457	130	3
	X	X	0.299	0.551	0.095	0.112	0.172	1.173	0.400	263	3
X	X	X	0.300	0.790	0.068	0.154	0.139	1.012	0.410	253	3

To analyse the effects of the steps in general, the results of all of the model types have been averaged (Table 11.3). The most accurate estimations were produced with models that were trained according to the methodology, but without GA in IVS. Using GA in the selection of the input variables produced in some cases, especially with MLP, more inaccurate estimates for the evaluation data sets. As a consequence, the GA may not be suitable for modelling methods that are prone to overfitting problems. Generally the models trained according to the methodology were significantly less complex, i.e. the models had fewer LVs, neurons, and/or states. Two individual steps clearly contributing in this reduction of model complexity were the IVS and calculated variables. The modified forward selection IVS method (section 3.4.1) was also the most important step in reducing the estimation

errors. Using calculated variables also had positive effects on the estimation accuracies of the models, but a more significant improvement was their ability to reduce the dimension of the measurement data set. The delay compensation was proved to be an effective way to improve the estimation accuracy of the models in general. The static models, PLS, MLP and SOM benefit most from the delay compensation, but also SMI produced more accurate state-space models when delay compensated data was used.

The detailed RMSE corresponding to all other tested model types are given in Appendix C.

Table 11.3: Average RMSEs of PLS, MLP, SMI, SOM, PLS-MLP and SMI-MLP models for IBP and FP

Delay compensation	Input variable selection	Calculated variables	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Average degrees of freedom*
Meth.			0.427	1.081	0.201	0.252	0.296	1.394	0.614		3.750
Ref.			0.577	1.933	0.158	0.282	0.340	1.722	0.835	745	9.000
X			0.388	1.606	0.102	0.231	0.248	1.613	0.698	742	8.250
	X		0.474	1.124	0.120	0.244	0.271	1.613	0.641	3424	3.833
		X	0.559	1.995	0.140	0.264	0.303	1.715	0.829	836	6.750
X	X		0.433	1.339	0.096	0.224	0.267	1.298	0.610	3451	8.000
X		X	0.374	1.556	0.129	0.214	0.255	1.602	0.688	519	8.833
	X	X	0.407	1.088	0.133	0.241	0.259	1.680	0.635	4828	5.750
X	X	X	0.403	1.168	0.102	0.237	0.267	1.384	0.594	4662	5.167

* The degrees of freedom for the tested models are the number of latent variables (PLS), number of neurons in the hidden layer (MLP), number of states (SMI), and size of the SOMs.

12 Conclusion

In this work a methodology is presented for constructing a fault detection (FDI) system for a complex industrial process utilising data-based modelling methods. The possibilities of using prior knowledge in data-based modelling have been studied. The importance of the selection of a proper data-based modelling method has been emphasised and general guidelines have been presented for partly solving the problem. Indices for describing the integrity of data sets and the reliability of detected faults have been introduced. In addition, information about the FDI has been used to achieve fault tolerant control (FTC) of the monitored process. In order to validate the methodology, three separate FDI/FTC systems for a dearomatisation process have been created utilising data created with a dynamic simulator and real process data collected from the Neste Oil Oyj Naantali refinery. The performance of the FDI/FTC system, created with real dearomatisation process history data, was validated onsite at the refinery.

The hypotheses presented in Chapter 1 are: (1) information provided by data-based modelling methods can be utilised to achieve FTC, (2) the estimation accuracy of the linear models (PLS and SMI) can be improved by combining them with a nonlinear ANN model (MLP), and (3) better input variable sets are achieved by including calculated variables. The hypotheses have been verified using the results of three experimental FDI systems for the dearomatisation process.

In the future, the tasks related to data-based modelling that need to be further researched include more rigorous methods for selecting optimal modelling methods based on the properties of the process history data. Also the possibilities for simultaneously using different types of models to perform FDI need to be considered in more detail. The third important research topic is the more efficient use of existing prior knowledge in data-based modelling. Creating calculated variables that

describe complex, unmeasured process phenomena offers one possibility for more accurate and robust data-based modelling.

References

[1] A. AlGhazzawi and B. Lennox. 2008. Monitoring a complex refining process using multivariate statistics. Control Engineering Practise 16, pages 294–307.

[2] C. Aubrun, P. De Cuypere, and D. Sauter. 2003. Design of a supervised control system for a sludge dewatering process. Control Engineering Practice 11, pages 27–37.

[3] A. D. Back and A. S. Weigend. 1997. A First Application of Independent Component Analysis to Extracting Structure from Stock Returns. International Journal of Neural Systems 8, pages 473–484.

[4] B. R. Bakshi. 1998. Multiscale PCA with Application to Multivariate Statistical Process Monitoring. AIChE Journal 44, pages 1596–1610.

[5] B. Ballé, M. Fisher, D. Fussel, O. Nelles, and R. Isermann. 1998. Integrated Control Diagnosis and Reconfiguration of a Heat Exchanger. IEEE Control Systems Magazine 18, pages 52–64.

[6] Y. H. Bang, C. K. Yoo, and I-B. Lee. 2003. Nonlinear PLS modeling with fuzzy inference system. Chemometrics and intelligent laboratory systems 64, pages 137–155.

[7] A. Bhagwat, R. Srinivasan, and P. R. Krishnaswamy. 2003. Fault detection during process transitions: a model-based approach. Chemical Engineering Science 58, pages 309–325.

[8] S. A. Billings and S. Y. Fakhouri. 1982. Identification of systems containing linear dynamic and static nonlinear elements. Automatica 18, pages 15–26.

[9] B. V. Bonnlander and A. S. Weigend. 1994. Selecting input variable using mutual information and nonparametric density estimation. In: Proc. of the

International symposium on artificial neural networks (ISANN 94), pages 42–50.

[10] R. Bro. 1996. Multiway calibration. Multilinear PLS. Journal of Chemometrics 10, pages 47–61.

[11] D. A. Brydon, J. J. Ciliers, and M. J. Willis. 1997. Classifying pilot-plant distillation column faults using neural networks. Control Engineering Practice 5, pages 1373–1384.

[12] C-W. Chen and D-Z. Chen. 2001. Prior-knowledge-based feedforward network simulation of true boiling point curve of crude oil. Computers & Chemistry 25, pages 541–550.

[13] J. Chen, R. J. Patton, and H. Z. Zhang. 1996. Design of unknown input observers and robust fault detection filters. International Journal of Control 63, pages 85–105.

[14] R. H. Chen, D. L. Mingori, and J. L. Speyer. 2003. Optimal stochastic fault detection filter. Automatica 39, pages 377–390.

[15] H. Cheng, M. Nikus, and S-L. Jämsä-Jounela. 2008. Fault diagnosis of the paper machine short circulation process using novel dynamic causal digraph reasoning. Journal of Process Control, doi:10.1016/j.jprocont.2007.12.003 .

[16] Y. Chetouani. 2004. Fault detection by using the innovation signal: application to an exothermic reaction. Chemical Engineering and Processing 43, pages 1579–1585.

[17] L. H. Chiang and L. F. Colegrove. 2007. Industrial implementation of on-line multivariate quality control. Chemometrics and intelligent laboratory systems 88, pages 143–153.

[18] L. H. Chiang, E. L . Russell, and R. D. Braatz. 2001. Fault detection and Diagnosis in Industrial Systems. Springer-Verlag, London, 279 pages.

[19] A. Chiuso and G. Picci. 2005. Consistency analysis of some closed-loop subspace identification methods. Automatica 41, pages 377–391.

[20] S. W. Choi, C. Lee, J-M. Lee, J. H. Park, and I-B. Lee. 2005. Fault detection and identification of nonlinear processes based on kernel PCA. Chemometrics and intelligent laboratory systems 75, pages 55–67.

[21] S. W. Choi, J. Morris, and I-B. Lee. 2008. Nonlinear multiscale modelling for fault detection and identification. Chemical Engineering Science 63, pages 2252–2266.

[22] C. T. Chou and M. Verhaegen. 1997. Subspace Algorithms for the Identification of Multivariable Dynamic Errors-in-Variables Models. Automatica 33, pages 1857–1869.

[23] F. N. Chowdhury and J. L. Aravena. 1998. A modular methodology for fast fault detection and classification in power systems. IEEE Transactions on Control Systems Technology 6, pages 623–634.

[24] M. A. Costa, A. de Pádua Braga, and B. R. de Menedez. 2007. Improving generalization of MLPs with sliding mode control and the Levenberg-Marquardt algorithm. Neurocomputing 70, pages 1342–1347.

[25] S. Dash and V. Venkatasubramanian. 2000. Challenges in the industrial applications of fault diagnostic systems. Computers & Chemical Engineering 24, pages 785–791.

[26] M. Daszykowski, K. Kaczmarek, Y. Vander Heyden, and B. Walczak. 2007. Robust statistics in data analysis - A review Basic concepts. Chemometrics and intelligent laboratory systems 85, pages 203–219.

[27] B. S. Dayal and J. F. MacGregor. 1997. Recursive exponentially weighted PLS and its applications to adaptive control and prediction. Journal of Process Control 7, pages 169–179.

[28] D. Dong and T. J. McAvoy. 1996. Nonlinear principal component analysis - Based on principal curves and neural networks. Computers & Chemical Engineering 20, pages 65–78.

[29] C. Edwards and S. K. Spurgeon. 1994. On the development of discontinuous observers. International Journal of Control 59, pages 1211–1229.

[30] C. Edwards, S. K. Spurgeon, and R. J. Patton. 2000. Sliding mode observers for fault detection and isolation. Automatica 36, pages 541–553.

[31] B. Fritzke. 1995. Growing Grid - a self-organizing network with constant neighborhood range and adaptation strength. Neural Processing Letters 2, pages 9–13.

[32] A. Georgakis, C. Kotropoulos, A. Xafopoulos, and I. Pitas. 2004. Marginal median SOM for document organization and retrieval. Neural Networks 17, pages 365–377.

[33] J. Gertler. 1997. Fault detection and isolation using parity relations. Control Engineering Practice 5, pages 653–661.

[34] R. Gnanadesikan. 1977. Methods for Statistical Data Analysis of Multivariate Observations. Wiley & Sons, New York, 311 pages.

[35] D. J. Greeff and C. Aldrich. 1998. Empirical modelling of chemical process systems with evolutionary programming. Computers & Chemical Engineering 22, pages 995–1005.

[36] I. Guyon and A. Elisseeff. 2003. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3, pages 1157–1182.

[37] S. Haykin. 1995. Neural Networks: A Comprehensive Foundation, 2nd ed. Prentice-Hall, Upper Saddle River, 842 pages.

[38] K. B. Helland, H. E. Berntsen, O. S. Borgen, and H. Martens. 1992. Recursive algorithm for partial least squares regression. Chemometrics and Intelligent Laboratory Systems 14, pages 129–137.

[39] A. Hirose and T. Nagashima. 2003. Predictive Self-Organizing Map for Vector Quantization of Migratory Signals and Its Application to Mobile Communications. IEEE Transactions on Neural Networks 14, pages 1532–1540.

[40] H. Hotelling. 1933. Analysis of a Complex of Statistical Variables with Principal Components. Journal of Educational Psychology 24, pages 498–520.

[41] W. W. Hsieh. 2007. Nonlinear component analysis of noisy data. Neural Networks 20, pages 434–443.

[42] Y. Huang, J. Gertler, and T. J. McAvoy. 2000. Sensor and actuator fault isolation by structured partial PCA with nonlinear extensions. Journal of Process Control 10, pages 459–469.

[43] D-H. Hwang and C. Han. 1999. Real-time monitoring for a process with multiple operating modes. Control Engineering Practice 7, pages 891–902.

[44] H. Hyötyniemi. 2001. Multivariate Regression - Techniques and Tools. Helsinki University of Technology, Espoo, 159 pages.

[45] A. Hyvärinen and E. Oja. 2000. Independent component analysis: algorithms and applications. Neural Networks 13, pages 411–430.

[46] M. Iri, K. Aoki, E. O'Shima, and H. Matsuyama. 1979. An algorithm for diagnosis of system failures in chemical processes. Computers & Chemical Engineering 3, pages 489–493.

[47] R. Isermann. 1997. Supervision, Fault-Detection and Fault-Diagnosis Methods. Control Engineering Practice 5, pages 639–652.

[48] R. Isermann and P. Balle. 1997. Trends in the Application of Model-based Fault Detection and Diagnosis of Technical Processes. Control Engineering Practise 5, pages 709–719.

[49] J. E. Jackson. 1959. Quality Control Methods for Several Related Variables. Technometrics 1, pages 359–377.

[50] J. E. Jackson. 1991. A user's guide to principal components. Wiley & Sons, New York, 569 pages.

[51] J. E. Jackson and G. S. Mudholkar. 1979. Control procedures for residuals associated with principal component analysis. Technometrics 21, pages 341–349.

[52] C. M. Jaeckle and J. F. MacGregor. 2000. Product Transfer Between Plants Using Historical Process Data. AIChE Journal 46, pages 1989–1997.

[53] M. E. Janusz and V. Venkatasubramanian. 1991. Automatic generation of qualitative descriptions of process trends for fault detection and diagnosis. Engineering Applications of Artificial Intelligence 4, pages 329–339.

[54] F. Jia, E. B. Martin, and A. J. Morris. 1998. Non-linear principal components analysis for process fault detection. Computers & Chemical Engineering 22, pages S851–S854.

[55] L. Jia, M-S. Chiu, and S. S. Ge. 2005. A noniterative neuro-fuzzy based identification method for Hammerstein processes. Journal of Process Control 15, pages 749–761.

[56] W. Joerding and J. Meador. 1991. Encoding A Priori Information in Feedback Networks. Neural Networks 4, pages 847–856.

[57] G. H. John, R. Kohavi, and K. Pfleger. 1994. Irrelevant features and the subset selection problem. In: Proc. of the 11th international conference on machine learning, pages 121–129.

[58] S-L. Jämsä-Jounela, M. Vermasvuori, P. Enden, and S. Haavisto. 2003. A process monitoring system based on the Kohonen self-organizing maps. Control Engineering Practice 11, pages 83–92.

[59] E. Järvinen, M. Sourander, and T. Liikala. 2006. Evaluation of Different FTC Strategies for a Refinery Process. In: Proc. of the the 2nd NeCST Workshop, pages CD–ROM.

[60] J. A. Kangas, T. K. Kohonen, and J. T. Laaksonen. 1990. Variants of Self-Organizing Maps. IEEE Transactions on Neural Networks 1, pages 93–99.

[61] M. Kano, S. Hasebe, I. Hashimoto, and H. Ohno. 2001. A new multivariate statistical process monitoring method using principal component analysis. Computers & Chemical Engineering 25, pages 1103–1113.

[62] M. Kano, S. Hasebe, I. Hashimoto, and H. Ohno. 2002. Statistical process monitoring based on dissimilarity of process data. AIChE Journal 48, pages 1231–1240.

[63] M. Kavcic and D. Juricic. 2000. CAD for fault tree-based diagnosis of industrial processes. Engineering Applications of Artificial Intelligence 14, pages 203–216.

[64] K. O. Kim and M. J. Zuo. 2007. Two fault classification methods for large systems when available data are limited. Reliability Engineering & System Safety 92, pages 585–592.

[65] M. Kinnunen. 2004. Master of Science Thesis: Integration of first-principles process models into data-based monitoring. Helsinki University of Technology, 96 pages.

[66] R. Kohavi and G. John. 1997. Wrappers for feature subset selection. Artificial Intelligence 97, pages 273–324.

[67] R. Kohavi and D. Sommerfield. 1995. Feature subset selection using the wrapper model: Overfitting and dynamic search space topology. In: Proc. of the 1st Int. Conf. on Knowledge discovery and Data mining (KDD-95), pages 192–197.

[68] T. Kohonen. 1990. The Self-Organizing Map. Proceedings of the IEEE 78, pages 1464–1480.

[69] T. Komulainen. 2003. Master of Science Thesis: Online-Monitoring of a Dearomatization Process. Helsinki University of Technology, 135 pages.

[70] T. Komulainen, M. Sourander, and S-L. Jämsä-Jounela. 2004. An online application of dynamic PLS to a dearomatization process. Computers & Chemical Engineering 28, pages 2611–2619.

[71] K. Kosanke, F.B. Vernadat, and M. Zelm. 1999. CIMOSA: enterprise engineering and integration. Computers in Industry 40, pages 83–97.

[72] K. A. Kosanovich, M. J. Piovoso, K. S. Dahl, J. F. MacGregor, and P. Nomikos. 1994. Multi-Way PCA Applied to an Industrial Batch Process. In: Proc. of the American Control Conference, pages 1294–1298.

[73] T. Kourti. 2002. Process Analysis and Abnormal Situation Detection: From Theory to Practice. IEEE Control Systems Magazine 20, pages 10–25.

[74] T. Kourti, P. Nomikos, and J. F. MacGregor. 1995. Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. Journal of Process Control 5, pages 277–284.

[75] J. R. Koza. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge Massachussets, 840 pages.

[76] M. Kramer. 1991. Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. AiChE Journal 37, pages 233–243.

[77] M. Kudo and J. Sklansky. 2000. Comparison of algorithms that select features for pattern classifiers. Pattern Recognition 33, pages 25–41.

[78] P. Kämpjärvi, M. Sourander, T. Komulainen, M. Nikus, N. Vatanski, and S-L. Jämsä-Jounela. 2007. Online Analyser Validation and Process Fault Diagnosis for Ethylene Cracking Process under MPC Feedback. Control Engineering Practice 16, pages 1–13.

[79] S. Laine, K. Pulkkinen, and S-L. Jämsä-Jounela. 2000. On-line determination of the concentrator feed type at Outokumpu Hitura Mine. Minerals Engineering 13, pages 881–895.

[80] S. Laksminarayanan, S. L. Shah, and K. Nandakumar. 1995. Identification of hammerstein models using multivariate statistical tools. Chemical Engineering Science 50, pages 3599–3613.

[81] P. Langley. 1994. Selection of relevant features in machine learning. In: Proc. of AAAI Fall Symposium on Relevance, pages 140–144.

[82] W. Larimore. 1990. Canonical variate analysis in identification, filtering, and adaptive control. In: Proc. of 29th IEEE conference on Decision and Control, pages 596–604.

[83] J-M. Lee, C. K. Yoo, and I-B. Lee. 2004. Statistical monitoring of dynamic processes based on dynamic independent component analysis. Chemical Engineering Science 59, pages 2995–3006.

[84] J-M. Lee, C. K. Yoo, and I-B. Lee. 2004. Statistical process monitoring with independent component analysis. Journal of Process Control 14, pages 467–485.

[85] R. P. Leger, W. J. Garland, and W. F. S. Poehlman. 1996. Fault detection and diagnosis using statistical control charts and artificial neural networks. Artificial Intelligence in Engineering 12, pages 35–47.

[86] B. Lennox and D. Sandoz. 2002. Recent experiences in the industrial exploitation of principle component based fault detection methods. In: Proc. of IEEE international symposium on computer aided control system design, pages xxxv–xliv.

[87] J. A. Leonard and M. A. Kramer. 1991. Radial basis function networks for classifying process faults. IEEE Control Systems Magazine 11, pages 31–38.

[88] W. Li and S. J. Qin. 2001. Consistent dynamic PCA based on errors-in-variables subspace identification. Journal of Process Control 11, pages 661–678.

[89] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin. 2000. Recursive PCA for adaptive process monitoring. Journal of Process Control 10, pages 471–486.

[90] T. Liikala. 2005. Master of Science Thesis: The use of fault detection in model predictive control of a refinery process unit. Helsinki University of Technology, 98 pages.

[91] W. Lin, Y. Qian, and X. Li. 2000. Nonlinear dynamic principal component analysis for on-line process monitoring and diagnosis. Computers & Chemical Engineering 24, pages 423–429.

[92] F. Lindgren, P. Geladi, and S. Wold. 1993. The kernel algorithm for PLS. Journal of chemometrics 7, pages 45–59.

[93] H. A. Linstone and M. Turof (Ed.s). 1975. The Delphi method: techniques and applications. Addison-Wesley, Massachusetts, 620 pages.

[94] C. H. Lo, Y. K. Wong, and A. B. Rad. 2004. Model-based fault diagnosis in continuous dynamic systems. ISA Transactions 43, pages 459–475.

[95] J. MacGregor, C. Jaeckle, C. Kiparissides, and M. Koutoudi. 1994. Process monitoring and diagnosis by multiblock PLS methods. AIChE Journal 40, pages 826–838.

[96] J. M. Maciejowski. 1999. Modelling and predictive control: enabling technologies for reconfiguration. Annual reviews in Control 23, pages 13–23.

[97] O. Marjanovic, B. Lennox, D. Sandoz, K. Smith, and M. Crofts. 2006. Real-time monitoring of an industrial batch process. Computers & Chemical Engineering 30, pages 1476–1481.

[98] E. C. Mathouse, A. C. Tamhane, and R. S. H. Mah. 1996. Nonlinear partial least squares. Computers & Chemical Engineering 21, pages 875–890.

[99] M.R. Maurya, R. Rengaswamy, and V. Venkatasubramanian. 2005. Fault Diagnosis by Qualitative Trend Analysis of the Principal Components. Chemical Engineering Research and Design 83, pages 1122–1132.

[100] M.R. Maurya, R. Rengaswamy, and V. Venkatasubramanian. 2007. A Signed Directed Graph and Qualitative Trend Analysis-Based Framework for Incipient Fault Diagnosis. Chemical Engineering Research and Design 85, pages 1407–1422.

[101] R. K. Mehra and J. Peschon. 1971. An innovations approach to fault detection and diagnosis in dynamic systems. Automatica 7, pages 637–640.

[102] M. R. Meireles, P. E. Almeida, and M. G. Simoes. 2003. A Comprehensive Review for Industrial Applicability of Artificial Neural Networks. IEEE Transactions on Industrial Electronics 50, pages 585–601.

[103] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter (Eds). 1996. Bounding approaches to system identification. Plenum Press, New York, 557 pages.

[104] M. Misra, H. H. Yue, S. J. Qin, and C. Ling. 2002. Multivariate process monitoring and fault diagnosis by multi-scale PCA. Computers & Chemical Engineering 26, pages 1281–1239.

[105] J. Moody and C. J. Darken. 1989. Fast Learning in Networks of Locally-tuned Processing Units. Neural Computation 1, pages 281–294.

[106] V. Moschou, D. Ververidis, and C. Kotropoulos. 2007. Assessment of self-organizing mapnext term variants for clustering with application to redistribution of emotional speech patterns. Neurocomputing 71, pages 147–156.

[107] M. Mrugalski, M. Witczak, and J. Korbicz. Confidence estimation of the multi-layer perceptron and its application in fault detection systems. Engineering Applications of Artificial Intelligence, doi:10.1016/j.engappai.2007.09.008 .

[108] S. G. Muñoz, J. F. MacGregor, and T. Kourti. 2005. Product transfer between sites using Joint-Y PLS. Chemometrics and Intelligent Laboratory Systems 79, pages 101–114.

[109] K. S. Narendra and P. G. Gallman. 1966. An iterative method for the identification of nonlinear systems using a Hammerstein model. IEEE Trans. Automatic Control 11, pages 546–550.

[110] P. Nomikos and J. F. MacGregor. 1994. Monitoring Batch Processes Using Multiway Principal Component Analysis. AIChE Journal 40, pages 1361–1375.

[111] P. Nomikos and J. F. MacGregor. 1995. Multi-way partial least squares in monitoring batch processes. Chemometrics and intelligent laboratory systems 30, pages 97–108.

[112] H. Noura, D. Sauter, F. Hamelin, and D. Theilliol. 2000. Fault Tolerant Control in Dynamic Systems : Application to a winding machine. IEEE Control Systems Magazine 20, pages 33–49.

[113] B. A. Ogunnaike and W. H. Ray. 1994. Process dynamics, Modeling and Control. Oxford University Press, New York, 1260 pages.

[114] E. S. Page. 1954. Continuous Inspection Schemes. Biometrika 41, pages 100–115.

[115] Y. H. Pao, S. M. Philips, and D. J. Sobajic. 1992. Neural-net computing and intelligent control systems. International Journal of Control 56, pages 263–289.

[116] Y. Papadopoulos. 2003. Model-based system monitoring and diagnosis of failures using statecharts and fault trees. Reliability Engineering & System Safety 81, pages 325–341.

[117] K. Patan and J. Korbicz. 2007. Fault detection in catalytic cracking converter by means of probability density approximation. Engineering Applications of Artificial Intelligence 20, pages 912–923.

[118] K. Patan and T. Parisini. 2005. Identification of neural dynamic models for fault detection and isolation: the case of a real sugar evaporation process. Journal of Process Control 15, pages 67–79.

[119] J. C. Patra and A van den Bos. 2000. Modelling of and intelligent pressure sensor using functional link artificial neural networks. ISA Transactions 39, pages 15–27.

[120] R. J. Patton and J. Chen. 1997. Observer-based fault detection and isolation: Robustness and applications. Control Engineering Practice 5, pages 671–682.

[121] K. Pearson. 1901. On lines and planes of closest fit to systems of points in space. Philosophical Magazine 2, pages 559–572.

[122] I. Pitas, C. Kotropoulos, N. Nikolaidis, R. Yang, and M Gabbouj. 1996. Order statistics learning vector quantizer. IEEE Transactions on Image Processing 5, pages 1048–1053.

[123] P. P. Polycarpou and A. J. Helmicki. 1995. Automated fault detection and accommodation: a learning systems approach. IEEE Transactions on Systems, Man, and Cybernetics 25, pages 1447–1458.

[124] M. Pottmann, H. Unbehauen, and D. E. Seborg. 1993. Application of a General Multi-Model Approach for Identification of Highly Nonlinear Processes - a Case Study. International Journal of Control 57, pages 97–102.

[125] J. Prakash, S. C. Patwardhan, and S. A. Narasimhan. 2002. Supervisory Approach to Fault-Tolerant Control of Linear Multivariable Systems. Industrial Chemical Engineering Research 41, pages 2270–2281.

[126] T. N. Pranatyasto and S. J. Qin. 2001. Sensor validation and process fault diagnosis for FCC units under MPC feedback. Control Engineering Practice 9, pages 877–888.

[127] P. Pudil, J. Novovicova, and J. Kittler. 1994. Floating search methods in feature selection. Pattern Recognition Letters 15, pages 1119–1125.

[128] V. Puig, M. Witczak, F. Nejjari, J. Quevedo, and J. Korbicz. 2007. A GMDH neural network-based approach to passive robust fault detection using a constraint satisfaction backward test. Engineering Application of Artificial Intelligence 20, pages 886–897.

[129] K. Pöllänen, A. Häkkinen, S.-P. Reinikainen, J. Rantanen, and P. Minkkinen. 2006. Dynamic PCA-based MSPC charts for nucleation prediction in batch cooling crystallization processes. Chemometrics and Intelligent Laboratory Systems 84, pages 126–133.

[130] S. J. Qin. 1993. Partial Least Squares Regression for Recursive System Identification. In: Proc. of the 32th conference on Decision and Control, pages 2617–2622.

[131] S. J. Qin. 1998. Recursive PLS algorithms for adaptive data modeling. Computers & Chemical Engineering 22, pages 503–514.

[132] S. J. Qin and T. J. McAvoy. 1992. Nonlinear PLS modelling using neural-networks. Computers & Chemical Engineering 16, pages 379–391.

[133] S. J. Qin, S. Valle, and M. J. Piovoso. 2001. On unifying multiblock analysis with application to decentralized process monitoring. Journal of Chemometrics 15, pages 715–742.

[134] H. J. Ramaker, E. N. M. van Sprang, S. P. Gurden, J. A. Westerhuis, and A. K. Smilde. 2002. Improved monitoring of batch processes by incorporating external information. Journal of Process Control 12, pages 569–576.

[135] C. R. Rao. 1964. The use and interpretation of principal component analysis in applied research. Sankhya, Ser. A 26, pages 329–359.

[136] A. Rauber, D. Merkl, and M. Dittenbach. 2002. The growing hierarchical self-organizing map: Exploratory analysis of high dimensional data. IEEE Transactions on Neural Networks 13, pages 1331–1341.

[137] H. E. Rausch. 1995. Autonomous control reconfiguration. IEEE Control Systems Magazine 15, pages 37–49.

[138] J. Reunanen. 2003. Overfitting in Making Comparisons Between Variable Selection Methods. Journal of Machine Learning 3, pages 1371–1382.

[139] F. Rosenblatt. 1958. The Perceptron: a Probabilistic model for lnformation Stogare and Organization in the Brain. Physiological Review 65, pages 386–408.

[140] D. Ruiz, J. M. Nougués, and L. Puigjaner. 2001. Fault diagnosis support system for complex chemical plants. Computers & Chemical Engineering 25, pages 151–160.

[141] E. L. Russell, L. H. Chiang, and R. D. Braatz. 2000. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. Chemometrics and Intelligent Laboratory Systems 51, pages 81–93.

[142] T. Scheffer and R. Herbrich. 1997. Unbiased assessment of learning algorithms. In: Proc. of the 15th joint conf on artificial intelligence (IJCAI 97), pages 798–803.

[143] B. Schölkopf, A. Smola, and K-R. Müller. 1998. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10, pages 1299–1319.

[144] H. Shah-Hosseini and R. Safabakhsh. 2003. TASOM: A New Time Adaptive Self-Organizing Map. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 33, pages 271–282.

[145] R. Shao, F. Jia, E. B. Martin, and A. J. Morris. 1999. Wavelets and non-linear principal components analysis for process monitoring. Control Engineering Practice 7, pages 865–879.

[146] J. J. E. Slotine, J. K. Hedrick, and E. A. Misawa. 1987. On sliding observers for nonlinear systems. Journal of Dynamic Systems, Measurements, and Control 109, pages 245–252.

[147] M. Sourander, T. Liikala, and K. Koivisto. 2006. FTC strategies in model predictive control of a dearomatisation process. In: In Proc. of 6th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, pages 325–330.

[148] G. Stephanopoulos and C. Han. 1996. Intelligent systems in process engineering: a review. Computers & Chemical Engineering 20, pages 743–791.

[149] Y. Takane and T. Shibayama. 1991. Principal component analysis with external information on both subjects and variables. Psychometrika 56, pages 97–120.

[150] C. P. Tan, F. Crusca, and M. Aldeen. 2008. Extended results on robust state estimation and fault detection. Automatica, doi:10.1016/j.automatica.2007.11.012 .

[151] C. P. Tan and C. Edwards. 2002. Sliding mode observers for detection and reconstruction of sensor faults. Automatica 38, pages 1815–1821.

[152] S. Tan and M.L. Mavrovouniotis. 1995. Reducing data dimensionality through optimizing neural network inputs. AIChE Journal 40, pages 1471–1480.

[153] M. Thompson and M. Kramer. 1994. Modeling Chemical Processes Using Prior Knowledge and Neural Networks. AIChE Journal 40, pages 1328–1340.

[154] R. J. Treasure, U. Kruger, and J. E. Cooper. 2004. Dynamic multivariate statistical process control using subspace identification. Journal of Process Control 14, pages 279–292.

[155] A. C. Tsoi and A. D. Back. 1994. Locally Recurrent Globally Feedforward Networks: A Critical Review of Architectures. IEEE Transactions on Neural Networks 5, pages 229–239.

[156] P. van Overschee and B. de Moor. 1994. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica 30, pages 75–93.

[157] P. van Overschee and B. de Moor. 1995. A unifying theorem for three subspace system identification algorithms. Automatica 31, pages 1853–1864.

[158] P. van Overschee and B. de Moor. 1996. Subspace identification for linear systems: Theory, Implementation, Applications. Kluwer Academic Publishers, 254 pages.

[159] N. Vatanski, M. Vermasvuori, R. Ylinen, S-L. Jämsä-Jounela, M. Sourander, J-B. Leger, and E. Rondeau. 2005. Definition of user needs and a functional architecture for the NeCST system. Project report of Networked Control Systems Tolerant to Faults (NeCST) EU IST-2004-004303.

[160] H. Vedam and V. Venkatasubramanian. 1997. Signed Digrahp Based Multiple Fault Diagnosis. Computers & Chemical Engineering 21, pages S655–s660.

[161] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri. 2003. A review of process fault detection and diagnosis Part II: Qualitative models and search strategies. Computers & Chemical Engineering 27, pages 313–326.

[162] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. 2003. A review of process fault detection and diagnosis Part I: Quantitative model-based methdos. Computers & Chemical Engineering 27, pages 293–311.

[163] M. Verhaegen. 1994. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. Automatica 30, pages 61–74.

[164] B. L. Walcott, M. J. Corless, and S. H. Zak. 1987. Comparative study of state observation techniques. International Journal of Control 45, pages 2109–2132.

[165] B. Walczak and D. L. Massart. 1996. Application of radial basis functions Ű partial least squares to nonlinear pattern recognition problems: diagnosis of process faults. Analytica Chimica Acta 331, pages 187–193.

[166] D. Wang and J. A. Romagnoli. 2005. Robust multi-scale principal component analysis with applications to process monitoring. Journal of Process Control 15, pages 869–882.

[167] J. Wang and J. Qin. 2002. A new subspace identification approach based on principal component analysis. Journal of Process Control 12, pages 841–855.

[168] X. Wang, U. Kruger, and B. Lennox. 2003. Recursive partial least squares algorithms for monitoring complex industrial processes. Control Engineering Practice 11, pages 613–632.

[169] J. Westerhuis, T. Kourti, and J. MacGregor. 1998. Analysis of multiblock and hierarchical PCA and PLS models. Journal of Chemometrics 12, pages 301–321.

[170] H. Wold. 1975. Path models with latent variables: The NIPALS approach. In: H.M. Blalock et al. Quantitative Sociology: International perspectives on mathematical and statistical model building, Academic Press, NY, pages 307–357.

[171] S. Wold. 1992. Nonlinear partial least squares modelling II. Spline inner relation. Chemometrics and Intelligent Laboratory Systems 14, pages 71–84.

[172] S. Wold, K. Esbensen, and P. Geladi. 1987. Principal component analysis. Chemometrics and Intelligent Laboratory Systems 23, pages 37–52.

[173] S. Wold, N. Kettaneh-Wold, and B. Skagerberg. 1989. Nonlinear PLS modeling. Chemometrics and Intelligent Laboratory Systems 7, pages 53–65.

[174] S. Wold, H. Martens, and H. Wold. 1983. The multivariate calibration method in chemistry solved by the PLS method. In: Proc. of Conf. Matrix Pencils Lecture Notes in Mathematics, pages 286–293.

[175] S. Wold, M. Sjöström, and L. Eriksson. 2001. PLS-regression: a basic tool of chemometrics. Chemometrics and intelligent laboratory systems 58, pages 109–130.

[176] Y. Yamamoto and V. Venkatasubramanian. 1990. Integrated approach using neural networks for fault detection and diagnosis. In: Proc. of the International Joint Conference on Neural Networks, pages 317–326.

[177] S. H. Yang, B. H. Chen, and X. Z. Wang. 2000. Neural network based fault diagnosis using unmeasurable inputs. Engineering Applications of Artificial Intelligence 13, pages 345–356.

[178] Z. Yang, K. Suzuki, Y. Shimada, and H. Sayama. 1995. Fuzzy fault diagnostic system based on fault tree analysis. In: Proc. of the 4th IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium, pages 165–170.

[179] S. Yoon and J. MacGregor. 2001. Incorporation of external information into multivariate PCA/PLS models. In: Proc. of the On-line fault detection and supervision in the chemical process industries workshop, pages 121–126.

[180] D. L. Yu, J. B. Gomm, and D. Williams. 1999. Sensor fault diagnosis in a chemical process via RBF neural networks. Control Engineering Practice 7, pages 49–55.

[181] L. Zhang, L. B. Jack, and A. K. Nandi. 2005. Fault detection using genetic programming. Mechanical Systems and Signal Processing 19, pages 271–289.

[182] P. Zhang and S. X. Ding. 2007. Disturbance decoupling in fault detection of linear periodic systems. Automatica 43, pages 1410–1417.

[183] Y. Zhang, M. Dudzic, and V. Vaculik. 2003. Integrated monitoring solution to start-up and run-time operations for continuous casting. Annual Reviews in Control 27, pages 141–149.

[184] Y. Zhou, J. Hahn, and M. S. Mannan. 2003. Fault detection and classification in chemical processes based on neural networks with feature extraction. ISA Transactions 42, pages 651–664.

Appendix A Mathematical descriptions of the utilised modelling methods

PCA is used to analyse only one data set, the process measurements. PCA can therefore be used to detect changes in the process, but it does not provide any information how those changes will affect the quality of the process product. The PCA is based on decomposing process history data $X(m, n)$ into principal component scores, T, and loadings P and residual term E (Jackson, 1990).

$$X = TP' + E \tag{A.1}$$

The PC scores and loadings can be determined by solving an eigenvalue decomposition of the sample covariance matrix S,

$$S = \frac{1}{n-1} X_s' X_s = V \Lambda V' \tag{A.2}$$

where the loading vectors are the orthogonal columns of the matrix V and the diagonal matrix Λ contains the non-negative real eigenvalues, λ_i, representing the variance in the direction of the corresponding loading vector. An alternative way to get the scores and the loadings is to use singular value decomposition:

$$\frac{1}{\sqrt{n-1}} X_s = U \Sigma V' \tag{A.3}$$

where the diagonal matrix Σ contains the non-negative real singular values, σ_i. Singular values and eigenvalues are related as $\lambda_i = \sigma_i^2$. The loading matrix P is constructed of $r < m$ column vectors of V corresponding to r largest eigenvalues (or singular values) and the score matrix $T = XP$. The i$^{\text{th}}$ transformed variable $t_i = x' p_i$ is called the i$^{\text{th}}$ principal component of x, where x is a column vector of measurements and p_i is the i$^{\text{th}}$ row of P. When the data matrix $\hat{X}$ is reconstructed (A.4), the residual E between the original and reconstructed data matrices can be determined (A.5) (Chiang *et al.*, 2001).

$$\hat{X} = TP' \tag{A.4}$$

$$E = X - \hat{X} \tag{A.5}$$

In PCA the first problem is to choose the optimal number of principal components (PCs). For this situation, different approaches have been proposed: (1) Select the first PCs until the break point in the slope of the cumulative explaining curve is reached. (2) Select the PCs which explain more than 1 % of the variation of the data. (3) Select the PCs so that the cumulative explaining capacity is more than 90 %. All the before mentioned methods for selecting the number of PCs serve only as guidelines and the optimal number of PCs should be determined by selecting different numbers of PCs and comparing the FDI performance of the resulting models.

PLS is in principle similar to PCA, but instead of using one data set it uses two, e.g. the process measurements and the product quality information. Utilising the quality information separately gives PLS an edge over PCA as the latent variables are formed so that they explain the variance of the product quality. The basic equations of PLS are given below (Chiang *et al.*, 2001). The auto-scaled predictor data set X_s is decomposed as

$$X_s = TP' + E \tag{A.6}$$

and the scaled predicted set Y_s as

$$Y_s = UQ' + \tilde{F} \tag{A.7}$$

where E and $\tilde{F}$ are residual matrices. The connection between (A.6) and (A.7) is the relation of the score vectors t_i and u_i shown in (A.8).

$$\hat{u}_i = b_i t_i \tag{A.8}$$

where $\hat{u}_i$ is the regression estimate of the score vector u_i and b_i are the regression coefficients. (A.7) can be written in a matrix form

$$\hat{U} = BT \tag{A.9}$$

which can then be used to substitute U in Equation A.7 resulting

$$Y_s = BTQ' + F \tag{A.10}$$

The basic problem in PLS-based modelling is to determine the optimal number of latent variables. This is similar situation to the previously presented selection of PCs and the same methods are used to solve the problem. The second problem is related to the data sets: for effective use of PLS the quality measurements should closely correlate to the process measurements. There should also be more than one quality parameter to get more accurate view of the quality (Nomikos and MacGregor, 1995).

Several subspace methods to determine the matrices A, B, C, and D for a state-space model have been presented in literature. The method used in this study utilises PCA in reducing the quantity of the states (Hyötyniemi, 2001). First, the state-space model is expressed as

$$\begin{cases} x(\kappa+1) = Ax(\kappa) + Bu(\kappa) + \varepsilon(\kappa) \\ y(\kappa) = Cx(\kappa) + Du(\kappa) + \varepsilon(\kappa) \end{cases} \tag{A.11}$$

where $u(\kappa)$ is the system input with dimension nx1, $x(\kappa)$ is the state vector with dimension dx1 and $y(\kappa)$ is the system output with dimension mx1. ε represent a white noise sequence. Input and output data matrices containing information of the past and the future are defined as

$$U_{\substack{past \\ k-2\beta+1}} = \begin{pmatrix} u_{1xn}(1) & u_{1xn}(2) & \dots & u_{1xn}(\beta) \\ \vdots & \vdots & \ddots & \vdots \\ u_{1xn}(k-2\beta+1) & u_{1xn}(k-2\beta+2) & \dots & u_{1xn}(k-\beta) \end{pmatrix} \tag{A.12}$$

$$Y_{past} = \begin{pmatrix} y(1) & y(2) & \dots & y(\beta) \\ \vdots & \vdots & \ddots & \vdots \\ y(k-2\beta+1) & y(k-2\beta+2) & \dots & y(k-\beta) \end{pmatrix} \tag{A.13}$$

$$U_{future} = \begin{pmatrix} u(\beta+1) & u(\beta+2) & \dots & u(2\beta) \\ \vdots & \vdots & \ddots & \vdots \\ u(k-\beta+1) & u(k-\beta+2) & \dots & u(k) \end{pmatrix} \tag{A.14}$$

$$Y_{future} = \begin{pmatrix} y(\beta+1) & y(\beta+2) & \dots & y(2\beta) \\ \vdots & \vdots & \ddots & \vdots \\ y(k-\beta+1) & y(k-\beta+2) & \dots & y(k) \end{pmatrix} \tag{A.15}$$

The model that is being created estimates the future outputs based on all inputs and past outputs. Thus using definitions

$$\chi = (Y_{past}|U_{past}|U_{future}) \;\; and \;\; Z = (Y_{future}) \tag{A.16}$$

there should be a mapping from χ to Z. The mapping can be performed with least squares method, resulting

$$Z_{est} = \chi_{est}(\chi^T\chi)^{-1}\chi^T Z = (Y_{past,est}|U_{past,est}|U_{future,est})(\chi^T\chi)^{-1}\chi^T Z \tag{A.17}$$

Because the values of $U_{future,est}$ are not known, the model is divided into parts

$$Z_{past} + Z_{future} = (Y_{past,est}|U_{past,est}|0)(\chi^T\chi)^{-1}\chi^T Z + (0|0|U_{future,est})(\chi^T\chi)^{-1}\chi^T Z \tag{A.18}$$

now the Z_{past} contains all available information from the past and a refined data matrix is determined as

$$X = Z_{past} = (Y_{past,est}|U_{past,est}|0)(\chi^T\chi)^{-1}\chi^T Z \tag{A.19}$$

This matrix is set as the preliminary system state matrix. The dimension of the highly redundant states is high and thus it is reduced with a suitable method, PCA. The final system matrices are determined by defining X^- as a submatrix of X with the last row eliminated and X^+ as a submatrix of X with the first row eliminated and the input and the output matrices (A.20)

$$U_{k-\beta xn} = \begin{pmatrix} u^T(\beta) \\ \vdots \\ u^T(k-\beta-1) \end{pmatrix} \;\; and \;\; Y_{k-\beta xm} = \begin{pmatrix} y^T(\beta) \\ \vdots \\ y^T(k-\beta-1) \end{pmatrix} \tag{A.20}$$

The model can be written as

$$\left(\begin{array}{c} x(\kappa+1) \\ \hline y(\kappa) \end{array}\right) = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right) \left(\begin{array}{c} x(\kappa) \\ \hline y(\kappa) \end{array}\right) \tag{A.21}$$

and there holds

$$(X^+|Y) = (X^-|U) \left(\begin{array}{c|c} A^T & C^T \\ \hline B^T & D^T \end{array} \right) \tag{A.22}$$

so, finally the matrices A, B, C and D can be solved in least squares sense:

$$\left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right) = (X^+|Y)^T (X^-|U)((X^-|U)^T (X^-|U))^{-1} \tag{A.23}$$

Multilayer perceptron is one of the most commonly used neural network algorithms, according to a study in 1995, 81,2 % of all ANN based applications used MLP structure (Haykin, 1995). The MLP models usually consist of 3 layers of perceptrons that perform the nonlinear input - output mapping. MLP is an interesting tool for process modelling as it can approximate functions to arbitrary accuracy if the number of perceptrons is sufficiently high. Unfortunately, this also makes the MLP models vulnerable to over fitting, as they can also model the noise in the training data. The output of a single perceptron is given by (A.24).

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) \tag{A.24}$$

where y is the output, w is a weight, x in an input, b is a bias term and φ is the nonlinear sigmoid activation function

$$y_s = \frac{1}{1 + e^{-a}} \tag{A.25}$$

For a single hidden layer MLP, the model mapping inputs to outputs can be presented as:

$$y = B\varphi(Ax + a) + b \tag{A.26}$$

where y is the vector of outputs, x the vector of inputs, A and a are the weight and bias vectors of the first layer, B and b are the weight and bias vectors of the second layer, φ denotes the nonlinear activation function. The training of the MLP can be performed with the Levenberg-Marquardt (LM) method in which the changes of weights, Δw, is determined by solving (A.27) (Costa *et al.*, 2007).

$$\Delta w = \left(J^T(w) J(w) + \mu I \right)^{-1} J^T(w) e(w) \tag{A.27}$$

where w is the weight vector, e is the vector of output errors, $J(w)$ is the Jacobian matrix of the partial derivativies of e with respect to the weights, I is an identity matrix and μ is a parameter that is adjusted during the training.

SOMs are two layer neural networks used for analysing the operating states and for classifying measurement vectors. SOM maps multidimensional measurement data (the first layer) into two-dimensional plane (the second layer) so that the topology of the data is preserved. One of the benefits of the SOM algorithm is that the two-dimensional representation of the data is visual and easy to understand compared to the original n-dimensional data. The size i.e. the maximum number of classes and the shape of the SOM are determined before the training. The SOM uses competitive, unsupervised training consisting of two parts: (1) determining the distances between the neurons and a new measurement vector (A.28) and (2) updating the weights of the neuron having the shortest distance with the measurement vector and the the weights of its neighbouring neurons with (A.29).

$$D(x, w^i) = \sum_{k=1}^{m} (k_k - w_k^i)^2 \tag{A.28}$$

where $D(x, w^i)$ is the distance between the measurement vector and a neuron i, m is the number of variables, x_k is the k^{th} element of the measurement vector and w_k^i is the k^{th} element of the weight vector of the neuron i.

$$w_k^i(t+1) = w_k^i(t) + h_{ci}(t)(x_k(t) - w_k^i(t)) \tag{A.29}$$

where h_{ci} is scalar Gaussian kernel function:

$$h_{ci} = \alpha(t) exp(-\frac{\|r_i - r_c\|^2}{2\sigma(t)^2}) \tag{A.30}$$

where α is the rate of learning foefficient, σ is the size of the affected neighbourhood and r's are coordinates of neurons. (Kohonen, 1990)

Appendix B Graphical presentations of the performance of the models for the second testing experiment

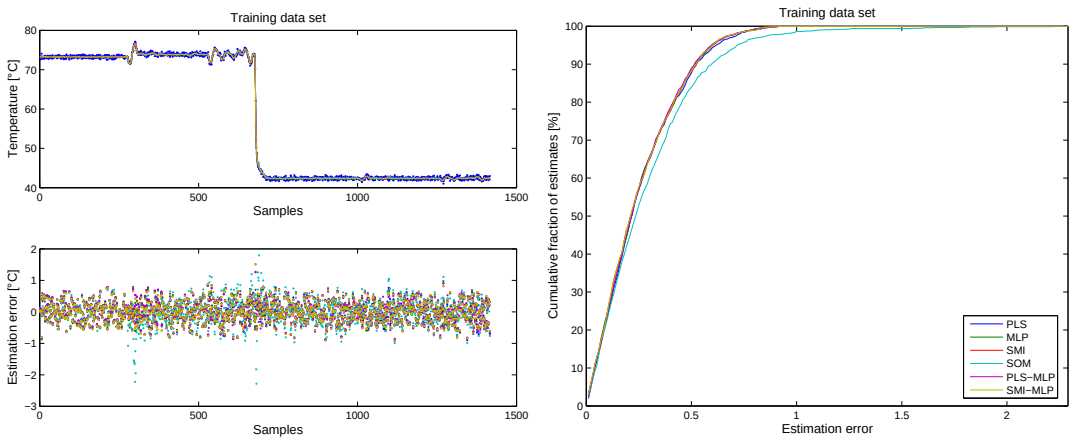

Figure B.1: Simulated and estimated values of FP for the training data set (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

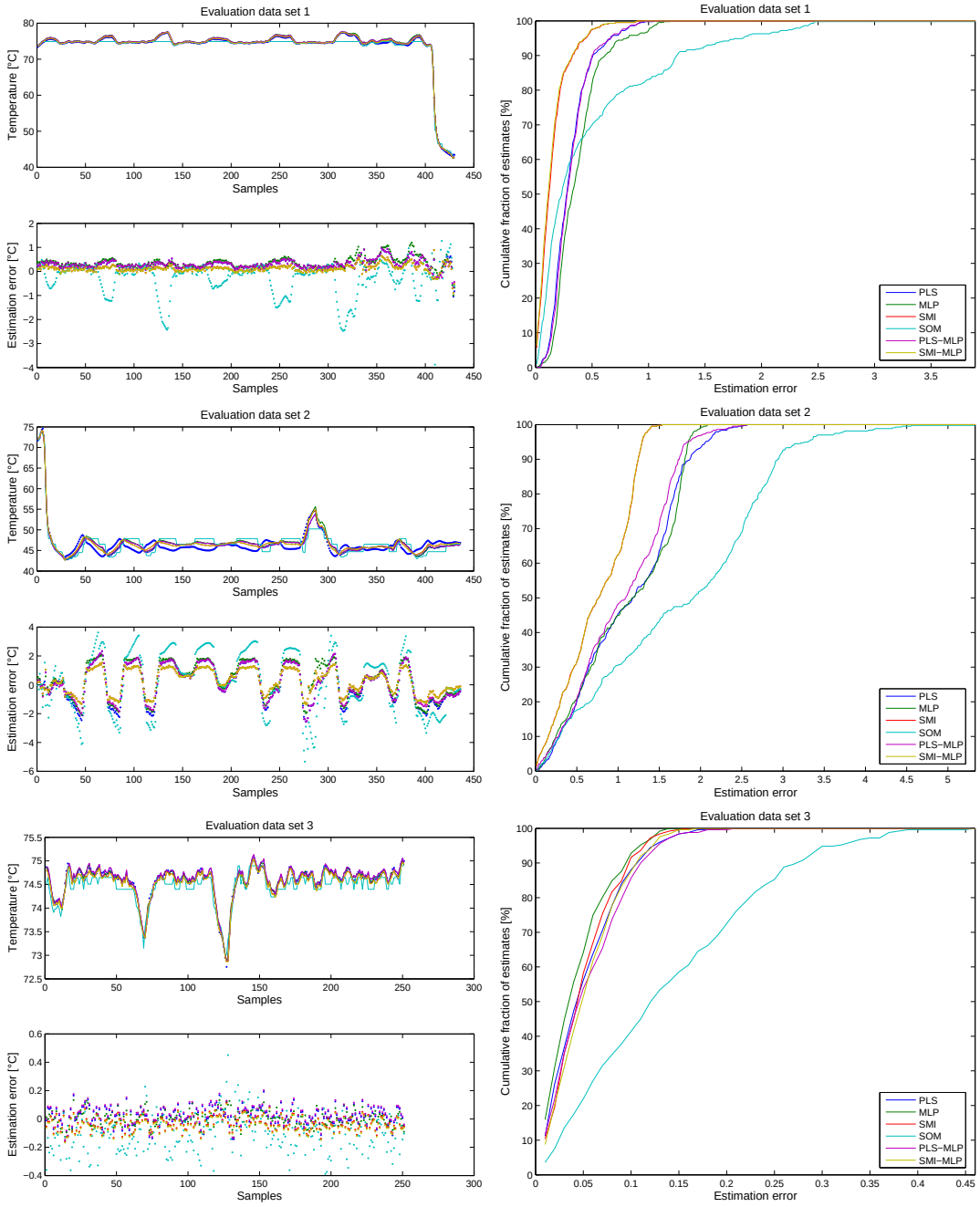

Figure B.2: Simulated and estimated values of FP for the evaluation data sets 1 to 3 (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

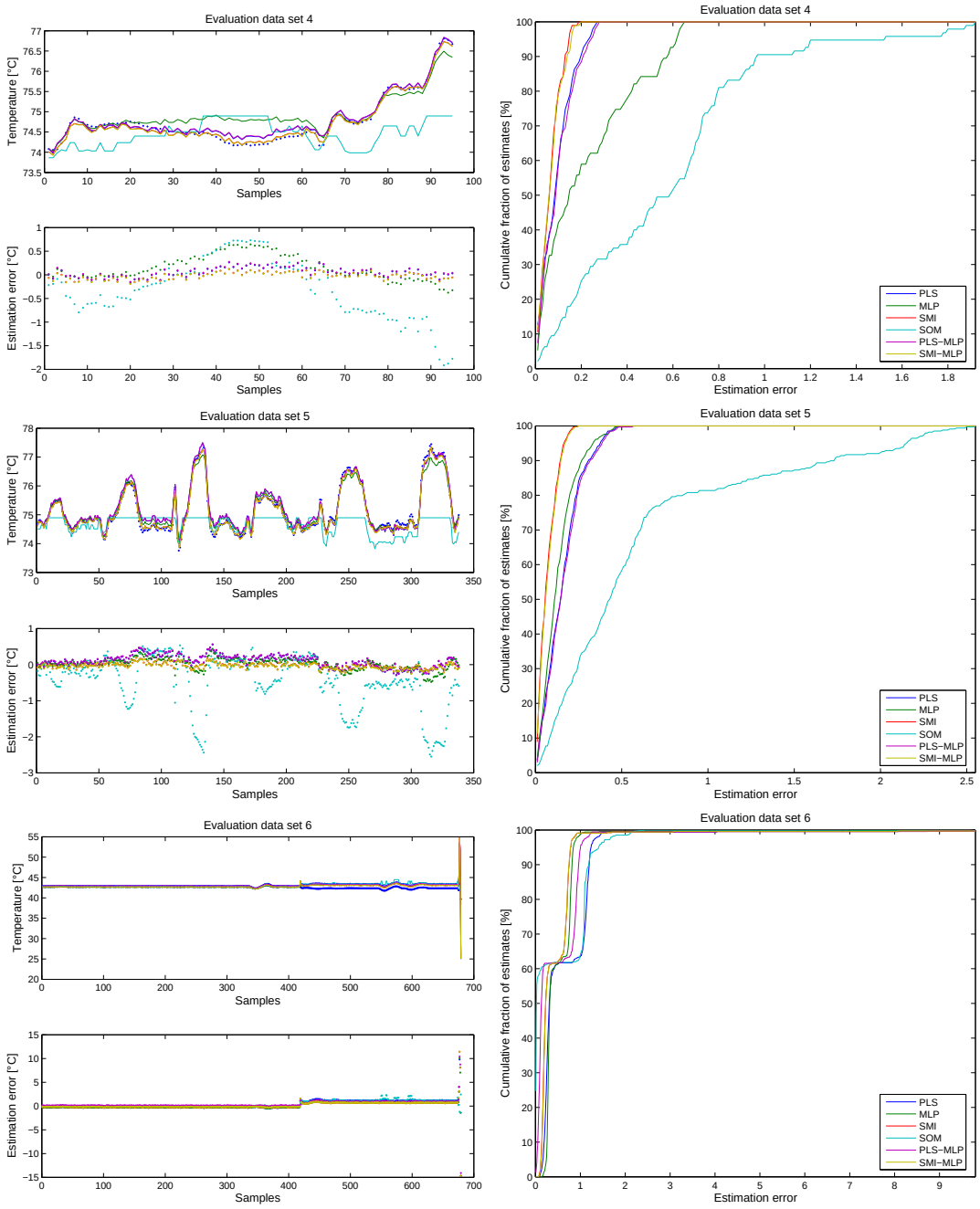

Figure B.3: Simulated and estimated values of FP for the evaluation data set 4 to 6 (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

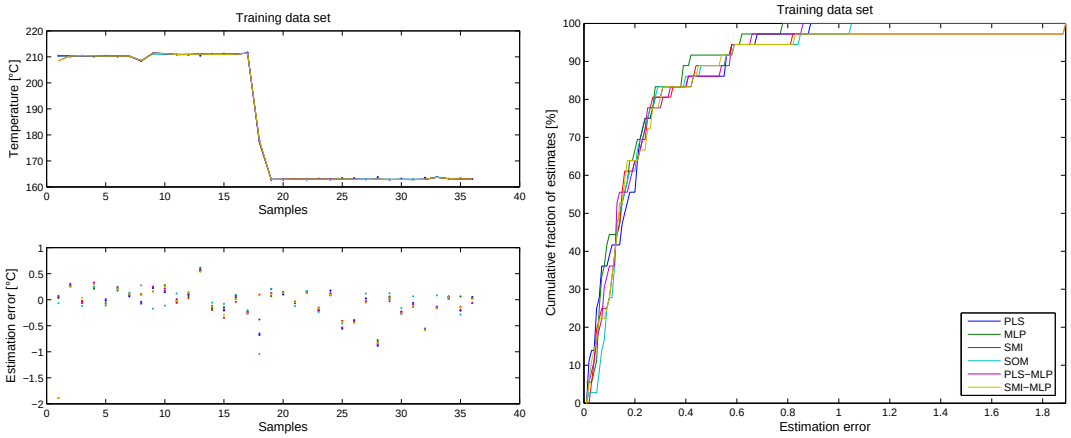

Figure B.4: Simulated and estimated values of IBP for the training data set (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

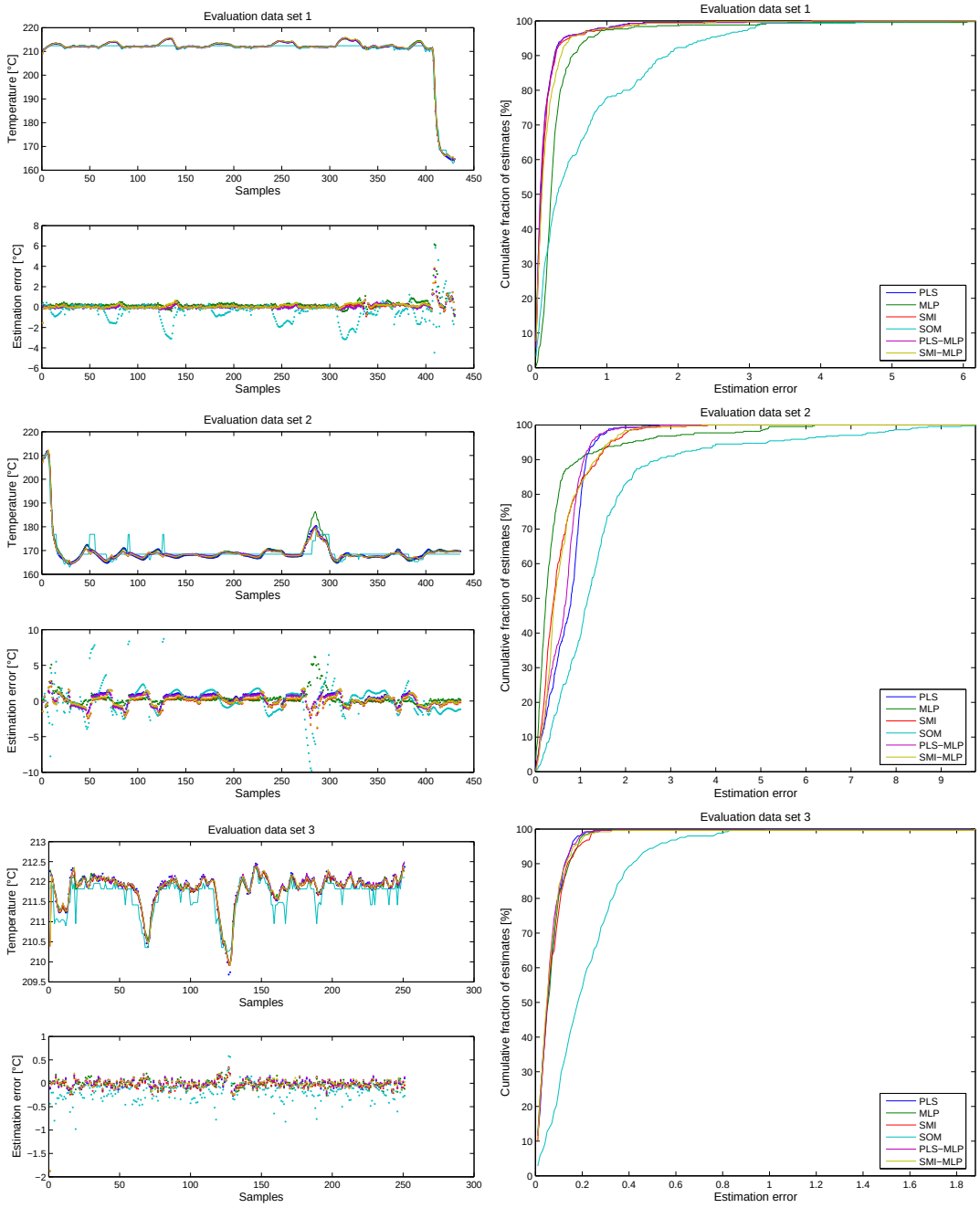

Figure B.5: Simulated and estimated values of IBP for the evaluation data sets 1 to 3 (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

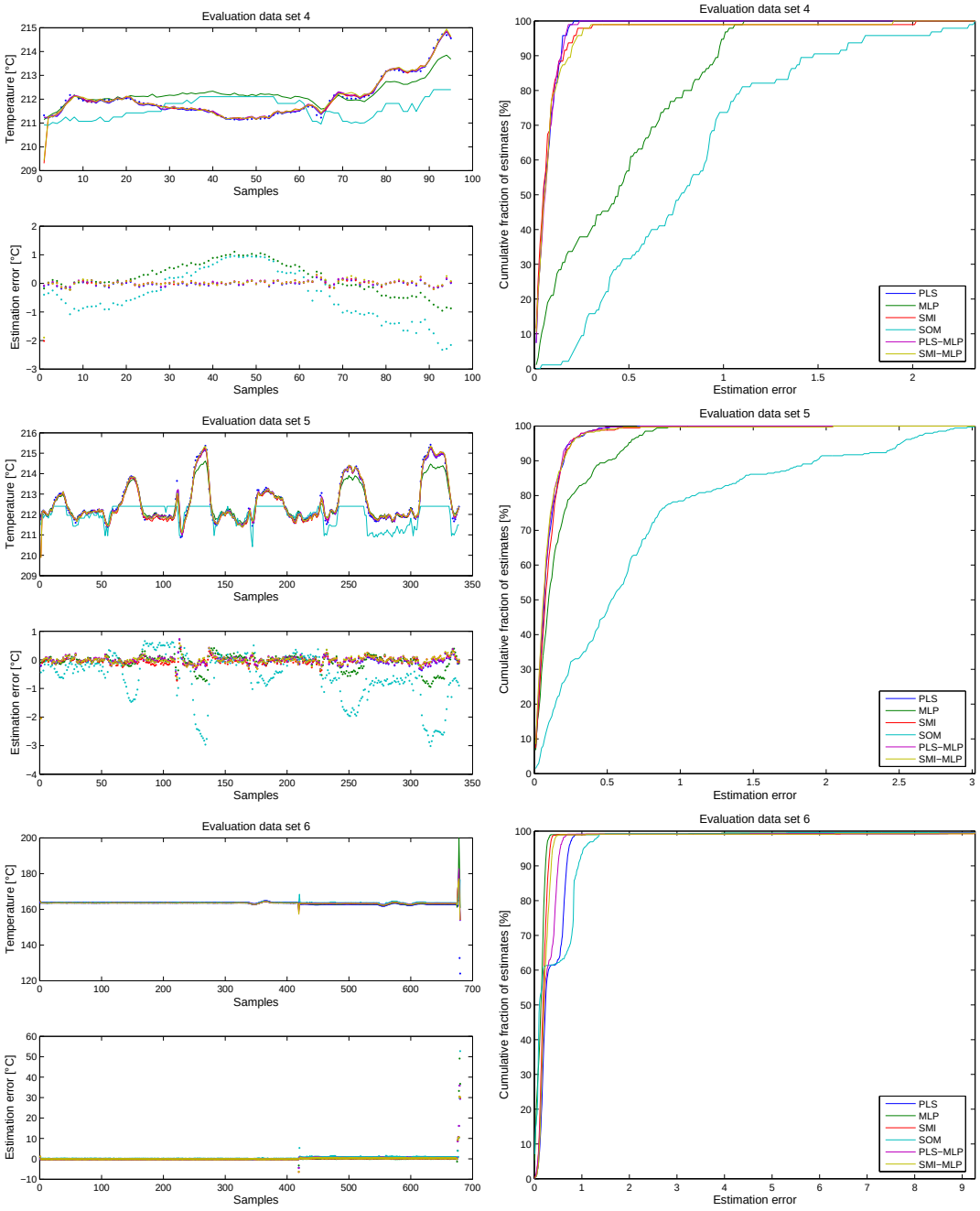

Figure B.6: Simulated and estimated values of IBP for the evaluation data set 4 to 6 (top left), corresponding estimation errors (bottom left) and the cumulative distribution of the estimation errors (right)

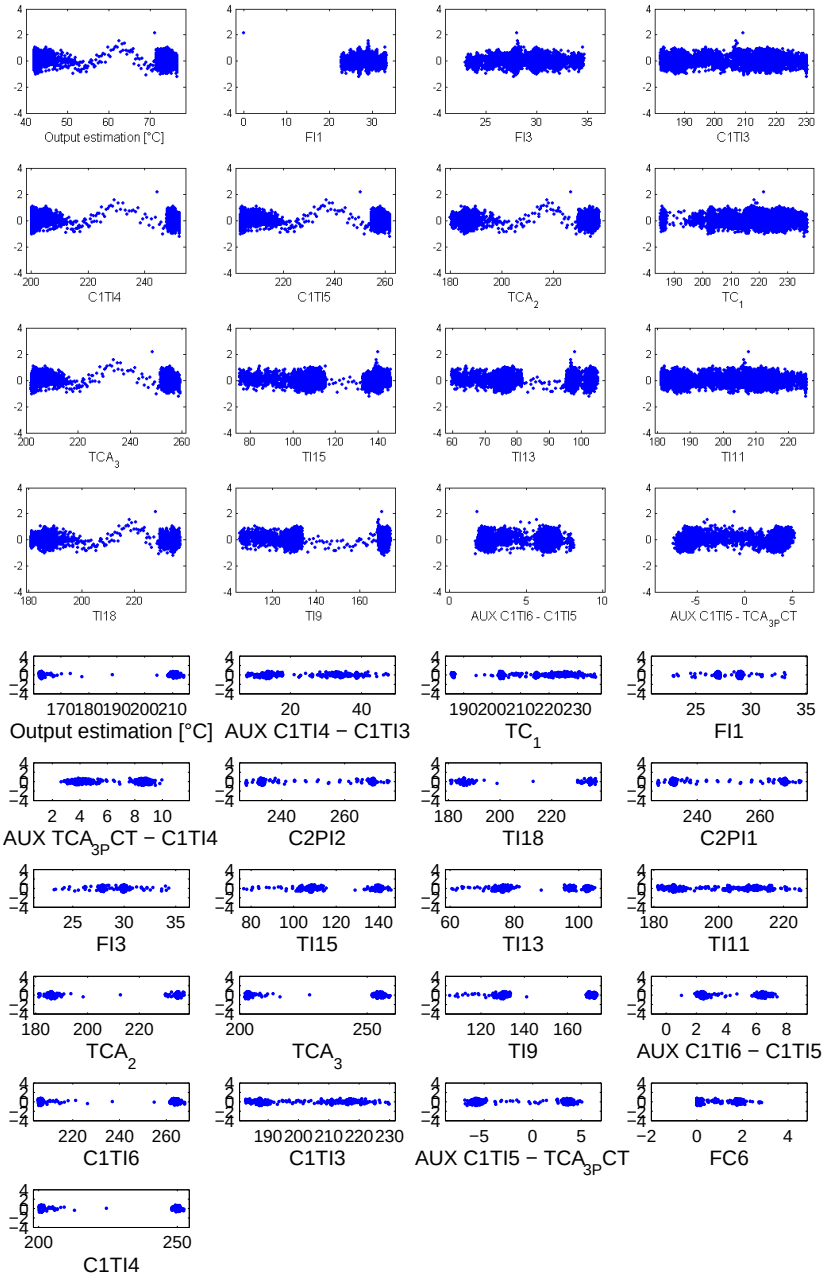

Figure B.7: Dependence of the estimation errors on the estimation and the input variables, training data, MLP FP model (top) and IBP model (bottom)

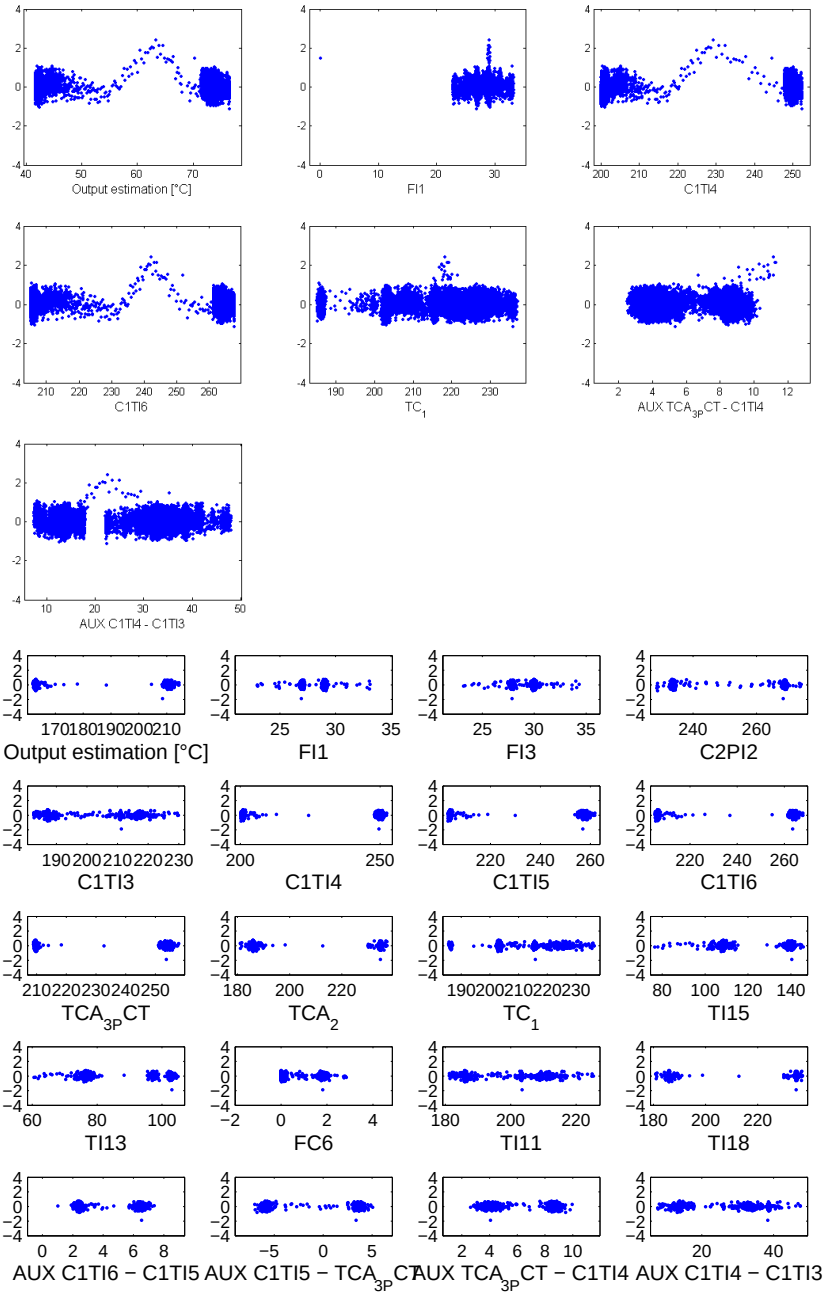

Figure B.8: Dependence of the estimation errors on the estimation and the input variables, training data, SMI FP model (top) and IBP model (bottom)

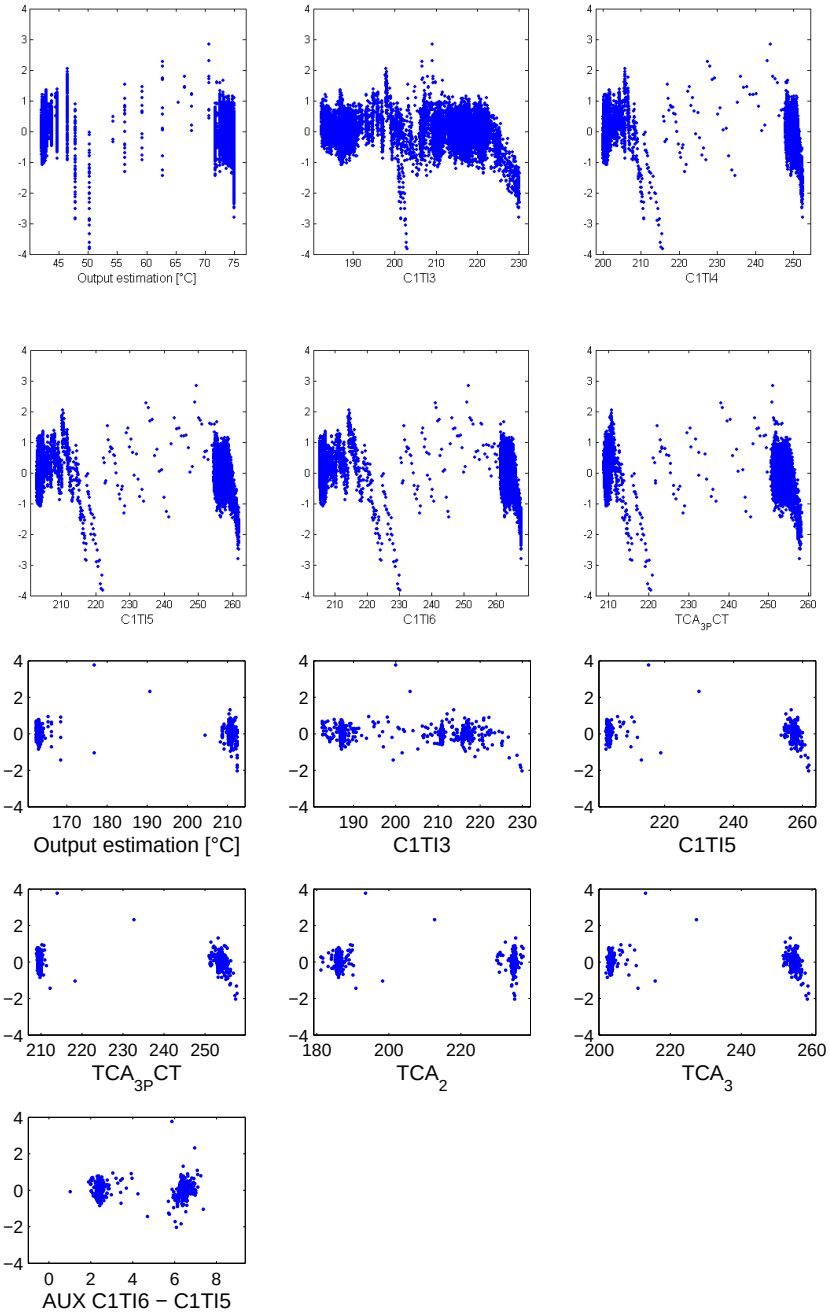

Figure B.9: Dependence of the estimation errors on the estimation and the input variables, training data, SOM FP model (top) and IBP model (bottom)

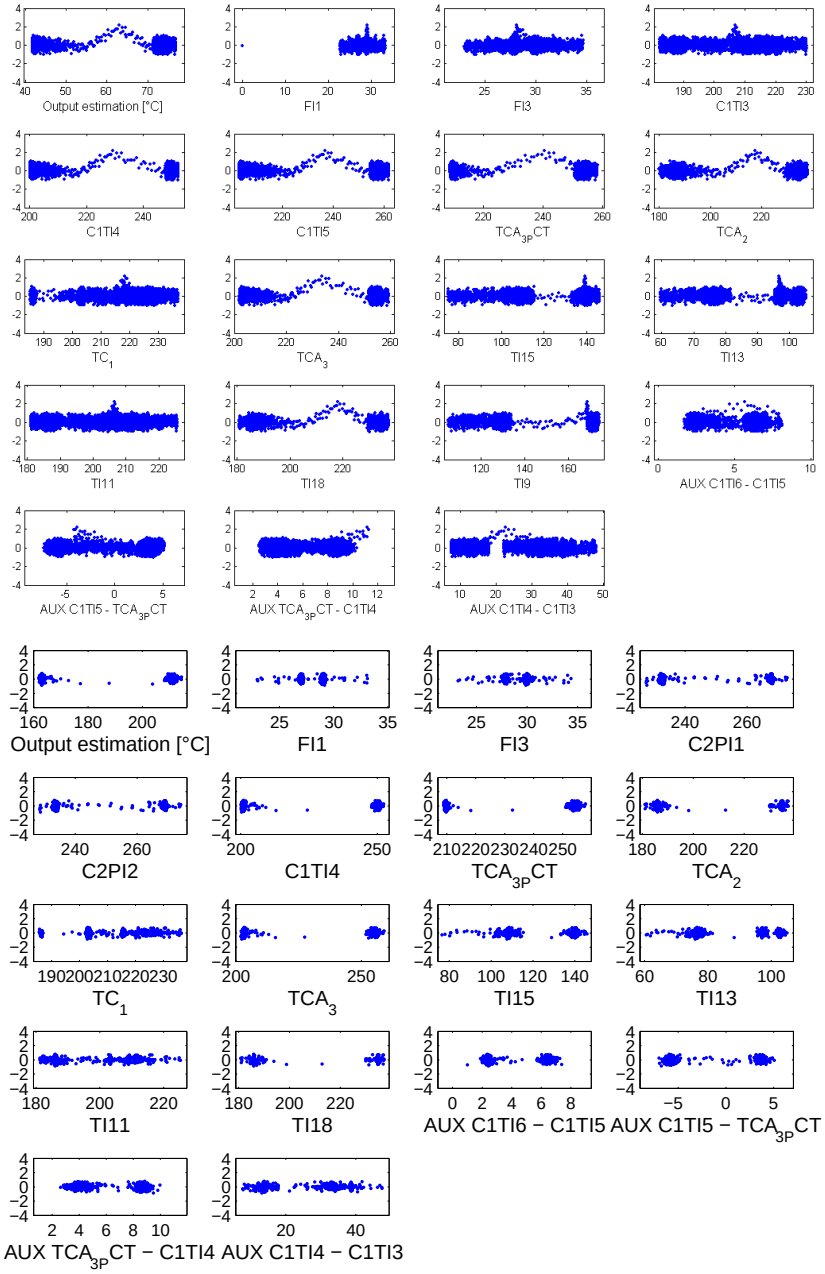

Figure B.10: Dependence of the estimation errors on the estimation and the input variables, training data, PLS-MLP FP model (top) and IBP model (bottom)

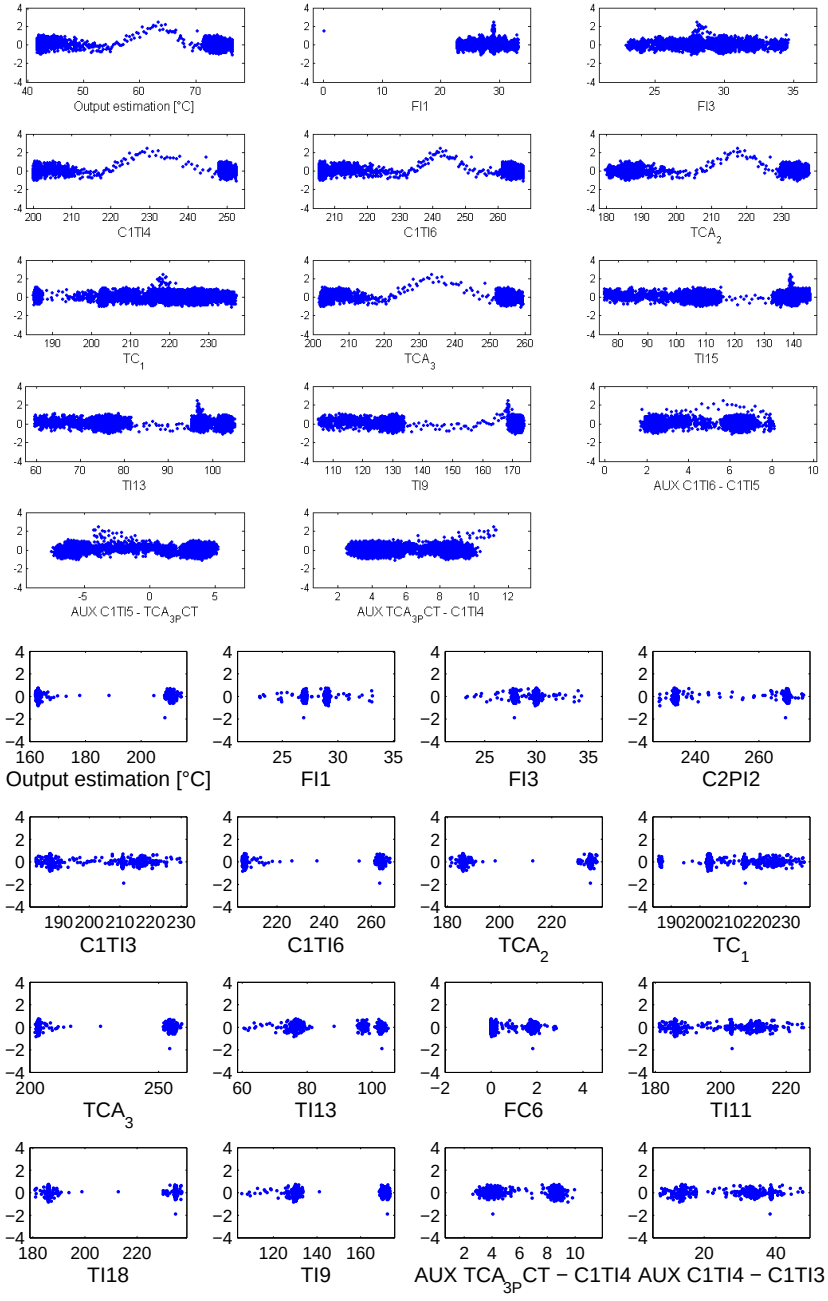

Figure B.11: Dependence of the estimation errors on the estimation and the input variables, training data, SMI-MLP FP model (top) and IBP model (bottom)

Appendix C Estimation accuracies of the FDI models

Table C.1: RMSEs of PLS models for IBP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of LVs
Methodology			0.273	0.294	0.826	0.086	0.081	0.127	1.888	0.550		4
Plain PLS			0.301	0.423	1.093	0.184	0.189	0.261	2.689	0.806	21	6
X			0.273	0.295	0.828	0.086	0.098	0.128	1.988	0.570	23	5
	X		0.290	0.445	1.000	0.094	0.107	0.145	2.842	0.772	31	6
		X	0.301	0.423	1.093	0.184	0.189	0.261	2.689	0.806	21	6
X	X		0.273	0.293	0.833	0.088	0.083	0.130	1.859	0.548	33	4
X		X	0.272	0.292	0.802	0.087	0.084	0.130	1.935	0.555	24	8
	X	X	0.294	0.408	0.976	0.140	0.172	0.207	2.749	0.775	36	5
X	X	X	0.273	0.293	0.833	0.088	0.083	0.130	1.859	0.548	37	4

Table C.2: RMSEs of PLS models for FP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of LVs
Meth.			0.370	0.352	1.278	0.066	0.112	0.186	1.099	0.515		2
Ref.				0.243	0.756	0.136	0.101	0.165	1.492	0.482	179	6
X				0.200	0.887	0.116	0.093	0.136	1.581	0.502	175	3
	X			0.251	0.862	0.126	0.091	0.152	1.615	0.516	981	3
		X		0.243	0.756	0.136	0.101	0.165	1.492	0.482	127	7
X	X			0.333	1.299	0.066	0.125	0.140	1.056	0.503	1038	5
X		X		0.200	0.887	0.116	0.093	0.136	1.581	0.502	125	3
	X	X		0.232	0.692	0.148	0.112	0.186	1.416	0.465	1058	3
X	X	X		0.333	1.299	0.066	0.125	0.140	1.056	0.503	1324	5

Table C.3: RMSEs of MLP models for IBP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of neurons
Meth.			0.269	0.560	1.030	0.087	0.560	0.262	2.606	0.851		1
Ref.				0.276	0.842	0.066	0.153	0.095	1.179	0.435	3163	12
X				0.226	1.213	0.064	0.161	0.088	1.258	0.502	2102	6
	X			0.520	0.933	0.077	0.352	0.153	0.559	0.432	12609	2
		X		0.286	1.125	0.064	0.141	0.098	0.580	0.382	3098	5
X	X			0.738	1.143	0.067	0.511	0.376	0.425	0.543	12844	13
X		X		0.186	1.186	0.068	0.178	0.099	1.611	0.555	1581	11
	X	X		0.362	1.310	0.049	0.163	0.139	0.609	0.439	18667	3
X	X	X		0.392	1.084	0.062	0.250	0.126	1.260	0.529	18294	2

Table C.4: RMSEs of MLP models for FP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of neurons
Meth.			0.317	0.421	0.404	1.280	0.051	0.329	0.142	0.498		1
Ref.				0.334	1.968	0.063	0.199	0.135	0.789	0.582	2307	13
X				0.187	1.607	0.069	0.160	0.097	1.285	0.568	3196	5
	X			0.515	1.072	0.057	0.313	0.135	0.686	0.463	12866	1
		X		0.268	0.741	0.062	0.143	0.098	1.221	0.422	2783	5
X	X			0.274	1.735	0.058	0.123	0.127	0.896	0.536	13232	3
X		X		0.269	1.351	0.067	0.145	0.090	0.906	0.471	1364	5
	X	X		0.269	0.908	0.058	0.177	0.081	0.875	0.395	19547	3
X	X	X		0.287	0.965	0.051	0.170	0.123	1.276	0.479	17941	2

Table C.5: RMSEs of SMI models for IBP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of states
Meth.			0.269	0.306	0.785	0.092	0.095	0.134	1.614	0.504		8
Ref.				1.103	4.027	0.183	0.428	0.200	2.443	1.397	95	2
X				0.295	0.886	0.090	0.105	0.133	1.731	0.540	107	14
	X			0.348	0.679	0.144	0.215	0.165	1.795	0.558	194	1
		X		1.103	4.027	0.183	0.428	0.200	2.443	1.397	97	2
X	X			0.280	0.724	0.082	0.090	0.116	1.651	0.491	218	6
X		X		0.286	0.779	0.082	0.093	0.116	1.834	0.532	103	14
	X	X		0.317	0.783	0.146	0.211	0.169	1.479	0.518	229	3
X	X	X		0.274	0.780	0.087	0.104	0.136	1.586	0.495	243	10

Table C.6: RMSEs of SOM models for IBP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Size of the map
Meth.			0.371	1.072	2.176	0.259	0.962	1.032	2.172	1.279		11x11
Ref.				0.929	3.493	0.156	0.377	0.521	2.263	1.290	234	23x23
X				0.689	3.088	0.189	0.386	0.603	2.223	1.196	284	24x24
	X			0.787	2.823	0.170	0.448	0.731	2.113	1.178	1088	16x16
		X		0.929	3.493	0.156	0.377	0.521	2.263	1.290	241	23x23
X	X			0.835	2.293	0.226	0.474	0.776	2.283	1.148	1281	16x16
X		X		0.689	3.088	0.189	0.386	0.603	2.223	1.196	274	24x24
	X	X		0.787	2.823	0.170	0.448	0.731	2.113	1.178	1384	16x16
X	X	X		0.835	2.293	0.226	0.474	0.776	2.283	1.148	1535	16x16

Table C.7: RMSEs of SOM models for FP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	
Meth.			0.498	0.748	2.047	0.169	0.704	0.870	0.762	0.883		10x10
Ref.				0.767	1.719	0.178	0.522	0.839	1.164	0.865	727	19x19
X				0.785	2.352	0.179	0.542	0.869	1.282	1.001	671	24x24
	X			0.992	1.586	0.214	0.674	1.047	0.855	0.895	7005	12x12
		X		0.767	1.719	0.178	0.522	0.839	1.164	0.865	722	19x19
X	X			0.593	1.449	0.148	0.400	0.674	0.859	0.687	6626	23x23
X		X		0.785	2.352	0.179	0.542	0.869	1.282	1.001	663	24x24
	X	X		0.633	1.383	0.292	0.800	0.674	1.950	0.955	7872	24x24
X	X	X		0.813	1.236	0.124	0.571	0.923	0.925	0.765	8375	12x12

Table C.8: RMSEs of PLS-MLP models for IBP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of neurons
Meth.			0.365	0.354	1.195	0.070	0.117	0.192	0.998	0.488		2
Ref.				0.235	0.629	0.132	0.094	0.276	1.475	0.473	1814	2
X				0.207	0.803	0.088	0.126	0.219	1.485	0.488	1929	5
	X			0.254	0.795	0.122	0.091	0.148	1.658	0.511	5569	1
		X		0.316	0.593	0.082	0.113	0.112	1.497	0.452	2557	2
X	X			0.471	1.127	0.059	0.236	0.104	0.808	0.467	5246	9
X		X		0.240	0.814	0.100	0.132	0.124	1.490	0.484	1664	8
	X	X		0.521	0.662	0.091	0.132	0.088	1.283	0.463	8166	2
X	X	X		0.264	0.932	0.057	0.257	0.129	0.856	0.416	7001	5

Table C.9: RMSEs of PLS-MLP models for FP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of neurons
Meth.			0.268	0.298	0.751	0.084	0.085	0.124	1.865	0.535		1
Ref.				0.429	1.085	0.185	0.117	0.318	2.477	0.768	62	8
X				0.298	0.831	0.086	0.160	0.141	1.761	0.546	63	9
	X			0.470	1.026	0.099	0.138	0.144	2.848	0.787	178	1
		X		0.403	1.044	0.125	0.114	0.232	2.655	0.762	66	2
X	X			0.293	0.797	0.086	0.083	0.129	1.833	0.537	186	2
X		X		0.299	0.769	0.117	0.091	0.157	1.926	0.560	64	3
	X	X		0.383	0.906	0.123	0.133	0.264	2.729	0.756	229	3
X	X	X		0.292	0.845	0.085	0.083	0.130	1.858	0.549	227	1

Table C.10: RMSEs of SMI-MLP models for FP trained with different combinations of steps described in methodology

Delay compensation	Input variable selection	Calculated variables	Training data	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4	Evaluation 5	Evaluation 6	Eval average	Execution time [s]	Number of neurons
Meth.			0.268	0.322	0.780	0.090	0.102	0.129	1.625	0.508		1
Ref.				1.113	2.506	0.147	0.650	0.595	1.699	1.118	91	9
X				0.865	4.236	0.131	0.634	0.320	2.682	1.478	98	2
	X			0.619	1.831	0.107	0.281	0.141	1.560	0.757	203	1
		X		0.835	4.262	0.165	0.597	0.484	1.652	1.333	94	3
X	X			0.500	2.668	0.137	0.268	0.361	1.731	0.944	173	5
X		X		0.898	5.107	0.358	0.652	0.478	1.463	1.493	99	2
	X	X		0.444	1.562	0.121	0.262	0.181	2.586	0.859	271	2
X	X	X		0.492	1.950	0.248	0.438	0.342	1.627	0.850	263	1

ISBN 978-951-22-9683-5
ISBN 978-951-22-9684-2 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)