SLOVAK UNIVERSITY OF TECHNOLOGY
IN BRATISLAVA

FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-4622-76835

Data-driven Design of Linear Soft Sensors

DISSERTATION THESIS

Bratislava 2023 **Ing. Martin Mojto**

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-4622-76835

Data-driven Design of Linear Soft Sensors

DISSERTATION THESIS

Study programme:	Process Control
Study field:	Cybernetics
Training workspace:	Department of Information Engineering and Process Control
Supervisor:	doc. Ing. Radoslav Paulen, PhD.
Consultants:	prof. Ing. Miroslav Fikar, DrSc.
	Ing. Karol Ľubušký

Bratislava 2023 **Ing. Martin Mojto**

Slovak University of Technology in Bratislava
Department of Information Engineering and Process Control

Faculty of Chemical and Food Technology
Academic year: 2022/2023
Reg. No.: FCHPT-4622-76835

STU FCHPT

DISSERTATION THESIS TOPIC

Student:	**Ing. Martin Mojto**
Student's ID:	76835
Study programme:	Process Control
Study field:	Cybernetics
Thesis supervisor:	doc. Ing. Radoslav Paulen, PhD.
Head of department:	doc. Ing. Martin Klaučo, PhD.
Workplace:	Department of Information Engineering and Process Control, Slovnaft, a.s.

Topic: **Data-driven Design of Linear Soft Sensors**

Language of thesis: English

Deadline for submission of Dissertation thesis:	31. 05. 2023
Approval of assignment of Dissertation thesis:	07. 05. 2023
Assignment of Dissertation thesis approved by:	prof. Ing. Miroslav Fikar, DrSc. – Chairperson of Field of Study Board

Acknowledgment

I am deeply grateful to the numerous individuals who have supported me directly or indirectly throughout my PhD studies. Foremost among them is my supervisor, Dr. Radoslav Paulen. His unwavering patience, guidance, and dedication have been instrumental in my academic journey. Dr. Paulen not only introduced me to the beauty of research but also motivated me to push beyond my limits and achieve better results than I ever thought possible. I am also grateful for his personal support during challenging times.

I would also like to express my heartfelt appreciation to my secondary supervisor, Professor Miroslav Fikar. I consider myself privileged to have had regular opportunities to discuss my research with such a renowned expert. His willingness to share his knowledge and provide invaluable advice has played a crucial role in shaping my academic progress.

Furthermore, I want to extend my gratitude to my consultant, Mr. Karol Ľubušký, for his expert knowledge and unwavering support during my PhD studies. The consultations with Mr. Ľubušký have significantly deepened my understanding of industrial practise and guided me in aligning the direction of my research.

I would also like to acknowledge Professor Benoît Chachuat for providing me with the opportunity to undertake a research stay at Imperial College London during my PhD studies. This invaluable 10-month experience has greatly enhanced the quality of my research outcomes and results.

Lastly, I am deeply thankful to my family and close friends for their unwavering support and dedication. I could always rely on their encouragement and assistance, especially during challenging times.

Martin Mojto
Bratislava 2023

Abstract

This thesis focuses on the design of data-driven linear soft sensors for real-world case studies in the petrochemical industry. Soft sensors are essential for monitoring important process variables by leveraging measurements from related variables like temperatures, pressures, and flow rates. The design procedure involves seven consecutive steps, including data inspection, historical data selection, preprocessing, input structure and model selection, model training, model validation, and sensor maintenance. Our primary objective is to use this procedure to design soft sensors specifically for industrial distillation columns. We employ various multivariate data treatment approaches to mitigate outliers and systematic errors in the dataset. The training of the soft sensors encompasses ordinary least squares as well as advanced variance-covariance methods and approaches focusing on achieving the desired sparsity in the model structure. We evaluate the performance of the designed soft sensors against reference sensors currently implemented in the refinery or developed based on expert knowledge. Additionally, we compare the performance of single-model and multi-model soft sensors, with the latter incorporating clustering and classification approaches. We propose novel approaches to address the limitations of existing methods in the design of multi-model soft sensors. Our research highlights the practical applicability of soft sensors, initially designed for prediction purposes, such as estimating the product concentration in two industrial distillation columns. Furthermore, we utilise the soft sensors for fault detection, specifically for detecting flooding in a third industrial distillation column using a data-driven indicator based on soft sensor principles.

Keywords: Classification, Data Treatment, Fault Detection, Key Process Variables, Petrochemical Industry, Soft (Inferential) Sensors

Abstrakt

Táto dizertačná práca sa zameriava na návrh lineárnych softvérových senzorov založený na dátach pre prípadové štúdie z petrochemického priemyslu. Softvérové senzory majú zásadný význam pri monitorovaní dôležitých procesných veličín využitím meraní korelovaných veličín, ako sú teploty, tlaky a prietoky. Návrhový postup softvérových senzorov pozostáva zo siedmich po sebe nasledujúcich krokov, vrátane počiatočnej analýzy dát, výberu historických dát, pred-spracovania dát, výberu vstupnej štruktúry a modelu, trénovania modelu, validácie modelu a údržby softvérových senzorov. Naším hlavným cieľom je aplikovať tento postup na návrh softvérových senzorov pre konkrétne priemyselné destilačné kolóny. Využívame rôzne prístupy spracovania viacrozmerných dát na zníženie počtu outlierov a systematických chýb v meraných dátach. Trénovanie softvérových senzorov zahŕňa bežnú metódu najmenších štvorcov, ako aj pokročilejšie metódy založené na variancii-kovariancii a prístupy, ktoré kladú dôraz na dosiahnutie požadovanej riedkosti štruktúry modelu. Efektívnosť navrhnutých softvérových senzorov porovnávame s referenčnými senzormi, ktoré sú momentálne implementované v rafinérii alebo sú vyvinuté na základe odborných poznatkov z praxe. Okrem toho porovnávame výkon jedno-modelových a viac-modelových softvérových senzorov, pričom viac-modelové senzory zahŕňajú prístupy zhlukovania a klasifikácie. Navrhujeme nové prístupy na odstránenie obmedzení existujúcich metód pri návrhu viac-modelových softvérových senzorov. Naša výskumná práca zdôrazňuje praktickú použiteľnosť softvérových senzorov, pričom ich najprv navrhujeme pre účely predikcie, ako napríklad odhad koncentrácie produktov v dvoch priemyselných destilačných kolónach. Okrem toho využívame softvérové senzory na detekciu porúch, konkrétne na detekciu zaplavovania v tretej priemyselnej destilačnej kolóne pomocou indikátora založeného na dátach a navrhnutého na základe princípov softvérových senzorov.

Kľúčové slová: Klasifikácia, Spracovanie údajov, Detekcia porúch, Kľúčové procesné veličiny, Petrochemický priemysel, Softvérové (inferenčné) senzory

Contents

Acknowledgment iii

Abstract v

Abstrakt vii

1 Introduction 1

I Methodology 11

2 Soft-sensor Theory Foundations 13

2.1 Introduction to Soft Sensors 13

2.1.1 Training, Validation, and Testing Datasets 14

2.1.2 Soft-sensor Implementation 15

2.2 Statistics for Soft-sensor Design 17

2.2.1 Normal Distribution 17

2.2.2 Covariance and Correlation Matrices 19

2.2.3 Hotelling's Distribution 20

2.3 Regression, Clustering, and Classification 21

2.3.1 Regression 21

2.3.2 Regression Performance Indices 22
2.3.3 Clustering 24
2.3.4 Clustering Performance Indices 24
2.3.5 Classification 25
2.3.6 Classification Performance Indices 25

3 Soft-sensor Algorithm Foundations 27

3.1 Ordinary Least Squares Regression 27
3.2 Variance-covariance Algorithms 28
3.2.1 Principal Component Analysis 28
3.2.2 Partial Least Squares 30
3.3 Sparsity-based Algorithms 30
3.3.1 Cross-Validation 31
3.3.2 Least Absolute Shrinkage and Selection Operator 31
3.3.3 Subset Selection 33
3.4 Clustering Algorithms 35
3.4.1 k-means Clustering 35
3.4.2 Density-based Spatial Clustering of Applications with Noise . . 36
3.4.3 Minimum Covariance Determinant 37
3.5 Classification Algorithms 38
3.5.1 Support Vector Machine 38
3.5.2 Logistic Regression 39

4 Soft-Sensor Design 41

4.1 Initial Data Inspection 42
4.2 Selection of Historical Data 43

4.3 Data Pre-processing 43
4.3.1 Identification of Linear Correlations 44
4.3.2 Incorporation of Nonlinear Transformations 45
4.3.3 Data Treatment 46
4.4 Model and Input Structure Selection 47
4.4.1 Model Selection 47
4.4.2 Input Structure Selection 48
4.5 Model Training 49
4.5.1 Single-model Soft Sensor 50
4.5.2 Multi-model Soft Sensor 51
4.6 Model Validation 55
4.7 Soft-sensor Maintenance 55
4.7.1 Recursive Estimation 55
4.7.2 Bias Correction 56

II Contributions 59

5 Data-driven Design of Soft Sensors for Petrochemical Industry 61

5.1 Problem Definition 61
5.1.1 FCC unit 61
5.1.2 VGH Unit 63
5.2 Solution Approach 64
5.3 Results 65
5.3.1 Implementation details 65
5.3.2 Soft Sensors for the FCC Unit 67
5.3.3 Design of Soft Sensors for the FCC Unit using Time Series Data 69

5.3.4 Design of Soft Sensors for the FCC Unit using Randomly Distributed Data . . . 73
5.3.5 Soft Sensors for the VGH Unit . . . 76
5.3.6 Design of Soft Sensors for the VGH Unit using Time Series Data 78
5.3.7 Design of Soft Sensors for the VGH Unit using Randomly Distributed Data . . . 82
5.4 Discussion . . . 84

6 Data-driven Design of Multi-model Soft Sensors 87

6.1 Problem Definition . . . 87
6.2 Solution Approach . . . 87
6.2.1 Design of MMS with Continuous Switching . . . 88
6.2.2 Design of MMS with Optimized Data Labelling . . . 89
6.3 Results . . . 91
6.3.1 Implementation Details . . . 91
6.3.2 Design of Soft Sensors for Pressure-Compensated Temperature 92
6.3.3 Design of Soft Sensors for Vacuum Gasoil Hydrogenation Unit 97
6.4 Discussion . . . 104

7 Data-Driven Indication of Flooding in an Industrial Debutanizer Column 107

7.1 Problem Definition . . . 107
7.1.1 Industrial Debutanizer Column . . . 107
7.1.2 Flooding in the Industrial Debutanizer Column . . . 108
7.2 Solution Approach . . . 109
7.3 Results . . . 110
7.3.1 Data Treatment using MCD . . . 110
7.3.2 Training of Data-Driven Indicators . . . 112

7.4 Discussion 114

8 Conclusions and Future Research 115

A Curriculum Vitae 117

B Author's Publications 119

C Resumé 123

Bibliography 129

List of Figures

2.1 Soft sensor life cycle (implementation) in an industrial plant 16

3.1 The visualization of PCA/PLS principal components 29

3.2 The impact of different values of λ on LASSO performance 32

4.1 The procedure of the soft-sensor design 42

4.2 The design and prediction phases of MMS 51

4.3 The SMS design (PCT) 53

4.4 The MMS design (PCT) 54

5.1 A schematic diagram of the depropanizer column. 62

5.2 A schematic diagram of the VGH unit. 63

5.3 Bottom product temperature of the FCC unit. 68

5.4 The measurements retaining after MCD (depropanizer, FCC) 69

5.5 The soft sensors design on chronological data (depropanizer, FCC) . . 72

5.6 The soft-sensor design on 50 random datasets (depropanizer, FCC) . . 74

5.7 The soft sensors design on random data (depropanizer, FCC) 75

5.8 Data treatment of the data from the VGH unit 77

5.9 The measurements retaining after MCD (main fractionator, VGH) . . 79

5.10 The soft sensors design on chronological data (main fractionator, VGH) 81

5.11 The soft-sensor design on 50 random datasets (main fractionator, VGH) 83

5.12 The soft sensors design on random data (main fractionator, VGH) . . 85

6.1 The enhanced MMS design (PCT) 91

6.2 The datasets for studied scenarios (PCT) 93

6.3 The soft-sensor design for studied scenarios (PCT) 94

6.4 The SMS and MMS design for different noises (PCT) 96

6.5 The MMS design on the industrial data (main fractionator, VGH) . . 103

7.1 The data treatment by MCD (debutanizer, FCC) 111

7.2 The comparison of training and testing data (debutanizer, FCC) . . . 111

List of Tables

5.1 The soft-sensor design on chronological data (depropanizer, FCC) . . . 70

5.2 The soft-sensor design on random data (depropanizer, FCC) 73

5.3 The soft-sensor design on chronological data (main fractionator, VGH) 79

5.4 The soft-sensor design on random data (main fractionator, VGH) . . . 82

6.1 The SMS design on industrial data (main fractionator, VGH) 99

6.2 The MMS design on industrial data (main fractionator, VGH) 100

7.1 The design of indicators on the industrial data (debutanizer, FCC) . . 112

Acronyms

AC	Accuracy
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
CV	Cross-Validation
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
F1	F1-Score
FCC	Fluid Catalytic Cracking
FN	False Negative
FP	False Positive
GF	Gasoline Fraction
HGO	Hydrogenated Gasoil
LASSO	Least Absolute Shrinkage and Selection Operator
MCD	Minimum Covariance Determinant
ML	Machine Learning

MMS Multi-Model Soft Sensor

OLS Ordinary Least Squares

PC Principal Component

PCA Principal Component Analysis

PCT Pressure Compensated Temperature

PLS Partial Least Squares

PR Precision

RC Recall

RLS Recursive Least Squares

RMSE Root-Mean-Square Error

SAE Sum of Absolute Errors

SMS Single-model Soft Sensor

SS Subset Selection

SS-CV Subset Selection with Cross-Validation

SS-MOC Subset Selection with Model-Overfitting Criterion

SSE Sum of Squares Error

SVD Singular Value Decomposition

SVM Support Vector Machines

TN True Negative

TP True Positive

VGH Vacuum Gasoil Hydrogenation

Symbols

Latin Symbols

$\mathbf{0}$	All-zeros vector
$\mathbb{1}$	All-ones vector
$\mathcal{C}$	Cluster (class)
$\boldsymbol{\mathcal{C}}$	Set of clusters (classes)
crt	Criterion
$\mathcal{D}$	Set representation of the entire dataset
$\boldsymbol{d}_{\mathrm{CL}}$	Vector of k-means clustering distances, $\mathbb{R}^n$
$\boldsymbol{d}_{\mathrm{M}}$	Vector of Mahalanobis distances, $\mathbb{R}^n$
$\boldsymbol{d}_{\mathrm{MCD}}$	Vector of MCD distances, $\mathbb{R}^n$
$\boldsymbol{d}_{T^2}$	Vector of Hotelling's T^2 distances, $\mathbb{R}^n$
$\boldsymbol{e}$	Vector of slack variables, $\mathbb{R}^n$
E_{BC}	Effort (frequency) of bias correction
h	MCD tuning parameter
$\mathcal{I}$	Index set representation
$\mathbf{I}$	Indicator
K_{BC}	Gain (multiplier) for the bias correction
n	Number of elements in the sample

n_{cl}	Number of clusters (classes)
n_{p}	Number of parameters
n_{p}^*	Optimal number of parameters
n_{pc}	Number of principal components
n_{pc}^*	Optimal number of principal components
N	Number of elements in the population
$\mathbf{R}$	Sample correlation matrix, $\mathbb{R}^{n_{\text{p}} \times n_{\text{p}}}$
$\mathbb{R}$	Real-vector space
s	Sample standard deviation
$\boldsymbol{w}$	Vector of model parameters for linear separation hyperplane, $\mathbb{R}^{n_{\text{p}}}$
w_0	Constant (bias) term for linear separation hyperplane
$\bar{x}$	Sample mean value of input variable
$\boldsymbol{x}$	Vector of input variables, $\mathbb{R}^{n_{\text{p}}}$
$\bar{\boldsymbol{x}}$	Sample mean vector, $\mathbb{R}^{n_{\text{p}}}$
$\boldsymbol{x}_{\{i\}}$	The i^{th} input variable, $\mathbb{R}^n$
$\boldsymbol{X}$	Sample matrix of input variables, $\mathbb{R}^{n \times n_{\text{p}}}$
$\boldsymbol{X}_{\text{p}}$	Population matrix of input variables, $\mathbb{R}^{N \times n_{\text{p}}}$
y	Measured (observed) output variable
$\hat{y}$	Estimated output variable
$\bar{y}$	Sample mean value of output variable
$\boldsymbol{y}$	Sample vector of the measured (observed) output variable, $\mathbb{R}^n$
$\hat{\boldsymbol{y}}$	Sample vector of the estimated output variable, $\mathbb{R}^n$
$\hat{\boldsymbol{y}}_{\text{p}}$	Population vector of the estimated output variable, $\mathbb{R}^N$
$\boldsymbol{z}$	Vector of binary variables, $\mathbb{R}^n$
$\boldsymbol{z}_{\text{ss}}$	Vector of binary variables for subset selection, $\mathbb{R}^{n_{\text{p}}}$

Greek Symbols

α	Weighting parameter for normal vector of the separation plane
$\boldsymbol{\beta}$	Vector of model parameters, $\mathbb{R}^{n_\mathrm{p}}$
$\boldsymbol{\beta}^*$	Vector of optimal model parameters, $\mathbb{R}^{n_\mathrm{p}^*}$
β_0	Constant (bias) term
β_0^*	Optimal constant (bias) term
$\boldsymbol{\epsilon}$	Vector of random error terms, $\mathbb{R}^n$
γ	Weighting parameter for slack variables
μ	Population mean value
$\boldsymbol{\mu}$	Population mean vector, $\mathbb{R}^{n_\mathrm{p}}$
σ	Population standard deviation
$\boldsymbol{\Sigma}$	Sample covariance matrix, $\mathbb{R}^{n_\mathrm{p} \times n_\mathrm{p}}$
$\boldsymbol{\Sigma}_\mathrm{p}$	Population covariance matrix, $\mathbb{R}^{n_\mathrm{p} \times n_\mathrm{p}}$
$\boldsymbol{\Sigma}_{XY}$	Cross-covariance matrix, $\mathbb{R}^{n_\mathrm{p} \times 1}$

Subscripts

BC	Bias correction
cen	Centred
low	Lower dimensional
nor	Normalized
sel	Sparse representation
std	Standardized
s	Subset
S	Testing
T	Training
V	Validation

Chapter 1

Introduction

"If you can't measure it, you can't improve it."
William Thomson Kelvin (1824–1907)

The continuous growth of technology in the industrial sector and the need to meet specific requirements, such as improved profitability and safety, motivate researchers to implement innovative solutions and ideas. One crucial element in achieving these requirements is gaining a comprehensive understanding of process dynamics through online sensors or analysers. Soft sensors, also known as inferential sensors or virtual sensors, have emerged as widely recognised and frequently used analysers designed to estimate hard-to-measure process variables or aspects using related variables that are easier to measure. The soft sensors are represented by mathematical structures and provide a cost-effective alternative to physical sensing devices, potentially offering more precise and frequent indications of the desired variables (Mejdell and Skogestad, 1991; Kordon et al., 2003; Curreri et al., 2020).

In general, the production processes represent complex systems with many variables and interactions between these variables (Santander et al., 2022). They usually exhibit nonlinear behaviour resulting from the rich interactions of the involved physical phenomena. One would conjecture that a nonlinear soft sensor design (Zheng et al., 2022) is necessary. However, a typical industrial process is usually operated in some operating regime (steady state) to achieve the desired product specifications. Therefore, the nonlinear behaviour of the process variable can be often neglected, and the linear soft sensor can provide an accurate estimate of the desired variable. The advantages of a linear soft sensor over its nonlinear counterpart lie foremost in lower maintenance expenses, higher transparency, the possibility of physical insight, and lower computational effort for sensor training, validation, online evaluation, and further calculations (e.g., for optimisation and control). The latter aspect can be significant mainly when the estimated variable is an input for an advanced process controller (Botha and Craig,

2021) or is involved in a complex plant-wide optimization (Ge, 2017).

Linear soft sensors can be classified into two main categories: model-based (or mechanistic) (Doraiswami and Cheded, 2014; Torgashov and Skogestad, 2019) and data-driven soft sensors (King, 2011; Mojto et al., 2021; Sun and Braatz, 2021). Model-based soft sensors leverage knowledge about the process derived from first-principles models based on fundamental physical and chemical laws. These models can be directly employed in soft sensor design (Torgashov and Skogestad, 2019) or indirectly used through an observer (Doraiswami and Cheded, 2014). However, the complexity and scale of industrial processes often restrict the application of model-based soft sensors. On the other hand, data-driven soft sensors yield results that are less interpretable compared to model-based ones since they require less domain knowledge of the process. This makes the data-driven soft sensors preferable in industrial practise over the model-based soft sensors. Consequently, the popularity of data-driven soft sensors is increasing alongside advancements in sensor technology and plant digitalization. Nevertheless, the performance of data-driven soft sensors is closely tied to the quality of the data.

The bridge between model-based and data-driven soft sensors is established through hybrid soft sensors (Pan et al., 2013; Pla et al., 2018). These soft sensors combine data-driven models with complementary mechanistic correlation models to leverage available domain knowledge (Tahir et al., 2019; Zhuang et al., 2022). The mechanistic correlation models are typically derived from first-principles models and calibrated using process data.

The effectiveness and reliability of linear data-driven soft sensors heavily depend on the quality of the data used in their design. This is particularly significant due to the utilisation of linear regression, also known as ordinary least squares (OLS) regression, which is a fundamental and commonly used approach for training the model parameters in the linear soft-sensor structure. OLS regression aims to minimise the sum of squared error (SSE) criterion, which tends to amplify the negative impact of outliers and deviated measurements on the accuracy of the model. Moreover, the multivariate dataset employed for the design of linear soft sensors typically comprises signals from various online sensors. Therefore, it is crucial to subject the dataset to an effective multivariate data treatment analysis that can detect both systematic and random errors (Su et al., 2009). In industrial settings, several methods have been developed to address data pre-treatment. Among these methods, Hotelling's T^2 distribution (Hotelling, 1931), k-means clustering (Forgy, 1965), and minimum covariance determinant (Rousseeuw, 1984) are well-known and widely used for multivariate data treatment. These methods have been successfully applied in various industrial contexts (Alameddine et al., 2010; Xu et al., 2017; Frumosu and Kulahci, 2019; Azzaoui et al., 2019; Fontes et al., 2021).

As mentioned earlier, the available dataset for linear data-driven soft-sensor design typically consists of multiple input variables, originating from various online sensors (referred to as a multivariate input dataset). This complexity adds a layer of challenge to the soft-sensor design process, as it requires not only finding the most accurate model using OLS regression, but also determining the desired complexity of the model structure. Currently, the most widely used techniques for linear data-driven soft-sensor design are principal component analysis (PCA) regression and partial least squares (PLS) regression (Pearson, 1901; Wold et al., 1984, 2001). PCA regression employs unsupervised learning to reduce the dimensions of the input-variable space and performs regression on the reduced space. This approach has a long and successful history of industrial applications (Kadlec et al., 2009; Yuan et al., 2015; Yu et al., 2020). On the other hand, PLS regression considers both the input and output spaces, making it a supervised learning approach. The choice between PCA and PLS depends on the availability and quality of infrequently measured output variables.

Both PCA and PLS offer partial solutions to address overfitting in soft sensors by performing regression in reduced dimensions. However, the resulting sensor structures are generally not sparse, which may not be desirable or feasible, especially when using the designed soft sensor for advanced process control. To address this limitation, data-driven approaches that promote sparsity in soft-sensor design have been developed. One such approach is the Least Absolute Shrinkage and Selection Operator (LASSO) (Santosa and Symes, 1986; Tibshirani, 2011), which applies a 1-norm penalty to balance the accuracy and complexity of the soft sensor. The concept of sparse soft-sensor design is further advanced through subset selection (SS) methods, which aim to identify the best subset of explanatory variables from the multivariate input space. The original subset selection methodology proposed various stepwise approaches (backward, forward, and bi-directional) to select suitable input variables from a pool of candidates (Efroymson, 1960; Smith, 2018). Further studies (Miyashiro and Takano, 2015; Mencarelli et al., 2020) have shown that SS can be enhanced by incorporating model overfitting criteria such as adjusted R^2 (R^2_{adj}), corrected Akaike Information Criterion (AICc), or Bayesian Information Criterion (BIC). Additionally, the performance of SS can be improved by emulating the cross-validation process (Takano and Miyashiro, 2020).

As mentioned earlier, linear data-driven soft sensors in industrial practise often involve a trade-off between estimation accuracy and model complexity. While designing nonlinear soft sensors (Zheng et al., 2022) may seem like a straightforward solution to improve performance, it would require significant effort to determine an appropriate model structure. Additionally, the resulting structure would likely be more complex and less transparent compared to a linear model structure. An alternative approach

to enhancing the performance of linear data-driven soft sensors is to use a multi-model soft-sensor design (MMS) instead of the traditional single-model soft-sensor (SMS) design. MMS offers the potential to mitigate the negative impact of nonlinear process behaviour on prediction accuracy while maintaining a relatively simple model structure. Each model within the MMS structure can explain a particular operating regime. Therefore, these sensors find applications in complex industrial processes with a multitude of operating modes (Khatibisepehr et al., 2012; Jin et al., 2015; Wang et al., 2021).

The state-of-the-art MMS design consists of three sequential steps: (1) a priori labelling, (2) data classification, and (3) individual training of the soft sensor models. In the first step, a modeller searches for an appropriate number of models and assigns tags to available data to distinguish the models (classes) discovered. The popular approach for a priori labelling is k-means clustering (Forgy, 1965). A comparison of several other techniques for a priori labelling is shown in (Lü and Yang, 2014). The classification step employs an appropriate data-driven (machine-learning) approach to draw model-validity regions, i.e., the boundaries between the classes (models) that would later serve as switching conditions for using predictions from a particular model. A frequently used and well-known classification learner is Support Vector Machine (SVM) (Boser et al., 1992). The method designs classification hyperplanes in the context studied in this work. Lastly, the constituent models of MMS are individually trained for each class by using a suitable regression technique (Mojto et al., 2021). One of the recent examples of learning MMS is given in (Bemporad, 2022), where piecewise linear regression is considered together with classification based on softmax regression and labelling by the k-means algorithm. While the state-of-the-art approaches train MMS effectively, there are still a few drawbacks hindering the overall potential of MMS. The first drawback is that the continuity of the switching between the different MMS models is not guaranteed. This can have a negative impact on plant production. For example, a common issue that might arise is that advanced process controllers with MMS implemented might face stability issues because of soft sensor discontinuity. The second drawback originates from the a priori labelling that is unaware — like any other unsupervised learning approach — of its impact on MMS prediction accuracy. It is, therefore, not likely that the optimal allocation of the model-validity regions will be achieved.

The earlier description emphasizes the wide range of applications for both single-model soft sensors (SMS) and multi-model soft sensors (MMS) in monitoring key process variables. However, soft sensors have the potential to find applications in various other fields as well. One area experiencing increasing industrial demand for soft sensors is fault detection (Joe Qin, 2003; Serpas et al., 2013). A fault is defined as a deviation

from the intended characteristic that renders it unable to fulfill its purpose. While some research has been published in this area (Serpas et al., 2013; Lemos et al., 2021), it is still relatively new in terms of real-world industrial applications.

Fault detection can be employed to identify potential anomalies in the desired variables. In the industry, fault detection plays a significant role in detecting flooding within distillation columns (Pihlaja and Miller, 2012). Flooding is an undesirable phenomenon that occurs when the liquid level rises above a tray due to foaming or excessive downcomer fill-up (King, 2011). This condition leads to a significant loss in tray separation efficiency and, consequently, plant profitability. Early detection or prediction of flooding through fault detection is therefore crucial for maintaining a profitable and sustainable plant. Numerous studies have addressed the issue of flooding detection, focusing on the correlation between flooding and internal process variables, particularly the pressure difference (drop) across the column (Peiravan et al., 2020) and the time derivative of the pressure drop (Pihlaja and Miller, 2012). Industrial experts often leverage these findings and combine them with their understanding of the primary causes of flooding to develop tailored solutions for each column. However, the use of machine learning (ML) approaches (Mojto et al., 2021; Oeing et al., 2021; Fuentes-Cortés et al., 2022) could potentially streamline the process of creating customized solutions, saving time and effort. ML approaches encompass both unsupervised techniques (e.g., k-means clustering and PCA) that do not rely on prior knowledge of the model outcome and supervised techniques (e.g., SS and SVM) that utilize knowledge about the desired outcome for training.

In this work, we focus on designing SMS using various data-driven techniques. The methods examined include variance-covariance approaches (i.e., PCA and PLS), as well as more recent techniques that enforce model sparsity (i.e., LASSO and SS). The primary contribution of this research is the comparison of these methods in the specific context of industrial soft-sensor design. To evaluate the effectiveness of these techniques, we analyse their performance in two industrial distillation columns: the depropranizer column in the Fluid Catalytic Cracking (FCC) unit and the main fractionator in the Vacuum Gasoil Hydrogenation (VGH) unit, both provided by the oil refinery Slovnaft, a.s. in Bratislava, Slovakia. Although these distillation columns differ in complexity, their common objective is to monitor the product composition using the soft sensors we develop.

In our research, we build upon the SMS design by exploring the natural progression to the MMS design. To address the drawbacks associated with MMS, we propose novel approaches that mitigate these limitations. Firstly, our proposed approaches enable continuous switching between the MMS models by merging classification and model

training into a single decision problem. This is achieved by training an SVM separation hyperplane, which acts as a switching boundary between the MMS models. Secondly, we introduce an optimisation-based labelling approach that simultaneously conducts all sequential steps of the state-of-the-art MMS design procedure. We evaluate the performance of these approaches using both a synthetic dataset based on a pressure compensated temperature (PCT) model (King, 2011) and an industrial dataset from the main fractionator at the VGH unit in the oil refinery Slovnaft, a.s. in Bratislava, Slovakia. We compare the performance of MMS with (a) the reference SMS currently used in the refinery and (b) SMS designed using various techniques (i.e., OLS, PCA, PLS, LASSO, and SS). Our conclusions are based on standard indicators, including prediction accuracy (root-mean-square error, RMSE) and complexity (number of model input variables or principal components). Additionally, we perform a comprehensive analysis of the life cycle of the designed soft sensors by conducting a bias correction analysis (Quelhas, 2009).

We leverage the knowledge acquired from both SMS and MMS designs to tackle the problem of fault detection in a real-world industrial setting. Specifically, we focus on designing data-driven flooding indicators for an industrial debutanizer column located at the FCC unit in the oil refinery Slovnaft a.s. in Bratislava, Slovakia. To assess the performance of these indicators, we compare them against a reference indicator, which is considered the ground truth. The reference indicator is designed based on industry specifications and the expertise surrounding flooding detection. We employ data-driven approaches, including both unsupervised and supervised machine learning (ML) techniques, to develop the flooding indicators and evaluate their effectiveness.

This work is structured as follows. The first part provides an introduction to the methodology of soft-sensor design, covering the theoretical foundations and algorithms utilised in this field. It is then followed by a discussion of the state-of-the-art approaches that are employed in the overall soft-sensor design procedure. The second part of this work builds upon the methodology and presents the main findings and contributions of the research.

Motivation

In the current era of advanced technology and increased computational power, there is a tremendous opportunity to enhance industrial processes by implementing innovative ideas and technological advancements. These advancements have a profound impact on the overall profitability and sustainability of industrial operations. Soft sensors play a crucial role in monitoring hard-to-measure variables and other important controlled

variables (CVs) in industrial settings, making them a prime example of such technology.

While real-time sensing devices can provide measurements for hard-to-measure variables, they often have limitations in terms of accuracy and frequency. Moreover, the cost of these devices is typically significantly higher compared to soft sensors. By contrast, designing soft sensors can offer a cost-effective alternative, albeit requiring historical data for their development. While soft sensors may require periodic maintenance for model updates, this maintenance is generally more affordable compared to maintaining real sensing devices.

The motivation for studying data-driven approaches to soft-sensor design stems from the increased availability of inexpensive and relatively accurate online sensors, which provide the desired variables and sufficient measurements for research purposes. This flexibility implies that such research holds promise for the future, enabling increased process safety and improved process economics with the addition of each new measured variable.

At the heart of soft sensors lies their mathematical structure, which allows for the incorporation of a wide range of input variables. This design flexibility empowers us to leverage modern ML techniques to optimise the performance of soft sensors. Furthermore, the process of designing soft sensors enables the combination of various approaches and facilitates a comprehensive analysis of the obtained results.

However, a significant drawback of data-driven soft sensors is their individual performance in different industrial plants. To address this, we compare the performance of soft sensors designed using various data-driven methods, with the aim of developing a novel method that combines the favourable characteristics of the studied methods.

Lastly, the signals from soft sensors are often used as additional inputs to advanced process controllers. Given that soft sensors typically monitor key process variables, they play a vital role in the overall performance of advanced process controllers. As technology continues to evolve and enhance the performance of advanced process controllers, the demand for soft sensors is expected to increase in the future.

General Objectives

The primary objective of this work is to analyse the applicability of different recent data-driven approaches for linear single-model soft sensor (SMS) design. This analysis aims to provide insights into potential future research directions in the field of soft

sensor design. Subsequently, the scope of the research will be expanded to include the design of a multiple model soft sensor (MMS). For the MMS, effective methods for improving state-of-the-art methodologies will be explored, followed by an analysis of the performance of the proposed approaches using real industrial datasets.

To evaluate the performance of the studied approaches for SMS and MMS designs, two different case studies will be considered: the depropanizer column in the FCC unit and the main fractionator in the VGH unit at the oil refinery Slovnaft, a.s. in Bratislava, Slovakia. The objectives for each case study can be summarised as follows:

- Conduct data treatment analysis using various multivariate analyses, such as Hotelling's T^2 distribution, k-means clustering, and MCD.
- Determine the input structure and model selection for SMS and MMS.
- Train the selected model structure using the training dataset and validate it using the testing dataset.
- Compare the quality of the designed sensors not only based on accuracy (RMSE) but also taking into account the complexity of the soft sensor structure (number of resulting model parameters).
- Compare the designed soft sensors against the reference soft sensor, which represents the current performance in the refinery.
- Analyse the sustainability and performance of the soft sensor in the future, including bias correction and evaluating its effort.

Another objective is to analyse the effectiveness of soft sensors in the area of fault detection. Specifically, a data-driven indicator will be designed for detecting flooding within the industrial debutanizer column located at the FCC unit in the oil refinery Slovnaft, a.s. in Bratislava, Slovakia. The knowledge obtained from the SMS and MMS designs will be applied to develop the desired flooding indicators. Additionally, various unsupervised and supervised machine learning approaches will be used to design the indicators, and their performance will be compared against the ground truth provided by expert knowledge.

Publications

Content of this thesis is primarily based on the following publications:

1. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2023). **Data-based Design of Multi-model Inferential Sensors**. Computers & Chemical Engineering, (*under review*).

2. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2023). **Design of Multi-Model Linear Inferential Sensors with SVM-based Switching Logic**. IFAC World Congress, (*accepted*).

3. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2023). **Data-Driven Indication of Flooding in an Industrial Debutanizer Column**. Antonis Kokossis, Michael C. Georgiadis, Efstratios N. Pistikopoulos, editors, 33rd European Symposium on Computer Aided Process Engineering, volume 52 of Computer Aided Chemical Engineering. Elsevier, (*accepted*).

4. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2023). **Input Structure Selection for Soft-Sensor Design: Does It Pay Off?**. In Proceedings of the 24rd International Conference on Process Control, (*accepted*).

5. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2023). **Comparing Linear and Nonlinear Soft Sensor Approaches for Industrial Distillation Columns**. In 49th International Conference of the Slovak Society of Chemical Engineering SSCHE 2023, Slovak Society of Chemical Engineering, Bratislava, SK, pp. 159–159.

6. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2022). **Multi-Model Soft-Sensor Design for a Depropanizer Distillation Column**. In Advanced Process Modelling Forum 18–19 October 2022.

7. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2022). **Support Vector Machine-based Design of Multi-model Inferential Sensors**. Editor(i): Ludovic Montastruc, Stephane Negny, In 32nd European Symposium on Computer Aided Process Engineering, Elsevier, no. 1, vol. 32, pp. 1045–1050.

8. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2022). **Data-based Design of Inferential Sensors for an Industrial Depropanizer Column with Data Pre-treatment Analysis**. Editor(i): Mário Mihaľ, In 48th International Conference of the Slovak Society of Chemical Engineering SSCHE 2022 and Membrane Conference PERMEA 2022, Slovak Society of Chemical Engineering, Bratislava, SK, pp. 200.

9. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2021). **Data Treatment of Industrial Measurements: From Online to Inferential Sensors**. Editor(i):

R. Paulen, M. Fikar and J. Oravec, In Proceedings of the 23rd International Conference on Process Control - Summaries Volume, Slovak Chemical Library, Slovak University of Technology in Bratislava, Radlinského 9, SK812-37, Bratislava, Slovakia, pp. 52–53.

10. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2021). **Data-based Design of Inferential Sensors for Petrochemical Industry**. Computers & Chemical Engineering, vol. 153, pp. 107437.

11. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2021). **Data-based Industrial Soft-sensor Design via Optimal Subset Selection**. Editor(s): Metin Türkay, Erdal Aydin, In 31th European Symposium on Computer Aided Process Engineering, Elsevier, vol. 31, pp. 1247–1252.

12. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2020). **Advanced Process Control of an Industrial Depropanizer Column using Data-based Inferential Sensors**. Editor(s): Sauro Pierucci, Flavio Manenti, Giulia Luisa Bozzano, Davide Manca, In 30th European Symposium on Computer Aided Process Engineering, Elsevier, vol. 30, pp. 1213–1218.

13. Mojto, M., Ľubušký, K., Fikar, M., Paulen, R. (2019). **Design of Data-based Inferential Sensors for Industrial Depropanizer Column**. Editor(s): G. Léonard and F. Logist, In Computer Aided Process Engineering, CAPE Forum, pp. 12–13.

14. Mojto, M., Ľubušký, K., Paulen, R., Fikar, M. (2019). **Advanced Process Control of a Depropanizer Column**. Editor(s): M. Fikar and M. Kvasnica, In Proceedings of the 22nd International Conference on Process Control, Slovak Chemical Library, Štrbské Pleso, Slovakia.

15. Mojto, M., Paulen, R., Ľubušký, K., Fikar, M. (2019). **Modelling and Analysis of Control Pairings of an Industrial Depropanizer Column**. In Advanced Process Modelling Forum 26–27 March 2019, pp. 5–6.

Part I

Methodology

Chapter 2

Soft-sensor Theory Foundations

2.1 Introduction to Soft Sensors

The soft sensor is a mathematical structure that estimates the output variable y based on a vector of input variables $\boldsymbol{x}$ ($\boldsymbol{x} \in \mathbb{R}^{n_\mathrm{p}}$), where n_p represents the number of selected input variables. Two types of soft sensors can be distinguished: static (Kadlec et al., 2009; King, 2011) and dynamic (Cao et al., 2020) soft sensors. This thesis focuses on the design of soft sensors with static mathematical structures, which can be represented as follows:

$$\hat{y} = f\left(\boldsymbol{x}, \boldsymbol{\beta}\right), \tag{2.1}$$

where $\hat{y}$ denotes the estimated output variable, $\boldsymbol{\beta}$ is the vector of model parameters, and f is a mathematical function (linear or nonlinear) that establishes the relationship between the input and output variables.

The specific focus of this thesis is on the design of linear soft sensors, chosen for their higher transparency and lower complexity compared to nonlinear soft sensors. The multivariate model structure of linear soft sensors (considering one data point) can be expressed as:

$$\begin{aligned} \hat{y} &= x_1\beta_1 + x_2\beta_2 + \ldots + x_{n_\mathrm{p}}\beta_{n_\mathrm{p}} + \beta_0, \\ &= (x_1, x_2, \ldots, x_{n_\mathrm{p}})(\beta_1, \beta_2, \ldots, \beta_{n_\mathrm{p}})^\intercal + \beta_0 = \boldsymbol{x}^\intercal\boldsymbol{\beta} + \beta_0, \end{aligned} \tag{2.2}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{n_\mathrm{p}}$, and β_0 represents a constant (bias) term.

In order to analyse the multivariate model structure represented in (2.2), more data points should be considered. These data points come from either a sample of the population or the entire population. The sample is a subgroup selected from the population that is representative of it, while the population refers to the complete collection of elements or objects that share specific characteristics. Consequently, the sample size (n) is expected to be smaller than the population size ($N, N \geq n$).

For a sample of data points, the vector of input variables $\boldsymbol{x}$ should be expanded into a sample matrix of input variables $\boldsymbol{X}$ ($\boldsymbol{X} \in \mathbb{R}^{n \times n_{\mathrm{p}}}$) as shown below:

$$\boldsymbol{x} \rightarrow \boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^{\intercal}. \tag{2.3}$$

To simplify the notation and indicate the variables (columns) in $\boldsymbol{X}$ more explicitly, the following notation is used:

$$\boldsymbol{X} = \left(\boldsymbol{x}_{\{1\}}, \boldsymbol{x}_{\{2\}}, \ldots, \boldsymbol{x}_{\{n_{\mathrm{p}}\}}\right), \tag{2.4}$$

where $\boldsymbol{x}_{\{i\}}$ ($\boldsymbol{x}_{\{i\}} \in \mathbb{R}^n$) represents the i^{th} variable (column) of $\boldsymbol{X}$.

For the entire population, the vector of input variables $\boldsymbol{x}$ should be expanded into a population matrix of input variables $\boldsymbol{X}_{\mathrm{p}}$ ($\boldsymbol{X}_{\mathrm{p}} \in \mathbb{R}^{N \times n_{\mathrm{p}}}$) as follows:

$$\boldsymbol{x} \rightarrow \boldsymbol{X}_{\mathrm{p}} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)^{\intercal}. \tag{2.5}$$

Based on the input matrices $\boldsymbol{X}$ and $\boldsymbol{X}_{\mathrm{p}}$, the linear multivariate input structure in equation (2.2) can be generalised to handle more data points as follows:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{\beta} + \beta_0, \tag{2.6}$$

$$\hat{\boldsymbol{y}}_{\mathrm{p}} = \boldsymbol{X}_{\mathrm{p}}\boldsymbol{\beta} + \beta_0, \tag{2.7}$$

where $\hat{\boldsymbol{y}}$ ($\hat{\boldsymbol{y}} \in \mathbb{R}^n$) is the sample vector of the estimated output variable, and $\hat{\boldsymbol{y}}_{\mathrm{p}}$ ($\hat{\boldsymbol{y}}_{\mathrm{p}} \in \mathbb{R}^n$) is the population vector of the estimated output variable.

2.1.1 Training, Validation, and Testing Datasets

The standard procedure for designing a soft-sensor typically involves three datasets: training, validation, and testing. The training dataset is used for the training phase, while the testing dataset is used to evaluate the performance of the soft-sensor. The validation dataset is occasionally employed in specific approaches.

To differentiate between these datasets, corresponding index sets are created. Let $\mathcal{I}$ ($\mathcal{I} := \{1, 2, \ldots, n\}$) be an index set representation of the entire dataset, which includes both input and output variables. The training index set $\mathcal{I}_{\mathrm{T}}$ is defined as follows:

$$\mathcal{I}_{\mathrm{T}} \subset \mathcal{I}, \quad \mathrm{card}(\mathcal{I}_{\mathrm{T}}) = n_{\mathrm{T}}, \quad n_{\mathrm{T}} < n, \tag{2.8}$$

where n_{T} is the size of the training dataset.

The validation index set $\mathcal{I}_{\mathrm{V}}$ is a subset of the training index set, given by:

$$\mathcal{I}_{\mathrm{V}} \subset \mathcal{I}_{\mathrm{T}}, \quad \mathrm{card}(\mathcal{I}_{\mathrm{V}}) = n_{\mathrm{V}}, \quad n_{\mathrm{V}} < n_{\mathrm{T}}, \tag{2.9}$$

where n_{V} is the size of the validation dataset.

Finally, the testing index set $\mathcal{I}_{\mathrm{S}}$ is formulated as:

$$\mathcal{I}_{\mathrm{S}} := \mathcal{I} \setminus \mathcal{I}_{\mathrm{T}}, \quad \mathrm{card}(\mathcal{I}_{\mathrm{S}}) = n_{\mathrm{S}}, \quad n_{\mathrm{S}} < n, \tag{2.10}$$

where n_{S} is the size of the testing dataset.

These index sets enable the division of the dataset into distinct training, validation, and testing subsets. The training dataset consists of elements selected using the training index set $(\boldsymbol{X}(\mathcal{I}_{\mathrm{T}}), \boldsymbol{y}(\mathcal{I}_{\mathrm{T}}))$, the validation dataset consists of elements selected using the validation index set $(\boldsymbol{X}(\mathcal{I}_{\mathrm{V}}), \boldsymbol{y}(\mathcal{I}_{\mathrm{V}}))$, and the testing dataset consists of elements selected using the testing index set $(\boldsymbol{X}(\mathcal{I}_{\mathrm{S}}), \boldsymbol{y}(\mathcal{I}_{\mathrm{S}}))$.

The training dataset $(\boldsymbol{X}(\mathcal{I}_{\mathrm{T}}), \boldsymbol{y}(\mathcal{I}_{\mathrm{T}}))$ is used during the training phase of the soft sensor design procedure. It is typically larger than the testing dataset to provide sufficient information for the model to learn. It is important to ensure that the training dataset is representative of the overall population to enable the model to learn meaningful patterns.

The validation dataset $(\boldsymbol{X}(\mathcal{I}_{\mathrm{V}}), \boldsymbol{y}(\mathcal{I}_{\mathrm{V}}))$ is primarily used in cross-validation (see Sections 3.3.1 and 3.3.3). It helps mitigate overfitting (i.e., fitting the noise), facilitates reliable input structure selection, and supports the ongoing operation of the soft sensor.

The testing dataset $(\boldsymbol{X}(\mathcal{I}_{\mathrm{S}}), \boldsymbol{y}(\mathcal{I}_{\mathrm{S}}))$, also known as an unseen dataset, is not used during the validation phase of the soft-sensor design procedure. Its purpose is to assess how well the trained model generalises to new data. The testing data should be independent of the training dataset and accurately represent the normal operation or future operation of the process.

2.1.2 Soft-sensor Implementation

The life cycle, or implementation, of a soft sensor is illustrated in Figure 2.1. The process involves obtaining frequent measurements (solid line) of the input variables $\boldsymbol{x}$ and infrequent or irregular measurements (dashed line) of the output variable y. The frequent measurements are typically obtained from online sensors, while the infrequent measurements come from laboratory (or lab) analysis. These measurements are then stored in the industrial database, and the historical data from this database can be used for further designing or updating the soft sensor. Furthermore, real-time measurements from the input variables $\boldsymbol{x}$ are fed into the soft sensor to provide an online estimation of the output variable $\hat{y}$. At the same time, the process monitoring and control system

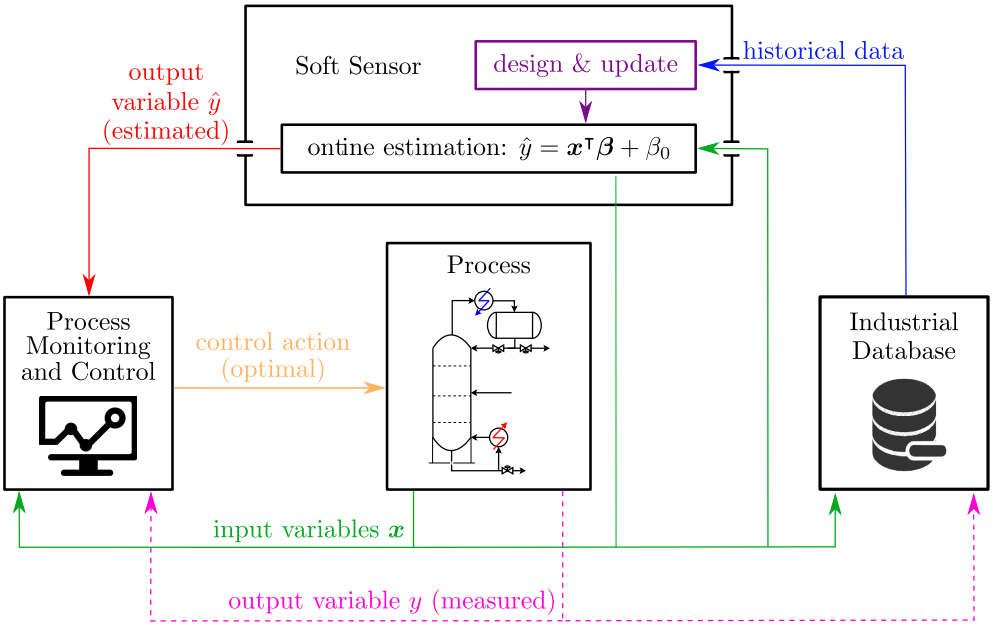


Figure 2.1: Soft sensor life cycle (implementation) in an industrial plant.

receives information from the process and the soft sensor. It ensures that optimal control actions are taken to maintain the desired operation of the process.

The most common application of soft sensors is for monitoring key process variables in various industries. These versatile sensors play a crucial role in obtaining estimates of hard-to-measure output variables ($\hat{y}$) by utilising measurements from easily measurable variables obtained from online sensors ($\boldsymbol{x}$). These easily measurable variables can include temperatures, pressures, flow rates, and other relevant parameters that provide valuable insights into the process. The hard-to-measure variables that soft sensors aim to estimate can vary depending on the specific application, ranging from the concentration of products to equipment ageing or the steady state of the process. By leveraging the relationships between easily measurable and hard-to-measure variables, soft sensors enable real-time monitoring and control of critical process variables, facilitating efficient and optimised operations.

In addition to their role in monitoring key process variables, soft sensors also find widespread application in fault detection systems. Faults in industrial processes typically indicate abnormal or undesirable operating conditions that can lead to inefficiencies, safety risks, or product quality issues. Soft sensors, with their ability

to accurately classify measurements as normal or abnormal, offer a valuable tool for detecting and diagnosing faults. These classification-based soft sensors utilise advanced algorithms and techniques to analyse the collected data and identify deviations from normal operation. By detecting anomalies in process variables, such as unexpected fluctuations or patterns, soft sensors can provide early warnings of potential issues and facilitate proactive maintenance or intervention. For example, in the petrochemical industry, soft sensors can be employed to detect flooding in industrial distillation columns, which can have severe consequences for process efficiency and product quality. Similarly, in industrial chemical reactors, soft sensors can identify dangerous operations or deviations from safe operating conditions, allowing operators to take prompt corrective actions and prevent accidents or costly disruptions. Overall, the integration of soft sensors in fault detection systems enhances process monitoring capabilities and contributes to improved operational performance and safety in various industries.

2.2 Statistics for Soft-sensor Design

This section provides an overview of the essential statistical concepts necessary for designing linear data-driven soft sensors. This includes the description of the normal distribution, introducing key statistical terms. Additionally, we delve into the statistics of multivariate systems, including the covariance and correlation matrices, as well as Hotelling's T^2 distribution.

2.2.1 Normal Distribution

The normal (Gaussian) distribution is a fundamental probability distribution widely used in statistical analysis. It is characterised by its symmetrical bell-shaped curve. The probability density function (PDF) of the normal distribution is defined by the following equation (Hastie et al., 2017):

$$f(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left(-\frac{(x-\mu_x)^2}{2\sigma_x^2}\right), \tag{2.11}$$

where x represents a random variable following the normal distribution, μ_x corresponds to the mean value (centre of the distribution), and σ_x denotes the standard deviation (measure of variability).

The normal distribution is of great importance in statistics and probability theory, as evidenced by the central limit theorem. This fundamental concept states that the

distribution of the sum or average of a large number of independent and identically distributed random variables tends to approximate a normal distribution, regardless of the shape of the original distribution. In simpler terms, when we take a large number of random samples (with a sample size greater than 30) from any population and compute the sum or average of those samples, the resulting distribution will closely resemble a normal distribution, irrespective of the shape of the original population.

The normal distribution serves as a foundational assumption in statistical analysis, providing the basis for various statistical techniques and models employed in this thesis. Moreover, it is a valuable tool for understanding and analysing data in a wide range of fields, including the social sciences, economics, engineering, and natural sciences. It enables researchers to apply statistical methods that rely on the assumption of normality and facilitates inference, hypothesis testing, and estimation of confidence intervals.

Mean Value The mean value is a measure of central tendency that represents the typical or average value of a set of numbers (Hastie et al., 2017). For the sample of data points, the sample mean value ($\bar{x}$) is given in the following form:

$$\bar{x} = \frac{1}{n} \sum_{k=1}^{n} x_k. \tag{2.12}$$

The mean value for the whole population (μ_x) has following form:

$$\mu_x = \frac{1}{N} \sum_{k=1}^{N} x_k. \tag{2.13}$$

Standard Deviation It is a measure that represents the amount of variation or dispersion within the studied set of data points (Hastie et al., 2017). The standard deviation for the sample from the population is as follows:

$$s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})^2}, \tag{2.14}$$

where s_x is the sample standard deviation.

With the respect to the whole population, the standard deviation achieves following

form:

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(x_k - \mu_x)^2}, \tag{2.15}$$

where σ_x is the population standard deviation.

2.2.2 Covariance and Correlation Matrices

To understand the relationships between the input variables in $\boldsymbol{X}$, we can calculate the (sample) covariance matrix $\boldsymbol{\Sigma}$ ($\boldsymbol{\Sigma} \in \mathbb{R}^{n_\mathrm{p} \times n_\mathrm{p}}$) using the following formula (Hastie et al., 2017):

$$\Sigma_{i,j} = \frac{1}{n-1}\sum_{k=1}^{n}(X_{k,i} - \bar{x}_{\{i\}})(X_{k,j} - \bar{x}_{\{j\}}), \quad \forall i, \forall j \in \{1, 2, \ldots, n_\mathrm{p}\}, \tag{2.16}$$

where $\Sigma_{i,j}$ is a covariance coefficient of $\boldsymbol{\Sigma}$, and $\bar{x}_{\{i\}}$ and $\bar{x}_{\{j\}}$ are mean values for the i^{th} and j^{th} input variables of $\boldsymbol{X}$ computed by (2.12).

For the matrix with the population of the input variables $\boldsymbol{X}_\mathrm{p}$, the population covariance matrix $\boldsymbol{\Sigma}_\mathrm{p}$ ($\boldsymbol{\Sigma}_\mathrm{p} \in \mathbb{R}^{n_\mathrm{p} \times n_\mathrm{p}}$) can be derived as:

$$\Sigma_{\mathrm{p},i,j} = \frac{1}{N}\sum_{k=1}^{N}\left(X_{k,i} - \mu_{\boldsymbol{x}_{\{i\}}}\right)(X_{k,j} - \mu_{\boldsymbol{x}_{\{j\}}}), \quad \forall i, \forall j \in \{1, 2, \ldots, n_\mathrm{p}\}, \tag{2.17}$$

where $\Sigma_{\mathrm{p},i,j}$ is a covariance coefficient of $\boldsymbol{\Sigma}_\mathrm{p}$, and $\mu_{\boldsymbol{x}_{\{i\}}}$ and $\mu_{\boldsymbol{x}_{\{j\}}}$ are mean values for the i^{th} and j^{th} input variables of $\boldsymbol{X}_\mathrm{p}$ computed by (2.13).

The covariance matrix $\boldsymbol{\Sigma}$ provides information about the covariance between pairs of input variables. It indicates the direction in which the variables move together. A positive covariance coefficient $\Sigma_{i,j}$ suggests that the corresponding input variables have a positive covariance and move in the same direction. Conversely, a negative covariance coefficient indicates the opposite. Furthermore, the magnitude of the covariance reflects the strength of the relationship between the variables.

Based on the sample covariance matrix $\boldsymbol{\Sigma}$, it is possible to derive the sample correlation matrix $\mathbf{R}$ ($\mathbf{R} \in \mathbb{R}^{n_\mathrm{p} \times n_\mathrm{p}}$) as follows:

$$r_{i,j} = \frac{\Sigma_{i,j}}{s_{\boldsymbol{x}_{\{i\}}} s_{\boldsymbol{x}_{\{j\}}}}, \quad \forall i, \forall j \in \{1, 2, \ldots, n_\mathrm{p}\}, \tag{2.18}$$

where $r_{i,j}$ is a correlation coefficient of $\mathbf{R}$, and $s_{\boldsymbol{x}_{\{i\}}}$ and $s_{\boldsymbol{x}_{\{j\}}}$ are standard deviations for the i^{th} and j^{th} input variables of $\boldsymbol{X}$ calculated using (2.14).

The correlation coefficient $r_{i,j}$ shares similar characteristics with the covariance coefficient $\Sigma_{i,j}$, but it is not affected by differences in the scale of the input variables. All covariance coefficients within $\mathbf{R}$ are normalised within the interval $[-1, 1]$, where -1 indicates a perfect negative linear correlation, 1 indicates a perfect positive linear correlation, and 0 indicates no linear correlation.

2.2.3 Hotelling's Distribution

The Hotelling's T^2 distribution (Hotelling, 1931) is a generalisation of the Student's t-distribution (Student, 1908). It originates from multivariate hypothesis testing, where Hotelling's T^2 statistic is defined as:

$$T^2 = n(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_x)^\intercal \boldsymbol{\Sigma}^{-1}(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_x), \tag{2.19}$$

where T^2 represents the Hotelling's T^2 statistic, $\bar{\boldsymbol{x}}$ ($\bar{\boldsymbol{x}} \in \mathbb{R}^{n_\mathrm{p}}$) is the sample mean vector computed by (2.12) for $\boldsymbol{X}$, and $\boldsymbol{\mu}_x$ ($\boldsymbol{\mu}_x \in \mathbb{R}^{n_\mathrm{p}}$) is the population mean vector computed by (2.13) for $\boldsymbol{X}_\mathrm{p}$.

The Hotelling's T^2 distribution follows the F-distribution, which is given by:

$$T^2 \sim T^2_{n_\mathrm{p},n-1} = \frac{n_\mathrm{p}(n-1)}{n - n_\mathrm{p}} F_{n_\mathrm{p},n-n_\mathrm{p}}, \tag{2.20}$$

where $F_{n_\mathrm{p},n-n_\mathrm{p}}$ represents the F-distribution.

The value of T^2 is proportional to the distance between the sample mean and population mean. The Hotelling's T^2 distance (T^2 distance) of each point from $\boldsymbol{\mu}_x$ can be evaluated as:

$$T^2_i = d_{T^2,i} = n(\boldsymbol{x}_i - \boldsymbol{\mu}_x)^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_x), \quad \forall i = \{1, 2, \ldots, n\}, \tag{2.21}$$

where $d_{T^2,i}$ represents i^{th} element of $\boldsymbol{d}_{T^2}$ ($\boldsymbol{d}_{T^2} \in \mathbb{R}^n$), and $\boldsymbol{x}_i$ represents the i^{th} row (data point) of $\boldsymbol{X}$.

Expanding the Hotelling's T^2 statistic stated in (2.19), we can derive the vector of Mahalanobis distances $\boldsymbol{d}_\mathrm{M}$ ($\boldsymbol{d}_\mathrm{M} \in \mathbb{R}^n$) using the following expression:

$$d_{\mathrm{M},i} = \sqrt{(\boldsymbol{x}_i - \boldsymbol{\mu}_x)^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_x)}, \quad \forall i = \{1, 2, \ldots, n\}. \tag{2.22}$$

The Mahalanobis distance $\boldsymbol{d}_\mathrm{M}$ quantifies how different or similar a data point is from a distribution or a reference set of data points.

From the perspective of data-driven soft-sensor design, both distances ($\boldsymbol{d}_{T^2}$ and $\boldsymbol{d}_\mathrm{M}$) can be utilised for data treatment analysis to detect outliers and systematic errors

within the industrial dataset. If the dataset is centred (with a zero mean, $\boldsymbol{\mu}_x = \mathbf{0}$), the distance represented by d_{T^2} or d_{M} can identify measurements that deviate the most from the centre, commonly known as outliers. An increasing value of this distance indicates a higher probability that the measurement is an outlier. The condition to differentiate between acceptable and unacceptable measurements is typically established using the empirical 3σ rule of thumb (which includes 99.7 % of the measurements) or the χ^2 test. However, some adjustments may be necessary depending on the ideality (normality) of the data. In practise, industrial datasets are often highly non-ideal with a significant presence of noise, requiring manual tuning of this condition.

2.3 Regression, Clustering, and Classification

The three fundamental tasks essential for soft-sensor design are regression, classification, and clustering. This section provides a brief description and mathematical representation of each of these tasks.

2.3.1 Regression

Regression, specifically linear regression (which is the focus of this thesis), is a supervised learning task that aims to predict the output variable y based on input variables $\boldsymbol{x}$. The basic mathematical formulation of a regression problem can be expressed as follows:

Given a set of input variables $\boldsymbol{X}$ and their corresponding output values $\boldsymbol{y}$, the goal is to find a function f that maps $\boldsymbol{X}$ to $\boldsymbol{y}$ using the equation:

$$\boldsymbol{y} = f(\boldsymbol{X}, \boldsymbol{\beta}, \beta_0) + \boldsymbol{\epsilon}, \tag{2.23}$$

where $\boldsymbol{\epsilon}$ ($\boldsymbol{\epsilon} \in \mathbb{R}^n$) represents a vector of random error terms.

In this work, we focus on the linear structure of the designed soft sensors. Therefore, the term $f(\boldsymbol{X}, \boldsymbol{\beta}, \beta_0)$ in (2.23) can be replaced with the vector of estimated output variables $\hat{\boldsymbol{y}}$ obtained from (2.6). The goal of regression is to minimise the difference between the measured (y) and estimated ($\hat{y}$) values of the output variable by finding the optimal estimates of the model parameters ($\{\boldsymbol{\beta}, \beta_0\} \rightarrow \{\boldsymbol{\beta}^*, \beta_0^*\}$, where $\boldsymbol{\beta}^* \in \mathbb{R}^{n_{\mathrm{p}}^*}$). The resulting model ($\boldsymbol{\beta}^*, \beta_0^*$) may not include all the original input variables ($n_{\mathrm{p}}^* \leq n_{\mathrm{p}}$). This is because some input variables may be eliminated during the regression process due to their low correlation with the output variable.

2.3.2 Regression Performance Indices

The quality of regression models is evaluated based on their accuracy and complexity. Accuracy refers to how closely the estimated values ($\hat{y}$) of the output variable match the measured values (y). Several criteria are commonly used to assess the accuracy of regression models:

- Sum of squares error (SSE): It quantifies the total sum of the squared differences between each measured value (y_k) and its corresponding estimated value ($\hat{y}_k$), given by:
$$\text{SSE} = \sum_{k=1}^{n} (y_k - \hat{y}_k)^2. \tag{2.24}$$
- Sum of absolute errors (SAE): It measures the mean of the absolute differences between each measured value (y_k) and its corresponding estimated value ($\hat{y}_k$), given by:
$$\text{SAE} = \frac{1}{n} \sum_{k=1}^{n} |y_k - \hat{y}_k|. \tag{2.25}$$
- Root-mean-square error (RMSE): It represents the square root of the mean of the squared differences between the measured values (y_k) and their corresponding estimated values ($\hat{y}_k$), calculated as:
$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (y_k - \hat{y}_k)^2}. \tag{2.26}$$
- Coefficient of determination R^2: It measures the proportion of the variance in the output (dependent) variable that can be explained by the input (independent) variables in a regression model, calculated as:
$$R^2 = 1 - \frac{\sum_{k=1}^{n} (y_k - \hat{y}_k)^2}{\sum_{k=1}^{n} (y_k - \bar{y})^2}, \tag{2.27}$$
where $\bar{y}$ is the mean value of the output variable.

The aforementioned performance indices provide measures of accuracy for regression models. Higher values of SSE, SAE, and RMSE indicate lower accuracy. SSE and RMSE penalise larger errors more heavily, while SAE treats all errors equally. The coefficient of determination, denoted by R^2, represents the proportion of the variance of the output variable explained by the input variables. A higher value of R^2 suggests a better fit of the model to the data.

The complexity of a regression model can be assessed by the number of parameters n_p^* within the resulting model structure. It is important to note that certain approaches, such as PCA regression (see Section 3.2.1) or PLS (see Section 3.2.2), may result in models composed of principal components that require the complete set of original input variables. When evaluating the overall performance of a regression model, it is desirable to minimise the complexity of the model structure. However, reducing complexity without control can lead to a significant decrease in accuracy. Therefore, finding the right balance between accuracy and complexity is crucial.

On an industrial scale, the performance of a regression model is closely related to the quality of the available dataset. Noise is an inevitable part of industrial data and is present to varying degrees in both input and output variables. As the complexity of the model structure increases, there is a higher tendency for the soft sensor to follow the noise rather than accurately capture the behaviour of the output variable. This phenomenon is known as model overfitting, which is an undesirable aspect of data-driven models. To avoid or reduce model overfitting, the following criteria can be considered:

- Corrected Akaike Information Criterion (AICc): A modification of the Akaike information criterion (AIC) that addresses the issue of small sample sizes. It adjusts the penalty term for model complexity to provide a more reliable estimate of model performance. The AICc can be formulated as follows:

$$\mathrm{AICc} = \mathrm{AIC} + \frac{2n_\mathrm{p}^*(n_\mathrm{p}^*+1)}{n-n_\mathrm{p}^*-1} = n\ln\left(\frac{\mathrm{SSE}}{n}\right) + \frac{2n_\mathrm{p}^*(n_\mathrm{p}^*+1)}{n-n_\mathrm{p}^*-1}, \tag{2.28}$$

  where n is the number of data points, n_p^* is the number of parameters in the model, and SSE is the sum of squares error. A lower AICc value represents a better trade-off between model accuracy and complexity.

- Bayesian Information Criterion (BIC): Another criterion for model selection that penalises model complexity. It is derived from a Bayesian perspective and provides a balance between model fit and complexity. The BIC is mathematically represented as:

$$\mathrm{BIC} = n\ln\left(\frac{\mathrm{SSE}}{n}\right) + \ln(n)n_\mathrm{p}^*. \tag{2.29}$$

- Adjusted Coefficient of Determination R^2_adj: An appropriate modification of the coefficient of determination R^2 that takes into account the model complexity. It is computed as:

$$R^2_\mathrm{adj} = 1 - \frac{(1-R^2)(n-1)}{n-n_\mathrm{p}^*-1} = 1 - \frac{\mathrm{SSE}}{n-n_\mathrm{p}^*-1}. \tag{2.30}$$

These criteria provide valuable insights into the balance between model accuracy and complexity, allowing for better decision-making in the selection and evaluation of regression models in industrial settings.

2.3.3 Clustering

Clustering is an unsupervised learning task that aims to group similar data points together based on their inherent characteristics or patterns. The basic mathematical formulation of a clustering problem can be described as follows:

Given a set of data points $\mathcal{D} \in \mathbb{R}^{n_\mathrm{p}}$, the objective is to partition the data into n_cl clusters, where each cluster represents a group of similar data points. This can be represented as:

$$\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{n_\mathrm{cl}}\}, \quad \mathcal{D} = \bigcup_i \mathcal{C}_i, \quad \forall i \in \{1, 2, \ldots, n_\mathrm{cl}\}, \tag{2.31}$$

where $\boldsymbol{\mathcal{C}}$ represents the set of clusters, and $\mathcal{C}_i$ represents the i^{th} cluster.

2.3.4 Clustering Performance Indices

Clustering performance indices are quantitative measures utilised to assess the quality of clustering outcomes. They offer valuable insights into the compactness and separation of clusters, enabling a comprehensive evaluation of clustering algorithms. Three widely recognised and frequently used clustering performance indices are following (Gan et al., 2020):

- Silhouette Coefficient: It evaluates clustering solutions by considering the compactness of data points within their clusters and the separation between different clusters. Ranging from -1 to 1, a higher value signifies superior clustering results. It takes into account the average distance between a data point and all other points within its cluster (intra-cluster distance) as well as the average distance to the nearest neighbouring cluster (inter-cluster distance).

- Dunn Index: It assesses clustering quality based on cluster compactness and separation. It computes the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance. A higher Dunn Index value indicates improved clustering results, characterized by more compact and well-separated clusters. The Dunn Index provides a global measure of clustering quality.

- Calinski-Harabasz Index: It measures the ratio of between-cluster dispersion to within-cluster dispersion. It quantifies cluster compactness and separation. A higher Calinski-Harabasz Index value suggests better clustering outcomes, with more distinct and well-separated clusters.

2.3.5 Classification

Classification is a supervised learning task where the goal is to assign input data points to predefined classes or categories. The basic mathematical formulation of a classification problem can be stated as follows:

Given dataset of input variables $\boldsymbol{X}$ and a set of predefined classes $\boldsymbol{\mathcal{C}}$, the task is to find a function f that maps the inputs to the corresponding classes. Mathematically, this can be represented as:

$$f : \mathbb{R}^{n_{\mathrm{p}}} \rightarrow \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{n_{\mathrm{cl}}}\}. \tag{2.32}$$

2.3.6 Classification Performance Indices

In this section, we derive performance indices for a system consisting of two classes: positive and negative. It is important to note that these indices can also be derived for multi-class problems with more than two classes. The outcomes of clustering and classification can fall into four categories:

- True positive (TP): instances that are truly positive and correctly classified as positive.
- True negative (TN): instances that are truly negative and correctly classified as negative.
- False positive (FP): instances that are actually negative but incorrectly classified as positive.
- False negative (FN): instances that are actually positive but incorrectly classified as negative.

Based on the aforementioned four categories, several well-known normalised performance criteria can be evaluated:

- Accuracy (AC): It measures the overall correctness of the classifier as follows:

$$\mathrm{AC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}. \tag{2.33}$$

- Precision (PR): It quantifies the precision or correctness of the true positive predictions as follows:

$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{2.34}$$

- Recall (RC): It measures how well the classifier predicts actual positive observations as follows:

$$\mathrm{RC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{2.35}$$

- F1-Score (F1): It is the harmonic mean between precision and recall as follows:

$$\mathrm{F1} = \frac{2 \times \mathrm{PR} \times \mathrm{RC}}{\mathrm{PR} + \mathrm{RC}}. \tag{2.36}$$

These performance indices, namely accuracy (AC), precision (PR), recall (RC), and F1-Score (F1), provide valuable insights into the effectiveness of clustering and classification models. They help assess the correctness and predictive power of the models.

Chapter 3

Soft-sensor Algorithm Foundations

The design of data-driven soft sensors is typically tailored to a specific application. This is because the performance of a data-driven soft sensor can be greatly influenced by the behaviour of the output variable being inferred or the quality of the available data for the soft sensor design. Therefore, it is important to have at least an intuitive knowledge of various potential algorithms that can be applied in different situations and circumstances. The following sections of this thesis describe fundamental and widely used approaches that are frequently employed in the design of linear data-driven soft sensors.

3.1 Ordinary Least Squares Regression

Ordinary least squares (OLS) regression is a commonly used statistical method for estimating the relationship between input (independent) variables and one or more output (dependent) variables. OLS regression aims to find the optimal parameters within the linear model (β_0 and $\boldsymbol{\beta}$ from (2.2)) while minimising the sum of squared errors (SSE) between the observed (y) and predicted ($\hat{y}$) output variables.

The vector of predicted output variables $\hat{\boldsymbol{y}}$ is given by (2.6). With respect to $\hat{\boldsymbol{y}}$, the objective of OLS regression is represented by the following expression:

$$\min_{\boldsymbol{\beta},\beta_0}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\beta_0)^{\intercal}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\beta_0)=\min_{\boldsymbol{\beta},\beta_0}\mathrm{SSE}, \tag{3.1}$$

where $\boldsymbol{y}$ ($\boldsymbol{y}\in\mathbb{R}^n$) is a vector of the observed output variable.

The objective function represents the sum of squared errors (SSE) between the observed and predicted values. By minimising this function with respect to the parameters $\boldsymbol{\beta}$ and β_0, we obtain the optimal estimates ($\{\boldsymbol{\beta},\,\beta_0\}\rightarrow\{\boldsymbol{\beta}^*,\,\beta_0^*\}$) that provide the best fit to the data.

3.2 Variance-covariance Algorithms

3.2.1 Principal Component Analysis

Principal component analysis (PCA) (Pearson, 1901) is one of the most important representatives of the unsupervised machine learning family of approaches. This statistical approach focuses on effective dimensionality reduction. PCA searches for a lower-dimensional representation of a high-dimensional dataset while preserving the most important information. This is achieved by identifying orthogonal vectors called principal components (PC) that explain the maximum variance in the data.

The visualisation of PCA, considering a two-dimensional example, is depicted in Figure 3.1. Since PCA is an unsupervised learning method, both dimensions are considered inputs. The dataset under study is represented by red dots and involves two principal components, namely PC_1 (indicated by the blue double arrow) and PC_2 (indicated by the black double arrow). The orientation and width of these principal components explain the covariance present in the dataset, which is represented by the variance-covariance matrix (depicted as a green ellipse).

The PCA procedure begins by computing the covariance matrix $\boldsymbol{\Sigma}$ (as shown in Equation (2.16)) from the available input dataset $\boldsymbol{X}$. In Equation (2.16), the mean value of each input variable within $\boldsymbol{X}$ is subtracted. This ensures that the resulting principal components are not influenced by the overall shift or centre of the input dataset.

Subsequently, singular value decomposition (SVD) is performed to factorise $\boldsymbol{\Sigma}$ as follows:

$$\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{M}\boldsymbol{V}^{\intercal}, \tag{3.2}$$

where $\boldsymbol{U}$ ($\boldsymbol{U} \in \mathbb{R}^{n \times n}$) is an orthogonal matrix containing the left singular vectors (eigenvectors), $\boldsymbol{M}$ ($\boldsymbol{M} \in \mathbb{R}^{n \times n_{\mathrm{pc}}}$) is a diagonal matrix containing the singular values (eigenvalues), and $\boldsymbol{V}$ ($\boldsymbol{V} \in \mathbb{R}^{n_{\mathrm{pc}} \times n_{\mathrm{pc}}}$) is an orthogonal matrix containing the right singular vectors (eigenvectors), n_{pc} is the number of principal components.

After performing SVD, the original input dataset $\boldsymbol{X}$ can be projected into a lower-dimensional representation $\boldsymbol{X}_{\mathrm{low}}$ ($\boldsymbol{X}_{\mathrm{low}} \in \mathbb{R}^{n \times n^*_{\mathrm{pc}}}$) considering n^*_{pc} ($n^*_{\mathrm{pc}} \leq n_{\mathrm{pc}}$) dimensions (principal components) as follows:

$$\boldsymbol{X}_{\mathrm{low}} = \boldsymbol{U}^{\intercal}_{\mathrm{low}}[\boldsymbol{X} - (\mathbb{1}\bar{\boldsymbol{x}}^{\intercal})^{\intercal}], \tag{3.3}$$

where $\boldsymbol{U}_{\mathrm{low}}$ ($\boldsymbol{U}_{\mathrm{low}} \in \mathbb{R}^{n \times n^*_{\mathrm{pc}}}$) is lower-dimensional matrix considering only first n^*_{pc} columns of $\boldsymbol{U}$.

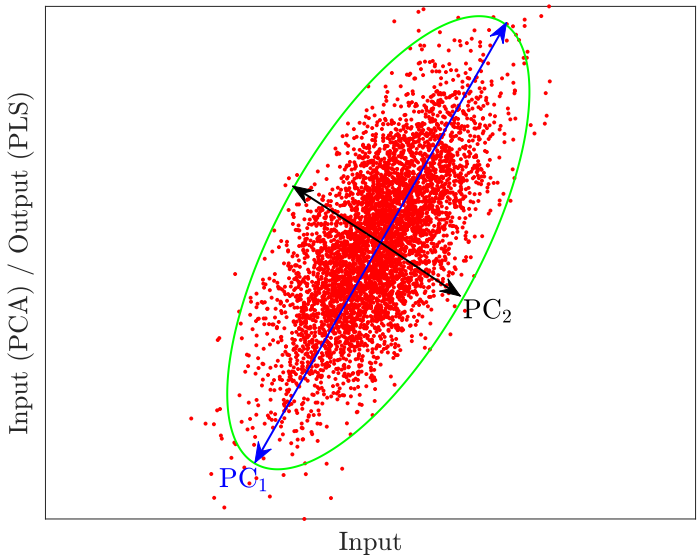


Figure 3.1: The visualization of principal components (PC_1, PC_2) of the input variable with another input variable (PCA) or the output variable (PLS).

Each eigenvector (within $\boldsymbol{U}$ or $\boldsymbol{V}$) represents one principal component that explains a certain amount of the data variance. The desired amount of total variance can be captured by selecting several (n_{pc}^*) principal components within the eigenvectors with the maximum explained variance. Regression can then be carried out over the selected principal component subspace by solving Equation (3.1) (OLS regression), considering n_{pc}^* parameters. This interconnection of PCA and OLS approaches is called PCA regression.

The use of PCA regression represents an advantage, mainly in cases where there is an insufficient amount of output data. This is often the situation in the industry, where measuring the desired output variable is either expensive or rare. Such a situation can lead to the performance deterioration of many data-driven methods for soft sensor design, as they usually require a large number of measurements. The popularity of PCA regression has grown due to its ability to learn from measurements of online sensors, enabling it to outperform other data-driven methods in certain cases.

3.2.2 Partial Least Squares

Partial least squares (PLS) regression (Wold et al., 1984) is a supervised machine learning method that aims to create a linear regression model for predicting the output variable based on the input variables. PLS regression is similar to principal component analysis (PCA) in that both methods reduce the dimensionality of the problem.

In general, the PLS approach repeats the following steps multiple times to obtain the desired number of principal components, denoted as n_{pc}^{*}:

1. Identifying the directions of maximum covariance between the input dataset ($\boldsymbol{X}$) and the output dataset ($\boldsymbol{y}$).
2. Applying OLS regression on the input scores, which are the projections of $\boldsymbol{X}$ onto the identified directions.
3. Deflating (subtracting the estimated component) the input and output datasets based on the obtained component.

There are variations in the specific details of this procedure, and two commonly used approaches are nonlinear iterative partial least squares (NIPALS) and simple partial least squares (SIMPLS) (de Jong, 1993). Both NIPALS and SIMPLS iteratively calculate the principal components.

To determine the maximum covariance between the input and output datasets, the cross-covariance matrix $\boldsymbol{\Sigma}_{XY}$ ($\boldsymbol{\Sigma}_{XY} \in \mathbb{R}^{n_{\text{p}} \times 1}$) is evaluated using the following formula:

$$\boldsymbol{\Sigma}_{XY} = \frac{1}{n-1}(\boldsymbol{X}^{\intercal} - \mathbb{1}\bar{\boldsymbol{x}}^{\intercal})(\boldsymbol{y} - \bar{y}). \tag{3.4}$$

The desired number of principal components is then selected using a similar approach as in PCA. These principal components are subsequently utilised in designing the soft sensor, similar to the application of PCA. Figure 3.1 provides a visualisation of the performance of PCA with PLS. The key distinction between these approaches is that PCA considers only the input variables, while PLS incorporates both the input and output variables.

3.3 Sparsity-based Algorithms

During the training of the model structure, there is a tendency for overfitting (as discussed in Section 2.3.2) due to the presence of noise. Therefore, it is essential to

find an appropriate balance between prediction accuracy and model complexity. The following approaches focus on achieving this balance by determining an effective input structure with the desired sparsity based on specific criteria.

3.3.1 Cross-Validation

Cross-validation (CV) is a commonly employed technique for assessing model performance and mitigating the issue of overfitting. It involves dividing the available training dataset into training and validation subsets, allowing the model to be trained on one subset and evaluated on the other.

The most widely used CV approach is k-fold cross-validation, where the dataset is divided into k equally sized folds (subsets). The model is trained on $k-1$ folds and evaluated on the remaining fold. This process is repeated k times, with each fold serving as the validation set once. Performance metrics (as discussed in Section 2.3.2), such as SSE, SAE, RMSE, or R^2, are computed in each iteration, and the average performance across all iterations is determined.

By utilising CV, the risk of overfitting is significantly reduced, as it provides a more reliable assessment of the behaviour of the model within the available dataset. The consistency of performance across different folds helps identify input model structures that exhibit high stability and robustness. An ideal input structure should consistently yield good performance, even when applied to unseen or testing datasets.

3.3.2 Least Absolute Shrinkage and Selection Operator

The least absolute shrinkage and selection operator (LASSO) method (Santosa and Symes, 1986; Tibshirani, 1996) is a regularisation technique that addresses the problems of model selection and parameter estimation. It solves the following optimisation problem:

$$\min_{\boldsymbol{\beta},\beta_0} \frac{1}{2n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\beta_0\|_2^2+\lambda\|\boldsymbol{\beta}\|_1, \tag{3.5}$$

where λ is a weight that controls the trade-off between model accuracy and model complexity.

The objective function consists of two terms: the least squares term, which measures the discrepancy between the predicted values and the actual output values, and the penalization term, which is the sum of the absolute values of the coefficients multiplied by λ. The use of the ℓ_1-norm penalization encourages sparsity in the model, leading to some coefficients being exactly zero.

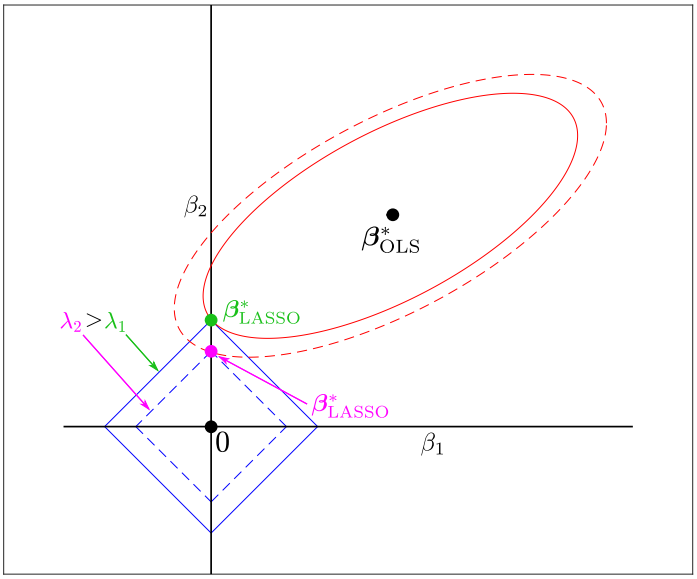


Figure 3.2: The impact of different values of λ on LASSO performance.

The tuning of the LASSO parameter λ is crucial in achieving an optimal balance between model complexity and goodness of fit. Smaller values of λ result in higher accuracy but may lead to overfitting, while larger values of λ promote sparsity at the expense of accuracy. To determine an appropriate value for λ, model-overfitting criteria (see Section 2.3.2) or cross-validation techniques (see Section 3.3.1) can be employed.

The LASSO method belongs to the family of regularised regression techniques. Other important methods in this family include ridge regression (Hoerl and Kennard, 1970) and elastic net (Zou and Hastie, 2005). Ridge regression uses the ℓ_2-norm penalization to shrink the values of all parameters and is particularly useful when the input variables are highly correlated. Elastic net, on the other hand, combines the ℓ_1-norm and ℓ_2-norm penalties to strike a balance between sparsity and parameter shrinkage.

The performance of LASSO is illustrated in Figure 3.2. The penalty term in the LASSO objective function, as shown in equation (3.5), promotes sparsity in the solution. It is evident from the plot that the resulting parameters obtained from ordinary least squares (OLS) (β^*_{OLS}) are non-zero, whereas one of the resulting parameters obtained from LASSO (β^*_{LASSO}) is zero. Furthermore, it is apparent that increasing the value of λ enhances the sparsity of the LASSO model. The magenta β^*_{LASSO} corresponding to

λ_2 is closer to zero compared to the green β^*_{LASSO} corresponding to λ_1, where $\lambda_2 > \lambda_1$.

3.3.3 Subset Selection

Subset selection (SS) is a class of methods that aim to find the simplest input structure for the soft sensor based on specific criteria or objectives. It addresses two tasks simultaneously: (a) determining the simplest input structure and (b) estimating the model parameters. To quantify the complexity and facilitate mathematical calculations, a binary variable vector $\boldsymbol{z}_{\text{ss}}$ ($\boldsymbol{z}_{\text{ss}} \in \mathbb{R}^{n_{\text{p}}}$) is introduced. This allows us to formulate the following bi-level optimisation problem (Berger et al., 2016):

$$\min_{\boldsymbol{\beta},\beta_0,\boldsymbol{z}_{\text{ss}}\in\{0,1\}^{n_{\text{p}}}} J(\boldsymbol{y},\boldsymbol{\beta},\beta_0,\boldsymbol{z}_{\text{ss}}), \tag{3.6a}$$

$$\text{s.t.} \quad \{\boldsymbol{\beta},\beta_0\} \in \arg\min_{\tilde{\boldsymbol{\beta}},\tilde{\beta}_0} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\tilde{\boldsymbol{\beta}}-\tilde{\beta}_0\|_2^2, \tag{3.6b}$$

$$\text{s.t.} \quad -\bar{\beta}z_{\text{ss},j} \le \tilde{\beta}_j \le \bar{\beta}z_{\text{ss},j}, \quad \forall j \in \{1,2,\ldots,n_{\text{p}}\}, \tag{3.6c}$$

where $\bar{\beta}$ represents an upper bound on $\|\boldsymbol{\beta}\|_\infty$ that needs to be tuned, and $J(\boldsymbol{y},\boldsymbol{\beta},\beta_0,\boldsymbol{z}_{\text{ss}})$ is the objective function in for of SSE for the ordinary subset selection approach.

The bi-level optimization problem (3.6) consists of the upper-level objective (3.6a) and the lower-level objective (3.6b). The upper-level objective seeks the optimal input structure ($\boldsymbol{z}^*_{\text{ss}}$), while the lower-level optimisation estimates the model parameters ($\boldsymbol{\beta}$ and β_0). The resulting complexity of the model structure can be expressed as:

$$n^*_{\text{p}} = \mathbb{1}^\intercal \boldsymbol{z}^*_{\text{ss}}, \tag{3.7}$$

where n^*_{p} (where $n^*_{\text{p}} \le n_{\text{p}}$) is the resulting number of the input variables that achieves the desired objectives.

The bi-level optimization problem (3.6) can be effectively solved using standard MIQP solvers with big-M reformulation, as demonstrated in (Takano and Miyashiro, 2020). In this thesis, SS is combined with the model-overfitting criterion (SS-MOC) presented in Section 2.3.2 and with cross-validation (SS-CV) as discussed in Section 3.3.1.

Subset Selection with Model-overfitting Criterion The SS optimisation problem presented in (3.6) offers flexibility in adapting its performance to specific situations. One natural modification is to set the objective function $J(\cdot)$ in (3.6a) to be one of the MOC criteria discussed in Section 2.3.2 (e.g., AICc, BIC, or R^2_{adj}). By considering these criteria, the resulting input structure provided by SS should be less complex

and less affected by model overfitting compared to the ordinary SS approach based on the SSE criterion. This reduced complexity in the input structure can lead to lower implementation and maintenance costs. The resulting optimisation problem of SS with the model-overfitting criterion (SS-MOC) can be effectively solved using the same solvers as the standard SS approach mentioned in Section 3.3.3.

Subset Selection with Cross-validation Subset selection with cross-validation (SS-CV) (Smith, 2018; Takano and Miyashiro, 2020) addresses the same tasks as SS-MOC, but it employs standard cross-validation (see Section 3.3.1) instead of the MOC criteria. At the beginning, the available training dataset represented by $\mathcal{I}_{\mathrm{T}}$ index set (see Section 2.1.1) is divided into K smaller subsets $\mathcal{I}_{\mathrm{s},k}$, defined as follows:

$$\mathcal{I}_{\mathrm{T}} = \bigcup_{k \in K} \mathcal{I}_{\mathrm{s},k}, \quad \mathcal{I}_{\mathrm{s},k} \cap \mathcal{I}_{\mathrm{s},k'} = \emptyset, \quad \forall k \neq k', \quad K \geq 2. \tag{3.8}$$

The data is distributed into training ($\mathcal{I}_{\mathrm{T},k}$) and validation ($\mathcal{I}_{\mathrm{V},k}$) index sets as follows:

$$\mathcal{I}_{\mathrm{V},k} := \mathcal{I}_{\mathrm{s},k}, \quad \mathcal{I}_{\mathrm{T},k} := \mathcal{I}_{\mathrm{T}} \setminus \mathcal{I}_{\mathrm{s},k}, \quad \mathrm{card}(\mathcal{I}_{\mathrm{T},k}) \geq n_{\mathrm{p}}, \quad \forall k \in K. \tag{3.9}$$

The different index sets $\mathcal{I}_{\mathrm{V},k}$ in (3.9) contain unique indices, while the $\mathcal{I}_{\mathrm{T},k}$ sets involve recurring indices from the original training index set $\mathcal{I}_{\mathrm{T}}$. The optimal approach for SS-CV is formulated as (Takano and Miyashiro, 2020):

$$\min_{\substack{\boldsymbol{\beta}^{(k)}, \beta_0^{(k)}, \forall k \in K \\ \boldsymbol{z}_{\mathrm{ss}} \in \{0,1\}^{n_{\mathrm{p}}}}} \frac{1}{2} \sum_{k=1}^{K} \| \boldsymbol{y}(\mathcal{I}_{\mathrm{V},k}) - \boldsymbol{X}(\mathcal{I}_{\mathrm{V},k}) \boldsymbol{\beta}^{(k)} - \beta_0^{(k)} \|_2^2, \tag{3.10a}$$

$$\text{s.t.} \quad \forall k \in K: \ \{\boldsymbol{\beta}^{(k)}, \beta_0^{(k)}\} \in \arg\min_{\tilde{\boldsymbol{\beta}}, \tilde{\beta}_0} \frac{1}{2} \| \boldsymbol{y}(\mathcal{I}_{\mathrm{T},k}) - \boldsymbol{X}(\mathcal{I}_{\mathrm{T},k}) \tilde{\boldsymbol{\beta}} - \tilde{\beta}_0 \|_2^2, \tag{3.10b}$$

$$\text{s.t.} \ -\bar{\beta} z_{\mathrm{ss},j} \leq \tilde{\boldsymbol{\beta}} \leq \bar{\beta} z_{\mathrm{ss},j}, \ \forall j \in \{1, \ldots, n_{\mathrm{p}}\}. \tag{3.10c}$$

The problem (3.10) can be solved for various values of K, taking into account constraints on parameter identifiability, such as the cardinality condition in (3.9). Additionally, different randomly generated data distributions into $\mathcal{I}_{\mathrm{T},k}$ and $\mathcal{I}_{\mathrm{V},k}$ sets can be considered. The resulting structure of the soft sensor is determined by the most frequently selected inputs from the calculated sensors. Once the optimal sensor structure is obtained, a least-squares fitting of the model is performed using the entire training dataset to determine the parameters of the designed soft sensor. Similar to problem (3.6), problem (3.10) can be effectively solved using standard MIQP solvers.

3.4 Clustering Algorithms

Clustering is one of the fundamental tasks in machine learning that involves organising the data into meaningful clusters (see Section 2.3.3) with respect to various criteria (see Section 2.3.4). The following section represents the well-known and frequently used clustering algorithms.

3.4.1 *k*-means Clustering

k-means clustering (Forgy, 1965) is an unsupervised learning approach used for data clustering and segmentation. The goal is to assign each data point to one of the predefined classes or sets $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{n_{\mathrm{cl}}}\}$ based on similarity or proximity. The procedure for k-means clustering includes the following steps:

1. Randomly select n_{cl} initial centroids ($n_{\mathrm{cl}} \leq n$).
2. Assign each data point to the closest centroid using a chosen distance metric (e.g., squared Euclidean distance, absolute differences).
3. Update the centroids by calculating the mean of the data points assigned to each cluster.
4. Repeat steps 2 and 3 until the convergence condition is satisfied.

The second step of the aforementioned procedure of the k-means clustering assigns the data point to the closest centroid according to the selected distance metric (d_{CL}). The most popular one is squared Euclidean, as follows:

$$\begin{aligned} d_{\mathrm{CL},i} = (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)^{\intercal}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j), \quad \boldsymbol{x}_i \in \mathcal{C}_j, \\ \forall i \in \{1, 2, \ldots, n\},\ j \in \{1, 2, \ldots, n_{\mathrm{cl}}\}, \end{aligned} \tag{3.11}$$

where $d_{\mathrm{CL},i}$ is i^{th} distance within the vector of squared Euclidean distances $\boldsymbol{d}_{\mathrm{CL}}$ ($\boldsymbol{d}_{\mathrm{CL}} \in \mathbb{R}^n$) evaluated for each considered data point, and $\bar{\boldsymbol{x}}_j$ represents the mean (centroid) of points within the j^{th} cluster given as follows:

$$\bar{\boldsymbol{x}}_j = \frac{1}{\mathrm{card}(\mathcal{C}_j)} \sum_{\boldsymbol{x} \in \mathcal{C}_j} \boldsymbol{x}. \tag{3.12}$$

The objective of k-means clustering is to minimise the within-cluster sum of squared distances for the set of data points $\mathcal{D} \in \mathbb{R}^{n_{\mathrm{p}}}$ as follows:

$$\min_{\boldsymbol{\mathcal{C}}} \sum_{j=1}^{n_{\mathrm{cl}}} \sum_{\boldsymbol{x} \in \mathcal{C}_j} \|\boldsymbol{x} - \bar{\boldsymbol{x}}_j\|^2, \quad \boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{n_{\mathrm{cl}}}\}, \quad \mathcal{D} = \bigcup_{j=1}^{n_{\mathrm{cl}}} \mathcal{C}_j. \tag{3.13}$$

By iteratively solving equation (3.13), k-means clustering aims to generate a desired number of clusters (n_{cl}) with minimal area, while ensuring that data points within different clusters are as distant from each other as possible. The initialization of k-means clustering involves randomly guessing the initial locations of the cluster centres. Therefore, multiple runs of the algorithm with different initial guesses should be performed to obtain more reliable and statistically supported results.

k-means clustering can be effectively used for data treatment. It is observed that clusters primarily composed of outlier measurements and systematic errors contain a smaller amount of data compared to other clusters.

The selection of the desired number of clusters (n_{cl}) is closely related to the data quality. In straightforward cases, the value of n_{cl} can be determined through visual inspection, where data points form visible groups representing different operating conditions of the unit. In more complex cases, the value of n_{cl} can be determined using methods such as the elbow method or various goodness-of-fit criteria (Kodinariya and Makwana, 2013).

3.4.2 Density-based Spatial Clustering of Applications with Noise

The density-based spatial clustering of applications with noise (DBSCAN) algorithm is a clustering approach that can find clusters of arbitrary shapes and sizes, unlike traditional clustering algorithms that assume spherical clusters of similar sizes.

The DBSCAN procedure involves the following steps:

1. Initialization with an arbitrary data point.
2. Expansion of the cluster by connecting neighbouring core points.
3. Repeat the clustering process with unvisited data points until no more points can be added to the cluster.
4. Repeat the previous steps until only noise points remain.

DBSCAN is advantageous because it does not require the number of clusters to be specified beforehand (compared to k-means clustering) and can handle datasets with varying densities. It is particularly effective for identifying clusters in complex and noisy datasets, where traditional methods may struggle.

3.4.3 Minimum Covariance Determinant

The minimum covariance determinant (MCD) method (Rousseeuw, 1984) is one of the first tools for the outliers detection with high robustness. The distance metric used in MCD (d_{MCD}) is represented by the so-called Mahalanobis distance (d_{M}) given by (2.22). As it is shown in Section 2.2.3, d_{M} is closely related to Hotteling's T^2 distance d_{T^2}. Despite the similarity of the distance metrics of these methods, the principle of MCD is different from Hotteling's T^2 method. MCD looks for the subset of measurements with the minimum determinant of the corresponding covariance matrix. In other words, the resulting subset of measurements should occupy the smallest volume possible (determinant of the covariance matrix). The algorithm can be viewed as an enhancement of Hotteling's T^2 distance method.

The iterative algorithm of MCD starts with a random guess of the initial subset. Subsequently, the sample mean vector $\bar{\boldsymbol{x}}$ and sample covariance matrix $\boldsymbol{\Sigma}$ of the initial subset are calculated. According to the calculated $\bar{\boldsymbol{x}}$ and $\boldsymbol{\Sigma}$, it is possible to evaluate $\boldsymbol{d}_{\mathrm{MCD}}$ from (2.22) for each measurement (not only for the selected subset). Subsequently, the new subset of h measurements with the smallest distances $\boldsymbol{d}_{\mathrm{MCD}}$ is selected from the whole set. If the covariance determinant of the new subset is decreased compared to the covariance determinant of the previous subset, the new subset is used in the next iteration of the MCD algorithm. Otherwise, the sought subset has been found (as the previously selected subset) and the MCD algorithm is terminated. The tuning parameter of this scheme is represented by the least number of the retained measurements h from the treated dataset. This parameter is usually adjusted according to the interval $\frac{n+n_{\mathrm{P}}+1}{2} \leq h \leq n$ (Hubert and Debruyne, 2010) or it can be adjusted by the user e.g., based on the visual inspection of the time series of some crucial variables.

Due to the random character of the MCD method, it is desired to perform several runs with different initial guesses to avoid local minima. According to the results from different runs of the MCD method, it is possible to derive a final subset. The vector of distances $\boldsymbol{d}_{\mathrm{MCD}}$ is evaluated in each iteration. The measurements with the smallest distances create a new subset for the next iteration of MCD. This process is terminated when the determinant of the covariance matrix does not decrease anymore.

The sample mean vector $\bar{\boldsymbol{x}}$ and the sample covariance matrix $\boldsymbol{\Sigma}$ of the final subset are subsequently used to evaluate $\boldsymbol{d}_{\mathrm{MCD}}$ from (2.22) for the whole set. According to the values of $\boldsymbol{d}_{\mathrm{MCD}}$, it is possible to determine the most deviated measurements (outliers) from the centre. The condition to separate admissible and inadmissible measurements by MCD is established by appropriate distribution (Hardin and Rocke, 2005) considering the desired confidence level.

3.5 Classification Algorithms

Classification aims to model a function that maps a dataset to specific classes. There are numerous classification algorithms available, and the choice of the appropriate approach should be based on the specific needs and characteristics of the application. In the following sections, two well-known approaches, support vector machine (SVM) and logistic regression, will be described. In this thesis, the SVM approach is utilised in the design of linear soft sensors because it directly designs a linear separation hyperplane.

3.5.1 Support Vector Machine

Support vector machines (SVM) (Boser et al., 1992) belong to the supervised learning family. SVM searches for a linear hyperplane (classifier) that separates data into different classes. The hyperplane can be represented as:

$$\boldsymbol{x}^\intercal \boldsymbol{w} + w_0 = 0, \tag{3.14}$$

where $\boldsymbol{w}$ is the vector of model parameters and w_0 is the constant (bias) term of the hyperplane.

The assignment of each data point to a class is determined by the sign of the expression $\boldsymbol{w}^\intercal \boldsymbol{x} + w_0$. To simplify the mathematics, a binary variable vector $\boldsymbol{z}$ ($\boldsymbol{z} \in \mathbb{R}^n$) is defined as:

$$z_i = \begin{cases} 1, & \text{if } \boldsymbol{x}_i^\intercal \boldsymbol{w} + w_0 > 0, \\ 0, & \text{if } \boldsymbol{x}_i^\intercal \boldsymbol{w} + w_0 < 0, \end{cases} \quad \forall i \in \{1, 2, \ldots, n\}. \tag{3.15}$$

The optimization problem solved by SVM can be formulated as:

$$\min_{\boldsymbol{w}, w_0} \frac{1}{2} \|\boldsymbol{w}\|_2^2, \tag{3.16a}$$

$$\text{s.t.} \quad (2z_i - 1)(\boldsymbol{x}_i^\intercal \boldsymbol{w} + w_0) \geq 1, \quad \forall i \in \{1, 2, \ldots, n\}. \tag{3.16b}$$

By solving the optimisation problem (3.16), SVM attempts to find the best hyperplane that maximises the margin and accurately separates different classes. The objective of minimising $\boldsymbol{w}$ in (3.16) is to ensure that the separation hyperplane is positioned between the two farthest measurements with distinct labels. This process allows SVM to effectively distinguish between different classes by creating a well-defined boundary in the feature space.

The SVM design presented by (3.16) can be extended about the vector of slack variables

$\boldsymbol{e}$ ($\boldsymbol{e} \in \mathbb{R}^n$) as follows:

$$\min_{\boldsymbol{w}, w_0, \boldsymbol{e} \geq 0} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma\|\boldsymbol{e}\|_1, \tag{3.17a}$$

$$\text{s.t.} \quad (2z_i - 1)(\boldsymbol{x}_i^\intercal \boldsymbol{w} + w_0) \geq 1 - e_i, \quad \forall i \in \{1, 2, \ldots, n\}, \tag{3.17b}$$

where γ is weighting (penalty) parameter for slack variables.

By incorporating $\boldsymbol{e}$ in (3.17), it becomes possible to design a linear classification hyperplane even when dealing with a linearly inseparable dataset. To achieve this, it is necessary to penalise $\boldsymbol{e}$ in the objective function (3.17a) by γ. This penalty term encourages the SVM model to minimise errors or misclassification, thus finding a balance between achieving a wider margin and correctly classifying data points that may fall within or near the margin.

3.5.2 Logistic Regression

Logistic regression is statistical approach that aims to model the relationship between the input (independent) variables and the probability of a particular outcome or class. The algorithm uses the logistic function, also known as the sigmoid function. It can be used for predicting binary outcomes or performing multiclass classification by using multiple logistic regression models.

Logistic regression is a computationally efficient algorithm, particularly when dealing with large datasets. It offers a level of interpretability, as the coefficients can be interpreted as the impact of each feature on the predicted probability. Additionally, logistic regression can handle both categorical and continuous features. However, it is sensitive to outliers and multicollinearity among the features, which can affect its performance.

Overall, logistic regression is a widely used algorithm for classification tasks, providing a balance between computational efficiency and interpretability.

Chapter 4

Soft-Sensor Design

The overall procedure for the soft-sensor design consists of seven consequent steps (Kadlec et al., 2009; Khatibisepehr et al., 2013; Botha and Craig, 2021; Mojto et al., 2021). These steps are as follows:

1. Initial data inspection: This step involves gaining an overview of the data structure and identifying any obvious problems, such as constant variables, NaN values, missing data, and linear dependencies.
2. Selection of historical data: Here, the data to be used for the soft-sensor design is selected.
3. Data pre-processing: In this step, the data is transformed to make it suitable for processing by the model. This may involve incorporating nonlinear transformations, and data treatment.
4. Model and input structure selection: The optimal input structure (i.e., feature selection) and type of model for the soft sensor are determined.
5. Model training: The model parameters are determined by minimising a specific criterion (see Section 2.3.2).
6. Model validation: The performance of the designed soft sensor is evaluated on a testing dataset that was not used during training.
7. Soft-sensor maintenance: The model parameters are retrained regularly to account for drifts and other changes in the data.

The procedure of the soft-sensor design is visualised in Figure 4.1. In the following sections, the selected parts of this procedure will be described in detail, focusing on their relevance to the objectives of the thesis.

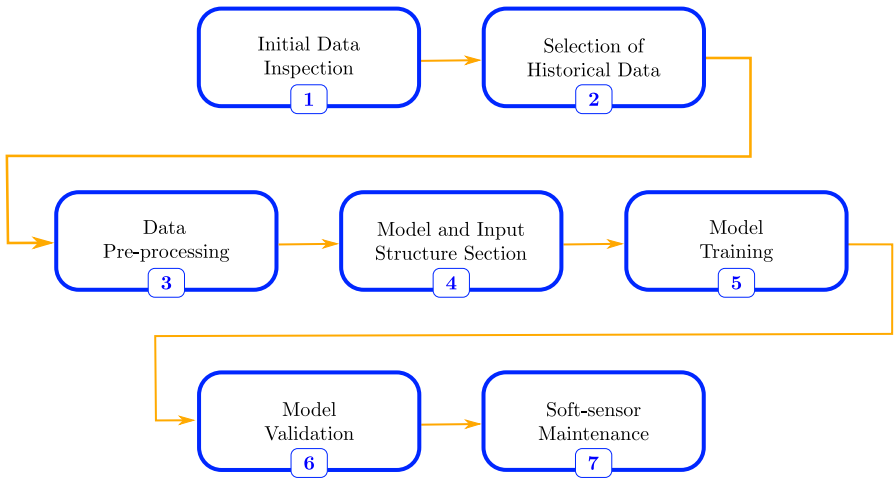


Figure 4.1: The procedure of the soft-sensor design.

4.1 Initial Data Inspection

The main purpose of this stage in soft-sensor development is to gain an overview of the data structure. It is crucial to address any obvious problems that could hinder further analysis if left unattended. These problems may include constant variables with no contribution, NaN values within the measurements causing numerical issues, missing data primarily due to sensor failures, and linear dependencies that increase the computational burden for many algorithms. Although these issues typically arise from the input variables, it is important to examine the output variables as well, particularly assessing whether they exhibit sufficient variance for subsequent soft-sensor design.

Another objective of this stage is to assess the requirements for model complexity, such as determining whether a linear or nonlinear model structure is necessary based on the demands of the particular application. However, it is recommended to compare the performance of the designed model (based on the selection made at this stage) against other models developed in later stages of the development process. There are cases when the industry directly specify the desired complexity the soft-sensor model structure.

4.2 Selection of Historical Data

At this stage, it is necessary to select data from the entire historical dataset for the subsequent stages of soft-sensor design. The selected data will serve as the basis for the training and testing datasets that will be used later.

Soft-sensor design often involves uneven sampling of input and output variables. The output (desired) variable is typically measured less frequently compared to the input variables. This is mainly due to the increased costs associated with measuring the output variable, which is precisely the motivation behind soft-sensor design. Therefore, in this development stage of soft-sensor design, it is necessary (especially for supervised learning approaches) to perform time pairing of the input and output variables.

In most cases, this stage involves identifying and selecting the stationary parts of the dataset. The selection of stationary measurements is typically done through manual analysis of the data. Alternatively, an automatic approach could involve filtering and variance analysis of the variables within the dataset. However, there are situations where it may not be desirable or possible to perform a steady-state indication, even when designing a steady-state soft sensor. This can occur when the available data size is small, which is often the case when designing a soft sensor for a hard-to-measure variable.

4.3 Data Pre-processing

The design of the soft sensor is significantly influenced by the suitability of the available dataset. To ensure the desired dataset properties, it is necessary to properly preprocess the data before further analysis. The data preprocessing phase can vary depending on the specific application objectives. For this thesis, the data preprocessing involves the following analyses: (a) identification of linear correlations; (b) incorporation of nonlinear transformations; and (c) data treatment.

These analyses form the core of the data preprocessing stage. Initially, several basic transformations of the dataset may be applied. One such transformation is data centring, which can be performed as follows:

$$\boldsymbol{X}_{\text{cen}} = \boldsymbol{X} - (\mathbb{1}\bar{\boldsymbol{x}}^{\intercal})^{\intercal}, \quad \boldsymbol{y}_{\text{cen}} = \boldsymbol{y} - \bar{y}, \tag{4.1}$$

where $\boldsymbol{X}_{\text{cen}}$ ($\boldsymbol{X}_{\text{cen}} \in \mathbb{R}^{n \times n_{\text{P}}}$) represents a matrix of centred input variables, $\bar{\boldsymbol{x}}$ ($\bar{\boldsymbol{x}} \in \mathbb{R}^{n_{\text{P}}}$) is sample mean vector, $\boldsymbol{y}_{\text{cen}}$ ($\boldsymbol{y}_{\text{cen}} \in \mathbb{R}^{n}$) represents a vector of centred output variable, and $\bar{y}$ is a sample mean value of the output variable y.

Centring the data makes it less susceptible to variations in the scales of the input and output variables compared to the original dataset. Additionally, if the dataset is centred around zero, the linear models (relevant to the thesis) can be simplified by omitting the constant (bias) term β_0 in (2.2). Furthermore, the centred dataset can be standardised as follows:

$$\boldsymbol{x}_{\mathrm{std}\{i\}} = \frac{\boldsymbol{x}_{\mathrm{cen}\{i\}}}{s_{\boldsymbol{x}\{i\}}}, \quad \boldsymbol{y}_{\mathrm{std}} = \frac{\boldsymbol{y}_{\mathrm{cen}}}{s_{\boldsymbol{y}}}, \quad \forall i \in \{1, 2, \ldots, n_{\mathrm{p}}\}, \tag{4.2}$$

where $\boldsymbol{x}_{\mathrm{std}\{i\}}$ ($\boldsymbol{x}_{\mathrm{std}\{i\}} \in \mathbb{R}^n$) represents vector of i^{th} standardized input variable, and $\boldsymbol{y}_{\mathrm{std}}$ ($\boldsymbol{y}_{\mathrm{std}} \in \mathbb{R}^n$) represents a vector of standardized output variable.

The standardised dataset has a mean of zero and a standard deviation of one, making it even less affected by variations in scales and fluctuations of the input and output variables compared to the centred dataset. Moreover, the covariance and correlation matrices of the standardised dataset are the same, eliminating the need for further re-calculations in-between these matrix forms. An alternative transformation to standardisation is normalisation, which can be expressed as follows:

$$\begin{aligned} \boldsymbol{x}_{\mathrm{nor}\{i\}} &= \frac{\boldsymbol{x}_{\{i\}} - \min\left(\boldsymbol{x}_{\{i\}}\right)}{\max\left(\boldsymbol{x}_{\{i\}}\right) - \min\left(\boldsymbol{x}_{\{i\}}\right)}, \quad \forall i \in \{1, 2, \ldots, n_{\mathrm{p}}\}, \\ \boldsymbol{y}_{\mathrm{nor}} &= \frac{\boldsymbol{y} - \min(\boldsymbol{y})}{\max(\boldsymbol{y}) - \min(\boldsymbol{y})}, \end{aligned} \tag{4.3}$$

where $\boldsymbol{x}_{\mathrm{nor}\{i\}}$ ($\boldsymbol{x}_{\mathrm{nor}\{i\}} \in \mathbb{R}^n$) represents vector of i^{th} normalized input variable, and $\boldsymbol{y}_{\mathrm{nor}}$ ($\boldsymbol{y}_{\mathrm{nor}} \in \mathbb{R}^n$) represents a vector of normalised output variable measurements.

By using (4.3), the resulting normalised input and output variables are within the interval $[0, 1]$. Normalisation ensures additional properties, such as the non-negativity of measurements, without losing the information content within the dataset. The choice of an appropriate scaler for normalisation should be based on the specific application requirements.

The data preprocessing phase in data-driven soft sensor design typically involves dividing the available dataset into the training dataset ($\mathcal{I}_{\mathrm{T}}$) and testing dataset ($\mathcal{I}_{\mathrm{S}}$). This division helps mitigate the risk of model overfitting, thereby enhancing the robustness and reliability of the soft sensors (see Section 2.1.1).

4.3.1 Identification of Linear Correlations

The effectiveness and sustainability of the linear data-driven soft sensor are greatly influenced by the selection of appropriate input variables. To identify the optimal set

of input variables, it is important to address and eliminate highly correlated variables. Multivariate datasets containing linearly correlated variables can lead to various issues, including increased computational burden, overfitting, incorrect model parameters, and inappropriate input structure.

There are several approaches to identifying linear correlations in multivariate datasets. One straightforward method is as follows:

1. Calculate the correlation matrix ($\boldsymbol{\Sigma}$) of the input dataset $\boldsymbol{X}$ using (2.18).
2. Identify pairs of input variables with a linear correlation exceeding a certain criterion (crt) for correlation coefficients (e.g., $|\Sigma_{i,j}| > \text{crt}, \text{crt} \in [0, 1]$), considering the off-diagonal elements of $\boldsymbol{\Sigma}$.
3. Remove redundant input variables based on their relevance to the output variable or expert knowledge.

4.3.2 Incorporation of Nonlinear Transformations

The original (industrial) input dataset comprises input variables obtained directly from online sensors. However, the information content provided by these variables, which can explain the output variable behaviour, is often limited by noise and external disturbances. To enhance the information content of the input dataset, it is possible to expand it by incorporating appropriate nonlinear transformations of the original input variables. The selected nonlinear transformations subsequently represent the indirectly measured input variables for the soft-sensor design.

Incorporating effective nonlinear transformations is particularly relevant for linear soft-sensor design, as they enable the linear models to account for the nonlinear behaviour of the output variable, which is strongly present in the industry. However, when extending the input dataset with additional variables (degrees of freedom), there is a higher risk of overfitting, especially in the case of nonlinear soft sensors.

Identifying effective nonlinear transformations can be achieved through brute force analysis, where the impact of a selected set of structures (e.g., logarithms, exponentials, powers, power roots, and inverses) on the output variable is sequentially evaluated. Another approach to exploring nonlinear transformations is through the feature selection methods described in Section 4.4.2. Additionally, an advanced method for identifying effective nonlinear transformations is automated learning of algebraic models (ALAMO) (Wilson and Sahinidis, 2017). ALAMO is specifically designed to

identify linear combinations of considered nonlinear transformations that enable a linear model to better approximate the complex behaviour observed in real processes.

4.3.3 Data Treatment

Industrial data often contains both systematic and random errors (Su et al., 2009). Systematic errors can arise from non-standard or infrequent situations in the industrial unit, which may be expected (e.g., maintenance) or unexpected (shutdown or plant tripping). Sensor failures and inaccuracies can also contribute to systematic errors.

The detection of certain types of systematic errors can be done through visual inspection of time series plots (Alves and Nascimento, 2007). If a consistent interval of significantly deviated measurements is observed across all variables, it indicates a potential source of systematic errors that needs to be addressed before designing the soft sensor. This type of error is relatively easy to detect with the naked eye.

However, there are situations where the sudden failure or malfunction of one or more online sensors can introduce systematic errors that are difficult or impossible to identify through visual inspection. In such cases, several multivariate data treatment methods can be employed to reduce the number of systematic errors remaining after visual detection. The representative of these approaches are as follows:

- T^2 distance (see Section 2.2.3) represents the metric of each measurement or data point from the centre ($\boldsymbol{\mu}_x = \mathbf{0}$) of studied dataset. High values of T^2 distance indicate measurements that deviate significantly from the centre, making them likely outliers. The T^2 distance considers the covariance matrix $\boldsymbol{\Sigma}$ of the treated dataset, allowing it to handle multivariate datasets. The covariance matrix $\boldsymbol{\Sigma}$ remains constant throughout the data treatment analysis.

- k-means clustering (see Section 3.4.1) can be applied for the data treatment analysis. This method involves clustering the measurements into n_{cl} clusters, aiming to include all measurements from the dataset. Outliers can be identified by examining the clusters with the smallest sizes. The initialization of k-means clustering involves randomly selecting the n_{cl} cluster centres, so it is recommended to perform multiple runs and average the results for increased credibility.

- DBSCAN (see Section 3.4.2) is another suitable algorithm for outlier detection. Unlike k-means clustering, DBSCAN does not require a predetermined number of clusters. Instead, it operates based on the neighbourhood of each data point.

Points that do not belong to any cluster are considered outliers. DBSCAN is particularly useful for noisy datasets.

- MCD (see Section 3.4.3) can also be employed for data treatment analysis. Similar to the T^2 distance, MCD considers a single centre for the dataset and computes the covariance matrix $\mathbf{\Sigma}$. However, the advantage of MCD is its iterative approach, aiming to minimise the covariance matrix $\mathbf{\Sigma}$ and find the subset of the predefined size h that has the smallest possible volume.

Considering the nature of these data treatment methods, one can expect the performance of MCD to be at least as good as that of the T^2 distance method. The performance of k-means clustering and DBSCAN can surpass that of the other methods if the measurements exhibit clear distinctions between operating points (steady states) in a specific unit.

4.4 Model and Input Structure Selection

This phase of the soft-sensor design is critical as it determines the input structure and model of the soft sensor, which form the core of the design. In this research, the emphasis is on the selection of the input structure (or feature selection) for linear data-driven soft-sensor design, with relatively less focus on model selection. However, we will provide a brief description of the model selection before delving into the details of the input structure selection.

4.4.1 Model Selection

The model selection process for soft-sensor design is often influenced by the past experiences of the developer, which can compromise the quality of the solution for a specific application. Relying on personal preferences for selecting the soft sensor model structure is not recommended, and a more sophisticated approach should be adopted.

An intuitive approach to model selection is to start with the simplest possible model and gradually increase its complexity until a significant improvement in performance is observed. However, it is important to evaluate the performance of the model on a testing dataset (unseen data) to avoid overfitting. Overfitting occurs when the model becomes too complex and starts to fit the noise in the data.

In certain situations, especially in industrial settings, it can be challenging to obtain a sufficient amount of historical data, often due to the expensive measurement of the

output variable. In such cases, the cross-validation approach (see Section 3.3.1) can be employed to effectively utilise the available data by creating validation datasets $(\boldsymbol{X}(\mathcal{I}_{\mathrm{V}}), \boldsymbol{y}(\mathcal{I}_{\mathrm{V}}))$.

4.4.2 Input Structure Selection

The industrial dataset may consist of hundreds of online measured variables (inputs) that can be used in the soft-sensor structure. Handling such a large dataset can impose a significant computational burden on the soft-sensor design, particularly during the model training stage. Therefore, the purpose of input structure selection is to determine which input variables should be included in further analysis. This step is also known as feature selection. The main focus of input structure selection is to assess the correlation between the input variables and the output variables.

As mentioned earlier, the task is to select a subset of input variables (n_{p}^*) from the available dataset (n_{p}) in order to reduce the dimensionality (complexity) of the problem being solved ($n_{\mathrm{p}}^* \leq n_{\mathrm{p}}$). This task involves monitoring criteria to ensure that the selected subset retains sufficient information compared to the original dataset. Another criterion is to search for the subset of input variables with the highest potential to explain the desired variable. These criteria are considered in the approaches suitable for input structure selection presented later.

Before applying any approach, initial structure selection should be based on expert knowledge from the industry. Engineers and operators are usually able to narrow down the entire dataset to a smaller set ("shortlist") of input variables that they believe are related to the output variable. This step in the input structure selection process can help address issues related to the linear correlations between input variables.

The fundamental approach for input structure selection is OLS regression (see Section 3.1). However, this approach primarily focuses on minimising the SSE (model accuracy) and can eliminate inputs that are not correlated with the output variable. OLS, however, does not effectively balance model accuracy and complexity.

More advanced methods for input structure selection include PCA (see Section 3.2.1) and PLS (see Section 3.2.2). Both methods are based on variance-covariance analysis. The main difference is that PCA is an unsupervised learning approach, while PLS is supervised. PCA is preferable when a large input dataset is available or when the measurements of the output variable are unreliable. On the other hand, PLS is suggested when the output variable is accurately measured. Both methods consider principal components sorted in decreasing order of explained variance and aim to

select a subset of these components without significant loss of explained variance. Various approaches can be used for selection, such as the elbow method, specifying a desired amount of explained variance, or selecting only the first principal component. It is important to note that when considering the principal components within the soft-sensor structure, all original variables are required to evaluate each principal component. Therefore, the signals from all online sensors involved are required, even if only one principal component is considered.

Another approach to input structure selection is sparsity-based methods, presented in Section 3.3. The LASSO approach (see Section 3.3.2) balances between model accuracy and complexity using the weighting parameter λ. This parameter can be tuned using model-overfitting criteria presented in Section 2.3.2 or through cross-validation (see Section 3.3.1). Additionally, the ℓ_1-norm penalization element within the objective function of LASSO can be replaced by the ℓ_2-norm in ridge regression. The elastic net approach provides a way to balance between the ℓ_1-norm and ℓ_2-norm.

The group of sparsity-based methods suitable for input structure selection includes the family of subset selection approaches (see Section 3.3.3). The ordinary subset selection, as represented by equation (3.6), aims to find the input structure with the highest accuracy. However, this approach can potentially lead to model overfitting, especially when dealing with non-ideal industrial datasets. Therefore, it is recommended to combine subset selection with a model overfitting criterion (see Section 3.3.3), which takes into account criteria such as AICc, BIC, or R^2_{adj} in the upper-level objective defined in equation (3.6a). In cases where the dataset size is small, subset selection can be combined with cross-validation (see Section 3.3.3). Cross-validation allows for a more effective use of the training dataset compared to other subset selection approaches, as it considers validation subsets extracted from the training dataset.

4.5 Model Training

This stage of the soft-sensor design is dedicated to finding the optimal values of the model parameters based on specific objectives. These objectives are specific to each approach utilised in this stage of the soft-sensor design. The dataset used for model training is referred to as the training dataset.

In this thesis, soft-sensor training is divided into two categories based on the number of trained models. The first category focuses on the design of single-model soft sensors (SMS). This type of soft-sensor design is well-established and commonly used in the industry to estimate key process variables. The second category involves the design of

multi-model soft sensors (MMS). This type of soft-sensor design is suitable for processes with multiple operating regimes or when a linear soft-sensor is needed to estimate a nonlinear process variable. By considering these two categories, we can explore different approaches to soft-sensor design and provide insights into their application in various scenarios.

4.5.1 Single-model Soft Sensor

The single-model soft sensor (SMS) is a standard and frequently used sensing technique in the industry for estimating or inferring key or hard-to-measure variables using measurements from other related easy-to-measure variables (Khatibisepehr et al., 2013; Doraiswami and Cheded, 2014; de Morais et al., 2019; Botha and Craig, 2021). As the name implies, this group of soft sensors considers only one model within the soft-sensor structure.

In this thesis, the objective of model training is to determine the optimal values of the model parameters ($\boldsymbol{\beta}^*$ and β^*) for the linear multivariable soft-sensor structure represented by Equation (2.6). The desired model parameters are determined based on the chosen training approach. The representative approaches for training SMS are as follows:

- OLS (or SMS_{OLS}): The basic design of SMS involves using OLS regression (see Section 3.1) to train the model parameters. The trade-off for this relatively simple model training approach is the potential for model overfitting.
- PCA (or SMS_{PCA}): The SMS design alone cannot be performed using the PCA approach (see Section 3.2.1) due to its unsupervised nature. However, PCA can be effectively combined with another approach that utilises the principal components to fit the output measurements. The most common combination is PCA with OLS, resulting in PCA regression.
- PLS (or SMS_{PLS}): Another approach suitable (and frequently used) for SMS training is PLS (see Section 3.2.2). This approach is based on the same principle of variance-covariance analysis as PCA but considers the output measurements during model training. Therefore, it is a supervised learning-based approach, unlike PCA.
- LASSO (or SMS_{LAS}): The SMS training can also be provided by the LASSO approach (see Section 3.3.2). It balances between model accuracy and complexity through the λ parameter. The complexity of the model structure is penalised by the ℓ_1-norm.

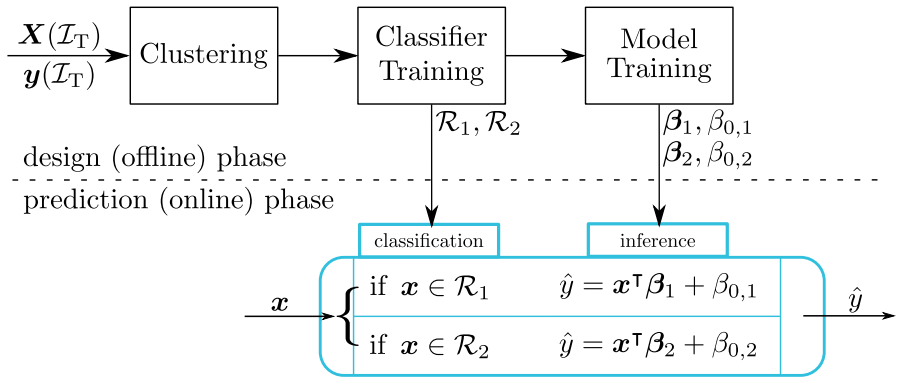


Figure 4.2: The design and prediction phases of MMS.

- SS (or $\mathrm{SMS}_{\mathrm{SS}}$): The family of SS approaches (see Section 3.3.3) can also be utilised in the SMS design. Due to the flexible structure of the ordinary SS optimisation problem (3.6), it is possible to consider various modifications of the standard SS approach as well. In the case of noisy data, it is possible to combine SS with a model overfitting criterion (see Section 3.3.3). Moreover, SS can effectively work in combination with cross-validation as well. This can be considered in the case of a dataset with a small size.

4.5.2 Multi-model Soft Sensor

Prediction capability of a linear soft sensor can be improved when considering a multi-model soft-sensor (MMS) structure. The MMS formulation with two models can be written as follows (Mojto et al., 2022):

$$\hat{y}_i = \begin{cases} \boldsymbol{x}_i^\intercal \boldsymbol{\beta}_1 + \beta_{0,1}, & \text{if } \boldsymbol{x}_i \in \mathcal{R}_1, \\ \boldsymbol{x}_i^\intercal \boldsymbol{\beta}_2 + \beta_{0,2}, & \text{if } \boldsymbol{x}_i \in \mathcal{R}_2, \end{cases} \quad \forall i \in \{1, 2, \ldots, n_\mathrm{T}\}, \tag{4.4}$$

where regions of individual model validity denoted as $\mathcal{R}$ represent convex polyhedra such that $\mathcal{R}_1 \bigcap \mathcal{R}_2 = \emptyset$. Consideration of more than two models is possible in a similar setup.

The design (offline) and prediction (online) phases of MMS are illustrated in Figure 4.2. The design phase begins with the clustering of the available training dataset ($\boldsymbol{X}(\mathcal{I}_\mathrm{T})$, $\boldsymbol{y}(\mathcal{I}_\mathrm{T})$). The training dataset, along with the assigned data labels, is then used for classifier design. The designed classifier defines different regions ($\mathcal{R}_1$, $\mathcal{R}_2$) that can be considered for the designed models. In the prediction phase, these regions

are utilised for classification. Simultaneously or after classifier design, the MMS models are constructed in the model training block (refer to Figure 4.2). The model parameters determined during this process are subsequently employed for inference in the prediction phase. Once the inference is complete, the MMS provides predictions, and the estimated output variable $\hat{y}$ becomes available.

State-of-the-Art Approach The workflow of the state-of-the-art approach for MMS design consists of the following three steps:

1. *A priori labelling of the training dataset:* The labelling is based on the characteristics of the dataset, such as the distinction between operating points. An appropriate clustering approach, such as k-means clustering, can be used.

2. *Classifier design based on the labelled training dataset:* The classifier determines the corresponding model class of a measurement point. In this paper, we consider the support vector machines (SVM) approach using linear separators (Boser et al., 1992) to describe the model-validity regions.

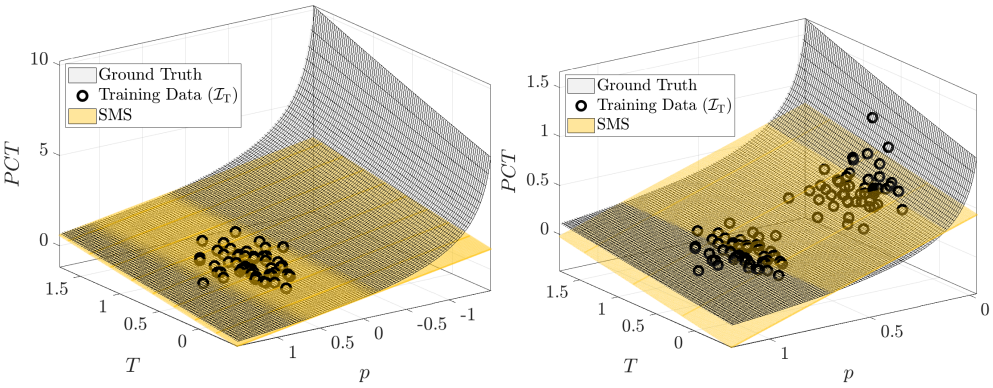
3. *Training of the individual soft-sensor models:* The individual MMS models for each class can be fitted using one of the SMS training methods (see Section 4.5.1).

In the following, we abbreviate the sensor designed by this procedure as $\mathrm{MMS}_{\mathrm{SotA}}$ and refer to it as a state-of-the-art approach, although the presented procedure is our contribution. This is because, to the best of the authors' knowledge, there is no consistent (agreed-upon) technique for MMS design.

Illustrative example. We consider a problem of designing a soft sensor for the pressure compensated temperature (PCT) model to provide the visual comparison of the SMS and MMS designs. The PCT model is frequently used in low-pressure petrochemical distillation columns (Pan et al., 2019). A combination of the Antoine and Clausius-Clapeyron equations forms the following mathematical representation (King, 2011):

$$\frac{1}{PCT} = \frac{R}{H_v} \ln\left(\frac{p}{p_{\mathrm{ref}}}\right) + \frac{1}{T}, \tag{4.5}$$

where PCT is pressure compensated temperature, H_v is the heat of vaporization, R is the universal gas constant, p_{ref} is the reference pressure, p is the absolute pressure, and T is the absolute temperature.



(a) One cluster (RMSE ($\mathcal{I}_T$)=0.028). **(b)** Two clusters (RMSE ($\mathcal{I}_T$)=0.064).

Figure 4.3: The ground truth PCT model with SMS designed on different datasets.

The ground truth model of the PCT is considered with $R = 8.3\,\mathrm{J/mol/K}$, $H_v = 55,940.6\,\mathrm{J/mol}$ and $p_{\mathrm{ref}} = 145.3\,\mathrm{Pa}$ over the operating region:

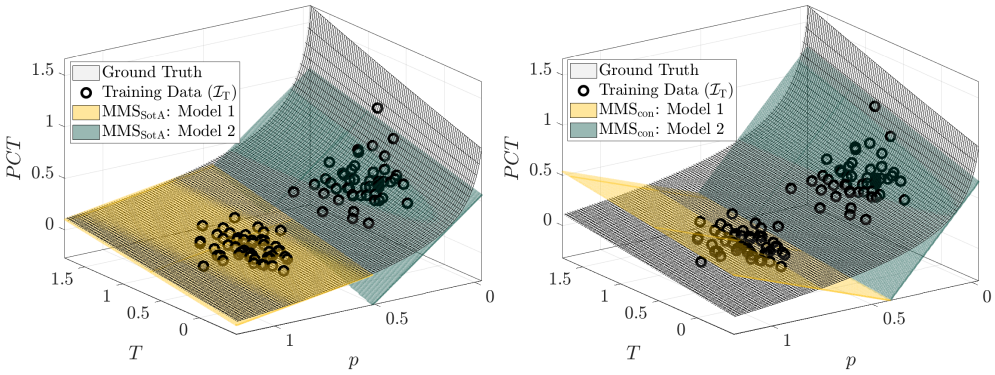
$$\begin{aligned} 523.2\,\mathrm{K} \leq T \leq 573.2\,\mathrm{K}, \\ 0.4\,\mathrm{Pa} \leq p \leq 15\,\mathrm{Pa}, \\ 635.3\,\mathrm{K} \leq PCT \leq 1151.4\,\mathrm{K}. \end{aligned} \tag{4.6}$$

For sensor training and evaluation, all input variables (p, T, PCT) are scaled (normalized) to the interval $[0, 1]$.

The SMS performance is shown in Figure 4.3 on two training datasets, which simulates the process working in one (one data cluster in Figure 4.3a) and two distinct operating regimes (two data clusters in Figure 4.3b), respectively. Noise is added to the input and output data to represent typical industrial datasets. The SMS accuracy, measured by the root mean squared error (RMSE) and tested on fresh data within the training regions, is significantly reduced (more than 2-fold deterioration) when the process runs in two operating regimes. This stems from inappropriateness of a single linear model to describe a nonlinear behaviour of PCT.

The usage of the $\mathrm{MMS}_{\mathrm{SotA}}$ on the PCT dataset is shown in Figure 4.4a. The designed models are presented with a yellow surface (Model 1) and a dark green surface (Model 2). The example considers a priori labelling by k-means clustering.

The advantage of using MMS is obvious, as its accuracy (RMSE ($\mathcal{I}_T$) = 0.032) significantly outperforms the best SMS (Figure 4.3b, RMSE ($\mathcal{I}_T$) = 0.064). This



(a) MMS_{SotA} (RMSE $(\mathcal{I}_{\text{T}}) = 0.032$). **(b)** MMS_{con} (RMSE $(\mathcal{I}_{\text{T}}) = 0.079$).

Figure 4.4: The ground truth model of PCT with MMS_{SotA} and MMS_{con} designed on the dataset with two distinct clusters.

confirms that the MMS models can better explain the nonlinear behaviour of PCT compared to SMS. Furthermore, the structure of MMS is flexible as it can involve more models. There are two primary limitations (challenges) of MMS_{SotA}: (a) the designed models are not necessarily continuous, and (b) a priori labelling is unaware of its impact on the accuracy of the resulting soft sensor. The first drawback can be seen in Figure 4.4a. There is a visible discrepancy between the designed models of MMS_{SotA} at the intersection of the surfaces. This behaviour can cause issues with the stability of the control strategy if the MMS is involved.

A glimpse of the proposed solution in Figure 4.4b (approach MMS_{con} will be introduced in detail in Section 6.2.1) reveals that it is possible to achieve continuous switching between the MMS models, yet potentially, at the expense of model accuracy. In the studied example, the discrepancy between ground truth and the designed models originates from the rotation of Model 1 (the yellow surface in Figure 4.4b) to achieve the desired continuity with Model 2 (the dark green surface). The rotation can be reduced by putting more weight on accuracy and relaxing the continuity constraint when designing a continuous MMS (as discussed in Section 6.2.1) or optimizing a priori labelling (as discussed in Section 6.2.2).

4.6 Model Validation

After the model training phase, it is crucial to assess the performance of the model on a separate testing dataset that was not used for training. This evaluation aims to determine how well the trained models can generalise beyond the training data, which is essential for assessing their robustness and future performance. Various performance indices presented in Section 2.3.2, such as SSE, SAE, RMSE, and R^2, can be utilised to evaluate the performance of the model.

In addition to performance indices, the quality of the model structure can also be assessed through visual examination. One approach is to create a parity plot that compares the measured and estimated values of the output variable. Another useful visualisation is a time series chart that displays the measured and estimated values of the output variable over time. This chart can reveal temporal patterns or effects, such as different behaviours during the day and night. It is important to note that visual inspection requires the input of the model developer, and therefore, the final judgement is subjective and relies on the expertise of the model developer.

4.7 Soft-sensor Maintenance

The final stage of soft-sensor development is maintenance, which should be performed regularly. This is necessary because the data from the process may experience drifts and other changes over time. To ensure the continued performance of the soft sensor, it is important to compensate for these changes by adapting or redeveloping the model. Failure to do so may result in performance deterioration of the soft sensor. Regular maintenance helps keep the soft sensor updated and aligned with evolving process conditions.

Automated mechanisms for soft-sensor maintenance are not widely implemented, and there is generally limited interest in them. As a result, most soft-sensor applications in the industry rely on manual quality control and maintenance, which can be a significant cost factor. Soft-sensor adaptation is often based on the judgement of the model operator and is performed through visual inspection of the deviation between the measured and estimated values of the output variable.

4.7.1 Recursive Estimation

The basic approach for redeveloping the soft-sensor model is known as recursive estimation, also referred to as recursive or online estimation. This approach enables

the estimation of model parameters in real-time as new data becomes available. Unlike batch estimation, which requires all the data to be present for estimation, recursive estimation updates the parameter estimates incrementally as each new data point is observed.

One commonly used method for recursive estimation is the recursive least squares (RLS) method. The RLS method aims to minimise the sum of squared errors between the measured and estimated output variables. It achieves this by assigning weights to the current estimate and the new data, allowing for weighted updates of the parameters.

Recursive estimation finds application in various fields, including control systems, signal processing, and machine learning. It is particularly valuable in situations where data is continuously streaming or evolving, as it enables real-time adjustments and monitoring of system or model parameters.

4.7.2 Bias Correction

The performance of industrial soft sensors can be adjusted during operation through adaptive bias correction, also known as bias update (Quelhas, 2009). Many industrial software solutions offer this form of soft sensor maintenance to account for changes in operating conditions. The purpose of bias correction is to enhance the accuracy of subsequent predictions by adjusting the constant (bias) term β_0 in the soft sensor structure (2.6) using the following equation:

$$\beta_{0,i+1} = \beta_{0,i} + \Delta\beta_{0,i}, \tag{4.7}$$

where i is the measurement index, $\beta_{0,i+1}$ is the adjusted (biased) constant term for the next prediction, and $\Delta\beta_{0,i}$ is the increment of the bias term.

The value of $\Delta\beta_{0,i}$ is calculated using the following equation:

$$\Delta\beta_{0,i} = K_{\mathrm{BC}}(y_i - \boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\beta} - \beta_{0,i}), \tag{4.8}$$

where K_{BC} is the gain or multiplier for the bias correction.

It is recommended to select the value of K_{BC} from the interval $[0, 1]$ (for normalised or standardised datasets), but this can vary depending on the specific application. To prevent deterioration of the β_0 term caused by incorrect or inaccurate measurements of the desired variable in lab analysis, the value of y_i must be within the desired interval $[y_{\min}, y_{\max}]$. Otherwise, the value of $\Delta\beta_{0,i}$ is set to zero.

There are additional constraints on $\Delta\beta_{0,i}$, as shown in equations (4.9):

$$\Delta\beta_{0,i} = 0, \qquad \text{if} \quad |\Delta\beta_{0,i}| < \Delta\beta_{0,\min}, \tag{4.9a}$$

$$\Delta\beta_{0,i} = \operatorname{sgn}(\Delta\beta_{0,i})\Delta\beta_{0,\max}, \qquad \text{if} \quad |\Delta\beta_{0,i}| > \Delta\beta_{0,\max}, \tag{4.9b}$$

where $\Delta\beta_{0,\min}$ represents the minimum accessible increment (bias increment dead-band) and $\Delta\beta_{0,\max}$ represents the maximum accessible increment.

The complete procedure of bias correction for i^{th} measurement, including the individual steps, is outlined in Algorithm 1. The inputs are three variables (i.e., y_i, $\boldsymbol{x}_i$, and $\beta_{0,i}$) and six parameters (i.e., $\boldsymbol{\beta}$, $y_{\min}$, $y_{\max}$, $\Delta\beta_{0,\min}$, $\Delta\beta_{0,\max}$, and K_{BC}). The output is biased constant term $\beta_{0,i+1}$ for the proceeding estimate.

Algorithm 1 The algorithm of the bias correction for i^{th} measurement

Input: y_i, $\boldsymbol{x}_i$, $\beta_{0,i}$, $\boldsymbol{\beta}$, $y_{\min}$, $y_{\max}$, $\Delta\beta_{0,\min}$, $\Delta\beta_{0,\max}$, K_{BC}
Output: $\beta_{0,i+1}$

```
Δβ_{0,i} = 0
if y_i ∈ [y_min, y_max] then
    Δβ_{0,i} = K_BC(y_i − x_i^T β − β_{0,i})
    if |Δβ_{0,i}| < Δβ_{0,min} then
        Δβ_{0,i} = 0
    end if
    if |Δβ_{0,i}| > Δβ_{0,max} then
        Δβ_{0,i} = sgn(Δβ_{0,i}) Δβ_{0,max}
    end if
end if
β_{0,i+1} = β_{0,i} + Δβ_{0,i}
```

The performance of bias correction can be evaluated using two criteria: the accuracy of the biased estimates from the soft sensor (RMSE_{BC}) and the effort or frequency of bias correction (E_{BC}). The accuracy of the biased estimates represents the effectiveness of combining a specific soft sensor with bias correction. While RMSE is commonly used for evaluating accuracy, other performance criteria such as SSE, SAE, and R^2 (see Section 2.3.2) can also be applied. It is desirable to increase the accuracy of the soft sensor through bias correction, but this improvement should be considered in relation to the credibility or reliability of the measured output variable. The effort or frequency of bias correction, E_{BC}, is determined as the percentage of cases where condition (4.9a) is not satisfied out of all the cases. In other words, E_{BC} quantifies the amount of effort required for bias correction within the analysed dataset. The effort of bias correction serves as a measure of how frequently the plant operating conditions

change and the ability of the sensor to adapt to these changes. The plant operators tend to prefer an soft sensor with less frequent bias updates.

Part II

Contributions

Chapter 5

Data-driven Design of Soft Sensors for Petrochemical Industry

The main focus of this contribution is to design soft sensors using industrial datasets, incorporating different approaches for multivariate data treatment and model training. The findings and outcomes of this research have been previously published in (Mojto et al., 2021).

5.1 Problem Definition

The objective of this study is to design linear multivariate soft sensors, as defined by Equation (2.6). These sensors are specifically tailored for real-world case studies. To achieve this, data from two industrial distillation columns at the oil refinery Slovnaft, a.s. in Bratislava, Slovakia, has been collected and used for analysis. The first column under consideration is the depropanizer column, which is part of the Fluid Catalytic Cracking (FCC) unit. The second column is the product fractionator located in the Vacuum Gasoil Hydrogenation (VGH) unit. Detailed specifications of the case studies are presented in the subsequent sections.

5.1.1 FCC unit

This unit serves to convert heavy hydrocarbon fractions (vacuum distillates) of the crude oil incoming from the entire refinery to more valuable products, such as gasoline or olefins. The FCC unit is separated into several individual sections (sub-units). One of these sub-units includes several interconnected distillation columns (e.g., debutanizer or depropanizer) to process light hydrocarbons C2–C6. The measured (observed) output variable (y) to be inferred by the soft sensor is the composition (main impurity) of the bottom product x_{B} of the depropanizer column shown in Figure 5.1.

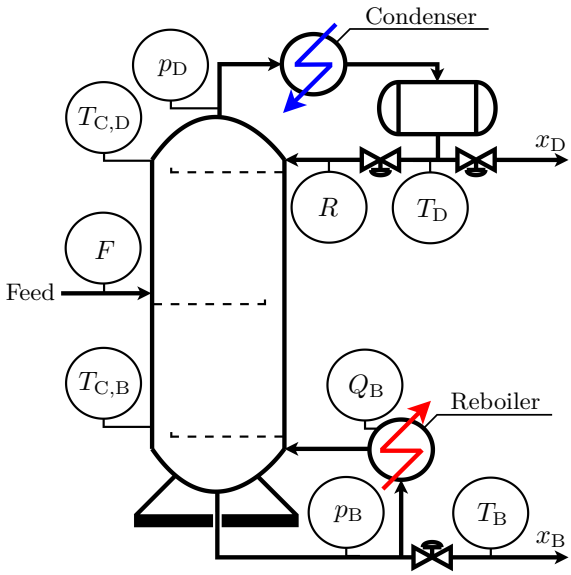


Figure 5.1: A schematic diagram of the depropanizer column.

The studied depropanizer column processes the feed mixture of nine hydrocarbons C3–C5. The purpose of this column is to separate the feed into C3-fraction-rich distillate product x_D and to C4/C5-fraction-rich bottom product x_B. The available operational degrees of freedom are feed flowrate F, bottom product flowrate B, distillate flowrate D, reflux flowrate R, heat duty in the reboiler Q_B, and heat duty in the condenser Q_D. Most of these variables are available as historical data. These are marked correspondingly in Figure 5.1. The plant measurements, also available from historical data, are pressure at the top of the column p_D, pressure at the bottom of the column p_B, and temperatures of distillate T_D, of bottoms T_B, at the top of the distillation column $T_{C,D}$ and at the bottom of the distillation column $T_{C,B}$. The vector of eleven available input variables is given as:

$$\boldsymbol{x} = \left(F, R, Q_B, p_D, p_B, T_D, T_B, T_{C,D}, T_{C,B}, \frac{R}{F}, \frac{Q_B}{F}\right)^\intercal . \tag{5.1}$$

The use of the thermodynamic properties model to monitor top/bottom stream compositions is prohibitive in this case, even under any appropriate ideality assumptions. This is because there are too many degrees of freedom for the treated multi-component mixture that cannot be inferred from plant data. The considered nonlinear transformations (see Section 4.3.2) within the vector of input variables $\boldsymbol{x}$ (i.e., R/F, Q_B/F) are selected according to expert knowledge from refinery and literature (King, 2011). The

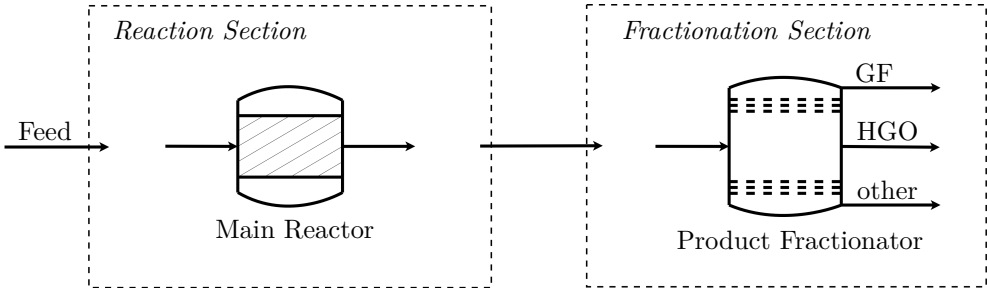


Figure 5.2: A schematic diagram of the VGH unit.

current soft sensor (denoted as Ref), applied in the refinery, uses three out of eleven variables and is designed according to King (2011) as follows:

$$x_{\mathrm{B}} = \beta_1 p_{\mathrm{B}} + \beta_2 T_{\mathrm{C,B}} + \beta_3 \frac{Q_{\mathrm{B}}}{F} + \beta_0. \tag{5.2}$$

This problem represents a rather standard and well-studied case study of designing a soft sensor.

5.1.2 VGH Unit

The purpose of this unit is to process the vacuum distillates by hydrotreating. This unit is separated into a high-pressure reaction section and a low-pressure fractionation section (see scheme in Figure 5.2). The main part of the reaction section is represented by the main reactor that hydrogenates the feed. This operation refines the feed from impurities, e.g., nitrogen and sulfur. The reaction section feeds the downstream fractionation section. Here the products are separated into a gasoline fraction (GF), a hydrogenated gasoil (HGO) and other (secondary) products.

Beside the main reactor, the VGH unit involves dozens of low-/high-pressure tanks, heat exchangers, coolers, and several distillation columns and furnaces. Furthermore, the unit contains many sensors, control devices (mostly PI controllers), and instrumentation to provide desired operating conditions and products. Overall, there are approximately 1,000 online measured variables available. Therefore, the soft sensor design for the VGH unit represents a much more challenging problem compared to the case of the FCC unit (11 variables measured at one distillation column).

The variable to be inferred by the soft sensor is HGO product purity expressed in terms of 95 % point of distillation curve $T_{95\%,\mathrm{HGO}}$. The design of a soft sensor is performed on the subset of the input variables selected from the whole available

dataset. The candidate inputs are selected based on consultation with operators and plant management. The resulting set of 30 candidate inputs is following:

$$\begin{aligned}\boldsymbol{x} = (&PCT_{\mathrm{HGO}}, PCT_{\mathrm{GF}}, T_{\mathrm{ex},1}, T_{\mathrm{ex},2}, T_{\mathrm{ex},3}, T_{\mathrm{ex},4},\\ &T_{\mathrm{wabt},1}, T_{\mathrm{wabt},2}, T_{\mathrm{wabt},3}, T_{\mathrm{wabt},4}, T_{\mathrm{wabt},5},\\ &RX_1, RX_2, RX_3, RX_4, RX_5, RX_6, RX_7,\\ &x_{\mathrm{H2}}, T_{\mathrm{frac},1}, T_{\mathrm{frac},2}, F_{\mathrm{frac,heat}}, p_{\mathrm{frac}},\\ &F_{\mathrm{f,rec}}, F_{\mathrm{f}}, x_{\mathrm{f,N2}}, x_{\mathrm{f,S}}, T_{\mathrm{f,5p}}, T_{\mathrm{f,50p}}, T_{\mathrm{f,95p}})^{\intercal},\end{aligned} \tag{5.3}$$

with pressure-compensated temperature PCT (see Eq. (4.5)), exotherms for the reactors T_{ex}, weighted average bed temperatures in the reactors T_{wabt}, ratios of gas/liquid phases in different sections RX, content of the hydrogen in the reaction section $x_{\mathrm{H_2}}$, temperatures in the main fractionator T_{frac}, flow rate of heat medium for main fractionator $F_{\mathrm{frac,heat}}$, pressure in the main fractionator p_{frac}, feed flowrate reconciled $F_{\mathrm{f,\ rec}}$, feed flowrate F_{feed} and content of impurities in the feed $x_{\mathrm{f,N2}}$, $x_{\mathrm{f,S}}$, T_{f}.

Current soft sensor (Ref) used in the refinery is of the following linear structure:

$$T_{95\%,\mathrm{HGO}} = \beta_1 PCT_{\mathrm{HGO}} + \beta_0. \tag{5.4}$$

The operators in the refinery have a good past experience with its performance. However, some recent operating conditions and changes to feedstock in the VGH unit caused significant deviations between estimated values from the reference soft sensor and the values obtained by the lab analysis. The plant management is unsure about the cause and so this study looks at the whole unit and its operation within up- and down-stream sections.

5.2 Solution Approach

This contribution focuses on the analysis of data treatment and model training in the overall soft-sensor design procedure (refer to Figure 4.1). The data treatment analysis aims to identify systematic errors and outliers present in the industrial datasets. Since these datasets involve multiple variables, three multivariate data treatment approaches are considered: T^2 distance, MCD, and k-means clustering (refer to Section 4.3.3 for details).

After the multivariate data treatment analysis, the retained data is utilised for soft-sensor design. The model training phase employs popular approaches such as OLS

regression (refer to Section 3.1), PCA regression (refer to Section 3.2.1), PLS (refer to Section 3.2.2), LASSO (refer to Section 3.3.2), as well as SS-MOC and SS-CV approaches from the family of SS (refer to Section 3.3.3). The performance of the designed soft sensors is compared with that of the reference (Ref) soft sensor currently implemented in the studied processes.

The design of the soft sensors is performed using datasets from both industrial case studies, specifically the depropanizer column from the FCC unit and the product fractionator from the VGH unit. To ensure a comprehensive analysis, the data is distributed to training ($\mathcal{I}_\mathrm{T}$) and testing ($\mathcal{I}_\mathrm{S}$) datasets in both chronological and random manners. The effectiveness of the designed soft sensors is evaluated based on appropriate performance indices (refer to Section 2.3.2).

5.3 Results

We present the results for both the presented use cases. We compare the performance of the presented data treatment methods and methods for soft-sensor design. Due to data confidentiality, the graphical representations of the results use the normalization of variables in the interval $[0, 1]$.

5.3.1 Implementation details

The implementation of all the presented methods is performed in MATLAB. For the initial data treatment, we use the Hotelling's T^2 distance considering χ^2-distribution with the probability of including 99.7 % measurements. For the MCD method, we select the value of parameter h as a midpoint of the interval $\frac{n+n_\mathrm{p}+1}{2} \leq h \leq n$ [1]. The outliers are determined by MCD considering an approximation of F-distribution [2] with the same probability as in the T^2 distance method. As a preliminary analysis suggested, the industrial data seem to be not normally distributed. Therefore the T^2 distance method considering χ^2-distribution tends to remove larger portions of data than MCD with F-distribution. The number of the desired clusters for the k-means clustering is determined using the elbow method. The results of the MCD method and the k-means clustering are gathered and averaged over 100 different runs of the respective algorithms. This is because of the inherent randomness of these methods, as mentioned above.

For the soft-sensor design, we set the variance-covariance methods (PCA and PLS) to select the amount of the variance explained by the principal components to at least 98 %.

The PLS method uses SIMPLS approach from MATLAB. We use Yalmip (Löfberg, 2004) and Gurobi (Gurobi Optimization, LLC, 2023) to solve various instances of the problems (3.1), (3.5), (3.6), and (3.10).

We will examine two different scenarios of soft-sensor design for each use case, focusing on the division of data into training and testing subsets, each comprising 50 % of the data. In both scenarios, the training set is utilized for designing the reference as well as the other soft sensors under study. In the first scenario, the available dataset is split into subsets based on the time series. The data from an earlier time period is used for training, while the data from a later time period is allocated for testing. This setup simulates soft-sensor design at a specific point in time using historical training data. The testing phase serves as a simulation of future sensor performance, where the sensor is deployed without any modification to its structure, even in the face of potential variations in plant operating conditions.

The second scenario groups the available data among training/testing subsets randomly. The results thus reveal the potential of the studied sensor-design methods for adaptation of the sensor structure to the changing operating conditions. In this scenario, the final results are gathered from 50 runs with different training/testing dataset distributions.

In order to tune the value of λ in (3.5) we use the goodness-of-fit criteria (2.28)–(2.30) and cross-validation (see Section 3.3.1) on the training set ($\mathcal{I}_{\mathrm{T}}$). We first obtain the candidate values of λ that minimize one of the goodness-of-fit criteria by training the sensors on the whole training set. Subsequently, we generate twenty different distributions of the training data into two subsets (similar to the SS-CV method). The candidate values of λ are used for regression and cross-validation on the generated subsets and the best performing value is used for the final sensor training.

When determining the final design of the soft sensor according to SS-CV, we take a median of $\mathbb{1}^{\mathsf{T}}\boldsymbol{z}^*_{\mathrm{ss}}$ from the results of the different runs (different validation data distribution and different values of K: $K \leq 6$ for the FCC unit, $K \leq 4$ for the VGH unit) to obtain the $n^*_{\mathrm{p}} \leq n_{\mathrm{p}}$, i.e., the number of inputs of the final sensor. Subsequently, we select the n^*_{p} most frequent inputs from the results of the different runs to finalize the sensor structure.

The complexity of each designed soft-sensor structure is determined according to the number of input variables n^*_{p}. We measure the impact of a particular input on the soft-sensor performance by the value of $|\beta_i|$. If the impact of a particular term is less than 0.1 % of the maximum value of the desired inferred variable, we neglect the corresponding part of the soft sensor.

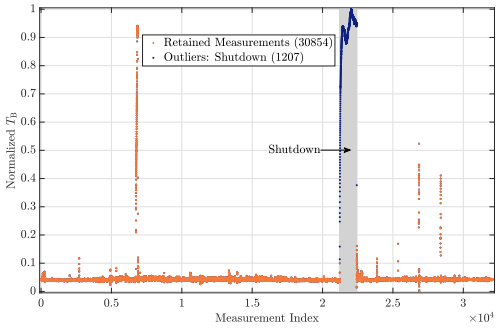
The accuracy of the soft sensors is evaluated and compared by the root mean square error (RMSE) of the sensor prediction on the testing dataset. The performance of the industrial soft sensors can be adjusted during the operation by an adaptive bias correction (see in Section 4.7.2), also called bias update. Therefore, in addition to the soft-sensor complexity (n_{p}^{*}) and accuracy (RMSE), we evaluate the effort of the bias correction ($\mathrm{E_{BC}}$) by simulating a bias correction procedure in parallel, i.e., without affecting the prediction error of the sensor evaluated by RMSE. The measure of the bias-correction effort is expressed as the percentage of measurement-based sensor corrections occurrences in the testing dataset.

5.3.2 Soft Sensors for the FCC Unit

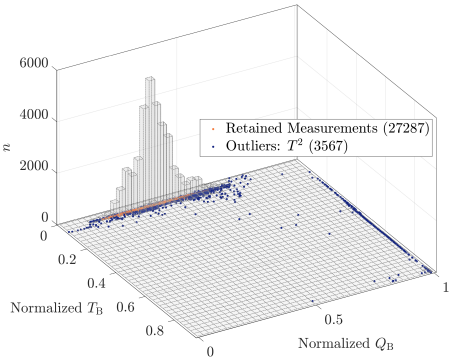
The available historical data involving 32,061 measurement points from online sensors (candidate input variables) represents more than two years of production in the period 2016–2019. This time span contains 181 lab measurements of the bottom product concentration x_{B} (output variable).

We first perform the data treatment to reduce the amount of systematic and gross errors. Figure 5.3a shows visualization of the data treatment results on the normalized temperature of the bottom product T_{B}. The visual inspection of the time series of the available data (data pre-treatment) reveals the initial set of systematic errors with significantly deviated data, which corresponds to the shutdown period of the unit. This is marked as a thick gray bar in Figure 5.3a. The unit operators confirmed in consultation the correctness of omission of the corresponding 1,207 data points from the further processing.

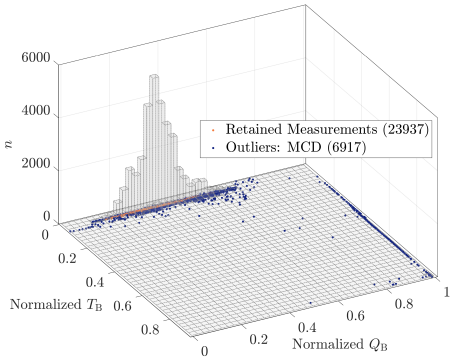
Subsequently, we applied the T^2 distance, MCD, and k-means clustering methods to detect outliers in the dataset. The performance of these methods is individually visualized and compared in Figures 5.3b, 5.3c and 5.3d for lucidity. Each figure shows a histogram of data points of bottom product temperature vs. reboiler heat duty. All the methods clearly identify the most distinct outliers. The results further show that k-means clustering (Figure 5.3d) might be overly conservative as it selected significantly fewer outliers than the other two methods. The low performance of this method is caused by the complex tuning (e.g., number of clusters). The k-means clustering method detects only five data clusters, which results in the low number of indicated outliers by this method. The number of outliers indicated by the MCD method (Figure 5.3c) is almost twice higher compared to the T^2 distance method (Figure 5.3b). The MCD method thus appears as a reasonable choice here as it removes a significant amount of outliers, yet retains reasonable number of data points, of which it guarantees better quality than the T^2 distance approach.



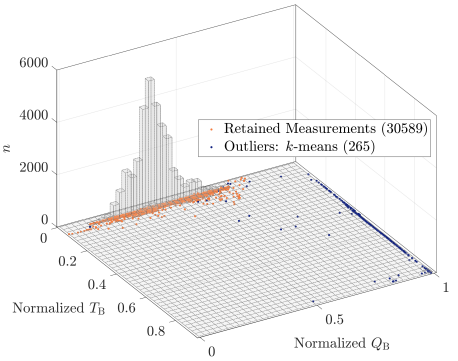
(a) Data pre-treatment by visual inspection detecting plant shutdowns.



(b) Data treatment by the T^2 distance method.



(c) Data treatment by the MCD method.



(d) Data treatment by the k-means clustering method.

Figure 5.3: **(a)** Normalized bottom product temperature of the FCC unit vs. measurement index. **(b)**, **(c)**, **(d)** Histogram of the bottom product temperature vs. reboiler heat duty of the FCC unit and retained measurement vs. outliers as detected by data treatment methods.

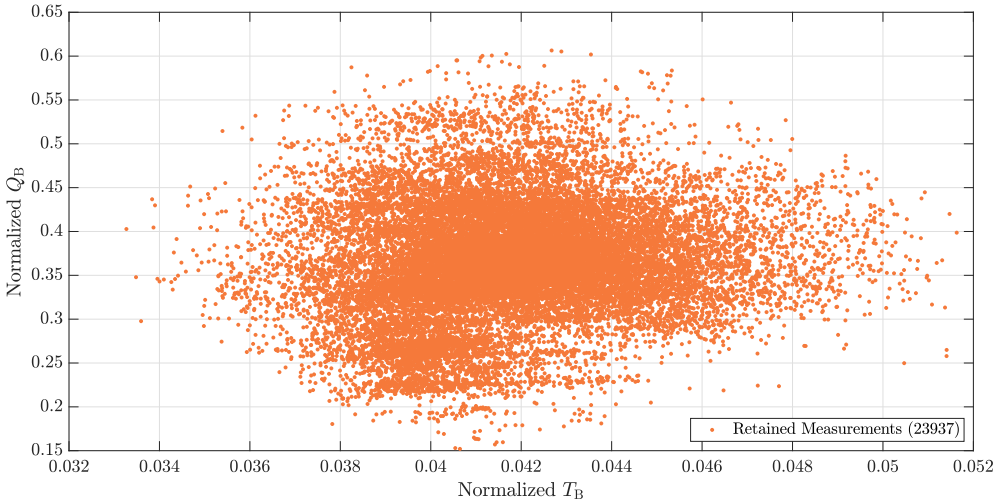


Figure 5.4: The retained online measurements (by the MCD method) of the bottom product temperature and reboiler heat duty of the FCC unit.

It is obvious that the majority of identified outliers (blue points in Figure 5.3c) by the MCD method deviates from the area with the highest density of the online measurements. On the same line, the approved measurements (green points in Figure 5.3c) are located inside or are very close to this area. This also indicates the good performance of the MCD method. The final set of the retained measurements by this method for the soft-sensor design is shown in Figure 5.4. It is evident that the MCD method provides well-poised data set, which appears to be close to normal distribution. We can conclude that the available industrial data are of good quality and that the conducted data treatment was able to reveal the high-quality data.

5.3.3 Design of Soft Sensors for the FCC Unit using Time Series Data

We first study a scenario where the (chronologically) first 50 % of the available data is assigned to the training set and the last 50 % of data is assigned to the testing set.

Soft-sensors designed by PCA and PLS require six and seven principal components, respectively, to explain 98 % of the variance in the data. This relatively high number of principal components suggests, on the one hand, to use a more complex structure of soft sensor than the reference soft sensor. On the other hand, sensors designed by

Table 5.1: Comparison of the number of inputs n_{p}^{*} (n_{pc}^{*} for PCA and PLS shown in brackets), sensor accuracy (RMSE) and bias correction relative effort or frequency ($\mathrm{E_{BC}}$) using time series data for the FCC unit.

	OLS	PCA	PLS	LASSO	SS-MOC	SS-CV	Ref
n_{p}^{*}	11	11 (6)	11 (7)	5	4	4	3
RMSE	0.120	0.096	0.104	0.099	0.099	0.099	0.117
$\mathrm{E_{BC}}$ [%]	29.7	21.6	24.3	20.3	23.0	23.0	28.4

these methods might be overfitted.

When designing a soft sensor by the SS methods, we compared the performance of the presented overfitting criteria (R^2_{adj}, AICc, BIC). We used the principle of parsimony. The simplest sensor yet the best performing one is designed by SS with R^2_{adj} criterion. This sensor is the same as suggested by SS with cross-validation in this case and it is selected for further performance analysis.

A comparison of the designed sensors in terms of their complexity (n_{p}^{*}), accuracy (RMSE), and the effort of the bias correction ($\mathrm{E_{BC}}$) is shown in Table 5.1. The results clearly suggest to enrich the structure of reference soft sensor to include at least one extra variable in order to improve its performance (see n_{p}^{*} in Table 5.1). The least complex sensors are suggested by the LASSO and SS methods. These methods suggest replacing bottom pressure p_{B} by temperatures $T_{\mathrm{C,D}}$ and $T_{\mathrm{C,B}}$ (LASSO selected also the ratio R/F). These sensors (including PCA) exhibit a reduced amount of bias correction compared to all others.

Overall, the accuracy of the reference soft sensor (see RMSE in Table 5.1) shows almost the worst performance. Only the (most likely overfitted) soft sensor designed by OLSR is worse in this comparison, despite using all the possible eleven inputs. The overfitting by OLSR can be documented by worsened accuracy and also by a high effort of the bias correction.

The highest sensor accuracy is achieved for the PCA-based soft sensor. The improvement compared to the reference soft sensor is approximately 18 %. Other proposed advanced sensors show similar performance (improvements of at least 15 %).

Looking at the amount of bias correction, we can see that the most frequently corrected soft sensor is designed by OLSR, while the soft sensor designed by LASSO requires the bias correction less frequently than others. The best sensor would be selected as

a compromise between accuracy, complexity, and maintenance (E_{BC}) effort. In this respect, all the advanced designed soft sensors represent good candidates.

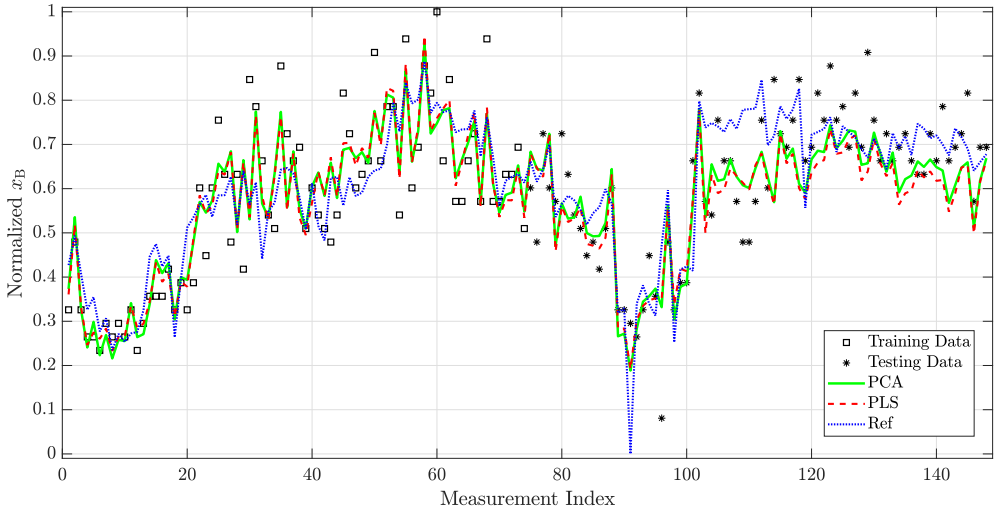
In order to provide a more comprehensive comparison of the soft sensors, Figure 5.5 visualizes their predictive performance on the output variable. The lab-analysis data is shown as black squares (training dataset) and black stars (testing dataset), respectively. The data show significant variability indicating several changes of the operating conditions within the studied time window, in both training and testing datasets. This means that the trained sensors face a rich portfolio of situations and thus a trained sensor can be expectedly valid for a long time after its commissioning. This is confirmed by the aforementioned good performance of the designed sensors and by the relatively low effort of the bias-update mechanism.

Figure 5.5 further presents the training and testing (predictions) performance of the designed advanced soft sensors, by PCA and PLS (Figure 5.5a; green solid line and red dashed line, respectively) and by LASSO and SS-CV (Figure 5.5b; magenta solid line and green dashed line, respectively), compared in both figures to the reference soft sensor (blue dotted line).

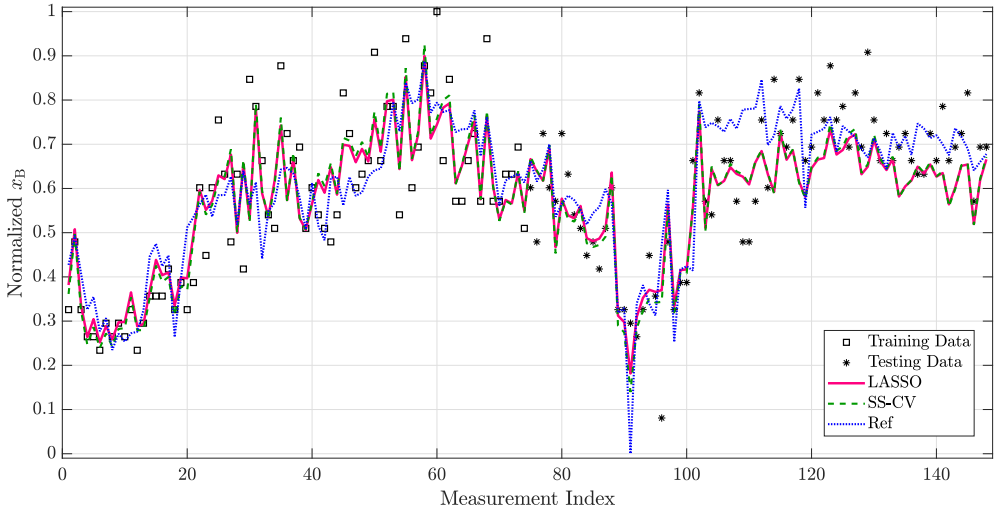
When looking at the performance of the reference sensor in both plots, one can clearly identify several points, where the reference sensor is not able to explain the measurements yet the advanced sensors are. This is present throughout the whole studied time window but it is most evident in the testing phase (around the measurements 80–120).

We can see that despite the behavior of the soft sensors designed by PCA and by PLS being similar in the training phase, the evolution of the predictions of these sensors on the testing data is quite different. This also explains differences in the accuracy and frequency of the bias correction. It also further supports our earlier conjecture of possible overfitting present in these sensors. This observation is in contrast with the bottom plot (LASSO and SS-CV), where the outputs of the visualized advanced sensors are almost identical.

A noticeable part of the testing phase is the last period (around the measurements 130–148), where it seems that the operating conditions in the FCC unit change considerably. There exist corresponding significant discrepancies between the measurements and values inferred by all the advanced soft sensors. The reference soft sensor, however, performs well here, which suggests good robustness properties of this sensor. All the advanced sensors exhibit a slower or faster drift from the measurements. This situation calls for sensor maintenance or complete structural change. It appears that a practical



(a) Training and prediction performance of the sensors designed by PCA and PLS methods and of the reference sensor.



(b) Training and prediction performance of the sensors designed by LASSO and SS-CV methods and of the reference sensor.

Figure 5.5: Comparison of the soft sensors for the FCC unit designed using time series data.

Table 5.2: Comparison of the number of inputs n_{p}^{*} (n_{pc}^{*} for PCA and PLS shown in brackets), sensor accuracy (RMSE) and bias correction relative effort or frequency ($\mathrm{E_{BC}}$) over 50 random training/testing data distributions for the FCC unit.

	OLS	PCA	PLS	LASSO	SS-MOC	SS-CV	Ref
n_{p}^{*}	11	11 (7)	11 (8)	7	6	5	3
RMSE	0.105	0.104	0.106	0.106	0.106	0.110	0.121
$\mathrm{E_{BC}}$ [%]	23.0	24.3	23.0	24.3	27.0	24.3	28.4

solution of performing bias update would be sufficient. We will revisit and analyze this issue in the following section in order to confirm whether the operating conditions change so dramatically that one would need to change the soft sensor structure.

5.3.4 Design of Soft Sensors for the FCC Unit using Randomly Distributed Data

We randomly distribute 50 % of the available data to the training set and the remaining data to the testing set. We generate 50 such distributions to increase the interpretabilily of the results. We then use the same workflow to design the soft sensors as outlined above.

We report averages of n_{p}^{*}, RMSE and $\mathrm{E_{BC}}$ for each soft sensor over the 50 data distributions in Table 5.2. According to the sensor complexity criterion (n_{p}^{*}), we can see that the designed soft sensors suggest more complex structure (at least two extra input variables) compared to the reference structure and also compared to the previous scenario with chronological training/testing data assignment. This suggests that varying operating conditions in the plant would require frequent revision of the sensor structure for better performance. The performance of the designed advanced sensors does not improve compared to the designs using chronological training/testing data distribution, which is a consequence of the overfitting implied by the increased complexity of the sensor. For example, LASSO and both SS methods commonly suggest including T_{D} and Q_{B} on top of the inputs suggested in the previous section. However, none of these variables seem to be significantly useful for the sensor overall. While, unlike for distillate temperature T_{D}, inclusion of Q_{B} would make sense from process viewpoint, its effect is already present in the input Q_{B}/F. Only the soft sensor designed by OLS exhibits improved accuracy compared to the design with chronologically distributed training/testing data. This is a consequence of providing

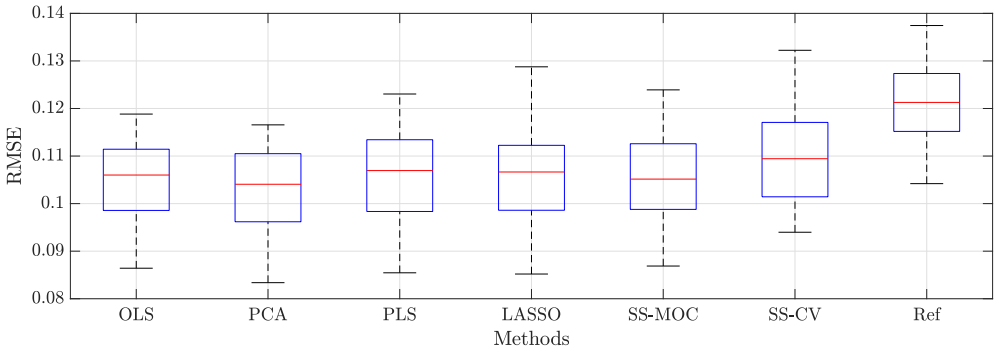


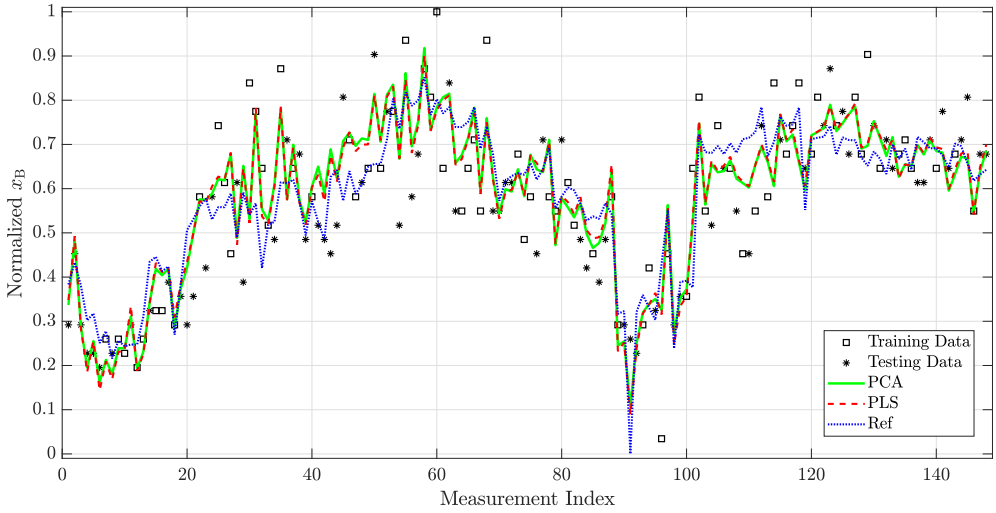
Figure 5.6: Comparison of accuracy of the designed soft sensors over 50 different random training/testing data distributions for the FCC unit.

better training data (more similar to testing ones) to the sensors, which reduces the overfitting effect. Designed advanced soft sensor (including PCA) shows the increased frequency of the bias correction, which can be attributed to the large noise magnitude in the lab data and overfitting.

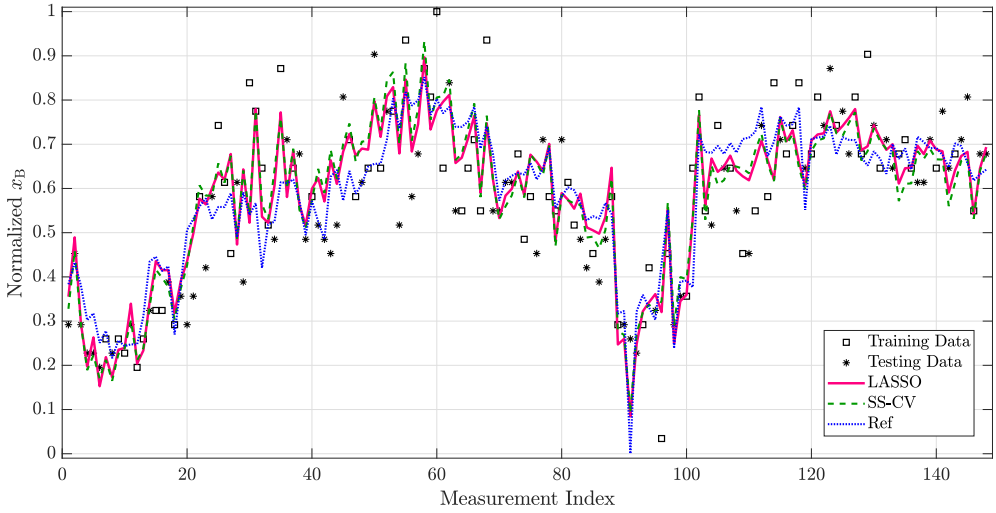
The performance features of the particular sensors remain practically the same as in the case of chronological training/testing data distribution. The soft sensor designed by PCA is slightly more accurate than other soft sensors and it improves the accuracy of the reference soft sensor by about 14 %. Yet the drop in this improvement confirms the overfitting. The structure of the soft sensor designed by SS-CV is less complicated than the structures of other designed soft sensors. As expected, the least complex sensor designed by SS-CV is again followed in terms of performance by design using the SS-MOC and LASSO methods, respectively.

Figure 5.6 visualizes the accuracy statistics of each soft sensor from the 50 randomly generated training/testing datasets using box plots. The central horizontal-line marker indicates the median, the bottom and top edges of the box indicate the 25^{th} and 75^{th} percentiles, respectively, the whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the '+' symbol. We can see that the median performance mostly copies the average performance of the designed soft sensors outlined in Table 5.2.

The accuracy variance seems to be considerable for all sensors, which confirms the aforementioned large noise in the samples the possible sensor overfitting. The least variance is present in the reference sensor, which is due to the aforementioned robustness properties.



(a) Training and prediction performance of the reference sensor and of the sensors designed by PCA and PLS methods.



(b) Training and prediction performance of the reference sensor and of the sensors designed by LASSO and SS-CV methods.

Figure 5.7: Comparison of the soft sensors using randomly distributed training/testing data for the FCC unit.

As in the previous section, we visualize the training and prediction performance of the designed soft sensors in Figure 5.7 for one representative random training/testing data distribution. We again show results obtained for the reference soft sensor (both plots; blue dotted line), the soft sensors designed by PCA and PLS (Figure 5.7a; green solid line and red dashed line, respectively), and the soft sensors designed by LASSO and SS-CV (Figure 5.7b; magenta solid line and green dashed lines, respectively).

As can be expected, the performance of the soft sensors is similar to the performance of the soft sensors designed by using time series data. The previously discussed discrepancy between the sensors and the measurements (around the measurements 130–148) is decreased. This, together with the increased complexity of the sensors designed using randomly distributed data, leads us to the conclusion that the performance of an advanced sensor can only be maintained if the sensor structure changes frequently or if the sensor parameters are frequently updated. Of course, in this particular case, the problem would be practically resolved by bias update.

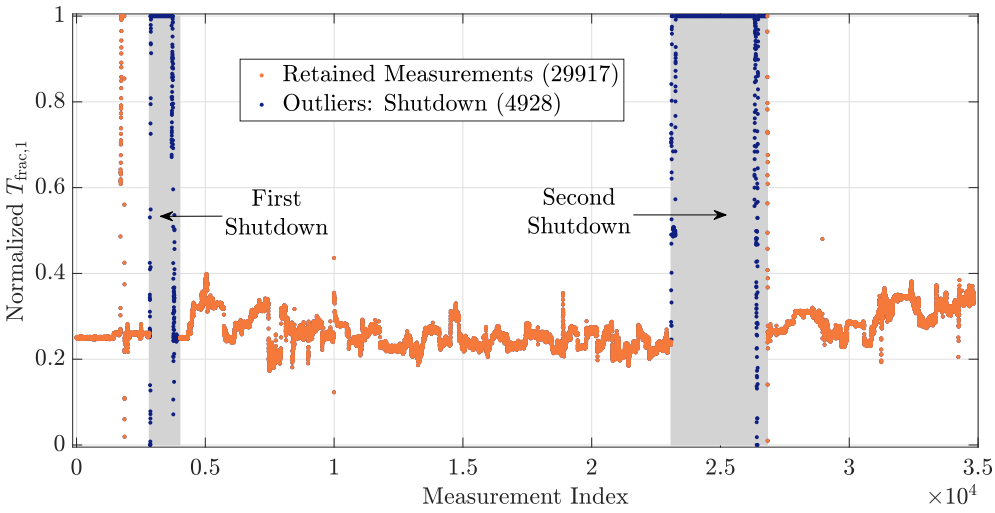
5.3.5 Soft Sensors for the VGH Unit

The available historical data encompasses almost two years of production in the period 2018–2019 with 34,845 time points of online measurements. This is a comparable amount of data as in the previous case study. The desired output variable $T_{95\%,\mathrm{HGO}}$, which, similarly to the previous use case, indicates the purity of the distillation product, is determined by the lab analysis. The mentioned time span involves 689 measurements of the output variable as it is measured more frequently than in the case of FCC.

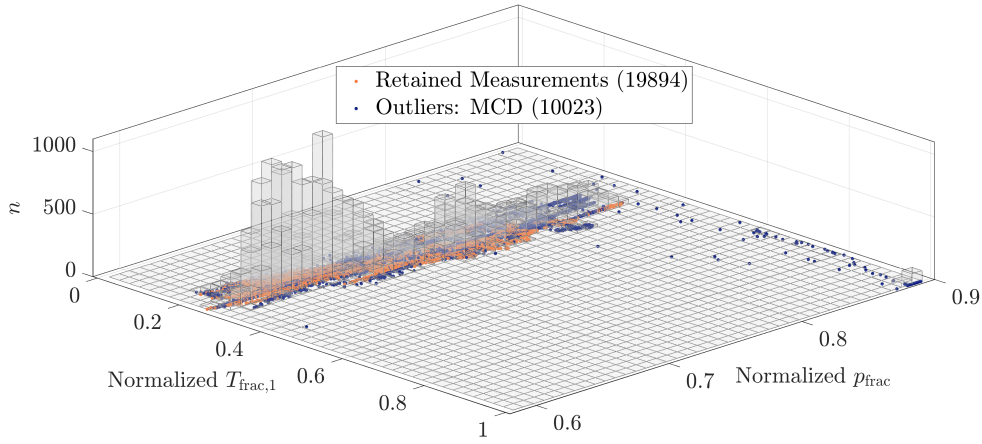
We first perform the pre-treatment of the available data. Based on the visual inspection of the time series of a temperature in the main fractionator (Figure 5.8a), we eliminated two intervals with obviously deviated measurements (see gray intervals in Figure 5.8a). The unit operators confirmed that the omitted 4,928 measurement points (black points in Figure 5.8a) correspond to the unit shutdowns.

Subsequently, the remaining data is processed using the T^2 distance, MCD, and k-means clustering methods. Although the visualized temperature data does not seem to be much qualitatively different in nature than the case of the FCC unit, there are more distinct variations and steady states. This feature causes that the T^2 distance method suggests removing more outliers than MCD and k-means clustering.

For the case of the T^2 distance method, 13,874 outliers is indicated, which represents almost half of the available pre-treated data. This behavior can be attributed to the previously observed distinct variations and steady states, which bias the statistics



(a) Data pre-treatment by visual inspection detecting plant shutdowns.



(b) Data treatment by the MCD method.

Figure 5.8: **(a)** Normalized temperature in the main fractionator of the VGH unit vs. measurement index. **(b)** Histogram of the temperature vs. pressure in the main fractionator of the VGH unit and retained measurements vs. outliers after data treatment.

used in the T^2 distance method. In fact, if we wanted to tune this method to the similar performance as the MCD method, we would require increasing the probability of measurements acceptance from 99.7 % to 99.9 %. This seemingly small alteration represents a significant increase in the acceptance, by one half of a standard deviation.

The MCD method indicates slightly more outliers (10,023 measurements) as in the case of the FCC unit (6,917 measurements), which may be caused by the worse quality of the data from the VGH unit. The k-means clustering method indicates much more outliers (11,229 measurements) than in the FCC unit (265 measurements). It uses 21 clusters (compared to five clusters detected for the FCC unit), which seems to be a consequence of the distinct variations and steady states. Nonetheless, the data distribution among the clusters exhibits certain uniformity, which further demonstrates the sensitivity of the k-means clustering method to tuning (e.g., number of clusters).

As in the case of the FCC unit, we again choose to remove the outliers labeled by the MCD method as it retains reasonable amount of data points. Even though the data quality (e.g., number of shutdowns, variations of the operating conditions) of the VGH unit is worse than the FCC unit, we can see more minor differences among the applied data-treatment methods. Therefore, only the performance of the MCD method is further shown via the histogram of data points of temperature vs. pressure in the main fractionator in Figure 5.8b. The blue points represent indicated outliers and the rest of the data (green points) is retained for the design of soft sensors (Figure 5.9). We can conclude that the marked outliers are mostly measurements deviated from the area with the highest density of the measurements. This proves the effectiveness of the MCD method to indicate deviated and undesirable measurements.

5.3.6 Design of Soft Sensors for the VGH Unit using Time Series Data

We design soft sensors in the same way as in Section 5.3.3. Therefore, we distribute (chronologically) first 50 % of the available time series data to training set and last 50 % of available time series data to the testing set.

Soft-sensors designed by PCA and PLS require thirteen and fifteen principal components, respectively, to explain 98 % of the variance in the data. This, on the one hand, suggests possible overfitting yet, on the other hand, there seems to be a good agreement between the advanced design methods on the number of important variables (or their combinations), i.e., 13–15. When designing a soft sensor by the SS methods, similar to the previous use case, we found that the combination with the R^2_{adj} criterion

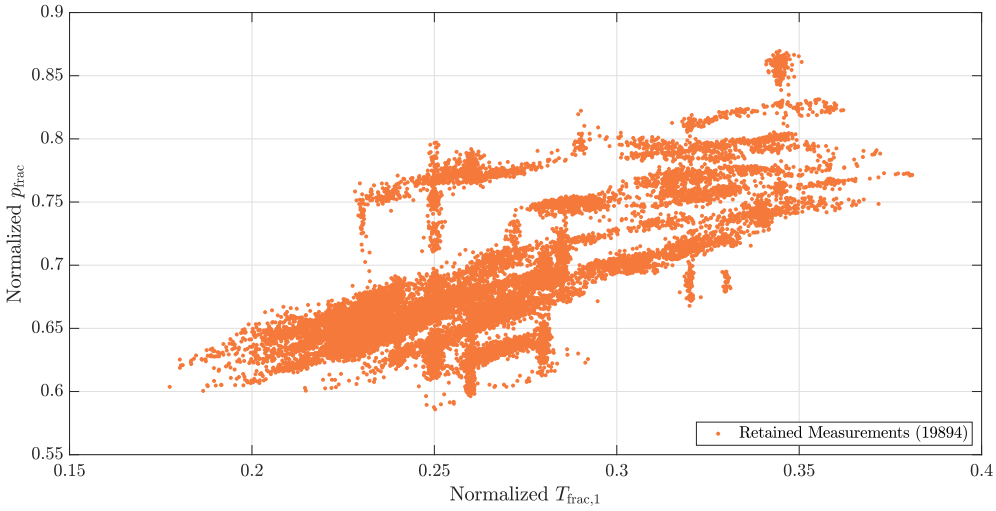


Figure 5.9: The retained online measurements (by the MCD method) of the temperature and pressure in the main factionator of the VGH unit.

Table 5.3: Comparison of the number of inputs n_{p}^* (n_{pc}^* for PCA and PLS shown in brackets), sensor accuracy (RMSE) and bias correction relative effort or frequency (E_{BC}) using time series data for the VGH unit.

	OLS	PCA	PLS	LASSO	SS-MOC	SS-CV	Ref
n_{p}^*	19	24 (13)	22 (15)	14	15	12	1
RMSE	0.184	0.103	0.158	0.145	0.190	0.182	0.114
E_{BC} [%]	82.8	85.3	80.2	74.6	82.3	79.3	75.4

gave the best results. Unlike in the case of the FCC unit, the SS-CV method proposes different sensor structure as the SS method (with R_{adj}^2 criterion).

A comparison of the designed sensors in terms of their complexity (n_{p}^*), accuracy (RMSE), and the amount of bias correction (E_{BC}) is shown in Table 5.3. As we can see, the suggested structure (n_{p}^*) of the designed soft sensors is much more complicated than the structure of the reference soft sensor. Out of 30 candidate inputs, the designed sensors suggest to include at least eleven more inputs. All the design methods (even OLS) are able to sparsify to a certain extent the structure of the full sensor (5.3). Beside PCT_{HGO} included in the reference sensor, LASSO suggests involving $T_{\text{frac},2}$, $T_{\text{frac},1}$, $T_{\text{f,50p}}$ and x_{H2} among the most influential variables. On contrary, the SS methods suggest including $T_{\text{frac},2}$, $T_{\text{frac},1}$, PCT_{GF} and $T_{\text{wabt},1}$. Despite the disagreement on

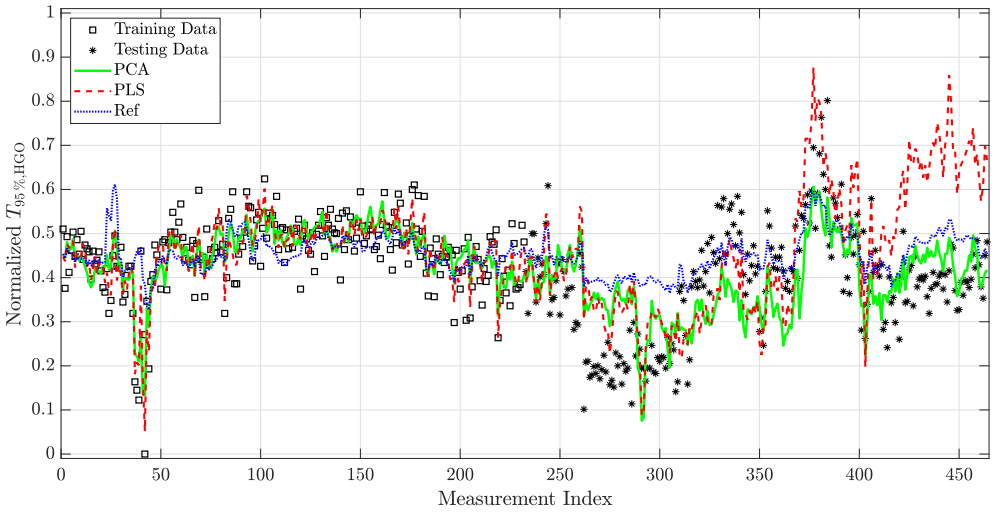
the added variables, it appears that certain variables from the reaction section of the plant could play a role in explaining the bad performance of the reference sensor when qualitatively different feedstock is used.

Overall, the accuracy of the designed soft sensors (see RMSE in Table 5.3) shows the best performance for the soft sensor designed by PCA, good performance of the soft sensors designed by the PLS and LASSO methods and the worst performance of the soft sensors designed by OLS and SS methods. Apparently, the reference sensor shows high robustness. The poor accuracy of the soft sensor designed by SS methods can be explained by the highly varying operating conditions of the plant. This can also be documented by the much increased amount of bias correction compared to the case of the FCC unit (see in Table 5.1).

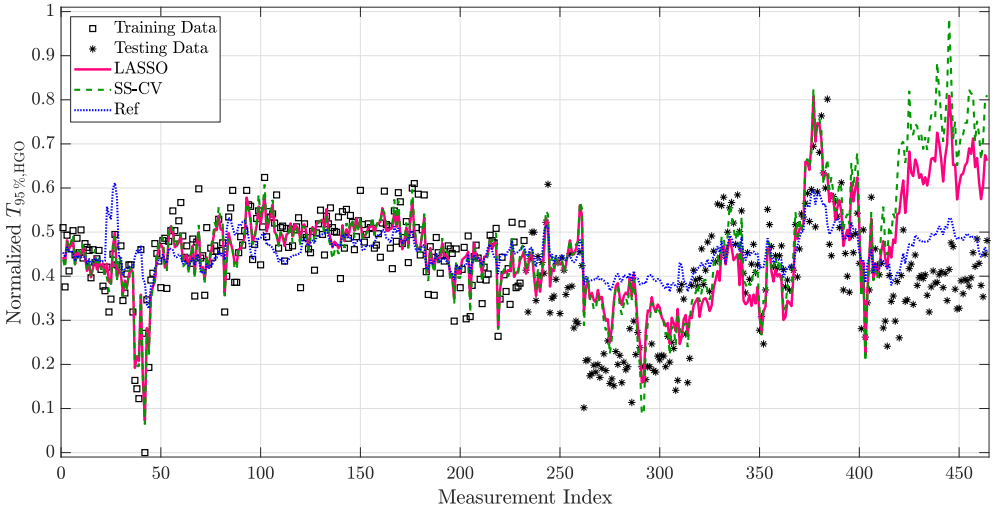
We can see that the soft sensor designed by OLS is much more complicated, less accurate and more frequently corrected than the reference soft sensor. The results show that PCA and PLS methods are not able to reduce the dimensionality of the soft sensor compared to OLS. The high number of principal components of these methods also suggests that a complex structure is required to express the behavior of the desired variable. The soft sensors designed by PCA, PLS and LASSO are more accurate than other designed soft sensors. Nevertheless, only the PCA sensor is more accurate (by about 10 %) than the reference soft sensor. According to the values of the E_{BC} criterion in Table 5.3, the soft sensor designed by PLS is more appropriate than the PCA soft sensor, although both sensors are more frequently corrected than the reference soft sensor. The further values of E_{BC} indicate that soft sensors designed by LASSO and SS-CV are corrected less frequently than other designed soft sensors, which results from their simple structure (and implied robustness).

In order to provide more comprehensive comparison of the soft sensors, we visualize their predictive performance on the output variable in Figure 5.10 using the same color coding as in the previous case study. The data shows high variability indicating several changes of the operating conditions within the studied time window, in both training and testing datasets. Nonetheless, the variability within the testing set appears to be higher. This might explain the poor performance of the designed advanced sensors and it is confirmed by the high effort of bias correction.

Figure 5.10a further presents the training and testing (predictions) performance of the designed advanced soft sensors, by PCA and PLS (Figure 5.10a) and by LASSO and SS-CV (Figure 5.5b), compared to the reference soft sensor. We can directly see the training performance of the designed advanced soft sensors being much better than the reference soft sensor. However, there are several sections in the testing dataset,



(a) Training and prediction performance of the sensors designed by PCA and PLS methods and reference (Ref) sensor.



(b) Training and prediction performance of the sensors designed by LASSO and SS-CV methods and reference (Ref) sensor.

Figure 5.10: Comparison of the soft sensors for the VGH unit designed using time series data.

Table 5.4: Comparison of the number of inputs n_{p}^{*} (n_{pc}^{*} for PCA and PLS shown in brackets), sensor accuracy (RMSE) and bias correction relative effort or frequency ($\mathrm{E_{BC}}$) over 50 random training/testing data distributions for the VGH unit.

	OLS	PCA	PLS	LASSO	SS-MOC	SS-CV	Ref
n_{p}^{*}	24	25 (15)	25 (17)	15	16	12	1
RMSE	0.086	0.087	0.085	0.086	0.086	0.087	0.105
$\mathrm{E_{BC}}$ [%]	88.8	86.6	86.6	90.1	89.7	87.1	91.0

where these soft sensors are not able to explain the behavior of the output variable. This is most prominent around the measurements 260–320 and 420–464. Interestingly, PCA-based soft-sensor performs relatively well in both the designated periods, which suggests that some process features were successfully caught in the sensor. On the other hand, it exhibits a relatively poor performance around measurement index 350, where it is outperformed by other sensors (even the reference sensor). These observations suggest that the training set is poor and should be expanded.

It appears that a practical solution of performing bias update would be sufficient in this situation. We will revisit and analyze this situation in the following section in order to confirm whether the operating conditions change so dramatically that one would need to vary the soft-sensor structure often.

5.3.7 Design of Soft Sensors for the VGH Unit using Randomly Distributed Data

Next, we design soft sensors using randomly distributed data. We assign 50 % of the available randomly distributed data to the training set and the remaining data to the testing set. We generate 50 such distributions and we use the same training/testing workflow as above. We finally present the average performance measures from the different runs of the corresponding soft-sensor design.

The comparison of soft sensors in Table 5.4 involves the same criteria (n_{p}^{*}, RMSE, $\mathrm{E_{BC}}$) as in the previous section. In terms of complexity of the designed sensors, we see similar trend as in the FCC use case. The overall complexity of the designed soft sensors is mostly higher compared to the soft sensors designed on chronologically distributed data (see Table 5.3). This is a recurring observation (from the first case study) and points at the need of enriching the number of explaining variables to adapt for varying

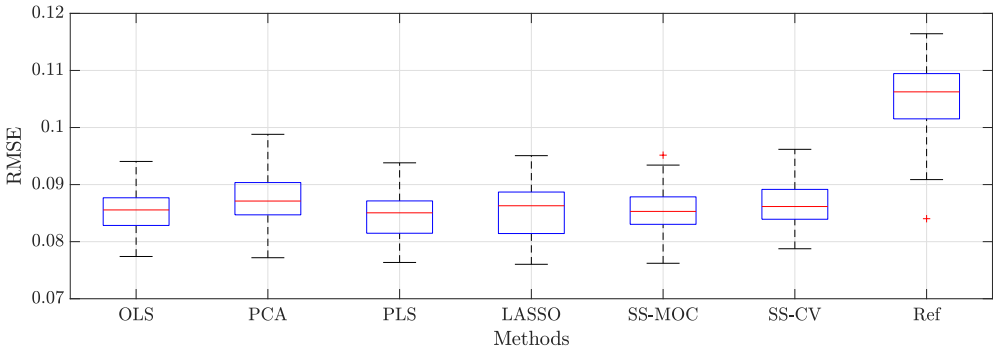


Figure 5.11: Comparison of the designed soft-sensor accuracy (RMSE) over 50 different randomly generated training/testing data distributions for the VGH unit.

plant operating conditions. Only the soft sensor designed by SS-CV is an exception and it even maintains exactly the same sensor structure. These observations reveal that despite SS-CV found good sensor structure in case of chronologically distributed data, the variation in the operating conditions would require to adapt sensor parameters. This also definitely proves high variation of operating conditions and its strong influence on the sensor performance. Similar to SS-CV, also for the rest of the designed sensors the most influential inputs selected by the design methods remain unchanged compared to the case of chronological training/testing data distribution. Each designed soft sensor shows the increased frequency of the bias correction, which can be attributed to the large noise magnitude in the lab data and to the need for adapting the sensor frequently due to operating conditions.

The accuracy of the designed soft sensors is essentially the same and each sensor is more accurate than the reference sensor. The most accurate sensor is designed by PLS and it improves the accuracy of reference sensor by about 19 %. A drop in this performance by PCA-based sensor can be attributed to significant changes in the operating conditions in combination with changes in the sensitivity of the output variable to different inputs (online measurements). The latter claim is supported by the comparatively better performance of the sensor designed by PLS.

Figure 5.11 visualizes the accuracy statistics using box plots of each soft sensor from the 50 randomly distributed training/testing datasets. We can see that the performance statistics of all the designed soft sensors mostly copies the conclusions reached in the discussion on the average performance (see Table 5.4). The results show similar accuracy variance of each soft sensor, which means that the variance is caused mainly

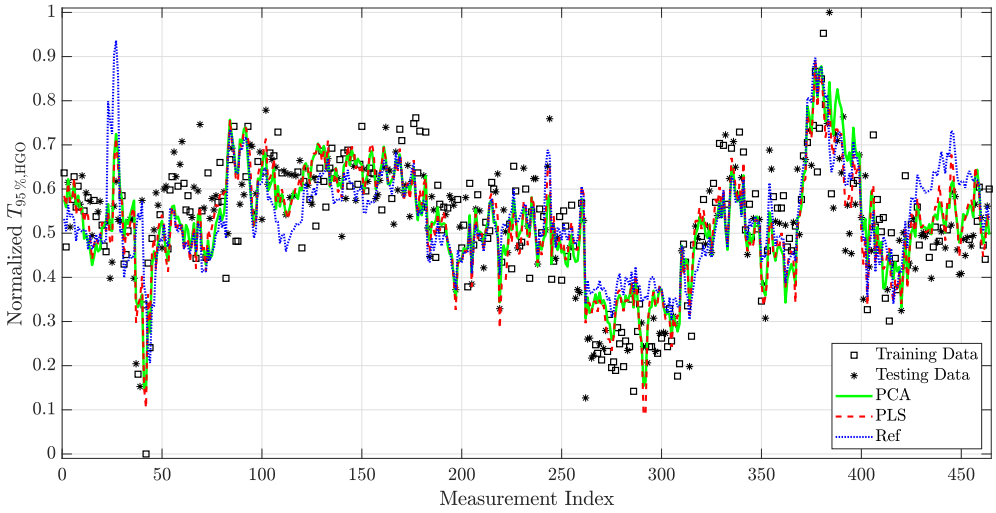
by the particular noise realizations in the data. The smallest variance is though achieved for the sensors found by SS methods.

As in the previous section, Figure 5.12 visualizes the training and prediction performance of the designed soft sensors for one representative random training/testing data distribution. Results are shown for the reference soft sensor (both plots), the soft sensors designed by PCA and PLS (Figure 5.12a), and the soft sensors designed by LASSO and SS-CV (Figure 5.12b). The performance improvement of the soft-sensors with randomly distributed data compared to chronological data is evident. We can observe this on previously mismatched measurements around markers 420–464. Yet, we can clearly identify the period of measurements 260–320 that still exhibits unsatisfactory sensor performance. This calls for another investigation at the plant and revision of the set of candidate sensor inputs.

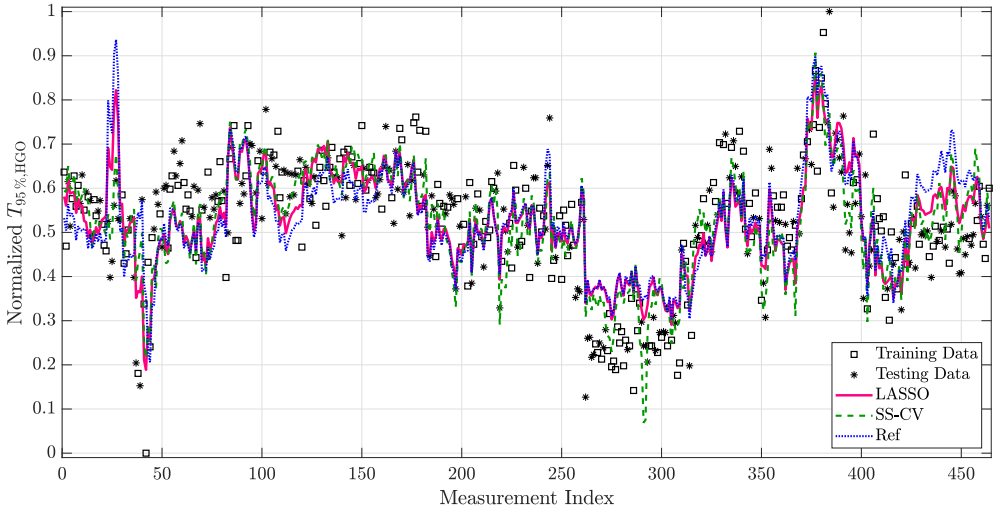
In conclusion, the advanced design methods show great potential for improving the sensor accuracy beside the good robustness properties of the reference sensor. Yet due to the complexity of the use case, the price to pay for the improved performance is paid in terms of higher sensor complexity. Moreover, due to varying operating conditions, the advanced sensors would need to be often updated or trained on a carefully selected training set.

5.4 Discussion

Overall, we can say that each studied data treatment method is able to a certain extent indicate outliers in the multivariate data. The advantage of the T^2 distance method lies in its simplicity. However, this method is strongly affected by the number of treated variables and by the data distribution. The T^2 distance method selects fewer outliers in the FCC unit data (3,567 outliers) than the VGH unit (13,874 outliers). The best results were achieved using the MCD method, which guarantees higher quality of the retained data than the T^2 distance method. The performance of this method seems to be consistent in both case studies. The MCD method indicates 6,917 outliers in the FCC unit and 10,023 outliers in the case of the VGH unit. The higher number of indicated outliers in the case of the VGH unit is caused mainly by the worse quality of the data. The treatment of the industrial data pointed out that k-means clustering is quite sensitive to tuning (e.g., number of clusters) that might lead to inferior-quality data treatment. We can see an even more significant discrepancy between the number of indicated outliers in the FCC unit and VGH unit (265 outliers and 11,229 outliers, respectively) by k-means clustering as in the T^2 distance method. It seems that the performance of this method should be adjusted to select more outliers in measurements



(a) Training and prediction performance of the sensors designed by PCA and PLS methods and reference (Ref) sensor.



(b) Training and prediction performance of the sensors designed by LASSO and SS-CV methods and reference (Ref) sensor.

Figure 5.12: Comparison of the soft sensors using randomly generated training/testing data distribution for the VGH unit.

in the case study on the FCC unit.

The performance of the soft sensors designed by the studied data-driven method (OLS, PCA, PLS, LASSO, SS-MOC and SS-CV) is compared against the reference (current) sensor in both case studies. The reference sensor has a relatively simple structure (three input variables) in the FCC unit and a simple structure (one input variable) in the VGH unit. The low structural complexity provides higher robustness of the soft sensors. We could see this robustness when the soft sensors were designed according to the chronological training/testing dataset of the VGH unit. In this case, the designed advanced soft sensors are more complex yet less accurate than the reference sensor in the final section of the testing dataset. It is most likely that the process deviates from the operating conditions present during the training phase and the advanced sensors would require frequent parameter adaptation to maintain the designed performance.

The results from chronological distribution of training/testing dataset indicate that the soft sensor designed by PCA achieved the highest accuracy. It outperforms the reference sensor by about 18 % in the FCC unit and by about 10 % in the VGH unit. Such sensor could be used for plant monitoring. On the other hand, if we also consider the sensor complexity, then the SS-CV method outperforms the rest of the approaches. A low-complexity sensor would be more suitable for optimization or advanced control.

The design of soft sensors considering both chronologically and randomly distributed training/testing datasets seems to be an effective way to determine the impact of changing operating conditions in the process. The results suggest that the soft sensors designed over the chronologically distributed training/testing dataset are less sensitive to overfitting than the randomly distributed training/testing dataset. This phenomenon supports the hypothesis of the occurrence of varying operating conditions since the trained sensors tend to involve more inputs to model the changing conditions.

Our investigation has also found that soft sensors commonly used in the petrochemical industry show high robustness and can give solid performance even long after their commissioning. On the other hand, the relative simplicity of the structure can be easily enhanced in simple cases (the FCC unit use case) by extension of the structure without much maintenance effort. Such sensors can also improve the trust of the operators in the sensors and the automation technology. For this purpose, advanced methods of soft-sensor design (LASSO and SS-MOC methods) show a good promise and even the associated computational burden is justified. In more complex cases, the studied design methods can be a promising technology for root-cause analysis.

Chapter 6

Data-driven Design of Multi-model Soft Sensors

This section of the thesis presents the second contribution, which focuses on extending the ordinary soft-sensor design from single-model (SMS) to incorporate multi-model soft-sensor (MMS) design. Novel approaches are proposed to enhance the current state-of-the-art MMS design. Currently, this contribution is undergoing the review process (Mojto et al., 2023a).

6.1 Problem Definition

We aim to design soft sensors that can predict a hard-to-measure process variable ($\boldsymbol{y}$) using the available dataset of easy-to-measure variables ($\boldsymbol{X}$). To compare the performance of single-model soft sensors (SMS) and multi-model soft sensors (MMS), we utilise datasets generated from the PCT model (see Section 4.5.2), as well as an industrial dataset from the VGH unit in the oil refinery Slovnaft, a.s. in Bratislava, Slovakia. The SMS design and its corresponding methodology are described in Section 4.5.1, while the state-of-the-art design of MMS is presented in Section 4.5.2.

6.2 Solution Approach

We propose a novel approach for the design of multi-model soft sensors (MMS), which addresses the limitations of the existing state-of-the-art MMS_{SotA} approach (see Section 4.5.2). Our proposed approach consists of two separate developments. The first part focuses on the design of MMS with continuous switching, referred to as MMS_{con}. This part aims to overcome the limitations of the MMS_{SotA} approach by introducing a continuous switching mechanism between different models within the MMS. The second part extends the MMS design further by incorporating optimised

data labelling, resulting in a $\text{MMS}_{\text{con,lab}}$ approach. This part addresses the challenge of effectively labelling the data used for training and switching between models within the MMS, aiming to improve the overall performance and accuracy of the soft sensor. Together, these two developments present a comprehensive and enhanced approach for the design of multi-model soft sensors, offering improved capabilities and addressing the limitations of the existing approaches.

6.2.1 Design of MMS with Continuous Switching

To deal with the limitation of discontinuity of MMS models, we propose a combination of the SVM-based classification of the data with the individual sensor training in the following optimization problem:

$$\min_{\substack{\boldsymbol{w},w_0,\boldsymbol{e}\geq 0\\ \boldsymbol{\beta}_1,\beta_{0,1},\boldsymbol{\beta}_2,\beta_{0,2}}} \quad \text{SSE}_1 + \text{SSE}_2 + \alpha\|\boldsymbol{w}\|_2^2 + \gamma\|\boldsymbol{e}\|_1 \tag{6.1a}$$

$$\text{s.t.} \quad (2z_i - 1)\left(\boldsymbol{x}_i^\intercal \boldsymbol{w} + w_0\right) \geq 1 - e_i, \quad \forall i \in \{1, 2, \ldots, n\}, \tag{6.1b}$$

$$\text{SSE}_1 = \sum_{i=1}^{n} z_i \left(y_i - \boldsymbol{x}_i^\intercal \boldsymbol{\beta}_1 - \beta_{0,1}\right)^2, \tag{6.1c}$$

$$\text{SSE}_2 = \sum_{i=1}^{n} (1 - z_i) \left(y_i - \boldsymbol{x}_i^\intercal \boldsymbol{\beta}_2 - \beta_{0,2}\right)^2, \tag{6.1d}$$

$$\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2 - \boldsymbol{w} = 0, \quad \beta_{0,1} - \beta_{0,2} - w_0 = 0, \tag{6.1e}$$

where $\boldsymbol{w}$ is a normal vector and w_0 constant off-set of the separation hyperplane, respectively, $\boldsymbol{e}$ is a vector of the slack variables, $\boldsymbol{z}$ is a vector of binary parameters that results from the data labelling procedure with $z_i = 1$ if $\boldsymbol{x}_i \in \mathcal{R}_1$ and $z_i = 0$ if $\boldsymbol{x}_i \in \mathcal{R}_2$, SSE is the sum of squared errors, α is a weighting parameter for normal vector of the separation plane and γ is a weighting parameter for vector of the slack variables.

The combination of the SVM-based classification of the data with the individual sensor training is represented by (6.1a)–(6.1d). The resulting optimization problem is extended with constraints (6.1e) which ensure the continuity at the switch between the two models. This is achieved by establishing the intersection of model surfaces to coincide with the determined switching hyperplane. Note that, we present the formulation of MMS_{con} in the simplest form (two MMS models, OLS setup), for brevity, yet it is possible to extend easily this formulation to multiple models and other training approaches (see Chapter 3).

As the a priori data labelling can be inappropriate for the design of a MMS with continuous switching, we allow small violations of the labelling using the slack variables

$\boldsymbol{e}$ in (6.1b). We also consider that the user can aim at giving up some portion of model (training) accuracy for the better separation by widening the separation band. The latter feature is established by minimizing $\|\boldsymbol{w}\|_2^2$ in (6.1a). The described features can be enforced/weakened by tuning the positive weights α and β.

Illustrative example The visualization of MMS_{con} on the *PCT* dataset can be seen in Figure 4.4b. Unlike the MMS_{SotA} approach (Figure 4.4a), there is no discrepancy at the intercept between the designed models (yellow and dark green surfaces) of MMS_{con}. This confirms that the proposed approach ensures continuous switching between designed models. The accuracy of MMS_{con} (RMSE ($\mathcal{I}_{\text{T}}$)=0.079) is significantly decreased compared to MMS_{SotA} (RMSE ($\mathcal{I}_{\text{T}}$)=0.032). This is a price to pay for MMS continuity and a design trade-off.

Naturally, the continuity constraints (6.1e) can be relaxed and introduced as soft constraints should one be willing to make the trade-off explicit for the MMS design. The accuracy and continuity of the MMS_{con} model can then be effectively tuned by varying the weights α and β according to the fidelity of a priori labelling and a desired level of discontinuity. The other way to improve the performance of the MMS_{con} approach represents an implementation of the optimization of a priori labelling into the MMS design, which is further explored in the following text.

6.2.2 Design of MMS with Optimized Data Labelling

In order to mitigate the inaccuracies caused by the a priori labelling of the training dataset, we propose the approach to design MMS with optimized data labelling ($\text{MMS}_{\text{con,lab}}$). This approach searches directly for the optimal data labelling by adding $\boldsymbol{z}$ among the optimized variables in (6.1a). The resulting optimization problem is following:

$$\min_{\substack{\boldsymbol{z}\in\{0,1\}^n,\boldsymbol{w},w_0,\boldsymbol{e}\geq 0\\ \boldsymbol{\beta}_1,\beta_{0,1},\boldsymbol{\beta}_2,\beta_{0,2}}} \text{SAE}_1 + \text{SAE}_2 + \alpha\|\boldsymbol{w}\|_1 + \gamma\|\boldsymbol{e}\|_1 \tag{6.2a}$$

$$\text{s.t.} \quad (2z_i - 1)\left(\boldsymbol{x}_i^{\intercal}\boldsymbol{w} + w_0\right) \geq 1 - e_i, \quad \forall i \in \{1, 2, \ldots, n\}, \tag{6.2b}$$

$$\text{SAE}_1 = \sum_{i=1}^{n} z_i \left|y_i - \boldsymbol{x}_i^{\intercal}\boldsymbol{\beta}_1 - \beta_{0,1}\right|, \tag{6.2c}$$

$$\text{SAE}_2 = \sum_{i=1}^{n} (1 - z_i) \left|y_i - \boldsymbol{x}_i^{\intercal}\boldsymbol{\beta}_2 - \beta_{0,2}\right|, \tag{6.2d}$$

$$\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2 - \boldsymbol{w} = 0, \quad \beta_{0,1} - \beta_{0,2} - w_0 = 0, \tag{6.2e}$$

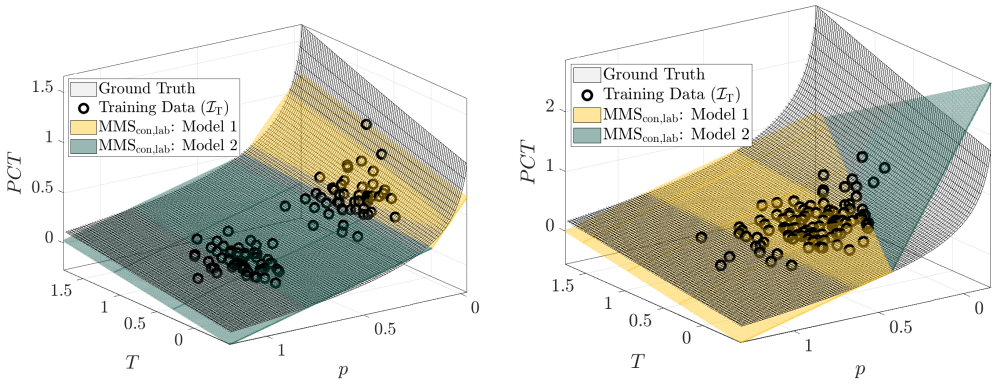
where SAE is a sum of absolute errors.

Although a formulation similar to (6.1) with SSE-based objective can be reused here, we adopt the SAE criterion to reduce the complexity. In a similar fashion, 2-norm is replaced for 1-norm to regularize the normal vector of the separating hyperplane. This is a standard approach (Song et al., 2002). The optimization problem (6.2) can thus be transformed to a mixed-integer linear program (MILP). The transformation uses: (a) the epigraph reformulation (Milano, 2012) of the absolute value, (b) the big-M method (Griva et al., 2008) to linearize the bilinear constraints. As the variables $\boldsymbol{z}$ are binary, the big-M method does not require any new integer variables. If SSE was used in the objective function, the optimization problem would turn into mixed-integer nonlinear program (MINLP), which might be challenging especially when n is high.

The problem (6.2) is primarily used to determine the data labels, which refer to how the training data is distributed and how the validity regions of the model are established. Once the values of $\boldsymbol{z}$ (data labels) are fixed, the final training of MMS models is performed by solving (6.1) using the SSE criterion. This ensures a fair comparison with other SMS and MMS approaches.

This two-step approach does not require a priori labelling of the training set and can provide an optimal MMS at the expense of increased computational burden. The optimization problem for the $\text{MMS}_{\text{con,lab}}$ design increases by one binary optimized variable per training data point. Therefore, the proposed approach is limited to relatively small-scale problems (tens to hundreds of measurements). However, this is typically sufficient for the design of soft sensors, where only a limited number of measurements is available for the desired (hard-to-measure) variable. Additionally, if a large dataset is available, a smaller size of the training dataset can be selected based on appropriate information criteria, similar to optimal design of experiments or sampling for surrogate model building Kamath (2022).

Illustrative example Figure 6.1 shows the design of $\text{MMS}_{\text{con,lab}}$ on the different *PCT* datasets. The results in Figure 6.1a show that the designed $\text{MMS}_{\text{con,lab}}$ has a high degree of flexibility and precision on the *PCT* dataset with two distinct clusters. The accuracy of this sensor (RMSE ($\mathcal{I}_{\text{T}}$) = 0.026) outperforms MMS_{SotA} (RMSE ($\mathcal{I}_{\text{T}}$) = 0.032) and MMS_{con} (RMSE ($\mathcal{I}_{\text{T}}$) = 0.079) shown in Figure 4.4. The accuracy improvement of $\text{MMS}_{\text{con,lab}}$ is ensured by optimizing a priori labeling instead of using k-means clustering. This can be indicated by comparing classification in Figure 6.1a against Figure 4.4. The results from Figure 6.1b indicate that the sensor returned by the $\text{MMS}_{\text{con,lab}}$ approach is designed effectively even when the considered



(a) MMS$_{con,lab}$ (RMSE ($\mathcal{I}_T$) = 0.023) designed on the *PCT* dataset with two distinct clusters.

(b) MMS$_{con,lab}$ (RMSE ($\mathcal{I}_T$) = 0.043) designed on the *PCT* dataset with indistinguishable clusters.

Figure 6.1: The ground truth model of *PCT* with MMS$_{con,lab}$ designed on different datasets.

dataset has no distinguishable clusters. This is typical for industrial datasets due to the presence of the significant level of noise and multitude of similar operating points.

6.3 Results

The design of single-model soft sensors (SMS) and multi-model soft sensors (MMS) is elaborated on two case studies. Both case studies have an industrial character and practical relevance. The first case study features a pressure compensated temperature *PCT*, which is briefly explored in Chapter 3. The purpose of this case study is to analyse the impact of data quality on the MMS design in multitude of simulations. The second case study involves an industrial dataset from the VGH unit, which is a part of the oil refinery Slovnaft, a.s. in Bratislava, Slovakia. This case study validates the applicability of the proposed MMS design approaches in practice.

6.3.1 Implementation Details

The presented design methods are implemented in MATLAB R2022a. To solve the involved optimization problems, we use the Yalmip package (Löfberg, 2004) and Gurobi solver (Gurobi Optimization, LLC, 2023). All the numerical results and graphical representations consider the normalization of variables within the interval $[0, 1]$ in both

case studies. The normalization (scaling) parameters are not disclosed for the dataset from the VGH unit due to data confidentiality.

Prior to the soft-sensor design, the entire available dataset is divided into the training and testing (unseen) datasets. The information contained within the training and testing datasets is one of the decisive factors directly affecting the performance of the designed soft sensors. Therefore, the effect of various ways of dividing the data into training and testing datasets on the SMS and MMS performances is further investigated on the *PCT* dataset in Section 6.3.2. The soft-sensor design on the industrial dataset from the VGH unit is based on the random distribution of measurements in the training and testing datasets.

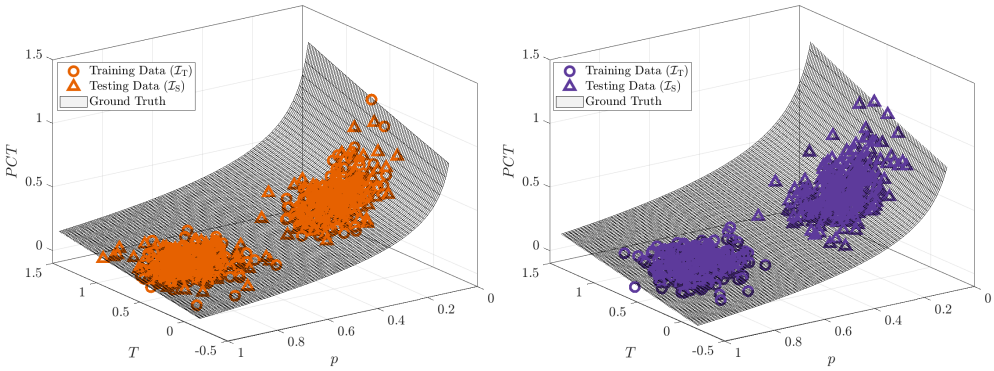
The design of MMS_{SotA} performs the a priori labelling by using k-means clustering. Subsequently, the linear classifier of MMS_{SotA} is designed by SVM. Finally, the model parameters within MMS_{SotA} models are calculated by OLS. The MMS_{con} design is performed according to (6.1). In order to reduce the computational effort, MMS_{con} is initialized by the results from (6.1) considering SAE instead of SSE within the objective function (6.1a). Subsequently, the $\text{MMS}_{\text{con,lab}}$ design from (6.2) is initialized by the results from MMS_{con}.

6.3.2 Design of Soft Sensors for Pressure-Compensated Temperature

We use the datasets generated by simulating the nonlinear model of *PCT* represented by (4.5) with respect to the parameters and specifications introduced in Chapter 3. We use this ideal case study to examine the impact of various factors on the performance of the soft sensor. Specifically, in this section, we analyze the impact of two factors on the SMS and MMS designs: (1) the method of data distribution into the training and testing datasets and (2) the noise variance in the output variable.

The considered case study involves two input variables (p and T) and one output variable (PCT), and therefore, it is unnecessary to consider the SMS approaches with advanced input structure selection (i.e., SMS_{PCA}, SMS_{PLS}, SMS_{LAS}, and SMS_{SS}). The set of compared soft sensors in this section involves SMS_{OLS}, MMS_{SotA}, MMS_{con}, and $\text{MMS}_{\text{con,lab}}$. Overall, the studied *PCT* datasets involve 620 measurements, which are equally distributed between the training and testing datasets.

To analyse the impact of the data distribution into the training and testing datasets, we generate datasets from the *PCT* model according to two different scenarios. The first scenario, or desirable scenario, considers that the *PCT* model operates in two different



(a) The desirable scenario. **(b)** The undesirable scenario.

Figure 6.2: The comparison of studied distributions of the PCT data into training and testing datasets.

operating regimes and measurements from both operating regimes are available in the training dataset, as illustrated in Figure 6.2a. This scenario occurs relatively frequently in the industry, and it assumes that the process operates only within known operating regimes, which is desired for the soft-sensor design. The second scenario, or undesirable scenario, assumes the same operating regimes within the PCT model as the first scenario, but the training dataset involves measurements from one operating regime only, and the testing dataset involves measurements from the other operating regime, as shown in Figure 6.2a. This scenario represents an undesirable, yet not unlikely, situation in the industry, when the process operates within a new operating state after the soft-sensor design.

The comparison in Figure 6.3 shows a statistical evaluation of the results obtained over 100 different datasets for each studied scenario. The datasets consider two classes of measurements, as shown in Figure 6.2, with different random distributions of measurements. The noise considered within the output variable is a random variable from $\mathcal{N}(\mu_{\text{noise}}, \sigma^2_{\text{noise}}) = \mathcal{N}(0, 5^2)$, where the value of the standard deviation σ_{noise} represents 0.67 % ($\sigma_{\text{noise}}/\mu_{PCT} \cdot 100\,\%$) of an averaged value from the original (prior to the normalization) output variable. The boxes in Figure 6.3 represent the 25^{th} and 75^{th} percentiles of the RMSE reached on the testing test. The red line within the box represents the median value of the considered set of results. The red crosses represent the statistical outliers. For each designed soft sensor (i.e., SMS_{OLS}, MMS_{SotA}, MMS_{con}, and $\text{MMS}_{\text{con,lab}}$), there are four pairs of orange (desirable scenario) and violet (undesirable scenario) boxes in Figure 6.3.

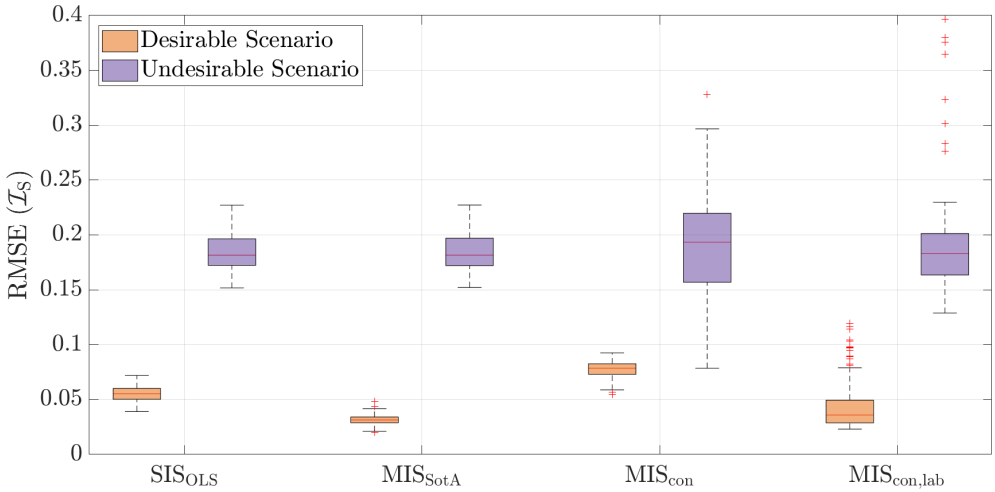


Figure 6.3: The statistical comparison of RMSE ($\mathcal{I}_S$) of designed soft sensors (SMS_{OLS}, MMS_{SotA}, MMS_{con} and $MMS_{con,lab}$) involving 100 different datasets for each studied scenario.

The results presented in Figure 6.3 demonstrate that the designed soft sensors exhibit better performance on datasets from the desirable scenario (represented by the orange boxes) compared to those from the undesirable scenario (represented by the violet boxes). These findings support our initial assumption regarding the impact of these scenarios on the performance of soft sensors.

The results from the desirable scenario further indicate a high degree of robustness (or low variance) in the SMS_{OLS} accuracy, as seen by the small height of the corresponding orange box in Figure 6.3. On the other hand, the accuracy of SMS_{OLS} is significantly outperformed by MMS_{SotA}. Furthermore, the performance of MMS_{SotA} is even smaller compared to that of SMS_{OLS}. The outstanding performance of MMS_{SotA} can be attributed to the nature of the datasets in the desirable scenario (as shown in Figure 6.2a) with distinguishable classes. One class involves measurements that precisely explain the behaviour of *PCT* in the (almost) linear section, while the other class involves mostly measurements from the highly nonlinear section of the *PCT* model range. Therefore, a priori labelling within the MMS_{SotA} approach (k-means clustering) provides appropriate data labels for the subsequent design of soft sensors. The results of MMS_{con} in the desirable scenario (as shown in the corresponding orange box in Figure 6.3) indicate a small variance in accuracy comparable to that of SMS_{OLS}. However, MMS_{con} achieves the lowest accuracy compared to other designed soft sensors on the datasets from the desirable scenario. This suggest that the a priori labelling is

not optimal w.r.t. design of a continuous multi-model sensor. Further analysis of the results reveals that the MMS_{con} accuracy (considering the desirable scenario) can be significantly improved by the $\text{MMS}_{\text{con,lab}}$. The results in the desirable scenario exhibit higher variability in the performance compared to all other designed soft sensors. The main reason for this is the complexity of the optimization problem that needs to be solved, which can lead to numerical inaccuracies. This assumption is supported by the increased occurrence of outliers in the $\text{MMS}_{\text{con,lab}}$ results in Figure 6.3 (represented by the red crosses). Despite the increased variance of the accuracy, $\text{MMS}_{\text{con,lab}}$ achieves comparable accuracy to MMS_{SotA} in the majority of the cases. This is notable given the similar values of the median (represented by the red lines) of these MMSs.

The results from the undesirable scenario (violet boxes) suggest that SMS_{OLS} and MMS_{SotA} perform similarly. Both soft sensors exhibit the lowest variance of accuracy (as indicated by the height of the corresponding violet boxes, similar to what we observed in the desirable scenario. The accuracy of MMS_{SotA} appears only slightly higher than that of SMS_{OLS}, unlike the observations from the desirable scenario. All the sensors pay the price for making extrapolated predictions. We can indicate very high variability in the MMS_{con} accuracy for the undesirable scenario. This stems from the requirement to design models with continuous switching, which determines the rotation and angle between the models. The nature of the undesirable scenario, where the testing dataset is completely unseen during training, gives MMS_{con} a chance (through the randomness of noise and k-means clustering) to fit the testing dataset significantly better or worse than other studied soft sensors. The results suggest that the variance and accuracy of MMS_{con} can be improved by using the $\text{MMS}_{\text{con,lab}}$ approach in the undesirable scenario. This improvement is achieved by optimizing the data labelling within the $\text{MMS}_{\text{con,lab}}$ approach, with a focus on explaining the nonlinear behaviour of the PCT model. However, the possibilities of $\text{MMS}_{\text{con,lab}}$ are limited due to the nature of the training dataset in this scenario. As a result, we observe similar accuracy of $\text{MMS}_{\text{con,lab}}$ compared to SMS_{OLS} and MMS_{SotA}, as indicated by the median values (red lines) within the corresponding violet boxes in Figure 6.3. We also note the occurrence of low-accuracy outliers within the $\text{MMS}_{\text{con,lab}}$ results, similar to the desirable scenario.

The previous analysis shows that the random distribution of data into the training and testing datasets should provide sufficient informative content for the training dataset. Therefore, we use this distribution in the following analysis focused on the impact of noise (of the output variable) on the performance of the designed soft sensors. The set of noise variances ($\sigma_{\text{noise}} \in \{0.1, 0.25, 2.5, 5, 10, 25, 50\}$) is selected based on the noise variances observed in the industrial dataset. The minimum, respectively maximum, considered noise variance ($\sigma_{\text{noise}} = 0.1$, respectively $\sigma_{\text{noise}} = 50$) represent

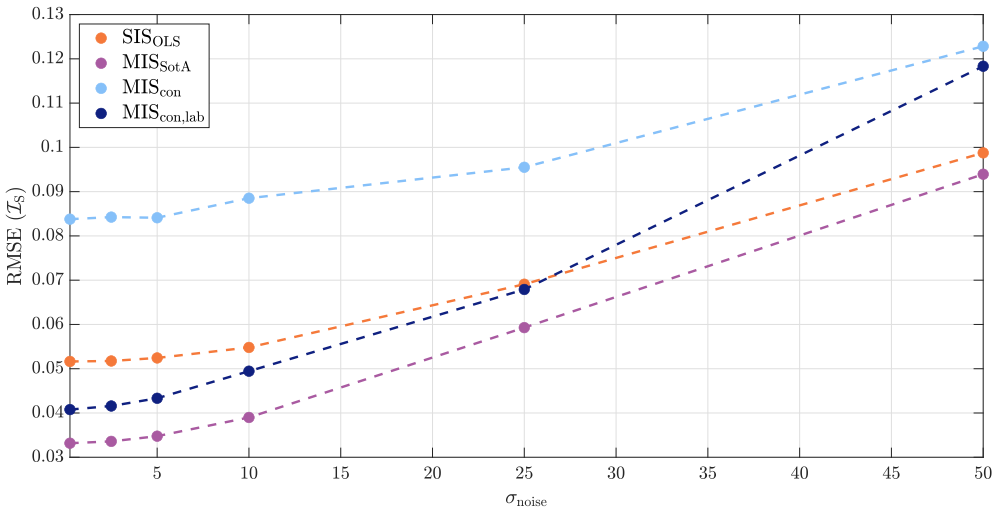


Figure 6.4: The averaged RMSE ($\mathcal{I}_{\mathrm{S}}$) value for designed soft sensors (SMS$_{\mathrm{OLS}}$, MMS$_{\mathrm{SotA}}$, MMS$_{\mathrm{con}}$, and MMS$_{\mathrm{con,lab}}$) from 100 different realizations of the noise for each studied σ_{noise} within the output variable.

approximately 0.1 %, respectively 6.7 %, of the mean value of the output variable from the considered datasets. To provide more representative results, we present the average RMSE ($\mathcal{I}_{\mathrm{S}}$) value from 100 different realizations of noise for each studied σ_{noise}.

The impact of the noise variance (within the output variable) on the accuracy (RMSE ($\mathcal{I}_{\mathrm{S}}$)) of the designed soft sensors is illustrated in Figure 6.4. In general, an increase in σ_{noise} leads expectedly to an increase in RMSE ($\mathcal{I}_{\mathrm{S}}$). The performance of SMS$_{\mathrm{OLS}}$ (orange points) appears to be relatively robust for smaller values of the noise variances ($\sigma_{\mathrm{noise}} < 10$), confirming conclusions from the previous analysis (see Figure 6.3). Moreover, the slope of SMS$_{\mathrm{OLS}}$ accuracy decrease within the interval $\sigma_{\mathrm{noise}} = (10, 50)$ is the smallest among the studied approaches, which further confirms good robustness properties of SMS$_{\mathrm{OLS}}$.

SMS$_{\mathrm{OLS}}$ is outperformed by MMS$_{\mathrm{SotA}}$ (magenta points in Figure 6.4) over the entire range of studied noise variances. The superior performance of MMS$_{\mathrm{SotA}}$ is expected due to the nature of the considered datasets (see Figure 6.2a) and nonlinearity of the ground-truth model, as discussed above.

The accuracy of MMS$_{\mathrm{con}}$ (represented by pale blue points) is lower compared to the other designed soft sensors, and this can be attributed to two main aspects. The first

aspect is the constraint to design models with continuous switching. The second aspect is that the objective function of the $\mathrm{MMS_{con}}$ approach (as shown in Equation (6.1a)) takes into account not only the model accuracy, represented by the sum of squared errors (SSE) of the designed models, but also the penalization of the separation hyperplane width and the penalization of the separation hyperplane violations.

The results of $\mathrm{MMS_{con,lab}}$ (blue points) indicate that the accuracy of $\mathrm{MMS_{con}}$ can be significantly improved by optimizing the data labelling. $\mathrm{MMS_{con,lab}}$ outperforms $\mathrm{SMS_{OLS}}$ when $\sigma_{\mathrm{noise}} < 29$. The accuracy of $\mathrm{MMS_{con,lab}}$ decreases steeper compared to other approaches when $\sigma_{\mathrm{noise}} > 29$. The increased flexibility of $\mathrm{MMS_{con,lab}}$ leads to an increased tendency for this approach to explain the noise within the training dataset, especially for significant noise variances. In other words, the $\mathrm{MMS_{con,lab}}$ approach has an increased tendency for overfitting.

6.3.3 Design of Soft Sensors for Vacuum Gasoil Hydrogenation Unit

The details and description of the VGH unit is stated in Section 5.1.2. The available industrial dataset involves measurements for 24 months of the VGH unit operation. The output (desired) variable to be estimated by soft sensors is the purity of the HGO product, represented by 95 % point of the distillation curve $T_{95\,\%,\mathrm{HGO}}$. The lab analysis of the HGO product is executed approximately once per day, and therefore, there are 621 measurements of the output variable available for the soft-sensor design. The input variables are measured every minute by online sensors. In order to reduce the impact of the measurement noise, the minute measurements are replaced by the averaged measurements from 30-minute intervals. The resulting input dataset involves 27,324 measurements available for the soft-sensor design.

The set of input candidates involves the following 35 variables:

$$\begin{aligned}\boldsymbol{x} = (\, & PCT_{\mathrm{HGO}}, PCT_{\mathrm{GF}}, T_{\mathrm{ex},1}, T_{\mathrm{ex},2}, T_{\mathrm{ex},3}, T_{\mathrm{ex},4}, \\ & T_{\mathrm{wabt},1}, T_{\mathrm{wabt},2}, T_{\mathrm{wabt},3}, T_{\mathrm{wabt},4}, T_{\mathrm{wabt},5}, \\ & RX_1, RX_2, RX_3, RX_4, RX_5, RX_6, RX_7, \\ & x_{\mathrm{H2}}, T_{\mathrm{frac},1}, T_{\mathrm{frac},2}, F_{\mathrm{frac,heat}}, p_{\mathrm{frac}}, \\ & F_{\mathrm{f,rec}}, F_{\mathrm{f}}, x_{\mathrm{f,N2}}, x_{\mathrm{f,S}}, T_{\mathrm{f,5p}}, T_{\mathrm{f,50p}}, T_{\mathrm{f,95p}}, \\ & F_{\mathrm{r}}, L_{\mathrm{reb}}, \mathrm{vo_r}, \mathrm{vo_{reb,heat,1}}, \mathrm{vo_{reb,heat,2}}\,)^{\mathsf{T}}\,,\end{aligned} \tag{6.3}$$

where F_{r} is a flowrate of the reflux stream L_{reb} is a liquid level in the reboiler, $\mathrm{vo_r}$ is a valve opening of the reflux stream, $\mathrm{vo_{reb,heat,1–2}}$ is a valve opening of the heating

medium for the reboiler (1 – input, 2 – output). The aforementioned five input variables were added to the original VGH dataset, as described in Section 5.1.2. This extension was proposed by our industrial partner as a means to enhance the performance of the soft sensors for the VGH unit. Furthermore, all five variables are directly associated with the product fractionator depicted in Figure 5.2, which is where the soft sensor design is conducted.

The supervised learning methods used for soft-sensor design require a paired input-output dataset, where the inputs and outputs correspond to the same measurement time. In this case, the input and output datasets consist of 621 measurements, which are subsequently divided randomly into training (311 measurements) and testing (310 measurements) sets. It is worth noting that this dataset is comparable in size to the PCT sets of data (620 measurements) used in the previous case study (Section 6.3.2). This allows us to explore any similarities between the results and conclusions of these case studies.

Currently, the refinery has implemented an SMS with the structure represented by (5.4). This structure is based on the expert knowledge of the operators and engineers in the refinery and is used as a reference in the comparison of designed soft sensors in this study. Therefore, we refer to this sensor as SMS_{Ref}. The OLS approach is used to evaluate the parameters in (5.4).

The set of approaches considered for SMS design (see Chapter 3) includes OLS (SMS_{OLS}), PCA (SMS_{PCA}), PLS (SMS_{PLS}), LASSO (SMS_{LAS}), and SS (SMS_{SS}). These approaches consider the entire set of input candidates ($n_{\text{p}} = 35$) and search for the optimal input structure (n_{p}^*) based on their specific objectives. The set of compared approaches for MMS design consists of MMS_{SotA}, MMS_{con}, and $\text{MMS}_{\text{con,lab}}$. The MMS approaches consider two candidate input structures: (1) the reference input structure (denoted as Ref) given by (5.4) with $n_{\text{p}} = 1$ (PCT_{HGO}) and (2) the input structure determined by the brute force approach (denoted as BF) with $n_{\text{p}} = 2$ (PCT_{HGO} and $\text{vo}_{\text{h},2}$). The brute force approach solves the MMS_{con} approach for each studied input structure, and the best input structure is determined based on the RMSE ($\mathcal{I}_{\text{T}}$) criterion. This approach explores all possible combinations of available candidate input variables within a linear structure with one or two variables. However, we do not explore more complex input structures using the brute force approach for three reasons. First, expert knowledge from the refinery suggests to consider the simple input structures (see reference structure in (5.4)). Second, the $\text{MMS}_{\text{con,lab}}$ approach tends to overfit, as indicated in Section 6.3.2. Third, the refinery demands that soft sensors have as simple an input structure as possible because each variable in the input structure requires measurements from a corresponding online sensor in the refinery.

Table 6.1: Comparison of the resulting number of input variables n_{p}^{*} / principal components n_{pc}^{*} (for PCA and PLS), accuracy (RMSE), accuracy with bias correction ($\mathrm{RMSE_{BC}}$) and effort of the bias correction ($\mathrm{E_{BC}}$) of designed single-model soft sensors (SMS) on the training ($\mathcal{I}_{\mathrm{T}}$) and testing ($\mathcal{I}_{\mathrm{S}}$) industrial datasets.

	$\mathrm{SMS_{OLS}}$	$\mathrm{SMS_{PCA}}$	$\mathrm{SMS_{PLS}}$	$\mathrm{SMS_{LAS}}$	$\mathrm{SMS_{SS}}$	$\mathrm{SMS_{Ref}}$	$\mathrm{SMS_{BF}}$
$n_{\mathrm{p}}^{*}/n_{\mathrm{pc}}^{*}$	31/-	35/6	33/6	13/-	17/-	1/-	2/-
RMSE ($\mathcal{I}_{\mathrm{T}}$)	0.077	0.109	0.084	0.083	0.08	0.111	0.104
RMSE ($\mathcal{I}_{\mathrm{S}}$)	0.145	0.105	0.097	0.097	0.211	0.1	0.095
$\mathrm{RMSE_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$)	0.135	0.085	0.081	0.083	0.198	0.081	0.079
$\mathrm{E_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$) [%]	50	48.71	47.1	48.39	50.32	48.06	48.71

To provide a fair comparison, the SMS set of approaches is extended to $\mathrm{SMS_{Ref}}$ and $\mathrm{SMS_{BF}}$, considering the same input structures as the MMS approaches.

Over its life cycle, the performance of the designed soft sensor can be further improved by using a well-known bias correction (or bias update) approach (see Section 4.7.2). The bias correction aims to improve the accuracy of the succeeding prediction of the soft sensor by adjusting the constant (bias) term β_0. The bias term change is evaluated when a new measurement from lab analysis is available. Taking into account the normalized values of the output variable, the considered value of the multiplier K_{BC} is 0.2, and the value of minimum accessible increment $\Delta\beta_{0,\mathrm{min}}$ is 0.01.

The resulting performance criteria of the studied single-model soft sensors (i.e., $\mathrm{SMS_{OLS}}$, $\mathrm{SMS_{PCA}}$, $\mathrm{SMS_{PLS}}$, $\mathrm{SMS_{LAS}}$, $\mathrm{SMS_{SS}}$, $\mathrm{SMS_{Ref}}$, and $\mathrm{SMS_{BF}}$) designed on the VGH dataset are shown in Table 6.1. The complexity of the soft sensors is represented by the resulting number of input variables n_{p}^{*} and the number of principal components n_{pc}^{*} (for $\mathrm{SMS_{PCA}}$ and $\mathrm{SMS_{PLS}}$). The results also involve the accuracy of soft sensors corrected by the bias correction ($\mathrm{RMSE_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$)) and the effort of the bias correction $\mathrm{E_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$) evaluated on the testing dataset.

The results presented in Table 6.1 indicate that $\mathrm{SMS_{OLS}}$ achieves the lowest RMSE ($\mathcal{I}_{\mathrm{T}}$) value among all studied SMSs, but it produces a relatively high RMSE ($\mathcal{I}_{\mathrm{S}}$) value and a complex input structure ($n_{\mathrm{p}}^{*} = 31$), which suggests overfitting. The variance-covariance approaches, $\mathrm{SMS_{PCA}}$ and $\mathrm{SMS_{PLS}}$, indicate the same input structure complexity ($n_{\mathrm{pc}}^{*} = 6$) but $\mathrm{SMS_{PLS}}$ performs better than $\mathrm{SMS_{PCA}}$ due to its supervised-learning nature. $\mathrm{SMS_{LAS}}$ shows accuracy similar to $\mathrm{SMS_{PLS}}$, but with a lower complexity ($n_{\mathrm{p}}^{*} = 13$) than $\mathrm{SMS_{PLS}}$ ($n_{\mathrm{p}}^{*} = 33$). $\mathrm{SMS_{SS}}$ achieves the highest RMSE ($\mathcal{I}_{\mathrm{S}}$) value

Table 6.2: Comparison of the resulting number of input variables n_{p}^{*}, accuracy (RMSE), accuracy with bias correction ($\mathrm{RMSE_{BC}}$) and effort of the bias correction $\mathrm{E_{BC}}$ of designed multi-model soft sensors (MMS) on the training ($\mathcal{I}_{\mathrm{T}}$) and testing ($\mathcal{I}_{\mathrm{S}}$) industrial datasets.

	$\mathrm{MMS_{SotA}}$		$\mathrm{MMS_{con}}$		$\mathrm{MMS_{con,lab}}$	
n_{p}^{*}	1 (Ref)	2 (BF)	1 (Ref)	2 (BF)	1 (Ref)	2 (BF)
RMSE ($\mathcal{I}_{\mathrm{T}}$)	0.098	0.097	0.109	0.130	0.108	0.1
RMSE ($\mathcal{I}_{\mathrm{S}}$)	0.094	0.092	0.113	0.133	0.105	0.098
$\mathrm{RMSE_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$)	0.082	0.078	0.092	0.090	0.087	0.081
$\mathrm{E_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$) [%]	50.00	45.81	54.52	55.81	50.32	49.36

among all studied SMSs, and it exhibits the same signs of overfitting as $\mathrm{SMS_{OLS}}$. $\mathrm{SMS_{Ref}}$ ($n_{\mathrm{p}}^{*} = 1$) and $\mathrm{SMS_{BF}}$ ($n_{\mathrm{p}}^{*} = 2$) have significantly lower complexities compared to other designed SMSs. Based on the results, it is possible to improve the performance of $\mathrm{SMS_{Ref}}$ (currently implemented in the refinery) by about 5 % using $\mathrm{SMS_{BF}}$. However, it would be required to maintain one extra online sensor (compared to $\mathrm{SMS_{Ref}}$), which is already installed in the refinery, to ensure the accuracy and reliability of $\mathrm{SMS_{BF}}$.

The results presented in Table 6.1 suggest that bias correction has the potential to improve the accuracy of the studied SMSs (RMSE ($\mathcal{I}_{\mathrm{S}}$) $\rightarrow$ $\mathrm{RMSE_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$)), at the cost of increased bias correction effort ($\mathrm{E_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$)). However, the bias correction is not effective in improving the accuracy of $\mathrm{SMS_{OLS}}$ compared to other SMSs, despite the high effort involved. This is likely due to the overfitted structure of $\mathrm{SMS_{OLS}}$. The results show that bias correction with $\mathrm{SMS_{PCA}}$ achieves similar performance to that of $\mathrm{SMS_{PLS}}$ and $\mathrm{SMS_{LAS}}$. Each of these soft sensors achieves lower $\mathrm{RMSE_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$) and $\mathrm{E_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$) than $\mathrm{SMS_{OLS}}$. It seems that $\mathrm{SMS_{PLS}}$ achieves the lowest $\mathrm{E_{BC}}$ ($\mathcal{I}_{\mathrm{S}}$) from all studied SMSs. The bias correction of $\mathrm{SMS_{SS}}$ suffers from overfitting, as $\mathrm{SMS_{OLS}}$, leading to the poorest accuracy after the bias correction, despite the highest effort required among all studied SMSs. The results also indicate that $\mathrm{SMS_{Ref}}$ and $\mathrm{SMS_{BF}}$ perform similarly to $\mathrm{SMS_{PCA}}$, $\mathrm{SMS_{PLS}}$, and $\mathrm{SMS_{LAS}}$ under bias correction. The highest accuracy of a bias-corrected soft sensor is achieved in the case of $\mathrm{SMS_{BF}}$.

Table 6.2 presents a comparison of the performance of MMSs ($\mathrm{MMS_{SotA}}$, $\mathrm{MMS_{con}}$, and $\mathrm{MMS_{con,lab}}$) designed for the VGH dataset. Each of these approaches shows the resulting quality of the designed soft sensors using both input structures (Ref and BF), and their accuracy is evaluated according to the same criteria as in the case of SMSs (Table 6.1). This enables a direct comparison of the results from the studied SMS and MMS approaches.

Table 6.2 suggests that MMS_{SotA} achieves the highest accuracy (RMSE ($\mathcal{I}_{\text{S}}$)) compared to other designed MMSs and SMSs, taking into account both input structures. These results confirm the excellent accuracy of the MMS_{SotA} approach on the *PCT* datasets (Section 6.3.2). Moreover, the accuracy of MMS_{SotA} is higher with the BF input structure than with the Ref input structure, suggesting that the BF input structure is more appropriate for the MMS design. The MMS with the Ref input outperforms the currently implemented soft sensor in the refinery (see SMS_{Ref} in Table 6.1) by about 6 % and with the BF input structure by about 8 %.

The accuracy of MMS_{con} appears worse compared to MMS_{SotA} for both input structures. This aligns with observations made on the *PCT* datasets (Section 6.3.2). The poor accuracy of MMS_{con} is primarily caused by the requirement to design models with continuous switching. The table suggests that the accuracy of MMS_{con} is decreased with the BF input structure compared to its performance with the Ref input structure. The additional variable within the BF input structure appears to be unhelpful for MMS_{con} accuracy, and it further increases the negative impact of the model continuity constraint on the MMS_{con} performance.

The results further indicate that the optimized data labeling within the $\text{MMS}_{\text{con,lab}}$ approach significantly improves its accuracy, considering both input structures. The accuracy of $\text{MMS}_{\text{con,lab}}$ is not as high as that of MMS_{SotA} but is comparable to that of SMS_{Ref} and SMS_{BF} in Table 6.1. Moreover, the $\text{MMS}_{\text{con,lab}}$ approach ensures continuous switching of the designed models, which can be crucial in specific applications, particularly if the soft sensor is part of a process control strategy. $\text{MMS}_{\text{con,lab}}$ achieves higher accuracy with the BF input structure than with the Ref input structure. Unlike the MMS_{con} approach, the optimized data labelling enables the $\text{MMS}_{\text{con,lab}}$ approach to effectively use the additional variable within the BF input structure with respect to the resulting accuracy of $\text{MMS}_{\text{con,lab}}$.

The results of the bias correction analysis suggest that the accuracy of the studied MMSs on the testing datasets can be further improved (RMSE ($\mathcal{I}_{\text{S}}$) $\rightarrow$ RMSE_{BC} ($\mathcal{I}_{\text{S}}$)). When comparing the bias-corrected MMSs with the SMSs (see Table 6.1), we can observe that the accuracy of the soft sensors is similar (RMSE_{BC} ($\mathcal{I}_{\text{S}}$) falls in the range 0.078–0.087). However, it seems that the bias correction effort (E_{BC} ($\mathcal{I}_{\text{S}}$)) required for MMSs is generally higher than that for SMSs. This increased effort is likely due to the multiple models within the MMS structure, especially if there is a significant discrepancy (discontinuity) between the MMS models. The bias correction of MMS_{SotA} with the Ref input structure supports this observation, as it achieves a slightly higher RMSE_{BC} ($\mathcal{I}_{\text{S}}$) and a higher E_{BC} ($\mathcal{I}_{\text{S}}$) compared to SMS_{Ref} (see Table 6.1). On the other hand, we can see that MMS_{SotA} with the BF input structure achieves a lower

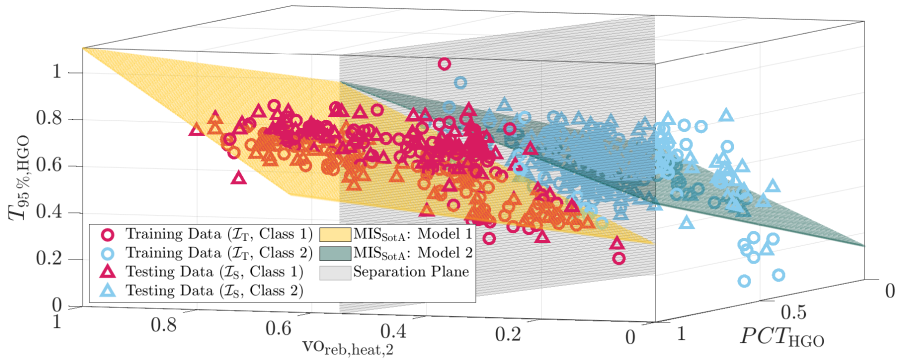
RMSE_{BC} ($\mathcal{I}_{\text{S}}$) and a lower E_{BC} ($\mathcal{I}_{\text{S}}$) than SMS_{BF} (see Table 6.1). This highlights that the choice of input structure for MMS can also significantly impact the effectiveness of the bias correction process.

The previous discussion indicates that MMS_{con} exhibits lower accuracy (RMSE ($\mathcal{I}_{\text{S}}$)) than the other MMS approaches due to the need to design models with continuous switching. The results in Table 6.2 indicate that bias correction is intended to improve the accuracy of MMS_{con} at the cost of higher E_{BC} ($\mathcal{I}_{\text{S}}$) than other MMS designs. However, despite these efforts, the bias-corrected accuracy (RMSE_{BC} ($\mathcal{I}_{\text{S}}$)) of MMS_{con} remains lower than that of other MMS approaches.

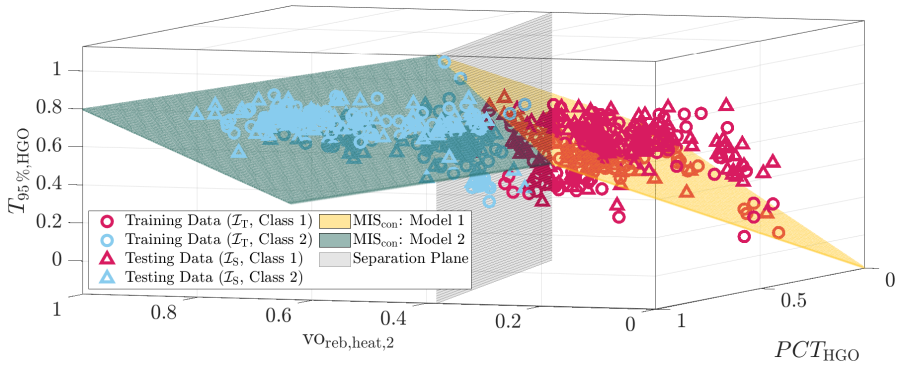
Based on the results presented in Table 6.2, it appears that the bias correction of $\text{MMS}_{\text{con,lab}}$ sensors is comparable to that of MMS_{SotA}. However, after the bias correction, the accuracy of $\text{MMS}_{\text{con,lab}}$ is slightly worse than that of MMS_{SotA} for both input structures. This is consistent with the accuracy of these soft sensors without bias correction, as shown in RMSE ($\mathcal{I}_{\text{S}}$) in Table 6.2. It is worth noting that the bias correction of the $\text{MMS}_{\text{con,lab}}$ sensors requires more effort compared to the MMS_{SotA}. This effort originates from the need to mitigates the impact of the additional requirement (to design models with continuous switching) on the accuracy of $\text{MMS}_{\text{con,lab}}$. Consequently, the corrected accuracy of $\text{MMS}_{\text{con,lab}}$ can compete with that of the most accurate soft sensors (i.e., MMS_{SotA}, SMS_{BF}, SMS_{PLS}, and SMS_{Ref}).

The designed MMSs considering the BF input structure ($n_{\text{p}}^{*} = 2$) are illustrated in Figure 6.5. The yellow and dark green surfaces represent the designed models within the MMS structure, and the gray vertical surface is the separation hyperplane. The circles represent the training dataset, and the triangles form the testing dataset. The pink colour of circles or triangles represents measurements from the first class, and the blue colour of circles or triangles indicates measurements from the second class.

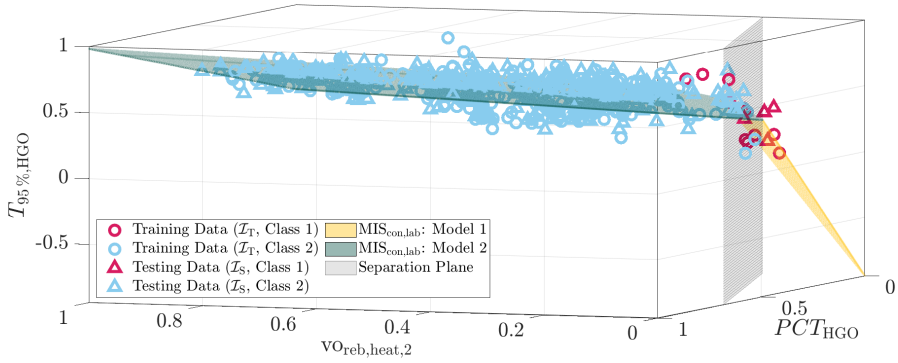
The models of MMS_{SotA} are shown in Figure 6.5a. The model surfaces exhibit an obvious discontinuity at the interface (separation plane) between models. The discontinuity appears to be beneficial for model accuracy. The MMS_{con} models are shown in Figure 6.5b. We can see that the designed models have continuous switching, which does not allow the arbitrary rotation of the MMS_{con} model surfaces. Therefore, the models of MMS_{con} deviate more from the measurements more than the models of MMS_{SotA}, resulting in lower accuracy of the MMS_{con} model compared to MMS_{SotA}. The designed models of $\text{MMS}_{\text{con,lab}}$ are shown in Figure 6.5c. We can observe the continuity at the switch between the $\text{MMS}_{\text{con,lab}}$ models. We can also see that one data class involves the majority of the measurements, while the remaining points are assigned to another, smaller class. This occurs when there are no discernible classes of



(a) $\mathrm{MMS}_{\mathrm{SotA}}$: model 1 (RMSE $= [0.091(\mathcal{I}_{\mathrm{T}}), 0.091(\mathcal{I}_{\mathrm{S}})]$), model 2 (RMSE $= [0.102(\mathcal{I}_{\mathrm{T}}), 0.094(\mathcal{I}_{\mathrm{S}})]$).



(b) $\mathrm{MMS}_{\mathrm{con}}$: model 1 (RMSE $= [0.136(\mathcal{I}_{\mathrm{T}}), 0.134(\mathcal{I}_{\mathrm{S}})]$), model 2 (RMSE $= [0.125(\mathcal{I}_{\mathrm{T}}), 0.133(\mathcal{I}_{\mathrm{S}})]$).



(c) $\mathrm{MMS}_{\mathrm{con,lab}}$: model 1 (RMSE $= [0.130(\mathcal{I}_{\mathrm{T}}), 0.211(\mathcal{I}_{\mathrm{S}})]$), model 2 (RMSE $= [0.099(\mathcal{I}_{\mathrm{T}}), 0.095(\mathcal{I}_{\mathrm{S}})]$).

Figure 6.5: The comparison of designed MMSs on the industrial dataset from the VGH unit.

measurements in the provided dataset (as seen in Figures 6.1b and 6.5c), although this is not always the case (as seen in Figures 6.1a). In the case of indistinct classes of measurements in the available dataset, $\text{MMS}_{\text{con,lab}}$ attempts to improve the accuracy of the model designed on the majority of measurements by assigning the most deviated measurements to the smaller class. As shown in Figure 6.1b, the measurements from the smaller class can explain the nonlinear nature of the estimated variable.

6.4 Discussion

The MMS design on the industrial dataset (see in Section 6.3.3) considered only the reference input structure given by (5.4) and the enhanced input structure determined by the brute force approach exploring all possible linear structures with one or two input variables (i.e., 630 structures). In the case of the VGH unit, the simple input structure seems to be desired, which is confirmed by the excellent performance of SMS_{Ref} considering only one input variable. However, the used brute force approach would cause significant computational load if a more complex input structure were demanded. In such a situation, it is possible to extend the objective function of MMS by adding an appropriate penalization element. The purpose of this penalization element is to reduce the value of the model parameters. This kind of penalization element is considered, for example, in the LASSO (ℓ_1-norm) or ridge (ℓ_2-norm) approaches. This enhanced form of the MMS design would directly provide the optimal input structure for the designed models and separation planes. Another possible approach to finding a suitable input structure for the MMS design is to combine cross-validation with some feature selection approach taking into account the objectives of the MMS design.

In this contribution, we assume that the MMS structure consists of two models. From an industrial perspective, this seems to be a reasonable assumption considering that most of the processes work within a specific operating regime. Therefore, it is expected that even a single model (SMS) should be efficient enough in such situations. The soft sensor consisting of two models should provide at least the same accuracy as the SMS, but additionally, it has the potential to reduce the number of input variables (i.e., required online sensors for the operation of the soft sensor). The presented methodology for the MMS design can be easily extended to involve more than two models in the soft-sensor structure. We have already designed MMS, considering more than two models for several simulation cases. It seems that the increasing number of designed models can further increase the accuracy of MMS. On the other hand, this type of MMS has higher requirements for the quantity and quality of data compared to the MMS with two models.

The presented case studies of the soft-sensor design provide several important insights about how to choose the appropriate MMS approach in a particular situation. The design of $\mathrm{MMS}_{\mathrm{SotA}}$ should be performed if the studied process requires a soft sensor with high accuracy, reliable knowledge about different operating regimes is provided, and discontinuous switching between the models cannot cause any (e.g., stability) issues within the considered process. If all previous specifications remain the same but continuous switching between the designed models is necessary, then $\mathrm{MMS}_{\mathrm{con}}$ should be designed. The results indicated that the continuous switching of the models is provided at the expense of the soft-sensor accuracy. In the case that reliable knowledge about different operating regimes within the process is not provided, then $\mathrm{MMS}_{\mathrm{con,lab}}$ represents the best option. In the case that the previous specification remains but the continuous switching between the models is not necessary, it is possible to solve problem (6.2) with a relaxation of continuity constrains (6.2e).

We showcased bias correction methodology as a part of the analysis of the soft-sensor design on the industrial dataset. The bias correction can significantly increase the accuracy of the soft-sensor estimates, especially when sudden changes in the process operation conditions occur. We expected that the MMS structure with two models would require more effort for the bias correction E_{BC} compared to the SMS structure. Nevertheless, the comparison of the results in Tables 6.1 and 6.2 suggests only a slightly increased effort for MMSs (except $\mathrm{MMS}_{\mathrm{SotA}}$) compared to SMSs, taking into account that $\mathrm{SMS}_{\mathrm{OLS}}$ and $\mathrm{SMS}_{\mathrm{SS}}$ are overfitted. It appears that the simplicity of the model structure is a key aspect of obtaining an effective combination of MMS with bias correction because both of them are sensitive to overfitting. The comparison of the bias correction for designed MMS (see in Table 6.2) indicates the highest accuracy yet lowest effort in the case of $\mathrm{MMS}_{\mathrm{SotA}}$. Although the performance of $\mathrm{MMS}_{\mathrm{SotA}}$ seems excellent, we believe that any (relatively) accurate SMS or MMS with continuous switching between models ($\mathrm{MMS}_{\mathrm{con}}$ and $\mathrm{MMS}_{\mathrm{con,lab}}$) should achieve less frequent occurrence of the bias correction compared to $\mathrm{MMS}_{\mathrm{SotA}}$. In our case, there is a relatively small discrepancy between the models of $\mathrm{MMS}_{\mathrm{SotA}}$ (see in Figure 6.5a) on the interface, which can be the reason for the low value of E_{BC}.

Chapter 7

Data-Driven Indication of Flooding in an Industrial Debutanizer Column

This chapter presents the results of the third contribution of the thesis, which focuses on the application of soft sensors in the area of fault detection. The objective is to design an effective indication system for detecting flooding within an industrial distillation column. These results have undergone review and have been accepted for publication (Mojto et al., 2023b).

7.1 Problem Definition

The problem of flooding indication (area of fault detection) can be framed as a binary classification task (see Section 2.3.5). Our objective is to design an indicator $\mathbf{I}$ that assigns a categorical label $\hat{y}$ based on the output of the classification model (classifier) $f(\boldsymbol{x}_{\text{sel}})$, defined as follows:

$$\hat{y} = \begin{cases} +1 \text{ (flooding)}, & \text{if } f(\boldsymbol{x}_{\text{sel}}) \geq 0, \\ -1 \text{ (normal operation)}, & \text{if } f(\boldsymbol{x}_{\text{sel}}) < 0, \end{cases} \tag{7.1}$$

where $\boldsymbol{x}_{\text{sel}} \in \mathbb{R}^{n_{\text{P}}^*}$ represents a subset (sparse representation, $n_{\text{p}} \geq n_{\text{p}}^*$) of online plant measurements $\boldsymbol{x} \in \mathbb{R}^{n_{\text{P}}}$ at a specific time instant.

In this contribution, we focus on a linear classifier represented by the mathematical form given in Equation (2.2).

7.1.1 Industrial Debutanizer Column

We study a debutanizer (distillation) column that is a part of the FCC unit of the refinery Slovnaft, a.s. in Bratislava, Slovakia. The column separates the C4/C5 fraction

into the C4-fraction-rich distillate product and the C5-fraction-rich bottom product. The column contains 40 trays.

The available dataset involves the measurements from January 2019 to April 2021 (28 months). The input variables are recorded every minute by online sensors, yet their 30-minute moving average values are considered in this study. Overall, the dataset involves 34,297 measurements. The measurements from two plant shutdowns (May – July 2019 and December 2020) are excluded.

The following 41 input variables are directly measured (by online sensors) at the column:

$$\begin{aligned}\boldsymbol{x} = (\, & vo_{\mathrm{B}}, vo_{\mathrm{D,1–3}}, vo_{\mathrm{R}}, vo_{\mathrm{reb,h}}, T_{\mathrm{col,1–5}}, T_{\mathrm{B,1–2}}, T_{\mathrm{D,1–3}}, T_{\mathrm{F}}, T_{\mathrm{reb,h,1–2}}, Q_{\mathrm{con}}, \\ & p_{\mathrm{col}}, p_{\mathrm{D,1–4}}, p_{\mathrm{df,col}}, p_{\mathrm{con}}, F_{\mathrm{R}}, F_{\mathrm{B}}, F_{\mathrm{D,1–4}}, F_{\mathrm{F}}, F_{\mathrm{reb,h,1–2}}, L_{\mathrm{reb,1–2}}, L_{\mathrm{con,1–3}} \,) ,\end{aligned} \tag{7.2}$$

where vo, T, Q, p, and F stand for a valve opening, temperature, heat input, pressure, and flow rate, respectively. Indices B, D, F, R, col, con, reb, and df represent a bottom section, distillate section, feed section, reflux section, column section, condenser section, reboiler section, and cross-column difference, respectively. Note that exact location of sensors cannot be disclosed due to the confidentiality reasons. The input set of directly measured variables is extended with important ratios ($F_{\mathrm{R}}/F_{\mathrm{F}}$, $F_{\mathrm{B}}/F_{\mathrm{F}}$, $Q_{\mathrm{con}}/F_{\mathrm{F}}$) and pressure compensated temperatures (PCT_{B}, PCT_{D}) represented by (4.5).

7.1.2 Flooding in the Industrial Debutanizer Column

The studied debutanizer column usually operates within the desired operating regime. At times, however, the operating conditions within the unit induce flooding. The envisioned low-cost solution to the flooding problem is to design a reliable indicator. The key aspect of this approach is that the designed indicator is not only used for monitoring the plant condition, but it can communicate directly with the advanced process controller that can provide a fast response.

The dataset does not contain any direct indication of flooding that could be used to label the data. However, it is possible to attribute flooding occurrence to the increased values of $p_{\mathrm{df,col}}$, F_{R}, $F_{\mathrm{reb,h,2}}$, and $T_{\mathrm{col,4}}$ and decreased values of $T_{\mathrm{reb,h,1}}$. We use this knowledge to design the reference indicator to provide the ground truth of the flooding indicator for our study.

7.2 Solution Approach

Data-driven indicators are designed using unsupervised ($\mathbf{I}^{\text{Uns}}$) and supervised ($\mathbf{I}^{\text{Sup}}$) ML approaches. For training ($\mathbf{I}^{\text{Sup}}$-type indicator) and testing, the ground truth is provided by the aforementioned reference indicator resulting from industrial knowledge about debutanizer flooding.

The design procedure of the data-driven indicator consists of three sequential steps:

1. Data processing (data filtering, data treatment, distribution to training/testing dataset).
2. A priori labelling of the training dataset (only applied for $\mathbf{I}^{\text{Uns}}$-type indicators).
3. Training of a classifier (calculation of the $f(\boldsymbol{x})$ parameters on the labelled training dataset).

After the standardization of the data set (removing the mean and scaling all the variables to unit variance), the aim of the data treatment (the 1$^{\text{st}}$ step) is to reduce the number of outliers. Due to the non-ideal (yet close normal) noise distribution within the industrial dataset, the minimum covariance determinant (MCD) approach is applied (see in Section 4.3.3). The outlier detection is performed using the F-distribution, retaining data with 99.9999 % probability. The high probability value follows from the need to eliminate only the most deviated measurements while maintaining the data representing the flooding, which can be otherwise seen as outliers.

It is optional to smoothen the dataset by filtering out the high-frequency noise that does not represent slower effects of flooding. Subsequently, as flooding is characterized by the changes of the process variables, we extend the dataset (here, 46 variables) by time differences of each variable:

$$\Delta x_i(k) = x_i(k) - x_i(k-1), \qquad \forall i = \{1, 2, \ldots, n_{\text{p}}\}, \tag{7.3}$$

where k is a time instant. The resulting dataset considers both, the original dataset and time differences, i.e., 92 variables in this study. Effectively, we assign $\boldsymbol{x} \leftarrow (\boldsymbol{x},\ \Delta\boldsymbol{x})$ in this step.

The 2$^{\text{nd}}$ step, applied to label the data for $\mathbf{I}^{\text{Uns}}$-type approaches, is performed by k-means clustering (Forgy, 1965) with the elbow method to determine the optimal number of clusters. The clusters with a low cardinality but large distance between the cluster centre and the dataset mean are considered to represent the debutanizer flooding.

The training phase needs to choose an appropriate indicator input space ($\mathbb{R}^{n_\mathrm{p}^*}$) among all the process variables and their time differences. The methods used in this study are:

1. Industrial patent by (Pihlaja and Miller, 2012), which exploits $\Delta p_{\mathrm{df,\ col}}$ only (referred to as $\mathbf{I}_{\mathrm{pat}}$).
2. Industrial experience (specific to the studied debutanizer) using $\Delta p_{\mathrm{df,col}}$, ΔF_{R}, $\Delta F_{\mathrm{reb,h,2}}$, $\Delta T_{\mathrm{col},4}$, and $\Delta T_{\mathrm{reb,h},1}$ (referred to as $\mathbf{I}_{\mathrm{ref}}$).
3. PCA approach (Pearson, 1901) presented in Section 3.2.1 (referred to as $\mathbf{I}_{\mathrm{PCA}}$).
4. SS-CV approach presented in Section 3.3.3, which determines the best subset of input variables via cross-validation and comparison of different input structures with $n_\mathrm{p}^* = \{1, 2, \ldots, 5\}$ (referred to as $\mathbf{I}_{\mathrm{SS}}$).

The finalization of the training phase designs a linear classifier (see Eq. (3.14)) based on the chosen input structure ($\boldsymbol{x}_{\mathrm{sel}} \in \mathbb{R}^{n_\mathrm{p}^*}$). To this end, we use support vector machines (SVM) presented in Section 3.5.1.

The quality of designed indicators is evaluated according to the basic classification performance indices presented in Section 2.3.6. In industrial conditions, it is much more important to warn about the potential of flooding and thus low value of FN (high RC) is preferred.

7.3 Results

7.3.1 Data Treatment using MCD

The results of data treatment using the MCD method are shown in Figure 7.1. The data values are anonymized for confidentiality reasons. As desired, only the most deviated measurements (0.75 %) are considered as outliers, and the rest of the measurements (99.25 %) is retained for further analysis. The dataset is further smoothened by filtering using a 10th-order low-pass Butterworth filter with a cut-off frequency of 0.028 mHz (with zero-phase distortion).

To guarantee fairness of indicator assessment, we distribute the retained data chronologically on an alternating monthly basis into the training and testing datasets (see Figure 7.2). From the entire dataset (25,775 measurement points), 12,781 and 12,994 points are assigned to the training and testing dataset, respectively. Figure 7.2 illustrates the training-testing data division together with (ground truth) labels assigned based on industrial experience with the reference indicator.

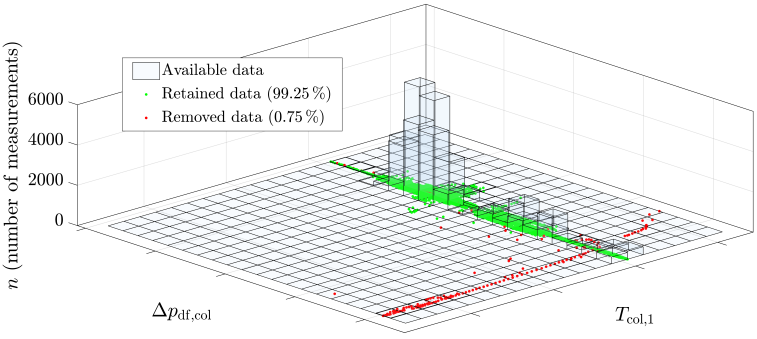


Figure 7.1: Histogram of two variables from the debutanizer dataset treated by the MCD method.

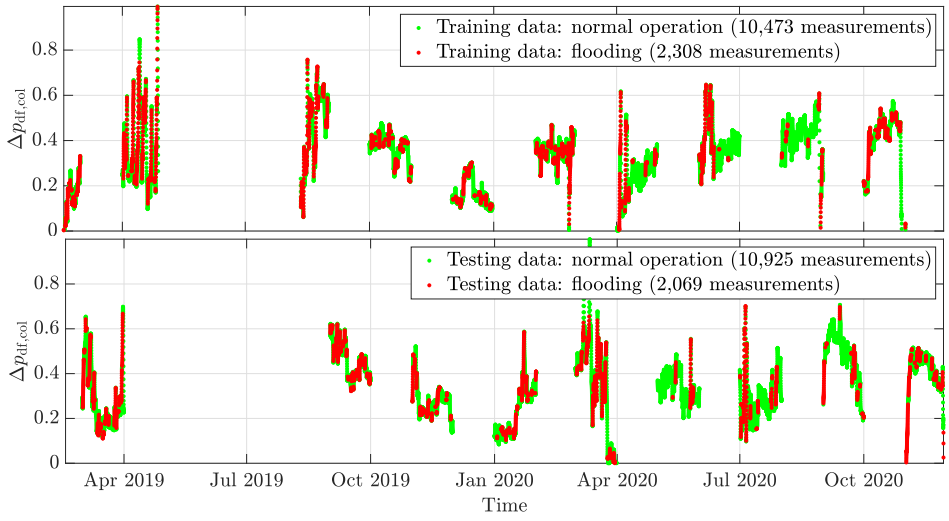


Figure 7.2: Visualization of training and testing datasets and ground truth labels.

Table 7.1: The comparison of the true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), accuracy (AC), precision (PR), recall (RC), and complexity (no. of input variables n_{p}^{*}, no. of principal components n_{pc}^{*}) of the designed data-driven indicators on the testing dataset.

ML method	Unsupervised learning			Supervised learning			
Structure	$\mathbf{I}_{\mathrm{pat}}^{\mathrm{Uns}}$	$\mathbf{I}_{\mathrm{ref}}^{\mathrm{Uns}}$	$\mathbf{I}_{\mathrm{PCA}}^{\mathrm{Uns}}$	$\mathbf{I}_{\mathrm{pat}}^{\mathrm{Sup}}$	$\mathbf{I}_{\mathrm{ref}}^{\mathrm{Sup}}$	$\mathbf{I}_{\mathrm{PCA}}^{\mathrm{Sup}}$	$\mathbf{I}_{\mathrm{SS}}^{\mathrm{Sup}}$
TP	1784	1704	618	1192	2031	1720	2029
FP	3828	2823	3147	1358	4	168	0
TN	7097	8102	7778	9567	10 921	10 757	10 925
FN	285	365	1451	877	38	349	40
AC	68.3	75.5	64.6	82.8	99.7	96	99.7
PR	31.8	37.6	16.4	46.7	99.8	91.1	100
RC	86.2	82.4	29.9	57.6	98.2	83.1	98.1
F1	46.5	51.7	21.2	51.6	99	86.9	99
$n_{\mathrm{p}}^{*}/n_{\mathrm{pc}}^{*}$	1	5	17	1	5	17	2

7.3.2 Training of Data-Driven Indicators

Design of the data-driven flooding indicators for the debutanizer column is conducted via MATLAB based on the methods from Section 7.2. MATLAB built-in routines for k-means clustering, PCA, and SVM are exploited. We design indicators based on unsupervised ML ($\mathbf{I}_{\mathrm{pat}}^{\mathrm{Uns}}$, $\mathbf{I}_{\mathrm{ref}}^{\mathrm{Uns}}$, $\mathbf{I}_{\mathrm{PCA}}^{\mathrm{Uns}}$, $\mathbf{I}_{\mathrm{SS}}^{\mathrm{Uns}}$) and supervised ML ($\mathbf{I}_{\mathrm{pat}}^{\mathrm{Sup}}$, $\mathbf{I}_{\mathrm{ref}}^{\mathrm{Sup}}$, $\mathbf{I}_{\mathrm{PCA}}^{\mathrm{Sup}}$, $\mathbf{I}_{\mathrm{SS}}^{\mathrm{Sup}}$). A main difference between these approaches is the use of the k-means algorithm to classify the data (used for $\mathbf{I}^{\mathrm{Uns}}$ indicators).

A key success factor of unsupervised ML is an appropriate data labelling. The results indicate that, unsurprisingly, the best results are obtained when the k-means clustering is performed on a dataset with reduced dimensionality (e.g., one variable for $\mathbf{I}_{\mathrm{pat}}^{\mathrm{Uns}}$ indicator or seventeen principal components determined for $\mathbf{I}_{\mathrm{PCA}}^{\mathrm{Uns}}$), with appropriate input structure. The clustering method reveals 4–5 clusters out of which 1–2 clusters are selected to represent flooding. This result suggest that merging of steps 1 and 2 mentioned in Section 7.2 is a sensible approach to successful indicator design. For this reason, we can expect PCA-based approaches to give inferior performance overall. Also, we exclude $\mathbf{I}_{\mathrm{SS}}^{\mathrm{Uns}}$ from further assessment as its performance would suffer from the inappropriate data labelling. A much more complicated design method (iterating over design steps 1–3 from Section 7.2) would be needed to construct a useful indicator.

The performance assessment of the designed indicators on the testing dataset is shown in Tab. 7.1, taking into account the so-called confusion matrix elements (i.e., TP, FP, TN, and FN) and performance criteria (i.e., AC, PR, RC, and F1). The complexity of designed indicators is represented by the number of principal components n_{pc} for PCA-based approach and by the number of input variables n_{p} for the rest of approaches. We can directly see that the supervised ML approaches outperform the unsupervised ones when we compare similar structures. The only exception appear to be the RC criterion when evaluated for $\mathbf{I}_{\mathrm{pat}}$ indicator. There are two reasons for this performance drop: 1. RC is given up in training for the AC and precision as the dataset is more populated with data points of normal operation; 2. the industrial data labels indicate flooding based on other variables than pressure (the sole input to $\mathbf{I}_{\mathrm{pat}}$ indicator) and thus $\mathbf{I}_{\mathrm{pat}}$ indicator falls short in terms of model (input) adequacy (some extra input variables would explain flooding better). Note that, the first reason can be remedied by a modification to SVM objective and some proper tuning, which, however, is beyond the scope of this study.

Among the $\mathbf{I}^{\mathrm{Uns}}$-type approaches, it is interesting that, although the structure of the reference indicator is optimal, the highest RC criterion (low number of FN) is achieved by $\mathbf{I}^{\mathrm{Uns}}_{\mathrm{pat}}$. Of course, this is paid off by worse accuracy as the classifier indicates flooding wrongly (high FP) more often overall. The PCA-based indicator appears to be the least effective (worst in all criteria). This is attributed to the aforementioned inappropriate labelling in high dimensions.

Unlike for the unsupervised learning approaches, the performance of the $\mathbf{I}^{\mathrm{Sup}}_{\mathrm{PCA}}$ indicator is sufficient. It also appears that the $\mathbf{I}^{\mathrm{Sup}}_{\mathrm{PCA}}$ is more efficient compared to the $\mathbf{I}^{\mathrm{Sup}}_{\mathrm{Pat}}$ indicator viewed by each performance criterion. The highest efficacy among supervised learning approaches is achieved for $\mathbf{I}^{\mathrm{Sup}}_{\mathrm{ref}}$ and $\mathbf{I}^{\mathrm{Sup}}_{\mathrm{SS}}$ indicators. These approaches already consider or can find the best possible input structure. It is noteworthy that $\mathbf{I}^{\mathrm{Sup}}_{\mathrm{SS}}$ achieves the best performance (almost 100 % in all performance criteria) using a very simple structure. This effectively tells that the reference structure is overly complicated (some inputs are redundant) and that it is possible to indicate flooding with data from just two sensors. It is also a very interesting result as it allows the industrial practitioners to concentrate efforts regarding sensor maintenance towards smaller subset of online sensors. Surprisingly, pressure is not among the inputs selected for the best indicator. The input structure involves reflux flow ΔF_{R} and the time difference of heating medium flowin the reboiler $\Delta F_{\mathrm{reb,h},2}$, which are both part of the reference indicator structure. It is possible that the two selected flow rates are measured with better precision and that they do not involve high-frequency fluctuations as pressure measurements do. These results need, of course, further validation in an industrial setup as the reference indicator (ground truth) involves the the same input variables

as the best indicators found.

7.4 Discussion

The distribution of the data in Figure 7.1 indicates that the dataset from the industrial debutanizer column exhibits strong non-ideality. The extent of non-ideality can be assessed by comparing the distribution to a normal distribution. By removing the outliers (0.75 %) identified using the MCD method, the resulting dataset (99.25 %) shows a closer resemblance to a normal distribution. This implies that the use of MCD has improved the quality of the industrial dataset. However, it is worth exploring alternative approaches such as k-means clustering (see Section 3.4.1) or T^2 distance (see Section 2.2.3), as they may offer better data treatment and lead to further improvements in the design of the flooding indicator $\mathbf{I}$.

The results presented in Table 7.1 indicate relatively low efficiency (i.e., AC, PR, RC, and F1) but high complexity for both $\mathbf{I}_{\mathrm{PCA}}$. This suggests that PCA, with its unsupervised learning nature, is not focused on explaining flooding specifically, but rather aims to reduce the complexity of the input dataset. Therefore, it may be beneficial to consider using PLS (see Section 3.2.2) instead of PCA.

Chapter 8

Conclusions and Future Research

The focus of this thesis was on designing linear soft (inferential) sensors to monitor hard-to-measure or completely unmeasurable variables in the petrochemical industry. Given the complexity of the studied processes, our research primarily centred around data-driven soft sensor design. The findings presented in this thesis validate its applicability and significance and can be summarised into three main contributions.

The first contribution (Mojto et al., 2021) presented the fundamental design of soft sensors for two case studies in the petrochemical industry: the depropanizer distillation column for the Fluid Catalytic Cracking (FCC) unit and the product fractionator from the Vacuum Gasoil Hydrogenation (VGH) unit at Slovnaft, a.s. in Bratislava, Slovakia. We examined the performance of basic data treatment methods and simple approaches in the design of linear data-driven soft sensors. This contribution offered a comprehensive design of data-driven soft sensors for specific case studies, shedding light on the potential challenges in the field of soft-sensor design.

In the second contribution (Mojto et al., 2023a), our research shifted towards designing multi-model soft sensors (MMS) for industrial case studies. We proposed innovative approaches that enable continuous switching between soft-sensor models and optimise data labelling in the training dataset. We compared the performance of the studied MMSs with single-model soft sensors (SMS) based on the approaches outlined in the first contribution. The case studies in this contribution involved the model of pressure-compensated temperature and the main fractionator from the VGH unit (identical to the first contribution). The results confirmed the superior potential of MMSs over SMSs.

The third contribution (Mojto et al., 2023b) of this thesis focused on designing a flooding indicator for the industrial debutanizer distillation column at Slovnaft, a.s. in Bratislava, a part of the FCC unit. We compared the performance of various unsupervised and supervised learning approaches against a ground-truth model designed based on expert

knowledge from the industry. As expected, the supervised learning approaches exhibited higher accuracy compared to the unsupervised methods. Additionally, the results identified the most relevant variables associated with the studied flooding phenomenon.

Overall, these contributions advance the understanding and practical implementation of soft sensors in the petrochemical industry, addressing the challenges of monitoring hard-to-measure variables and providing valuable insights for future research and application in this field.

This research has the potential for future extensions. The performance of the linear data-driven approaches studied here is strongly influenced by the quality of the data. While we have focused on basic data treatment methods with limited effectiveness, exploring more advanced techniques would be advantageous. These advanced approaches can further improve the quality of the available dataset, resulting in more reliable and accurate designs of linear soft sensors. Furthermore, our analysis has demonstrated that incorporating nonlinear transformations of the input variables can significantly enhance the performance of linear soft sensors. Therefore, it would be valuable to incorporate a suitable method for identifying appropriate nonlinear transformations, as this would greatly enhance the quality of our research. The ALAMO approach (Wilson and Sahinidis, 2017) shows promise as a foundation for addressing this challenge. In addition, we recognize the potential of multi-fidelity modelling (Perdikaris et al., 2017), which not only addresses the search for nonlinear transformations but also enhances the performance and sustainability of soft sensors by effectively combining information from various sources. This approach holds great potential for advancing the field of soft sensing.

Appendix A

Curriculum Vitae

Martin Mojto

Date of Birth:	December 16, 1994
Citizenship:	Slovakia
Email (work):	martin.mojto@stuba.sk
Email (personal):	mojtommartin@gmail.com
Homepage:	https://www.uiam.sk/ mojto
LinkedIn:	https://www.linkedin.com/in/martinmojto/

Education

- Doctoral Studies (September 2019 – Now)
  - Slovak University of Technology in Bratislava
  - Programme: Process Control
  - Thesis topic: Data-driven Design of Linear Soft Sensors
  - Thesis supervisor: doc. Ing. Radoslav Paulen, PhD.
- Master's Degree (September 2017 – May 2019)
  - Slovak University of Technology in Bratislava
  - Programme: Automation and Information Engineering in Chemistry and Food Industry
  - Thesis topic: Advanced Process Control of a Depropanizer Column
  - Thesis supervisor: prof. Ing. Miroslav Fikar, DrSc.
- Bachelor's Degree (September 2014 – July 2017)

  - Slovak University of Technology in Bratislava
  - Programme: Chemical Engineering
  - Thesis topic: Simulation of Azeotropic Distillation
  - Thesis supervisor: doc. Ing. Pavol Steltenpohl, PhD.

Foreign Internships

- Research Stay at Imperial College London (February 2022 – November 2022)
  - Research focus: Gaussian process-based multi-fidelity modelling
  - Personal aims: get knowledge about the academic standards at prestigious foreign universities and apply this knowledge to enhance the quality of home science, quality improvement of my scientific outputs

Research Fields

- Soft (inferential) sensors
- Regression and classification
- Distillation processes
- Data treatment
- Feature engineering
- Mathematical modelling

Digital Skills

- Matlab & Simulink (advanced)
- Python (intermediate)
- gPROMS (intermediate)
- UniSim (beginner)
- LaTeX (advanced)
- Windows (intermediate)
- Linux (beginner)
- MS Office (intermediate)
- ChatGPT (intermediate)
- Google (intermediate)

Language Skills

- Slovak (mother tongue)
- English (listening B2, reading C1, writing B2, spoken interaction C1)

Appendix B

Author's Publications

Articles in Journal

1. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Data-based Design of Multi-model Inferential Sensors. Computers & Chemical Engineering, 2023, (*under review*).

2. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Data-based Design of Inferential Sensors for Petrochemical Industry. Computers & Chemical Engineering, vol. 153, pp. 107437, 2021.

3. Mojto, M. – Horváthová, M. – Kiš, K. – Furka, M. – Bakošová, M.: Predictive control of a cascade of biochemical reactors. Acta Chimica Slovaca, no. 1, vol. 14, pp. 51–59, 2021.

Articles in Conference Proceedings

1. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Design of Multi-Model Linear Inferential Sensors with SVM-based Switching Logic. IFAC World Congress, 2023, (*accepted*).

2. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Data-Driven Indication of Flooding in an Industrial Debutanizer Column. Antonis Kokossis, Michael C. Georgiadis, Efstratios N. Pistikopoulos, editors, 33rd European Symposium on Computer Aided Process Engineering, volume 52 of Computer Aided Chemical Engineering. Elsevier, (*accepted*).

3. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Input Structure Selection for Soft-Sensor Design: Does It Pay Off?. In Proceedings of the 24rd International Conference on Process Control, 2023, (*accepted*).

4. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Comparing Linear and Nonlinear Soft Sensor Approaches for Industrial Distillation Columns. In 49th International Conference of the Slovak Society of Chemical Engineering SSCHE 2023, Slovak Society of Chemical Engineering, Bratislava, SK, pp. 159–159, 2023.

5. Fáber, R. – Ľubušký, K. – Mojto, M. – Paulen, R.: Enhancing Industrial Data Analysis through Machine Learning-based Classification of Petrochemical Datasets. In 49th International Conference of the Slovak Society of Chemical Engineering SSCHE 2023, Slovak Society of Chemical Engineering, Bratislava, SK, pp. 160–160, 2023.

6. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Support Vector Machine-based Design of Multi-model Inferential Sensors. Editor(i): Ludovic Montastruc, Stephane Negny, In 32nd European Symposium on Computer Aided Process Engineering, Elsevier, no. 1, vol. 32, pp. 1045–1050, 2022.

7. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Multi-Model Soft-Sensor Design for a Depropanizer Distillation Column. V Advanced Process Modelling Forum 18–19 October 2022, 2022.

8. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Data-based Design of Inferential Sensors for an Industrial Depropanizer Column with Data Pre-treatment Analysis. Editor(s): Mário Mihaľ, In 48th International Conference of the Slovak Society of Chemical Engineering SSCHE 2022 and Membrane Conference PERMEA 2022, Slovak Society of Chemical Engineering, Bratislava, SK, pp. 200, 2022.

9. Fikar, M. – Furka, M. – Horváthová, M. – Kiš, K. – Mojto, M.: Dynamic Optimisation Toolbox dynopt 5.0. Editor(s): R. Paulen and M. Fikar, In Proceedings of the 23rd International Conference on Process Control, IEEE, Slovak University of Technology, pp. 296–301, 2021.

10. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Data Treatment of Industrial Measurements: From Online to Inferential Sensors. Editor(i): R. Paulen, M. Fikar and J. Oravec, In Proceedings of the 23rd International Conference on Process Control - Summaries Volume, Slovak Chemical Library, Slovak University of Technology in Bratislava, Radlinského 9, SK812-37, Bratislava, Slovakia, pp. 52–53, 2021.

11. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Data-based Industrial Soft-sensor Design via Optimal Subset Selection. Editor(s): Metin Türkay, Erdal Aydin, In 31th European Symposium on Computer Aided Process Engineering, Elsevier, vol. 31, pp. 1247–1252, 2021.

12. Furka, M. – Kiš, K. – Horváthová, M. – Mojto, M. – Bakošová, M.: Identification and Control of a Cascade of Biochemical Reactors. In 2020 Cybernetics & Informatics (K&I), 2020.

13. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Advanced Process Control of an Industrial Depropanizer Column using Data-based Inferential Sensors. Editor(s): Sauro Pierucci, Flavio Manenti, Giulia Luisa Bozzano, Davide Manca, In 30th European Symposium on Computer Aided Process Engineering, Elsevier, vol. 30, pp. 1213–1218, 2020.

14. Mojto, M. – Ľubušký, K. – Fikar, M. – Paulen, R.: Design of Data-based Inferential Sensors for Industrial Depropanizer Column. Editor(s): G. Léonard and F. Logist, In Computer Aided Process Engineering, CAPE Forum, pp. 12–13, 2019.

15. Mojto, M. – Ľubušký, K. – Paulen, R. – Fikar, M.: Advanced Process Control of a Depropanizer Column. Editor(s): M. Fikar and M. Kvasnica, In Proceedings of the 22nd International Conference on Process Control, Slovak Chemical Library, Štrbské Pleso, Slovakia, 2019.

16. Mojto, M. – Paulen, R. – Ľubušký, K. – Fikar, M.: Modelling and Analysis of Control Pairings of an Industrial Depropanizer Column. In Advanced Process Modelling Forum 26–27 March 2019, pp. 5–6, 2019.

Appendix C

Resumé

Táto dizertačná práca sa zaoberá návrhom lineárnych softvérových (inferenčných) senzorov založených na dátach. Softvérový senzor je matematický model, ktorý využíva dostupné údaje z kontinuálne meraných veličín, ako sú teploty, tlaky, prietoky a iné, na odhad ťažko merateľných veličín, napríklad zloženia alebo koncentrácie produktov. Motiváciou návrhu inferenčných senzorov sú vysoké náklady spojené s kúpou a prevádzkovaním fyzických zariadení (senzorov) na indikáciu týchto ťažko merateľných veličín. V súčasnosti je zaznamenaný vzostup dopytu po lacnom a presnom spôsobe merania týchto veličín, pretože sú neodmysliteľnou súčasťou pokročilých procesných regulátorov. Tieto regulátory dokážu dosiahnuť optimálne prevádzkové podmienky, čo prispieva k zvýšeniu celkového zisku a udržateľnosti procesu.

V tejto práci je návrh softvérových senzorov založený na reálnych dátach z priemyselnej databázy. Takéto merania zvyčajne obsahujú veľké množstvo systematických chýb, ktoré môžu znížiť efektivitu návrhu softvérových senzorov. Preto sa v tejto práci zaoberáme rôznymi metódami na identifikáciu systematických chýb a štatisticky odchýlených meraní vo viacrozmernej dátovej množine. Prvou skúmanou metódou je Hotellingovo T^2 rozdelenie (T^2 vzdialenosť), ktoré indikuje systematické chyby na základe vzdialenosti meraní od stredu ($\boldsymbol{\mu}_x = \mathbf{0}$) mnohorozmernej dátovej množiny. Ďalšou metódou, ktorou sa zaoberáme, je metóda minimálneho determinantu kovariančnej matice (MCD), ktorá rovnako ako T^2 vzdialenosť berie do úvahy jedno stredovú hodnotu mnohorozmernej dátovej množiny. MCD metóda je sofistikovanejším prístupom k identifikácii odchýlených meraní v porovnaní s T^2 vzdialenosťou. Poslednou metódou, ktorou sa zaoberáme, je k-means klastrovanie, ktoré voči predchádzajúcim metódam uvažuje viacero (v závislosti od voľby) stredových hodnôt mnohorozmernej dátovej množiny pri identifikácii systematických chýb.

Návrh inferenčných senzorov možno rozdeliť na základe počtu uvažovaných modelov na jedno-modelové lineárne softvérové senzory (SMS) na viac-modelové lineárne softvérové senzory (MMS). SMS predstavujú štandardný a často využívaný spôsob monitorovania

ťažko merateľných veličín v priemyselnej praxi. Ich výhodou je jednoduchá a intuitívna štruktúra modelu. Operátori a inžinieri v priemyselnej praxi vedia pristúpiť k modelu SMS a vykonať validáciu jednotlivých parametrov na základe ich znalosti o správaní sa samotného procesu.

V tejto práci sa zaoberáme viacerými metódami trénovania jedno-modelových softvérových senzorov (SMS). Do množiny skúmaných metód patrí základná lineárna regresia (OLS), ktorá sa snaží nájsť model s najväčšou presnosťou vzhľadom na poskytnutú trénovaciu množinu údajov. Ďalšími skúmanými metódami sú analýza hlavných komponentov (PCA) a regresná analýza parciálnych najmenších štvorcov (PLS). Obe metódy sa zaoberajú koreláciami medzi veličinami v trénovacej množine údajov. Tieto metódy sa líšia v tom, že PCA uvažuje vo svojej analýze len vstupné dáta (kontinuálne merané veličiny), zatiaľ čo PLS regresia uvažuje aj výstupné dáta (ťažko merateľná veličina). Na základe tohto porovnania možno konštatovať, že použitie PCA je preferované vtedy, keď je dostupná veľká množina vstupný dát, zatiaľ čo PLS by sa mala použiť vtedy ak meranie výstupnej veličiny možno považovať za dôveryhodné. Ďalšou skúmanou metódou je LASSO, ktorá súčasne navrhuje štruktúru a parametre SMS. Táto metóda obsahuje vo svojej účelovej funkcii penalizačný element (ℓ_1-norm), ktorý odstraňuje (vynuluje) vstupné veličiny so zanedbateľným vplyvom na odhadovanú výstupnú veličinu. Na ladenie tohto parametra sa môžu využiť rôzne kritéria pretrénovania modelu (AICc, BIC, R^2_{adj}) alebo krížová validácia. Poslednou metódou je výber podmnožiny (angl. subset selection, SS), ktorá sa zaoberá výberom vhodnej podmnožiny veličín na návrh softvérového senzora. V tejto práci používame modifikovanú formu SS metódy, ktorá využíva kritéria pretrénovania modelu spomenuté vyššie a krížovú validáciu ako to bolo uvedené pri metóde LASSO.

Následne sa v tejto práci realizuje návrh viac-modelových lineárnych softvérových senzorov (MMS), ktoré môžu byť použité pre procesy s viacerými operačnými bodmi a pre odhad veličín s výraznejším nelineárnym správaním, kde jedno-modelové lineárne softvérové senzory (SMS) nedosahujú požadovanú presnosť odhadu. Štandardný návrh MMS (MMS_{SotA}) sa skladá z troch krokov: (1) počiatočné označkovanie trénovacej množiny údajov, (2) návrh klasifikátora na základe získaných značiek z predchádzajúceho bodu návrhu MMS_{SotA} a (3) trénovanie individuálnych modelov v štruktúre MMS_{SotA}. V tejto práci používame metódu k-means klastrovania na počiatočné označkovanie trénovacej množiny údajov (prvý krok návrhu MMS_{SotA}). Táto metóda nám umožňuje rozdeliť údaje do klastrov a každý klastr je označený konkrétnym identifikátorom (preto sa tento krok nazýva počiatočné označkovanie). V druhom kroku návrhu MMS_{SotA} využívame metódu support vector machines (SVM), ktorá nám zabezpečuje návrh lineárnych separačných klasifikátorov. Lineárna štruktúra klasifikátorov je dôležitá, pretože aj modely v MMS_{SotA} majú byť lineárne. V treťom

kroku návrhu MMS_{SotA}, ktorý sa zaoberá trénovaním lineárnych modelov v MMS_{SotA}, môžeme využiť niektorý z prístupov na trénovanie SMS, ktoré sme spomínali vyššie. V tejto práci sme sa rozhodli využiť metódu OLS na trénovanie modelov MMS_{SotA} pre jednoduchosť výpočtu. Možné využitie iných prístupov na trénovanie SMS v tomto kroku návrhu MMS_{SotA} by mohlo predstavovať potenciálne zlepšenie celkového výkonu MMS.

Skúmané metódy sú porovnávané na základe návrhu softvérových senzorov pre tri priemyselné prípadové štúdie z ropnej rafinérie Slovnaft, a.s. v Bratislave. Prvá štúdia sa zaoberá destilačnou kolónou depropanizérom, ktorá je súčasťou prevádzky fluidného katalytického krakovania (FCC). Táto kolóna spracováva ľahké uhľovodíky (C3-C5) a delí ich na destilát obohatený o C3 frakciu a zvyšok, obohatený o C4/C5 frakciu. Druhá priemyselná destilačná kolóna je debutanizér, ktorý sa tiež nachádza v prevádzke FCC. Tento debutanizér spracováva C4/C5 frakciu z depropanizéru a delí ju na destilát obohatený o C4 frakciu a zvyšok obohatený o C5 frakciu. Tretia priemyselná destilačná kolóna je hlavný frakcionátor na jednotke hydrogenácie vákuových destilátov (VGH). Hlavným cieľom prevádzky VGH je odstránenie nečistôt, ako je dusík a síra, z vákuových destilátov. Hlavný frakcionátor produkuje niekoľko produktov, pričom jeden z produktov je nástrek (surovina) pre prevádzku FCC. V prípade VGH prevádzky je odhadovanou veličinou čistota plynového oleja, ktorý je jeden z produktov hlavného frakcionátora. Návrh softvérových senzorov pre hlavný frakcionátor na prevádzke VGH zahŕňa oveľa viac procesov a zariadení v porovnaní s návrhom softvérových senzorov pre depropanizér a debutanizér v prevádzke FCC. Na druhej strane, hlavný frakcionátor VGH disponuje väčším počtom meraní odhadovanej veličiny z laboratórnej analýzy v porovnaní s depropanizérom na prevádzke FCC. Obidva tieto prístupy majú rovnaký typ návrhu softvérového senzora na indikáciu alebo odhad ťažko merateľnej veličiny. Návrh softvérového senzora pre debutanizér sa odlišuje tým, že má detekovať poruchy, ako je napríklad zaplavovanie, na danom zariadení, čo predstavuje odlišnú oblasť výskumu.

Táto dizertačná práca prezentuje výsledky troch hlavných príspevkov, ktoré boli publikované, akceptované alebo odoslané do časopisu počas štúdia. Prvý príspevok sa zameriaval na návrh jedno-modelových lineárnych softvérových senzorov (SMS) pre dve priemyselné prípadové štúdie: FCC (depropanizér) a VGH prevádzka (hlavná frakcionačná rektifikačná kolóna). Tento návrh sa zameral na hĺbkovú analýzu a porovnanie viacerých metód spracovania údajov, ako je T^2 vzdialenosť, MCD a k-means klastrovanie. Tieto metódy boli použité na odstránenie outlierov a systematických chýb a následne boli očistené údaje použité na trénovanie SMS na základe rôznych prístupov založených na dátach (OLS, PCA, PLS, LASSO a SS). Výsledky ukázali, že MCD prístup k spracovaniu údajov je viac prispôsobivý v porovnaní s T^2 vzdialenosťou alebo

k-means klastrovaním. V oboch prípadových štúdiách tento prístup indikoval relatívne rozumné množstvo údajov ako odchýlené merania, zatiaľ čo iné prístupy bolo treba doladiť, aby sa znížila ich agresivita pri detekcii odchýlených meraní. Výsledky z návrhu softvérových senzorov naznačujú, že metóda SS s krížovou validáciou dosiahla dobrý kompromis medzi zložitosťou a presnosťou modelu. Okrem spomenutých numerických výsledkov, význam tohto príspevku spočíva aj v tom, že poukázal na možné výzvy a potenciálne smerovanie výskumu v oblasti návrhu priemyselných softvérových senzorov.

Ďalší príspevok dizertačnej práce je zameraný na návrh viac-modelových lineárnych softvérových senzorov (MMS) a ich porovnanie s jedno-modelovými lineárnymi softvérovými senzormi (SMS). Pri návrhu MMS sme vylepšili štandardný MMS_{SotA} vlastnými metódami, ktoré odstraňujú nedostatky MMS_{SotA}. Prvá nami navrhnutá metóda zohľadňuje spojité prepínanie jednotlivých modelov v štruktúre MMS, preto sme tento prístup nazvali spojitý MMS (MMS_{con}). Spojité prepínanie modelov v MMS predstavuje výrazný prínos, najmä v prípade implementácie MMS v pokročilých priemyselných regulátoroch, kde nespojité prepínanie modelov MMS by mohlo viesť k nestabilite riadenia procesu. Druhá nami navrhnutá metóda rieši druhý nedostatok MMS_{SotA}, ktorý vyplýva z toho, že počiatočné označkovanie údajov (prvý bod návrhu MMS_{SotA}) nemá informácie o jeho vplyve na presnosť trénovaných modelov MMS (tretí bod návrhu MMS_{SotA}). Nami navrhnutá metóda efektívne spája všetky tri body MMS_{SotA} do jedného optimalizačného problému, kde sa všetky problémy návrhu MMS riešia súčasne. Preto je tento spôsob návrhu považovaný za MMS návrh s optimálnym počiatočným označkovaním údajov ($\text{MMS}_{\text{con,lab}}$). Okrem návrhu nových prístupov na návrh MMS sa tento príspevok dizertačnej práce zameriava na porovnanie SMS a MMS pomocou množiny údajov z priemyselnej jednotky VGH. Výsledky naznačujú, že uvažovanie MMS pri indikácii ťažko merateľnej veličiny na VGH jednotke môže viesť k zvýšeniu presnosti odhadu v porovnaní s prístupmi SMS.

Posledný príspevok dizertačnej práce sa zaoberá návrhom indikátora pre zaplavovanie debutanizéra na prevádzke FCC. Návrh tohto indikátora je založený na princípoch softvérových senzorov. Pri návrhu sa zvažujú metódy strojového učenia bez učiteľa (unsupervised learning) a s učiteľom (supervised learning). Efektívnosť navrhnutých prístupov je porovnávaná so skutočnými indikáciami zaplavovania v kolóne debutanizéra. Skutočné indikácie sú získané na základe vedomostí operátorov v rafinérii o tomto jave. Výsledky naznačujú, že indikátory zaplavovania vytvorené pomocou metód strojového učenia s učiteľom dosahujú presnejšie výsledky v porovnaní s metódami strojového učenia bez učiteľa. Toto pozorovanie bolo očakávané vzhľadom na to, že metódy strojového učenia s učiteľom majú priamy kontakt so skutočnými indikáciami zaplavovania pri návrhu indikátora. Výsledky tiež poukázali na veličiny, ktoré sa zdajú byť najrelevantnejšie pre vysvetlenie zaplavovania v kolóne debutanizéra.

Na základe uvedených výsledkov možno konštatovať, že ciele dizertačnej práce boli splnené. Prvotným cieľom bolo spracovanie údajov s využitím viacerých prístupov na spracovanie mnohorozmernej dátovej množiny. Za týmto účelom sme použili metódy, ako je T^2 vzdialenosť, MCD a k-means klastrovanie. Hlavným cieľom bolo porovnanie efektívnosti návrhu jedno-modelových softvérových senzorov (SMS) a viac-modelových softvérových senzorov (MMS). Na vykonanie vernej analýzy kvality SMS sme uvažovali viaceré metódy dátovo orientovaného návrhu, ako je OLS, PCA, PLS, LASSO a SS. Pri MMS sme tiež zvážili rôzne prístupy. Prvým je štandardný návrh ($\mathrm{MMS}_{\mathrm{SotA}}$), ktorý sme porovnali s našimi vlastnými modifikáciami štandardného prístupu: (1) prístup so spojitým prepínaním modelov ($\mathrm{MMS}_{\mathrm{con}}$) a (2) prístup s optimalizovaným označkovaním údajov ($\mathrm{MMS}_{\mathrm{con,lab}}$). Posledným cieľom dizertačnej práce bolo navrhnúť softvérový senzor pre problém zaplavovania priemyselnej rektifikačnej kolóny (detekcia poruchy). Tento cieľ sme dosiahli prostredníctvom rôznych prístupov strojového učenia s učiteľom aj bez učiteľa.

Bibliography

Alameddine, I., Kenney, M. A., Gosnell, R. J., and Reckhow, K. H. (2010). Robust multivariate outlier detection methods for environmental data. *Journal of Environmental Engineering*, 136(11):1299–1304.

Alves, R. M. B. and Nascimento, C. A. O. (2007). Analysis and detection of outliers and systematic errors in industrial plant data. *Chemical Engineering Communications*, 194(3):382–397.

Azzaoui, H., Mansouri, I., and Elkihel, B. (2019). Methylcyclohexane continuous distillation column fault detection using stationary wavelet transform and k-means. In Hajji, B., Tina, G. M., Ghoumid, K., Rabhi, A., and Mellit, A., editors, *Proceedings of the 1st International Conference on Electronic Engineering and Renewable Energy*, pages 399–411. Springer Singapore.

Bemporad, A. (2022). A piecewise linear regression and classification algorithm with application to learning and model predictive control of hybrid systems. *IEEE Trans. Autom. Control*, pages 1–16.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA. Association for Computing Machinery.

Botha, S. and Craig, I. K. (2021). An industrial implementation of a c4 hydrocarbon soft sensor to optimise a debutaniser column. *IFAC-PapersOnLine*, 54(21):180–185. Control Conference Africa CCA 2021.

Cao, L., Yu, F., Yang, F., Cao, Y., and Gopaluni, R. B. (2020). Data-driven dynamic inferential sensors based on causality analysis. *Control Engineering Practice*, 104:104626.

Curreri, F., Graziani, S., and Xibilia, M. G. (2020). Input selection methods for data-driven soft sensors design: Application to an industrial process. *Information Sciences*, 537:1–17.

de Morais, G. A., Barbosa, B. H., Ferreira, D. D., and Paiva, L. S. (2019). Soft sensors design in a petrochemical process using an evolutionary algorithm. *Measurement*, 148:106920.

Doraiswami, R. and Cheded, L. (2014). Robust model-based soft sensor: Design and application. *IFAC Proceedings Volumes*, 47(3):5491–5496. 19th IFAC World Congress.

Efroymson, M. (1960). *Mathematical Methods for Digital Computers*, chapter Multiple Regression Analysis. Wiley, New York, NY.

Fontes, C. H., Santos, I. C., Embiruçu, M., and Aragão, P. (2021). Pattern reconciliation: A new approach involving constrained clustering of time series. *Computers & Chemical Engineering*, 145:107169.

Forgy, E. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–769.

Frumosu, F. D. and Kulahci, M. (2019). Outliers detection using an iterative strategy for semi-supervised learning. *Quality and Reliability Engineering International*, 35(5):1408–1423.

Fuentes-Cortés, L. F., Flores-Tlacuahuac, A., and Nigam, K. D. P. (2022). Machine learning algorithms used in pse environments: A didactic approach and critical perspective. *Ind. Eng. Chem. Res.*, 61(25):8932–8962.

Gan, G., Ma, C., and Wu, J. (2020). *Data Clustering: Theory, Algorithms, and Applications, Second Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Ge, Z. (2017). Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometrics and Intelligent Laboratory Systems*, 171:16–25.

Griva, I., Nash, S. G., and Sofer, A. (2008). *Linear and Nonlinear Optimization (2. ed.)*. SIAM.

Gurobi Optimization, LLC (2023). Gurobi Optimizer Reference Manual.

Hardin, J. and Rocke, D. M. (2005). The distribution of robust distances. *Journal of Computational and Graphical Statistics*, 14(4):928–946.

Hastie, T., Friedman, J., and Tisbshirani, R. (2017). *The elements of Statistical Learning: Data Mining, Inference, and prediction.* Springer.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hotelling, H. (1931). The generalization of student's ratio. *Annals of Mathematical Statistics*, 2(3):360–378.

Hubert, M. and Debruyne, M. (2010). Minimum covariance determinant. *WIREs Computational Statistics*, 2(1):36–43.

Jin, H., Chen, X., Yang, J., Zhang, H., Wang, L., and Wu, L. (2015). Multi-model adaptive soft sensor modeling method using local learning and online support vector regression for nonlinear time-variant batch processes. *Chemical Engineering Science*, 131:282–303.

Joe Qin, S. (2003). Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 17(8-9):480–502.

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

Kamath, C. (2022). Intelligent sampling for surrogate modeling, hyperparameter optimization, and data analysis. *Machine Learning with Applications*, 9:100373.

Khatibisepehr, S., Huang, B., and Khare, S. (2013). Design of inferential sensors in the process industry: A review of Bayesian methods. *Journal of Process Control*, 23:1575–1596.

Khatibisepehr, S., Huang, B., Xu, F., and Espejo, A. (2012). A bayesian approach to design of adaptive multi-model inferential sensors with application in oil sand industry. *Journal of Process Control*, 22(10):1913–1929.

King, M. (2011). *Process Control: A Practical Approach.* John Wiley & Sons Ltd.

Kodinariya, T. and Makwana, P. (2013). Review on determining of cluster in k-means clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1:90–95.

Kordon, A., Smits, G., Kalos, A. N., and Jordaan, E. (2003). Robust soft sensor development using genetic programming. *Data Handling in Science and Technology*, 23:69–108.

Lemos, T., Campos, L. F., Melo, A., Clavijo, N., Soares, R., Câmara, M., Feital, T., Anzai, T., and Pinto, J. C. (2021). Echo state network based soft sensor for monitoring and fault detection of industrial processes. *Computers & Chemical Engineering*, 155:107512.

Löfberg, J. (2004). Yalmip: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan.

Lü, Y. and Yang, H. (2014). A multi-model approach for soft sensor development based on feature extraction using weighted kernel fisher criterion. *Chin. J. Chem. Eng.*, 22(2):146–152.

Mejdell, T. and Skogestad, S. (1991). Composition estimator in a pilot-plant distillation column using multiple temperatures. *Industrial & Engineering Chemistry Research*, 30(12):2555–2564.

Mencarelli, L., Pagot, A., and Duchêne, P. (2020). Surrogate-based modeling techniques with application to catalytic reforming and isomerization processes. *Computers & Chemical Engineering*, 135.

Milano, M. (2012). *Principles and Practice of Constraint Programming - CP 2012: 18th International Conference, CP 2012, Québec City, QC, Canada, October 8-12, 2012, Proceedings*. Lecture Notes in Computer Science. Springer.

Miyashiro, R. and Takano, Y. (2015). Mixed integer second-order cone programming formulations for variable selection in linear regression. *European Journal of Operational Research*, 247:721–731.

Mojto, M., Ľubušký, K., Fikar, M., and Paulen, R. (2021). Data-based design of inferential sensors for petrochemical industry. *Computers & Chemical Engineering*, 153:107437.

Mojto, M., Ľubušký, K., Fikar, M., and Paulen, R. (2022). Support vector machine-based design of multi-model inferential sensors. In Montastruc, L. and Negny, S., editors, *32nd European Symposium on Computer Aided Process Engineering*, volume 51 of *Computer Aided Chemical Engineering*, pages 1045–1050. Elsevier.

Mojto, M., Ľubušký, K., Fikar, M., and Paulen, R. (2023a). Data-driven design of multi-model soft sensors. *Computers & Chemical Engineering*. (*under review*).

Mojto, M., Ľubušký, K., Fikar, M., and Paulen, R. (2023b). Data-driven indication of flooding in an industrial debutanizer column. In Antonis Kokossis, Michael C. Georgiadis, E. N. P., editor, *33nd European Symposium on Computer Aided Process Engineering*, volume 52 of *Computer Aided Chemical Engineering*. Elsevier. (*accepted*).

Oeing, J., Neuendorf, L. M., Bittorf, L., Krieger, W., and Kockmann, N. (2021). Flooding prevention in distillation and extraction columns with aid of machine learning approaches. *Chemie Ingenieur Technik*, 93(12):1917–1929.

Pan, H., Wu, X., Qiu, J., He, G., and Ling, H. (2019). Pressure compensated temperature control of kaibel divided-wall column. *Chemical Engineering Science*, 203:321–332.

Pan, Y. C., Qin, S. J., Nguyen, P., and Barham, M. (2013). Hybrid inferential modeling for vapor pressure of hydrocarbon mixtures in oil production. *Industrial & Engineering Chemistry Research*, 52(35):12420–12425.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Peiravan, H., Ilkhani, A. R., and Sarraf, M. J. (2020). Preventing of flooding phenomena on vacuum distillation trays column via controlling coking value factor. *SN Applied Sciences*, 2(10):1–11.

Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N. D., and Karniadakis, G. E. (2017). Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2198):20160751.

Pihlaja, R. K. and Miller, J. P. (2012). Detection of distillation column flooding. US Patent 8,216,429.

Pla, D. L., Kamyar, R., Hashemian, N., Mehdizadeh, H., and Moshgbar, M. (2018). Moisture soft sensor for batch fluid bed dryers: A practical approach. *Powder Technology*, 326:69–77.

Quelhas, A. (2009). Soft sensor models: Bias updating revisited. *IFAC Proceedings Volumes*, 42.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880.

Santander, O., Kuppuraj, V., Harrison, C. A., and Baldea, M. (2022). An open source fluid catalytic cracker - fractionator model to support the development and benchmarking of process control, machine learning and operation strategies. *Computers & Chemical Engineering*, 164:107900.

Santosa, F. and Symes, W. W. (1986). Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330.

Serpas, M., Chu, Y., and Hahn, J. (2013). Fault detection approach for systems involving soft sensors. *Journal of Loss Prevention in the Process Industries*, 26(3):443–452. Papers presented at the 2011 Mary Kay O'Connor Process Safety Center International Symposium.

Smith, G. (2018). Step away from stepwise. *Journal of Big Data*, 5:32.

Song, M., Breneman, C. M., Bi, J., Sukumar, N., Bennett, K. P., Cramer, S., and Tugcu, N. (2002). Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. *Journal of Chemical Information and Computer Sciences*, 42(6):1347–1357.

Student (1908). The probable error of a mean. *Biometrika*, 6(1):1–25.

Su, H., Rong, G., and Chu, J. (2009). Industrial processes: Data reconciliation and gross error detection. *Measurement and Control*, 42:209–215.

Sun, W. and Braatz, R. D. (2021). Smart process analytics for predictive modeling. *Computers & Chemical Engineering*, 144:107–134.

Tahir, F., Islam, M. T., Mack, J., Robertson, J., and Lovett, D. (2019). Process monitoring and fault detection on a hot-melt extrusion process using in-line raman spectroscopy and a hybrid soft sensor. *Computers & Chemical Engineering*, 125:400–414.

Takano, Y. and Miyashiro, R. (2020). Best subset selection via cross-validation criterion. *TOP*, 28:475–488.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282.

Torgashov, A. and Skogestad, S. (2019). The use of first principles model for evaluation of adaptive soft sensor for multicomponent distillation unit. *Chemical Engineering Research and Design*, 151:70–78.

Wang, B., Wang, X., He, M., and Zhu, X. (2021). Study on multi-model soft sensor modeling method and its model optimization for the fermentation process of pichia pastoris. *Sensors*, 21(22).

Wilson, Z. T. and Sahinidis, N. V. (2017). The alamo approach to machine learning. *Computers & Chemical Engineering*, 106:785–795. ESCAPE-26.

Wold, S., Ruhe, A., Wold, H., and Dunn, III, W. J. (1984). The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743.

Wold, S., Sjöström, M., and Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130.

Xu, S., Lu, B., Bell, N., and Nixon, M. (2017). Outlier detection in dynamic systems with multiple operating points and application to improve industrial flare monitoring. *Processes*, 5(2).

Yu, Y., jun Peng, M., Wang, H., guo Ma, Z., and Li, W. (2020). Improved PCA model for multiple fault detection, isolation and reconstruction of sensors in nuclear power plant. *Annals of Nuclear Energy*, 148:107662.

Yuan, X., Ye, L., Bao, L., Ge, Z., and Song, Z. (2015). Nonlinear feature extraction for soft sensor modeling based on weighted probabilistic pca. *Chemometrics and Intelligent Laboratory Systems*, 147:167–175.

Zheng, J., Ma, L., Wu, Y., Ye, L., and Shen, F. (2022). Nonlinear dynamic soft sensor development with a supervised hybrid CNN-LSTM network for industrial processes. *ACS Omega*, 7(19):16653–16664.

Zhuang, Y., Liu, Y., Ahmed, A., Zhong, Z., del Rio Chanona, E. A., Hale, C. P., and Mercangöz, M. (2022). A hybrid data-driven and mechanistic model soft sensor for estimating co2 concentrations for a carbon capture pilot plant. *Computers in Industry*, 143:103747.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.