Gaussian Process Adaptive Soft Sensors and their Applications in Inferential Control Systems

Ali Abusnina

Doctor of Engineering

University of York
Computer Science

September 2014

Abstract

Soft sensors are increasingly gaining wide popularity in industrial processes. They are becoming essential tools that contribute to building ultimately optimised and fully controlled plants due to the soft sensor's ability to address numerous hardware sensor problems. Gaussian processes are among the techniques recently utilised for soft sensor building. This research reviews the use of this technique as an adaptive soft sensor building method. It investigates different model structures, addresses issues associated with this technique, introduces Gaussian process-based soft sensors in inferential control, and proposes a methodology to enhance the reliability of the introduced inferential control system. These are achieved by conducting various case studies and empirical experiments on real and artificial data retrieved from real and simulated industrial processes. The comparative case studies conducted on various Gaussian process model structures revealed that the Mateŕn class covariance functions outperform the predominantly used squared exponential functions, particularly in clean and properly pre-processed data sets. The results show the plausibility of Gaussian processes in building adaptive soft sensors, particularly those based on windowing techniques. Specifically, empirical results have revealed that the prediction accuracy of the sensor can be improved by considering window-updating criteria. The research results have also shown that the size of raw data used for soft sensor development can be significantly reduced while preserving the informative properties of the data. This results in a significant reduction in the associated computational cost of Gaussian process-based models. Simulated results have also revealed that an adaptive soft sensor with a high prediction capability can be integrated with Proportional Integral controllers to build inferential control systems. The robustness and reliability of such a control system can be enhanced using a hybrid Gaussian process and kernel Principle Component Analysis-based method. This allows the implementation of the control system in the industrial process during both healthy and faulty operating conditions.

Contents

List of Figures

List of Figures

List of Tables

List of Tables

Acknowledgements

The research and thesis included could not have been finished if not for the support, guidance, and help of my supervisors. I would like to express my gratitude to my academic supervisor Dr. Daniel Kudenko for all of his efforts through the research years starting from accepting me as an Engineering Doctorate student until critically reviewing this thesis. Thanks, Daniel, for your critical reviews.

I would also like to sincerely thank my industrial supervisors Dr. Rolf Roth for his patience, time, and continuous support during all the days of research. Thanks, Rolf, for the fruitful discussions.

My deepest gratitude is to my previous industrial supervisor Dr. Petr Kadlec, who organised all the collaboration between Evonik and the university. Thanks, Petr, for your confidence in me.

I would also like to extend my appreciation to Evonik Industries and all the team members in the Technology and Process Engineering group.

Further, I would like to express my warm thanks and deep appreciation to my sincere friend Ayoub Blat who has always helped, supported, and encouraged me during all the research days.

My sincere appreciation is to Mr. Azdeen Albarrah for the efforts he made and the help he offered in the first year of the research

Special thanks to my parents, brothers, and sisters for every single minute they spent thinking about me. Words can never be enough to express my gratitude. Thanks, my dear family.

Finally, this thesis would not have been in this form without the understanding, support, and encouragement of one person. Thanks, my beloved wife.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author.This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References. Some of the material contained in this thesis has appeared in the following published papers:

1. A. Abusnina and D. Kudenko, "Adaptive soft sensor based on moving Gaussian process window," in Industrial Technology (ICIT), 2013 IEEE International Conference on. IEEE, 2013, pp. 1051-1056.

2. A. Abusnina, D. Kudenko, and R. Roth, "Selection of covariance functions in Gaussian process-based soft sensors," in Industrial Technology (ICIT), 2014 IEEE International Conference on. IEEE, 2014, pp. 371-378.

3. A. Abusnina, D. Kudenko, and R. Roth, "Improved performance of sparse Gaussian process based adaptive soft sensors," in MENDEL - 20th International Conference on Soft Computing. Springer, 2014, pp. 371-380.

4. A. Abusnina, D. Kudenko, and R. Roth, "Gaussian process-based inferential control system," in International Joint Conference SOCO1́4-CISIS1́4-ICEUTE1́4. Springer, 2014, pp. 115- 124.

5. A. Abusnina, D. Kudenko, and R. Roth, "Improving robustness of Gaussian process-based inferential control system using kernel principle component analysis," in International Con- ference on Machine Learning and Applications. IEEE, 2014, p. to appear.

Chapter 1

Introduction

1.1 Problem Summary

In striving towards automated industrial processes and fully optimised, monitored, and controlled plants, diverse industries (from food processing and paper manufacturing to nuclear plants and chemical refineries) seek effective tools to achieve these productivity targets for increasingly complicated processes. To fulfil the requirements of control and optimisation systems and to achieve the above-mentioned targets in complicated industrial processes, different types of variables and quantities are measured. While some of these variables, such as pressure, temperature, and flow are easily measured by simply installing a hardware device, other variables, such as emissions, biomass, melt index, and concentration cannot be measured as easily.

Classical measuring approaches implemented hardware sensors, where a diversity of affordable, accurate, and rapid sensors and transducers exist for measuring variables, such as temperature or pressure. This make such variables easy to measure. However, there are some other measurements that cannot be taken by simply installing an instrument [1]. The reason for this could be the unavailability of a suitable instrument, the high cost of the measuring device (hardware sensor), or the unreliability of the sensor. For these reasons, such measurements are considered difficult-to-measure variables [2]. An alternative solution is to measure these variables by laboratory analysis techniques, which are based on samples taken from the process and sent to labs. Those samples are taken infrequently and are time-consuming to gather and test, which makes it difficult, if not impossible, to reach high levels of process monitoring and control. This has several undesirable impacts including an increased cost of production, a high number of off-specification products, and safety or environmental issues [3].

These difficulties categorise industrial process variables in terms of their measurability into easy-to-measure and difficult-to-measure variables [3]. Modelling the relationship between these types of variables is the central idea behind inferential sensing technology. This technology was originally developed to improve the control of chemical and biological processes in the 1970s [4] and gained a wider range of application and rapid development ever since. Inferential sensors, often referred to as soft sensors, are inferential models that estimate primary variables (difficult-to-measure variables) from process secondary variables (easy-to-measure variables).

As cited in [2], on a broad level, soft sensors are classified into two main categories. Model-driven soft sensors that can be defined as First Principle Models (FPM) that focus on the steady state of the processes, describe their physical and chemical backgrounds. Data-driven soft sensors, on which this research focuses, are based on data models that describe the actual conditions of the plant based on the true measurements available. Details of soft sensor types are given in Section 2.3.

Soft sensors play an important auxiliary role in hardware sensor validation when performance declines through fault accumulation [3], as they can be used for fault diagnosis purposes.

Soft sensors are considered a valuable alternative approach to achieve the above-mentioned tasks. In the last two decades, they have demonstrated the ability to palliate and (in some cases) eliminate the problems they are developed to address. However, there are many challenges that face the development, implementation, and introduction of soft sensors into practice. These challenges are [3, 5].

- Obtaining high-quality data to develop the soft sensors.
- Incorporating expert knowledge into the development and implementation stages.
- Choosing independent variables to be used for prediction.
- Model identification and model structure selection.

Beyond these challenges, there is another recognised problem that recently attracted researchers in the soft sensor field: adaptability or the ability of the soft sensor to change its behaviour in a changing environment [6]. The potential for the soft sensor to adapt becomes important if the sensor is deployed in rapidly changing processes. It would be useless to implement a soft sensor that reflects process conditions and statuses that are not valid. The invalidity of these

predictions and reflections are caused by the historical data used to develop the sensor, as the historical data cannot contain all process conditions, especially future ones. For this reason, soft sensors need to be adaptive to preserve their prediction validity.

In addition to these concerns, identifying an adequate technique for building the soft sensor is a crucial part in the soft sensor design phase. Multivariate statistics and artificial neural networks are among the techniques that are widely reported and extensively investigated for building soft sensors. Contrarily, Gaussian processes (GPs) are marginally explored, and their importance is less emphasised. They were first used in the field of geostatistics where they are termed kriging. It is only recently when Gaussian processes (GPs) attracted the machine learning community. Specifically, Gaussian processes (GPs) were first applied in machine learning in 1996, and ever since they have been an extensive research area. The structure of Gaussian process (GP)-based models is relatively easy to determine; moreover, model predictions are accompanied by a confidence level that can be further utilised. In addition to these, expert knowledge can be incorporated into the model through the Gaussian process (GP) prior. Despite this, Gaussian process (GP) in the soft sensor domain are not extensively studied nor thoroughly investigated.

This research is dedicated to reviewing Gaussian processes and determining their practicality and applicability in the soft sensor domain. This research assesses the adaptability of soft sensors and seeks to maximise the adaptive soft sensor prediction capability by maximising data informativeness .

A known limitation in Gaussian process models becomes more evident in adaptive soft sensors, and (more crucial in certain applications) the research reviews this issue and targets this limitation in order to explore Gaussian process soft sensor applicability in building inferential control systems.

1.2 Motivation

Developing a soft sensor that can aid process monitoring and improve control system capabilities is important, as it would reduce production and operation time as well as associated costs. In addition, measuring difficult-to-measure variables, such as product quality online not only reduces off-specification material, but also helps plant operators and experts take timely and effective corrective action. This increases productivity and facilitates a smooth operation. In addition, the

adaptability of soft sensors is a crucial attribute that enhances their value and usability and further increases the importance of their role in achieving optimised, fully controlled, and monitored plants.

Developing an adaptive soft sensor that reliably, accurately, and continuously predicts difficult-to-measure variables allows building inferential control systems that control inferentially measured variables. This contributes towards building fully controlled plants, optimises plant operations, and enhances the efficient use of resources.

1.3 Research Contributions

- Improvement of the moving window paradigm as a soft sensor adaptability mechanism to effectively reflect the actual process status and process current concepts under minor and major process changes is achieved by proposing window-updating criteria as an extra parameter in building adaptive soft sensor models.
- Studying Gaussian process model selection and devising a practical recommendation in the context of industrial applications, specifically in the absence of expert knowledge, is achieved by a thorough comparative study between the widely used squared exponential covariance function and the Mateŕn class type.
- Addressing the computational time demanded by a Gaussian process-based soft sensor, particularly when the soft sensor is applied online or offline when the training data exceeds a few thousand observations, is achieved by compressing and filtering the data to extract a representative and fully informative data subset.
- Utilising Gaussian process-based soft sensors in building data-driven inferential control systems is achieved by integrating the Gaussian process soft sensor with a PI controller, and further exploring and enhancing the inferential control robustness in faulty process conditions is attained by hybridising Kernel Principle Component Analysis with Gaussian process to build the control system.

1.4 Thesis Structure

This section briefly outlines the organisation of the thesis, describes the subsequent chapters, and highlights the focal points of each chapter as detailed below.

1.4.1 Chapter 2

The outset of the chapter introduces soft sensors on a general level and briefly describes the industrial environments where soft sensors are deployed. The chapter contains a critical analysis and analytical discussion on state-of-the art in the field of soft sensors, particularly data-driven soft sensors. It explores their design procedures and applications with an initial focus on data pre-processing, model selection, and identification. The chapter concludes with a summary of open issues and pinpoints some of the questions the research aims to answer.

1.4.2 Chapter 3

This chapter starts by introducing the most widely used soft sensor building techniques, and the remainder of the chapter is dedicated to detailing Gaussian process principles and the theory underpinning these models. It outlines their advantages and limitations in addition to their applications in regression problems. The chapter then reports a comprehensive critical review focused on related work on Gaussian process-based soft sensors and open issues in this area before concluding with a brief summary.

1.4.3 Chapter 4

This chapter is a two-part chapter. The first part details the research general methodology and describes the data sets used to conduct the experiments and studies throughout the research, while the second part of the chapter details the first contribution of the research that investigated Gaussian process model selection.

1.4.4 Chapter 5

This chapter utilises the results achieved in the previous chapter and further reviews soft sensor adaptation mechanisms, specifically mechanisms based on instance selection. The contribution made is detailed in this chapter, in addition

to dealing with an issue that hinders Gaussian process models. Addressing this issue is another contribution the research is presenting in this chapter.

1.4.5 Chapter 6

This chapter details and proposes a scheme to address the problem of controlling inferentially measured variables. The scheme proposed is based on utilising Gaussian process data-driven soft sensors developed based on the results achieved in previous chapters. Besides this contribution, the chapter discusses an enhanced version of the proposed control system before it concludes with summarising remarks.

1.4.6 Chapter 7

This is the final chapter of the thesis, where the overall concluding remarks are summarised and the most significant findings are outlined. Limitations of the research and open research issues are discussed with brief details on future research directions.

Chapter 2

Soft Sensors in Industrial Processes

2.1 Soft Sensors

Soft sensors have been shown to be valuable alternatives to their hardware counterparts. This is due to their ability to measure and predict important process quantities that are difficult to measure using hardware/physical sensors, such as measuring quality indices (e.g., melt index, kappa number, particle size, and viscosity) and pollutant emissions (e.g., sulphur oxides, nitrogen oxides, and ozone).

Measurement difficulties can be technical or economical limitations that include long time delays, reduction of the efficiency of feedback control policies, lack of appropriate measuring devices, unavailability of certain instruments, and the cost associated with some measuring devices.

Difficulties include reliability issues because measuring devices are often required to work in harsh environments that require very restrictive design standards and maintenance scheduling [3]. Real-time measurements of such important process quantities are key to building a successful monitoring and control system in industrial plants. In contrast, failure to measure these may cause product loss, energy loss, resource loss, toxic by-product generation, and other health and safety problems [7]. The difficulties that soft sensors overcome and the limitations that they address establish them as a widely adopted alternative to physical sensors and an active research area among researchers in the last two decades.

What Is a Soft Sensor?

Soft sensors are a subgroup of empirical/mechanical predictive models that estimate target variables based on their correlation with readily available process independent variables.

Difficult-to-measure variables are referred to as responses, dependent variables, target variables, primary variables, or model output; whereas easy-to-measure

Table 2.1: Hardware Sensor Problems

Percentage	Recognised Problem
27	Time-consuming maintenance
21	Need for calibration
15	Aged deterioration
13	Insufficient accuracy
10	Long down time, slow dynamics
8	Large noise
2	Low reproducibility
4	Other

variables are often referred to as independent variables, process features, predictors, secondary variables, covariates, or model inputs. Target variables and input variables are used consistently throughout the thesis.

Soft sensor applications are increasing, not solely in the process industry, but also in other engineering fields. An example of which is hydroinformatics, where soft sensors are referred to as data driven models and are undergoing a fast development for their ability to replace physical based model [8].

Bioinformatics is another field where soft sensors are referred to as data driven models, and are used to derive insights from large scale biological data like for example, using kinase-activity data to predict substrate phosphorylation using PLS data driven models as detailed in [9].

In addition, econometrics is an application area where data driven models are extensively used to examine economic data and derive empirical economic relations that can be used to make different types of predictions like stock prices. They are often referred to as estimators as described in [10]. However, the term soft sensors is commonly used in chemical process to describe data driven models.

Soft sensors are capable of addressing inadequacies in hardware sensors and conventional measuring techniques. As per the results of a survey conducted in 2004 on 26 companies in Japan, soft sensors can now solve or at least help problems reported in Table 2.1 [11]. Other than the above-mentioned problems that soft sensors can address, the major objective of using soft sensors in the process industry is to:

- Measure product quality stabilisation via online predictions.
- Reduce time and costs via effective operation.
- Online validation of analyses via parallel operation.

- Back-up and fault detection devices.

For the aforementioned advantages and being a key technology for the task of controlling infrequently measured variables, soft sensors are increasingly deployed in industrial plants, particularly in chemical processes as outlined in the literature review.

2.2 Chemical Processes

A chemical process is a transformation of raw materials into chemical products through a series of interconnected intermediate transformation steps, such as reaction, separation, heating, cooling, and mixing, where the interconnection between these steps provides the overall raw materials transformation as illustrated in Figure 2.1. Chemical processes are sought to [12]:

- Efficiently use raw materials to prevent resource loss and production waste.
- Consume as little energy as is economic and practicable to minimise carbon dioxide emissions.
- Sustainably consume water to preserve its source quality.
- Flexibly operate under different conditions.
- Meet health and safety criteria.

These are the basic evaluation criteria for the design performance of chemical processes. However, to further optimise the plant and improve its performance, other adjustments and variations are pursued including structure and parameter optimisation. Data-driven soft sensors are among the tools utilised for real-time process optimisation, online process operation monitoring, and process fault detection [13]. At a general level, chemical processes can be classified as batch/semibatch processes and continuous processes.

Continuous Process

A continuous chemical process is a continuously operated process in which the inlet and outlet feeds flow continuously throughout the duration of the process as illustrated in 2.2. From a dynamic prospective, continuous processes can be steady-state or transient. Steady-state implies that the process is stable around one point and operating within a stationary region, whereas transient implies that process variables are changing progressively with time [13]. An example of a continuous process is the Haber process for ammonia production [14]. From

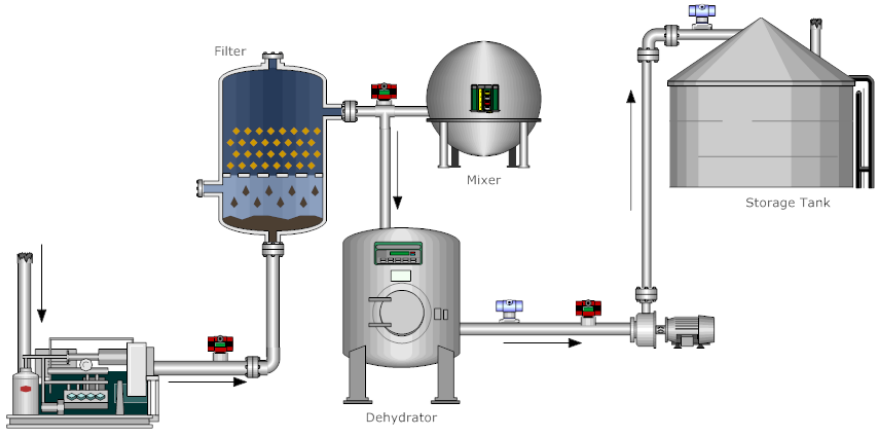


Fig. 2.1: An Example of a Chemical Process

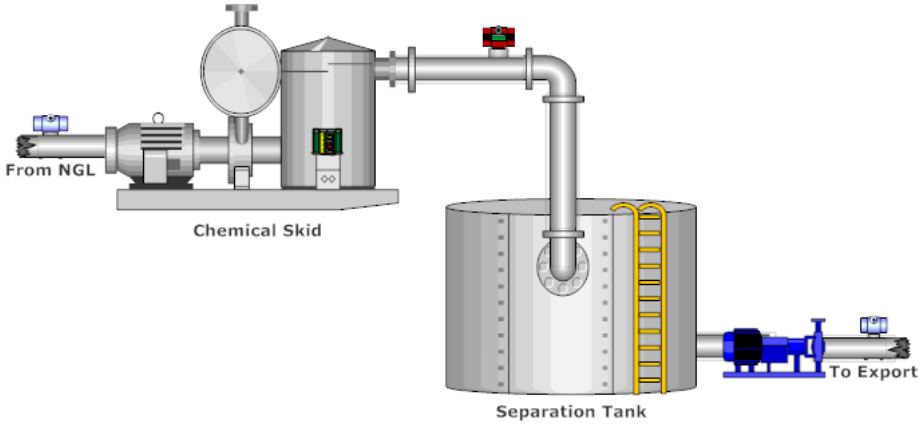


Fig. 2.2: Continuous Process Illustration

the perspective of a data-driven soft sensors, it is difficult to collect sufficiently informative data from continuous processes for soft sensor development. This is due to the dynamic behaviour of the steady-state processes, excluding start-ups, shut down phases, and abnormal operation conditions. A soft sensor based on such a dataset focuses on the description of the steady-state operation. For this reason, a non-adaptive soft sensor cannot tackle fluctuations or changes that might be progressively encountered in steady-state conditions.

Batch Process

A discontinuously operated process is where raw materials are charged at the beginning of the process, and products are delivered at the end of the batch cycle time as illustrated in 2.3. Batch processes have a definite duration and perform a fixed series of steps and tasks. In these processes, products are delivered in discrete amounts and in relatively small quantities. In contrast to continuous

processes, batch processes are always transient processes [15], and they play an important role in industry for their low-volume, high-value products. An example of these is the manufacturing of colourants, pharmaceutical products, and the biochemical industry [16]. Batch-to-batch conformity is an issue in batch processes where a subtle change in a batch operation causes deviations from different products of different batches. From a soft sensor modelling aspect, batch

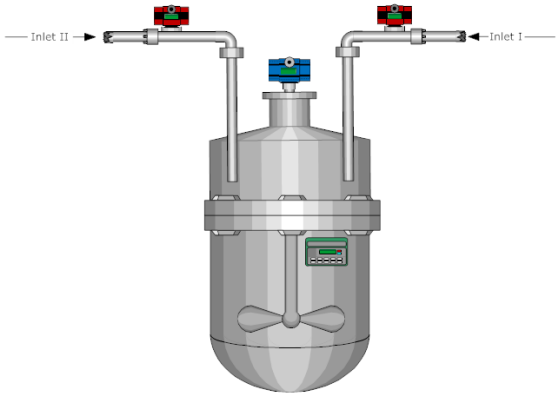


Fig. 2.3: Batch Process Illustration

processes are more challenging to model than continuous processes in that they require more sophisticated modelling techniques to account for the inherent characteristics of batch processes (e.g., process dynamics and stochastic disturbances) [17]. In addition, when developing a soft sensor for a batch process application, the developer has to take into account other factors, such as batch trajectory, high batch-to-batch variances, and starting conditions of the batches [18].

2.3 Types of Soft Sensors

Soft sensors can be categorised according to different criteria. Based on their modelling methodology, soft sensors are fundamentally classified into two primary categories: [2, 19]:

- **Data-driven Soft Sensors**
  These are black box data-based models, which describe the actual conditions of a plant, induced from input-output observed measurement data. Unlike model driven soft sensors, which will be described next, they resemble process states more realistically and describe input-output relationships more accurately. As their name implies, they are based on measured data, which can be either collected from the plant for which the soft sensor is

being developed for, or from especially designed experiments (experimental data) [2, 20].

- **Model Driven Soft Sensors**
  These are First Principle Models (FPM) that focus on the state of the processes, describing their physical and chemical backgrounds. Unlike data-driven models, this type of sensor is primarily based on deriving equations that can describe process characteristics, and due to this they are also referred to as knowledge-driven models or white box models [21]. Since industrial processes are so complex and a great amount of phenomenological knowledge about the process is required for developing such a sensor, they are often impractical, if not impossible to develop [20].

  A model that combines both empirical and physical models in estimating the target variable is termed a hybrid or grey box soft sensor [19]. Data-driven soft sensors are more likely to return more accurate predictions than their two counterparts because of the built-in flexibility; which is achieved on the expense of the significant amount and quality of training data needed during the design stage [19]. In addition, they are relatively easy to develop as intensive expert knowledge is not required. Conversely, model driven models are entirely based on phenomenological knowledge, which is often not attainable in a complex process. Consequently, the focus of this research is on data-driven soft sensors.

2.4 Data-Driven Soft Sensor Design

Data-driven soft sensors are designed to be valuable alternatives, not solely to hardware sensors, but also to model-driven soft sensors. This is primarily due to the relatively high cost for the former and the cost and complexity of the latter. In addition, modern measurement techniques in industrial plants enable data collection, storing, and analysing, which strongly suggests the use of data-driven soft sensors by manipulating the use of the available measurements in plant databases [22]. Consequently, the quality of a soft sensor is dependent on the quality of the data used for its development [23]. In addition, the design of a soft sensor largely depends on the purpose it is designed for and the tasks it is expected to accomplish. This means soft sensor design procedures for back-up purposes differ from those designed for online predictions. However, they all share a virtually unified framework that can be divided into stages as presented

in the diagram in Figure 2.4.

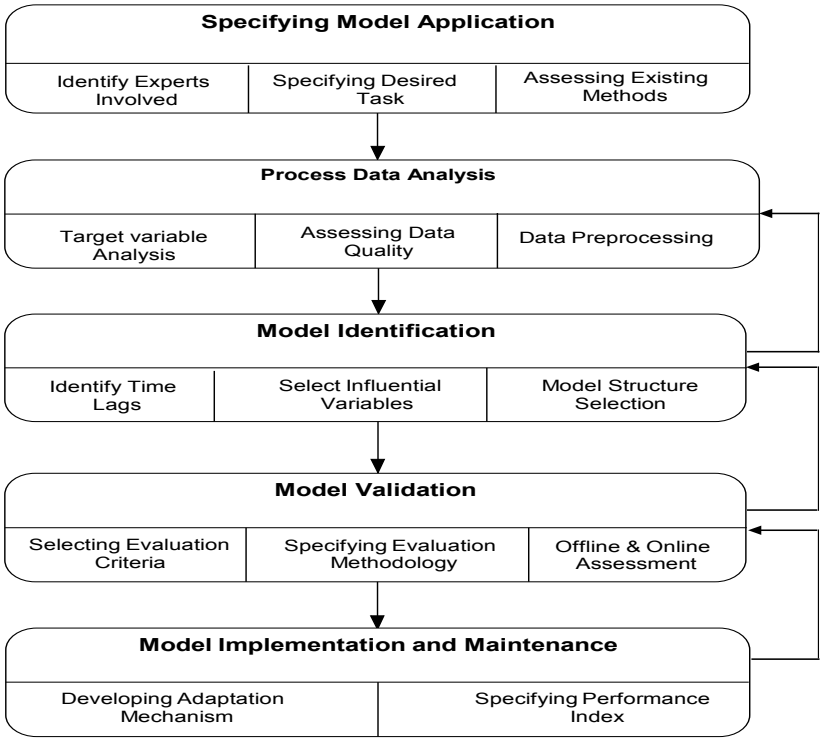


Fig. 2.4: Soft Sensor Design Block Diagram

The block diagram explains the basic classical steps of soft sensor design. These steps do not demonstrate methods to incorporate and integrate expert knowledge into those steps. Expert knowledge and use of the experience of plant experts, control room operators, and field operators cannot be emphasised enough, as it greatly improves the overall performance of the sensors and facilitates the design phase. It is overlooked in literature; however, its importance is recognised. Without expert help and knowledge, soft sensor design is a complicated and difficult task.

The design of a data-driven soft sensor is primarily based on historical data retrieved from industrial plant databases. In such plants, production is the primary goal of the process while the variable measurements, recording samples, data retrieving, reconciliation, and synchronisation are subsidiary goals [23].

This poses a number of challenges to soft sensor developers, such as evaluating data quality (e.g., reliability, accuracy, completeness, and representativeness), selecting relevant independent variables and measurements, identifying sampling rates and time lags, and extracting system dynamics from steady-state patterns of the data. As a result, data pre-processing steps that tackle these issues are

pivotal in the quality of the soft sensor as discussed in subsequent sections.

2.4.1 Data Collection and Inspection

Data inspection is a preliminary step in the design of data-driven soft sensors as it aids choosing the most influential independent variables, guides identifying data patterns, helps classifying process profiles, and allows extracting relevant information contained in the data [3]. It is usually performed as an initial step as it allows the designer to gain an overview of the data structure, for instance whether it was retrieved from a batch process or a continuous process. This step helps the developer to identify obvious problems that can be handled at an early stage. It helps to understand apparent relationships between process variables, and detect steady-state and time-varying variables. A variable x_t is said to be steady at t_0 if the ratio of change with time around t_0 is less than or equal to a certain threshold value T_x:

$$\frac{x_t - x_{t0}}{t - t_0} \leqslant T_x \tag{2.1}$$

Most of the techniques proposed in literature for Steady-State Detection (SSD) are based on Student's t tests performed on two means and the pooled standard deviation of two adjacent windows. A recently proposed method that adopts this procedure and accounts for the drift components is detailed in [13]. Detection of steady-state conditions is carried out solely for continuous processes, as batch process are transient processes and do not encounter steady-state conditions [12].

Data inspection involves a thorough assessment of the target variable to ensure that it is of a sufficient quality, variability, representativeness, and can be modelled [2]. As pointed out in [21] the target variable is most often measured using lab analysis techniques, and the quality of laboratory data is often negatively affected due to factors, such as:

- Redundant instances are not recorded.
- Human errors.
- Laboratory equipment calibration accuracy.

For these reasons, acquiring sufficient knowledge about the characteristics of the lab data, such as sampling intervals, sampling procedures, and techniques adopted, contributes to the quality of the developed sensor. Data inspection involves conducting a critical analysis in cooperation with process experts. The

cooperation normally takes the form of meetings, interviews, and informal discussions [3]. Although it is not reported in literature, this stage should involve selecting influential variables phenomenologically, running experiments, critically analysing results, and comparing the results to empirically selected influential variables.

A judicious investigation at this stage is important in order to determine whether the data collected contains any missing values, outliers, or collinearity. These data issues are results of failures in measurement devices and transmission nodes. Other operational conditions, such as process disturbances, offsets, and seasonal effects, are also possible causes of the above issues that soft sensor developers need to consider during the design stage.

Data inspection contributes to the assessment of the requirements of model complexity and determining reasonable decisions for the selection of the model and structure and whether to use simple linear models or more complex nonlinear models. In some data sets this can be done visually from the pattern of the target variable.

To sum up, this step gives the designer an idea about the procedure to be followed during data pre-processing and the solutions that are to be considered for solving data problems.

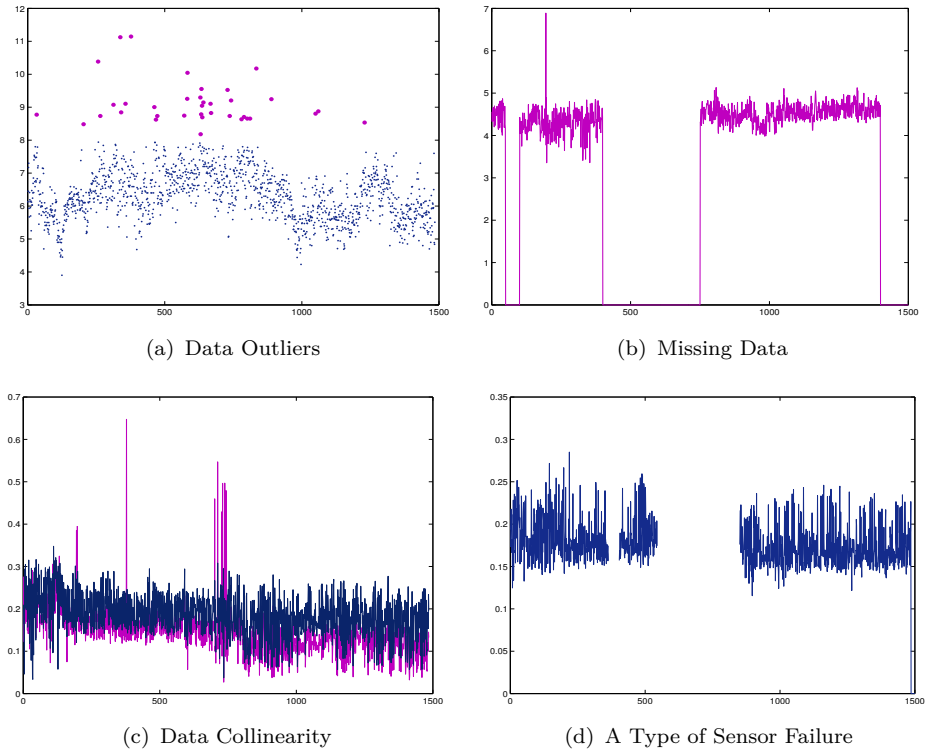
2.4.2 Data Pre-processing

Industrial datasets are collected from different sources, including online plant acquisition systems and plant laboratories. Data quality is crucial in building data-driven soft sensors and has a crucial significance on their performance [24]. In other words, a soft sensor cannot be better than the data used to develop it [3]. However, the quality of industrial data is susceptible to various factors including:

- Different types of sensor failures.
- Programming effects of plant archiving systems.
- Interaction with different sources of noise.
- Delays and sampling rates.

Therefore, data pre-processing is required to address the process data issues briefly mentioned above and visually depicted in Figure 2.5. The data pre-processing procedure involves the following steps:

- Variable Selection.
- Data missing value.



(a) Data Outliers

(b) Missing Data

(c) Data Collinearity

(d) A Type of Sensor Failure

Fig. 2.5: Common Process Data Issues

- Data outliers.
- Data collinearity.
- Delays and sampling rates.

Because all process variables are available regardless of their relevance and importance to predicting the target variable, phenomenological and empirical variable selection and identifying time lags are the very first steps considered when developing soft sensors.

Variable Selection

Variable selection or feature selection is defined in [25] as the optimal subset of the whole set. It is one of the crucial steps in soft sensor development and is directly related to the prediction accuracy of the soft sensor. Availability of all variables in plant databases does not imply that they are all influential on the target variable. Including irrelevant input variables or even less relevant inputs in the model deteriorates overall performance [26]. As cited in [27], incorrect variable selection may have unwanted consequences, such as singularity, over-parameterisation, and marked reduction of model prediction accuracy.

Reducing input space by excluding irrelevant variables simplifies model development, facilitates model structure, and reduces training time [27]. While in non-parametric models, such as Gaussian processes, this reduces the computational load associated with inverting the covariance matrix. In addition, in certain model structures, it reduces the number of hyper parameters, as the latter is determined by the input space dimensionality, and it is preferred to keep them as low as possible [28].

Striving towards the above advantages, different techniques and methods were developed for appropriate selection of relevant variables. Prior knowledge from plant experts is usually exploited initially to screen the collected variables. Then statistical or other machine learning techniques are used to refine the selection further. An example of one of those techniques is a Multi-Layer Perceptron (MLP) based on variable salience (measure of variable importance) and uses the mean squared error as a selection criterion [25]. In the MLP method, less relevant variables are excluded iteratively by a network backward search until a stop criterion is met. The latter method is computationally demanding, as the MLP is retrained whenever a variable is excluded [25]. This shortcoming is rectified in [29] through an Iteratively Adjusted Neural Network (IANN) algorithm that exploits the mean squared error and mutual information as selection criteria.

[30] argues that many approaches in literature use Mean Squared Error (MSE) as evaluation and selection criteria, which do not actually measure tracking precision between the observed target and the predicted one, instead the authors propose the Input Variable and Delay Selection (IVDS) algorithm that employs Relevance Variance Tracking Precision (RVTP) as a selection measure.

IVDS uses mutual information to select the best delay for each variable and then linearly redundant variables are removed. The algorithm then uses MLP for variable selection with MSE and RVTP as a stopping criterion. It showed that it is only suitable if the MLP model can accurately predict the target variable.

A stochastic optimisation-based method is detailed in [26] where the problem of selecting the minimised optimal subset of variables is combined with maximising the monitoring performance of a Principle Component Analysis (PCA) model. This multi-objective optimisation problem is approached using genetic algorithms and experimentally demonstrated that focusing on the relevant variables significantly improved the monitoring performance of the PCA model.

These methods assume that the data is retrieved from a single operating mode; however, this is not valid for batch processes where different operating conditions during different product profiles are encountered. For this reason, an

algorithm that selects variables for each operating mode (locally) is required. Such an algorithm is proposed in [31]. However, the selection criteria is based on the loading weights and scores of a Partial Least Square (PLS) regression model. This makes it inappropriate for nonlinear processes. The virtue of the suggested algorithm is that it does not assume a fixed correlation in the entire dataset, rather it utilises a moving window technique adaptively to select variables.

Conversely, most algorithms developed for variable selection do not consider the long-term changes in plant conditions or that these changes can influence and deteriorate the overall performance of the sensor. In [32] this issue is also dealt with by introducing an online adaptive technique such that variables are reselected in cases where new process dynamics are detected. The selection technique is a standard stepwise linear regression, while the adaptation is based on a moving window.

A novel technique that can be used to extract relationships and understand dependencies of process variables on one another is detailed in [33]. The technique is based on exploring the data by measuring the Maximal Information Coefficient (MIC). The authors argue that this measure of statistical dependence virtually fulfils generality (capturing a wide range of associations), and equality (similar noisy data with different types of relationships should receive scores similar to MIC). Although the authors demonstrated the significant performance of this technique and the usefulness of using the by-product statistical properties accompanied with the MIC calculation, it has not yet been applied when selecting variables in soft sensor design.

The procedure of selecting variables can vary depending on the problem being handled and the models being considered. [34] provides a review of variable selection in linear models, the review focuses on generalized information criterion, and regularization approaches, where selecting penalty parameters is discussed and dimensionality reduction and screening procedures are investigated.

Different factors should be considered when selecting input variables, an important one of which is the algorithm identifying the relationship between the variables. Methods successfully applied to select variables in linear models are not suited to non-linear models like ANNs. Authors in [35] review different variable selection algorithms for ANN based methods. The algorithms are categorized according to the criterion adopted, the relationship assumed, and the search procedure followed. Authors argue that a criterion like mutual information gives generic measure for both kinds of relationships linear and non-linear, in addition to being more reliable than other criteria for linear data analysis applications.

A review of the state of the art in MI feature selection methods is presented in [36] where the idea of maximizing relevance, minimizing redundancy, and identifying complementarity is clearly defined. The authors also further develops an existing unifying approach for mutual information based variable selection algorithms. The review also identifies open problems in the field such as the urgency to develop a more efficient and effective methods to select variables in high dimensional data.

A common critique of most of the variable selection algorithms is that the importance of selecting time delays before selecting input variables is overlooked. Variable selection should be accompanied by identification of corresponding time lags, where the latter has a direct substantial influence on the relationship between input variables and the target variable.

Sampling Rate and Time Lags

In chemical processes, target variables are infrequently and irregularly measured with random time delays. The irregularity and randomness arise from many factors, such as lab manual analysis, and uncertainty associated with the actual sample time. In contrast, the input variables are readily measured at various rates. Another factor contributing to the complication of the problem is the Process Information Management System (PIMS). It records new data samples solely if one of the input variables changes more than a pre-defined threshold value. These asynchronous sampling and recording times give rise to multi-rate systems, which are typical situations in industrial plants.

Therefore, one of the soft sensor design problems is the state estimation of multi-rate systems and the identification of time lags associated with each independent variable. This is as crucial as the selection of the independent variables. This step should be carried out iteratively with the variable selection stage; yet, it is not emphasised enough in the literature.

To achieve this the down-sampling technique is most frequently used; however, as emphasised in [2] extensive expert knowledge is required, which is not always available, particularly in complex plants. Consequently, a soft sensor designer appeals to empirical methods. Most methods proposed are based on simple correlation, partial correlation, and Mallows Coefficient. However, these methods are criticised in [37] as they are solely suitable for linear models. While Mallows Coefficient is suitable for non-linear models, it is computationally demanding as data dimensionality increases.

Recently, there has been a focus on exploiting Mutual Information (MI) to jointly select variables and their optimum time lags as presented in the previous section by the Input Variable and Delay Selection (IVDS) algorithm. In [37], a method is proposed that is also based on a multidimensionality mutual information estimator based on the $l-$ nearest neighbour. The two primary stages of the algorithm first transform the possible variables and their time lags into one set and (using a forward search) maximise the MI where the MI estimator is used as a cost function. The authors assert that this method has a low computation cost, it can detect non-linear relationships, and thus it can be used for nonlinear processes. Nonetheless, the suggested algorithm is tested on target variables measured using gas chromatographs. As such, it does not consider target variables and their associated uncertain random delays.

In [38] the authors suggest resolution to this issue by Expectation-Maximization (EM)-based algorithm that takes into account uncertain random delays. The algorithm treats these random delays as hidden states estimated as a model parameter. In [37] it is stressed that performing the selection of time lags before selection independent variables improves the overall performance of the sensor. Selecting variables before selection time lags may lead to biased results because variables with correct time lags may be more informative than those with incorrect time lags.

In [39] the authors recognised the significance of identifying time lags; however, the method proposed selects variables and then selects the best time lags using Lipschitz quotients. Because variables and their respective time delays are co-related, it is suggested in [40] that each time a variable is selected or deselected or a time delay is changed, the evaluation test should be repeated until the most influential variables with their corresponding time delays are identified.

Missing Values

Missing values are defined in [41] as either single or a set of values whose measurements do not represent the actual physical quantity being measured. Missing values are virtually always present in data retrieved from industrial plants. This is due to various reasons, including missing samples, measurement errors, sensor failures, sampling frequencies, database accessibility, and irregular measurement intervals, to name few. Missing values might not be present in the original dataset retrieved from the plant, rather it is introduced as a post processing step in the data [42]. The author in [27] claims that it is not surprising for many datasets

to have 10% - 30% of missing data, and for some variables (due to less frequent sampling) to have up to 90% of the data missing [27]. The relationship between a measured variable and the missing values is termed missing values mechanisms, which can be categorised as:

Randomly Missing Values (RMV)

This pattern is the simplest and infrequently found in practice. The missing values are randomly scattered over a vector of observations without a specific pattern. It typically arises in a process where different sensors are monitoring one quantity at a time when one of those sensors temporarily stops functioning. RMV may also occur in processes where quantities are measured at different intervals.

Randomly Missing Spectra (RMS)

This pattern resembles cases where the entire spectrum is randomly missing. It usually occurs in processes where a multivariate instrument (e.g., spectrometer) is monitoring a specific quantity in time. If for any reason the multivariate instrument malfunctions, the whole spectrum may not be measured. Analogously, if a single sensor is monitoring a certain quantity and it malfunctions without replacement until the end of the process, it causes the whole block of measurements to be missed.

Systematically Missing Value (SMV)

Systematic missing values appear if different sensors are measuring a certain quantity at different sampling rates, such that missing values are constant over the samples.

Different approaches have been suggested to remedy the issue of missing values. In [22] the authors developed a soft sensor for a cement kiln system to estimate two product-quality indices. The authors proposed a procedure to deal with the missing data. They defined a template using the kiln drive measurement to identify the length of the missing block; if the block was less than two hours, interpolation based on neighbouring values was inserted (likelihood), otherwise the entire block was discarded (case deletion). This is not adequate, particularly for chemical data sets, which are usually multivariate datasets, as the discarded block might have useful predictive information.

In [27] the authors compared and summarised different approaches to dealing with missing data that included single imputation and multiple imputation. In the former, missing values are replaced in single steps, which generally lead to data distortion and biased derived statistics of the data, while the latter replaced the missing values in a multiple iterative manner. This can produce unbiased and accurate estimates provided a sufficiently high number of imputations are drawn [43].

In [44] three different neural network-based soft sensors were developed for real-time prediction of cement clinker properties using three different data sets. The missing values were treated by linear interpolation imputation. Linear interpolation was demonstrated in [45] where it outperformed the straightforward mean imputation method. Although the latter is widely adopted, it was theoretically proven and empirically demonstrated in [46] that when the input variables are correlated and the mean imputation is adopted, soft sensor prediction accuracy rapidly deteriorates.

Data Outliers

Outliers are observations that abnormally deviate from the normal data variability such that they seem to be generated from a different mechanism, and they do not follow any pattern. They are generally caused by inaccurate observation, inappropriate recording, incorrect data entry, sensor malfunction, and transposing values when recording measurements [47]. Outliers can be divided into two categories [41, 48]:

- **Obvious Outlier**
  These are values that violate physical and technological limitations and can be detected by providing the detection system with upper and lower limiting values.

- **Non-obvious Outliers**
  These are values that lie within the typical ranges of the quantity being measured but do not reflect the correct variable status and do not violate any limitations. Such types of outliers are harder to detect.

At a very general level, outlier detection techniques are classified as follows [49]:

- Distribution Based
- Distance Based

- Clustering Based
- Depth Based
- Density Based

Detecting and handling outliers are critical steps in improving soft sensor prediction accuracy. In contrast, the failure to detect or correctly address outliers can have serious ramifications on the prediction accuracy of the soft sensor[41]. However, the influence of the outliers on prediction accuracy depends on the method used for soft sensor development [50]. For instance, in a multi-model-based soft sensor, data cannot be appropriately clustered if outliers exist, which leads to poorly predicting sub-models. Conversely, in Gaussian process regression models with Student's t noise, outliers can have a weaker effect, and prediction accuracy is widely unaffected [51]. That is because the effect of one single outlying observation on the predictive posterior distribution becomes more negligible as this observation tends towards infinity. In contrast to this, is the Gaussian noise model assumption, where every single observation influences the posterior regardless its location whether it is in the tails or near the mean of the distribution. This robustness of t-distribution can altered from very heavily tailed data to Gaussian by varying the degree of freedom [52].

A simple outlier detection technique that is frequently employed is the 3σ [22] and it is argued in [53] that it is the most widely used heuristics to detect outliers, in addition, authors in [54] declared that this technique is the best-known criterion to detect an outlier. The 3σ rule states that nearly all (99.73%) of data points lie within the boundaries of 3 standard deviations of the mean. If a data point exceeds these boundaries, it is classified as an outlier.

However, this approach cannot distinguish between outliers and extreme data variations, which results in detecting outliers in clean data that might be an informative observation. In addition, if it is employed online, it cannot distinguish between new process dynamics and outliers [22]. Moreover, it is likely to have poor performance in when the sample size is small [54]

This shortcoming is addressed in a more sophisticated, but effective technique proposed in [50], where outliers can be detected online, and the algorithm still discriminates between new process states and outliers. The suggested method is based on measuring the Squared Prediction Error (SPE) of the target variable and the input variables whenever a new data point is available. If the SPE violates a predefined threshold, it is classified as suspicious and is investigated further.

SPE is given by:

$$SPE = \sum_{i=1}(y_{new,i} - \hat{y}_{new,i})^2 \tag{2.2}$$

where $y_{new,i}$ and $\hat{y}_{new,i}$ are the ith real target and the ith soft sensor prediction, respectively. The basic assumption is that if the new data point is an outlier, then the time series in the SPE encounters a short step and the change will not be sustainable, while if the new data point is a new process state, the SPE encounters a ramp disturbance or a sustainable step.

These techniques are based on the detection strategy, and if outliers are inaccurately detected, soft sensor prediction accuracy will be susceptible to deterioration. For this reason, more robust algorithms are needed to accomplish the task.

Aiming to minimise the effect of outliers on the soft sensor model, the authors in [55] propose Least Square-Support Vector Machine (LS-SVM) based on a fuzzy pruning approach, where the sensitivity of the algorithm to outliers is reduced through membership scores assigned to data observations. Despite the availability of those methods, soft sensor designers still face the challenge of choosing the optimal technique that best suits their data and that best balances the sophistication of the method and ease of the implementation.

Data Collinearity

As far as process control is concerned, it is necessary to have detailed measurements of all process variables and to have redundant measurements for certain other variables. This delivers highly correlated measurements that are data rich, but information poor as described in [41]. This data characteristic is termed collinearity [56].

Collinearity and near-collinearity are issues encountered in data acquired from industrial plants. Whereas the former refers to the linear relationship between two or more independent variables [3], the latter refers to closely related linear variables (i.e., when the correlation matrix of a set of variables is near singular, the variables are said to be near collinear). This linear dependency can be a result of a duplicated variable or its offset version, or by one input variable that is equal to the linear combination of other variables.

Collinearity has negative effects on the data and causes instability in the coefficients of the least squares regression models. Therefore, in the presence of collinearity developing models (in particular Ordinary Least Squares (OLS) models) make the model predictions susceptible to deterioration whenever slight

disturbance occurs in process dynamics. Collinearity can take three distinct forms in the context of linear regression problems [57]:

- Little Variability
- Large Correlation
- Exact Linearity

Correlated variables increase the complexity of the model, reduce its interpretability, and may deteriorate its prediction accuracy [2]. The author in [58] explains that collinearity is a more of an obstacle in understanding relationships within process variables than an obstacle in building a perfect model.

As cited in [58] Hocking suggests a measure of collinearity for a set of variables by decomposing their correlation matrix using PCA to assess the degree of collinearity. Hocking argues that if the smallest eigenvalue is less than 0.05, then the variables are seriously collinear, and they are moderately collinear if the eigenvalues are less than 0.10.

The most common techniques for dealing with this issue are PLS and PCA. PCA is based on decomposing the data covariance matrix, where eigenvalues are then sorted in order of significance (i.e., the first principle component has the maximum variance). Transposing the new vector and multiplying by the original data results in a new informative uncorrelated dataset. An industrial case study is demonstrated in [59] in which a soft sensor based on PCA-ANN is developed to predict parameters in a petroleum reservoir. In the hybrid algorithm suggested, PCA is exploited to address collinearity in the input variables and extract the most relevant variables.

However, classical PCA has some drawbacks, for example, it has a poor interpretation of the actual process variables, it is prone to error in presence of outliers, it is inadequate in dealing with nonlinearities, and it is time invariant.

These issues are addressed by different extensions of PCA to nonlinear PCA as cited in [27]. Recursive PCA has been used to adapt to process changes and remedy the problem of time invariance. However, a common disadvantage to PCA-based methods is that these techniques do not account for correlation between input and target variables. Conversely, PLS methods simultaneously maximise the variance in input variables and maximise the correlation with the target.

In [60] a novel approach based on PLS and False Nearest Neighbour (FNN) was proposed to deal with variable selection (discussed in section 2.4.2); however,

the first part of the algorithm deals with the multi-collinearity problem. PLS is used to extract principle components and reduces the complexity of the model.

A recent algorithm proposes the use of mix-PLS to build a soft sensor designed to predict the viscosity in a polymerisation batch process. In the proposed algorithm, the collinearity of the input data is addressed using PLS, and it is argued that the use of PLS to address the collinearity allowed mix-PLS to be less susceptible to over-fitting [61].

2.4.3 Model Identification

Model identification is the procedure of building plant empirical/mechanical models based on data observations retrieved from that plant or based on phenomenological knowledge acquired from experts of that plant. This step comprises model structure selection and model parameter optimisation. As the focus of this research is on data-driven soft sensors, First Principle Models of soft sensors are not considered here. To accurately describe model identification from a dataset, we consider data acquired from a chemical process $\mathbb{D} = \{\mathbf{x}_i, y_i | i = 1, \cdots n\}$, after pre-processing, assuming a Multiple Input Single Output (MISO) model; it can be represented as an input matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ and an output vector $\mathbf{y} \in \mathbb{R}^{n \times 1}$.

Model identification is finding a function f that best approximates the dependence of the output vector on the input matrix $f \colon \mathbb{R}^{n \times 1} \rightarrow \mathbb{R}^{n \times m}$. The approximation function is used such that $y = f(x, \Theta)$, where y^* is the output corresponding to any unseen test input $\mathbf{x}^*$, and Θ is a vector of parameters.

As noted in [62], in data-driven models flexibility, generality, and simplicity are the three primary criteria for model structure selection. However, flexibility might reduce generality and compromise between the two is a key element in the model structure selection step.

The model structure should match the key characteristics of the process, such as whether the process is static or dynamic, linear or nonlinear, constant or time-variant. In industrial processes, dynamic models most often resemble the reality of the process more than static models. This is because even steady-state processes encounter dynamic behaviours during different operation phases, such as process fluctuations, shutdowns, start-ups, and depressurisation [63].

A theoretical methodology or superior approach for model selection does not exist; however, there are some techniques that can aid in the accomplishment of this task. Kadlec in [2] suggests starting with simple models, such as linear models; model complexity can be gradually increased by considering nonlinear

structures based on improved or deteriorated performance. Neural networks as declared in [64] are the most popular nonlinear approximators and represent a large class of model structures. Fuzzy approximators, on the other hand, can be successfully used to represent linguistic knowledge and add it into the model [65].

Model selection depends on the training data available, the type of application, the framework of the model structures considered, and the problem domain the soft sensor is aimed to solve [66]. K-fold cross-validation can be used for a proper model selection when there is insufficient data. This method can make use of all available data by partitioning it so that it can be used for validating the model [2, 19].

In Bayesian-based methods, a set of model structures are evaluated and selected based on Bayesian Information Criterion (BIC) given by 2.3.

$$BIC = L(\theta) - \frac{n_\theta}{2} logn \tag{2.3}$$

where $L(\theta)$ is the marginal likelihood of the model hyper parameters, n_θ is the number of hyper parameters in the model, and n is the number of observations in the training data [67]. BIC minimises the prediction error of the model and penalises for the increased number of parameters. Thus, it tends to select parsimonious and accurate models.

Model selection in Gaussian process mounts to the selection of the covariance function and the associated hyper parameters [68], and thus it is relatively simple to select Gaussian processes models. However, Gaussian processes are still not as widely adopted as other soft sensor building techniques (e.g., Artificial Neural Networks (ANN), PLS, and Principle Component Regression (PCR)). Despite the simplicity of their model selection, there is no clear methodology or criterion that guides the choice of the Gaussian process model. Contrarily, all reported Gaussian process-based soft sensors used a widely reported model that comprises squared exponential covariance function and a zero mean function. Therefore, a heuristic or/and meta-heuristic practical method is required to aid in the development process of Gaussian process-based soft sensors.

Once the model is selected, model parameters are optimised in the second step of the model identification procedure. In Gaussian process-based models, hyper parameters are usually optimised by maximising the marginal log likelihood of the model [69]. However, if the Gaussian process-based soft sensor is deployed online, it might be not computationally feasible to perform the optimisation for

every test point.

2.4.4 Model Validation

Model validation is the final step in soft sensor design and is defined in [70] as the level of agreement between the identified model and the system being modelled. Once the model is identified, its performance is evaluated on an unseen set of data to check model plausibility, model falseness, and model purposiveness [70].

Plausibility indicates whether the model conforms to the prior knowledge of the process (whether it behaves logically). Falseness reflects the agreement between the real process output and the model output, while purposiveness expresses model practicality and whether a model satisfies the purpose for which it is developed.

An identified model can perform satisfactorily on the identification/training data and simultaneously perform poorly on the validation or test data [21]. This phenomenon is known as over-fitting and indicates a poor model generalisation capability. In [3] model validation criteria are divided into two groups: techniques that derive an analysis to understand model residual properties and other techniques required by the soft sensor application.

In industrial processes it is frequently challenging to acquire a sufficient amount of data to develop the model and validate it; therefore, it is advised in [2] to resort to statistical techniques such as cross-validation or re-sampling methods. However, re-sampling techniques such as bootstrapping and bagging, as [71] concludes are effective but computationally demanding.

Model validation can be carried out quantitatively by measuring Pearson correlation-the most common measure of correlation in statistics - which evaluates the correlation between the observed and the predicted values and is given by:

$$R = \frac{\sum_{i=1}^{n}((y_i - \bar{y})(\hat{y}_i - \hat{\bar{y}}))}{\sqrt{\sum_{i=1}^{n}((y_i - \bar{y})^2 \sum_{i=1}^{n}((\hat{y}_i - \hat{\bar{y}})^2}} \tag{2.4}$$

Where $\hat{\bar{y}}$ is the mean of the predicted variable, and $\bar{y}$ is the mean of the observed variable.

Quantitative model validation can also be done by measuring MSE given in

2.5 to evaluate the overall prediction performance in terms of the accuracy and reliability of predictions.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.5}$$

where y is the observed target, while $\hat{y}_i$ is the predicted target, and n is the number of test samples. Alternatively, Relative Mean Squared Error (RMSE) given in 2.6 is often used as an alternative to MSE for its interpretability.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.6}$$

The size and the dimension of the target variable impact RMSE as criticised in [72]; therefore, the authors recommended using Relevance Variance Tracking Precision. RVTP given by (2.7) reflects whether the soft sensor preserves its prediction accuracy when input measurements vary. The best approximating model has a maximum value of RVTP.

RVTP is a measure of the precision of the soft sensor output. (i.e., how the precision changes when the output changes) and is given by:

$$RVTP = 1 - \frac{\sigma^2_{error}}{\sigma^2_{measurement}} \tag{2.7}$$

where σ^2_{error} and $\sigma^2_{measurement}$ are the prediction error variance and the measured output variance, respectively.

Another quantitative validation measure suggested in [73] is Theils Inequality Coefficient (TIC). It provides a measure of how well a predicted time series of values corresponds to another observed set of time series values. It is given by:

$$TIC = \frac{\sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_i)}}{\sqrt{\sum_{i=1}^{n} y_i^2} + \sqrt{\sum_{i=1}^{n} \hat{y}_i^2}} \tag{2.8}$$

where y and $\hat{y}$ are the observed target and the predicted target, respectively. TIC ranges between zero and one; the closer the TIC value is to zero, the better

the prediction quality of the model. It is asserted that values of TIC that are smaller than 0.3 indicate good model performance; however, such values depend on the application and the modelled problem. It should be emphasised that TIC is a more appropriate measure for comparison of different model than for indication of model quality in an absolute sense. Despite its applicability, TIC has not yet been reported for soft sensor validation or identification purposes.

It is more appropriate however to evaluate the performance of non-parametric models in terms of the entire predictive distribution which can be captured by the predictive density error given by:

$$LD = \frac{1}{2}\log 2\pi + \frac{1}{2N}\sum_{i=1}^{N}(\log \sigma_i^2 + \frac{e_i^2}{\sigma_i^2}) \tag{2.9}$$

where σ_i^2 is the (i-th) prediction variance and e_i^2 is the error between the $(i-th)$ actual output and its corresponding prediction. This loss function takes into account model accuracy by measuring the prediction error and model uncertainty by measuring the variance.

By subtracting the loss of the model that predicts using a Gaussian with the mean $E(\vec{y})$ and the variance σ_y^2 of the training data from the model LPD, The mean standardised log loss (MSLL) is obtained by as given below [69]:

$$\text{MSLL} = \frac{1}{2N}\sum_{i=1}^{N}\left[\log(\sigma_i^2) + \frac{(E(\hat{y}_i) - y_i)^2}{\sigma_i^2}\right] - \frac{1}{2}(\sigma_y^2 + E(\vec{y})). \tag{2.10}$$

The MSLL is approximately zero for simple models and negative for better performing ones.

Model validation can also be performed qualitatively. This involves plotting the time series of the model predicted target and the observed target. Overlaying the plots gives an immediate indication of the level of agreement between the two plots. It can also provide insight regarding model deficiencies and where improvements are necessary. Residual plots aid analysing any potential agreements and disagreements between the predicted and the observed targets [73]. Model validation is often under emphasised despite its importance [74]; the approach to this step depends on soft sensor specifications, customer requirements, and the application area for which the soft sensor is developed [2].

2.5 Soft Sensor Applications

Soft sensors have a broad range of applications and can fulfil a wide range of tasks. This is primarily due to their relatively low cost in comparison to their hardware counterparts; in addition, hardware sensors require acquisition, installation, operation, and maintenance costs. Soft sensors are becoming routine tools, not only as an informative source to control room operators, but also for closed loop inferential control [3].

It is reported in [75] that ten years ago 80% of soft sensors were deployed in open loop applications while 20% were used in closed loop applications, whereas now 80% are being used in closed loop control applications. Nonetheless, data-driven soft sensors (as yet) do not represent the reported percentage as detailed in Section 2.5.3.

The dominant application area of soft sensors is the online prediction of difficult-to-measure process variables. However, other application areas include process monitoring and fault detection [76]. A review of these applications is given below with a focus on soft sensors based on Gaussian processes as a building technique.

2.5.1 Online Predictions

Predicting process target variables in a real-time manner is an application area that elevated soft sensors as critical optimisation tools in industrial processes. In contrast to this approach, samples are taken to labs for offline analysis, which creates long time delays or variables are measured online by an analyser, which creates time delays and reliability issues.

Predicted targets are frequently quality indices or another key process variables and predicting these continuously online is of paramount significance for process operationalisation and resources optimisation. An example of a Gaussian process-based soft sensor tasked with the online prediction of a superheated steam flow in a power generation plant is reported in [77]. The suggested algorithm addresses online computational complexity with a two-fold procedure: i) the hyper parameters of the Gaussian process model are not re-optimised for every new data point added into the model training matrix; they are retained until an error threshold point is reached, and ii) the model training data matrix is updated recursively through a pruning technique that excludes redundant points. The limitation of the suggested procedure is that the pruned data is selected based on

an absolute error criterion, which does not account for redundancy. This can be addressed by introducing a similarity measure (e.g., Euclidean or Mahalanobis distance). Such measures account for the informativeness and data variability in the training data matrix.

Typical data-driven soft sensors developed for online predictions are based on supervised learning approaches; the training data comprises both input data and corresponding target variable measurements. This raises the issue of data scarcity, as the target variable is not measured as frequently as the input variables. To remedy this issue, [78] proposes a mixture of semi-supervised probabilistic principal component regression models that incorporate all input data, including the unlabelled inputs. From a probabilistic prospective, this approach to data utilisation improves the estimation for data distribution, which improves prediction accuracy. The idea behind the mixture of models is to address the issue of multi-mode nonlinear non-Gaussian variables. However, the algorithm lacks an adaptivity mechanism to account for process behaviour changes.

A recent and innovative ANN-based soft sensor was applied to an industrial case to estimate the heavy diesel 95% cut point of a Crude Distillation Unit (CDU) was proposed in [64]. The novelty in the suggested approach is the adoption of deep learning as a training strategy for deep neural networks. The virtue of the proposed algorithm is the network structure that: i) yields more accurate predictions, ii) handles highly correlated data, and iii) efficiently deals with big data. Although the novel deep neural network algorithm outperformed the traditional ANN, Support Vector Machine (SVM), PLS, and ANN-PLS, it is computationally more challenging. The authors argue that the algorithm reduced estimation errors such that it can be employed in a quality control loop (inferential controller). However, the computation time could impede the applicability of the algorithm in inferential control, particularly if the online computation time is longer than the sampling interval of the Distributed Control System (DCS) of the plant.

2.5.2 Process Monitoring

Product quality and process safety are two primary criteria considered during the design of industrial processes. It is often infeasible to have a fully monitored process, as this necessitates building heavily instrumented plants with high costs. Therefore, soft sensors are considered an alternative to enable monitoring process safety and enhancing product quality. Process monitoring is defined in [22] as de-

tecting abnormal process operations resulting from a shift in the mean or variance of one of the process variables. As noted in [79] process monitoring is also termed fault detection and diagnosis. Monitoring processes allows problem diagnostics and highlights process weak points. It provides a framework that can be utilised for product quality improvement. This is particularly useful in batch processes, as it reduces the number of rejected batches, allows fault detection before the batch is completed, and helps mitigate faults in subsequent batches[80].

The typical inherent characteristics of batch processes (e.g., finite duration, nonlinear behaviour, insufficient online sensors, and change of production grades) make the task of developing online monitoring soft sensors more complicated than in continuous processes.

The lack of appropriate monitoring techniques represents another obstacle to building an effective monitoring system. Although classical techniques of monitoring and fault detection are simple and reliable, they react late and in-depth diagnosis is usually not possible [81]. Therefore advanced techniques that can fulfil the following tasks are needed:

- Early detection of faults.
- Diagnosis of faults of sensors, actuators, and processes.
- Monitoring and supervision of processes.

Several techniques are used to accomplish the monitoring task and fulfil the aforementioned requirements; Multivariate Principle Component Analysis (MPCA) is among the techniques employed. The idea behind this method is extracting information from process data by compressing and projecting it onto a lower space. The data is stored in three items: variables, time, and batch number. Once the normal behaviour of the process is learnt, new batches are monitored by comparing the time progression of the projections in the reduced space with the normal batch data. However, multivariate statistical process control-based methods such as PCA and PLS are based on easily violated assumptions such as Gaussian distributed process data, linearly correlated process variables, and single-mode operated processes [79]. Although kernel PCA can address some of the aforementioned limitations (e.g., the linear correlation assumption), it cannot cope with multi-mode operated processes [82].

To rectify this issue, the authors in [80] modified the MPCA technique to build Multivariate Independent Component Analysis (MICA). While PCA searches for Gaussian components, ICA searches for non-Gaussian components. As such MICA provided a more meaningful online monitoring tool compared to the MPCA.

The limitations of ICA-based methods are that their monitoring results are unstable; in addition, it is difficult to select the number of independent components.

As Gaussian process is a relatively new option for soft sensor building, Gaussian process-based soft sensors for process monitoring have (as yet) merely been reported in [82]. Yet, the objective application of the proposed sensor is online prediction, and it was subsequently utilised for process monitoring. The proposed adaptive soft sensor, which is based on Gaussian mixture models, was outperformed by the widely used recursive PLS and recursive PCA. This could be due to the developing method adopted, where the objective application of the soft sensors is a primary factor that determines the development methodology.

To continuously monitor online NO_x and O_2 emissions from industrial boilers, [83] proposed an efficient Radial Basis Functions (RBF) network soft sensor to accomplish the emissions monitoring task. The soft sensor was developed offline using experimental data and validated using real data acquired from the boiler plant. The results confirmed soft sensor plausibility and purposefulness. In addition, the authors asserted that the soft sensor could be utilised in an inferential control system and be integrated with the boiler controller to optimise operation and maintain production.

2.5.3 Data-Driven Inferential Control

Building fully optimised and controlled processes involves building heavily instrumented plants and calls for monitoring and measuring key plant variables and quality indices (target variables). However, on one hand, such a task is frequently not feasible due to the difficulty of directly and continuously measuring these variables, and on the other hand, accomplishing the task contributes to a safer working environment, less environmental impacts, more enhanced product quality, more efficient use of resources, and lower levels of waste due to off-specification products.

Inferential control schemes were introduced to circumvent this problem and competently control the infrequently measured target variables. Systems controlling inferentially measured variables are termed inferential control systems. The term "data-driven inferential control" stems from the data-driven estimation technique employed for the prediction of the target variables.

The authors in [84] note that the two key elements of an inferential control system are the estimator (e.g., soft sensor) and the controller. Inferential control systems are employed in different schemes in industry. Foremost among them is

Model Predictive Control (MPC). It utilises an explicit process model to compute a control signal and optimise a cost function based on the desired output trajectory over a predefined length of prediction horizon [85]. They are extensively utilised due to their ability to handle constraints, uncertainties, and time-varying system dynamics[86].

Data-driven soft sensors have been integrated with this type of control scheme in designing inferential control systems. An example is detailed in [87], in which a data-driven Latent Variable-MPC system is proposed to control product composition in a distillation column (continuous process). Identification of the model cannot be effectively performed if the control variable is not sufficiently excited, and if not feasible to excite, the paper proposes performing the identification procedure in the latent variable space.

The other reason behind the implementation of MPC in the latent space is to reduce the computational complexity of the MPC and the correlation among the manipulated variables. The cost function that the MPC is minimising is the sum of the squared two-norms of prediction error and control actions, where predictions are attained by a multi-step ahead PLS soft sensor.

The proposed algorithm outperformed the conventional data-driven MPC detailed in [88]. However, as the accuracy or suitability of MPC actions are based on the prediction accuracy of the controlled variable, the overall system performance significantly deteriorates if the process exhibits any time-varying behaviour. Thus, the consideration of adaptivity helps to alleviate this issue. In addition, as the soft sensor is PLS-based, the suggested system is not suited to nonlinear processes. Nonlinear and dynamicity are typical characteristics of industrial processes.

[89] developed a Data-Driven Direct Predictive Control (DDPC), which is primarily based on multi-modal data-driven predictors developed using subspace identification, to address the above-identified shortcoming. The suggested control system eliminates the need for MPC modelling, and thus the problem of model mismatching is addressed, computational complexity is significantly improved, nonlinearity is tackled by the multi-modal approach, and the adaptivity is considered through an online update strategy. The algorithm is validated on a boiler turbine unit to regulate power output in safe and friendly-environmental conditions.

Although the suggested system fulfils the task it was developed for, it is demanding in terms of engineering and implementation. As indicated, the performance of the MPC-based control system is dependent on the validity of the

process model [90], where a model-process mismatch (resulting from employing a linear model in nonlinear processes) can deteriorate process behaviour predictions and consequently reduce control performance. To ensure that the model is not used outside the range of its validity, a movement suppression weighted term is used that sets the controller degree of freedom. However, difficulty adjusting the term and the controller sensitivity to the term are drawbacks of this strategy, which are addressed in [91], where the authors propose defining four validity indicators on model performance. The defined indicators are integrated with the MPC optimisation function as constraints to bound the controller decision space and ensure a valid utilisation for the model.

The drawback of the suggested method (as pointed out by the authors) is that the constraints result in a computationally expensive optimisation problem. An alternative and simpler indication of model validity and process-mismatch is offered by Gaussian process models. Such models provide confidence intervals (uncertainty) associated with model predictions. These uncertainties are harnessed in [92] to design a nonlinear model-based predictive control. The authors argue that the suggested strategy allows the MPC to optimise the manipulated variable such that it avoids regions with high uncertainty. This results in a more robust system that optimises actions according to model validity.

Controlling a continuous process is different from a batch process. In the former the control objective is to reach the equilibrium point, while in the latter the primary objective is to attain a specific product quality by the completion of the batch cycle [93]. Batch process direct control is frequently impractical; therefore, other approaches are utilised. An example of which is Statistical Process Control (SPC) and Midcourse Correction (MCC) in which the target variable is predicted (via a soft sensor) at every sample interval, and midcourse corrective actions are taken accordingly. However, as the target variable is estimated during the run of the batch, measurements of easy-to-measure variables are available solely up to the current sampling instance, and for predictions that are more accurate, future measurements are required. Consequently, control actions will be based on a poorly predicted quality.

To alleviate this problem, [93] proposed an algorithm that integrates two data-driven soft sensors with model-predictive control. The first sensor is a multimodal soft sensor that predicts the future process conditions based on the batch input trajectories, which are embedded in the MPC optimisation function (making use of MPC prediction horizon), while the second soft sensor relates these predicted batch conditions with the final product quality.

In addition to MPC-based controllers, a Proportional Integral Derivative (PID) controller is a type of feedback control that minimises the error between the measured variable and the process set point. It is widely adopted in industry and represents more than 95% of the control loops deployed, most of which are Proportional Integral (PI) controllers [94].

Soft sensors have been integrated with this type of controller in different industrial applications. An example was demonstrated in [95] where a Dynamic Partial Least Square (DPLS)-based soft sensor is designed to predict the kappa number (quality index) in a Kaymr digester process. The predictions are fed back to a Proportional Integral Derivative controller to minimise variations of the Kappa number and maximise pulp quality. Simulated studies demonstrate that the soft sensor can capture the process dynamics in both closed and open loop modes. However, because some chemical processes are of a higher order and in order to alleviate the influence of the initial state of the process, the past horizon of the DPLS model should be much larger than the process order. This necessitates including a large number of input variables including their lagged value, which results in a slower response to process change in addition to the difficulty of maintaining and building the model. Therefore, the authors proposed a Reduced Order Partial Least Square (RD-DPLS) that reduces the system order by estimating time delays and using time-shifted variables. It has been shown that the RD-DPLS has superior performance over the Dynamic Partial Least Square (DPLS). Nonetheless, both soft sensors cannot handle process drifts and do not account for process time-varying behaviour.

To remedy the drawbacks identified, [96] proposed a recursive RD-DPLS that is constantly updated online. The authors argued that the Proportional Integral Derivative (PID) fed back by the soft sensor predictions significantly outperformed the one fed back by kappa number lab measurements. Regardless, lab measurements are not as frequent as predictions for the soft sensor, consequently, the Proportional Integral Derivative (PID) fed back by those infrequent measurements performs incompetently.

In addition to multivariate statistical-based soft sensors, Artificial Neural Networks (ANN)-based soft sensors are also utilised in Proportional Integral Derivative (PID) closed loop controls. A Recurrent Neural Networks (RNN) soft sensor was reported in [97] to predict a column distillation product composition, which is then fed back to decentralised PI controllers to control a reactive distillation process. The suggested RNN approach was compared with Extended Kalman Filter (EKF) and Feed Forward Neural Network (FFNN) in open and closed loop

modes, where it quantitatively and qualitatively outperformed both types of neural networks, (RNN and FFNN). Yet the outperformance of the EKF is at the expense of computational complexity and the engineering and modelling effort involved.

In a comparable application area, specifically in a multicomponent distillation process, two different ANNs algorithms (Levenberg Marquardt (LM) and Adaptive Linear Network (ADALINE)) are compared in designing soft sensors. The sensors are then utilised as feedback elements in a Proportional Integral Derivative (PID) controller to control product composition in the mentioned process [98]. Experiments show that the ADALINE soft sensor's performance is superior to that of the LM soft sensor. Based on this result, the ADALINE soft sensor was further improved by developing a dynamic version that demonstrated even better performance. Despite the achieved results, ANN-based models have some limitations, such as random and uncontrolled convergence, local optima issues, and difficulty identifying model structure.

2.6 Summary

In terms of their metastability, industrial process variables can be categorised as easy-to-measure variables or difficult-to-measure variables. The importance of measuring difficult-to-measure variables arises in applications where such variables are the key to building successful monitoring and control systems and where the failure to measure them may result in undesired complications.

This chapter discusses the increasingly indispensable role of data-driven soft sensors in industry, particularly in chemical processes. It briefly describes industrial data issues, reviews the data pre-processing techniques that address them, and highlights some of the limitations of these techniques.

The review outlines the general procedure for soft sensor design and highlights some of the limitations therein. For instance, the current adopted procedure does not fully exploit expert domain knowledge, and is there no framework that formalises and systematises the procedure. Such a framework can be utilised with Gaussian process soft sensors as the latter allows prior expert knowledge incorporation in the form of a GP prior.

The reliability of data-driven soft sensors in open and closed loop applications is not rigorously validated despite the critical tasks some of the reported soft sensors are deployed to fulfil. In addition to validating the reliability, data

reconciliation procedures that can address data issues are obviously overlooked and are not described in soft sensor development procedures. Is this due to a misconception between data pre-processing and data reconciliation?

This chapter brings into focus the limited employment of Gaussian processes as a soft sensor building technique compared to other techniques that are extensively used, such as PLS, ANN, and PCR. As a result, the review identifies potential application areas where Gaussian process-based soft sensors can be utilised and critical tasks can be fulfilled. For instance, inferential control application can be further investigated using Gaussian processes data-driven soft sensors.

Chapter 3

Gaussian Processes as a Data-Driven Soft Sensor Modelling Technique

3.1 Introduction

Data driven soft sensor modelling is essentially analysing data and extracting an approximation function that approximates the functional dependence between the selected input variables (easy-to-measure variables) and the target variables (difficult-to-measure variables) [5]. They are more widely adopted than their model-driven counterparts because data-driven soft sensors are independent on a prior phenomenological knowledge. Various statistical and machine-learning inference techniques are employed for the modelling task; foremost among them are:

- **Multivariate Statistical Techniques**
  Multivariate statistical techniques are extensively utilised in soft sensor building because of their simplicity and clear mathematical background. They are based on projecting the input and output space into a lower dimensional space[5]. Among them are Principle Component Regression (PCR), which is primarily a Principle Component Analysis (PCA) integrated with a regression model Independent Component Analysis (ICA), Kernel Principle Component Analysis (KPCA), and Partial Least Square (PLS).

- **Artificial Neural Networks (ANN)**
  Artificial Neural Networkss are by far the most adopted nonlinear formalism in soft sensor modelling [64]. In the process industry and in soft sensors in particular, the most widely used paradigm is Multi-Layer Perceptron

(MLP). This is due to its inherent characteristics and ability to extract nonlinear relationships between independent and target variables. MLP (if properly trained) has an excellent generalisation capability; moreover, MLP can not only model Multiple Input Single Output (MISO) models, it can also model Multiple Input Multiple Output (MIMO) [99].

- **Gaussian processes (GPs)**
  Gaussian processes, as highlighted in [100], are utilised in various areas; however, this utilisation is largely meagre compared to their merits, virtues, and advantages. It is speculated that this is due to the confusion between the properties of the unknown function underlying the data and those of the optimal model for this unknown function. In the soft sensor context, Gaussian processes are a relatively new alternative to the aforementioned techniques, and recently a few publications have started to appear. This can be isolated to their computational complexity scale that might hinder them from being adopted more widely in soft sensor applications. Despite this, Gaussian processes have indispensable features over other machine learning techniques and in particular in the field of dynamic nonlinear modelling[70].

As the research focuses on Gaussian process as a soft sensor building technique, this chapter is dedicated to introducing them in more detail, outlining their advantages and limitations, highlighting the motivation behind adopting them, and summarising their applicability in the soft sensor domain.

3.2 Gaussian processes

A Gaussian process, which is a type of continuous stochastic process, attracted the attention of the machine learning community in the nineties when Neal in [101] demonstrated a link between neural networks and Gaussian processes.

A Gaussian process is a set of continuous random variables $\mathbf{f} = \{f_1, f_2, ..., f_N\}$ indexed by their inputs $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, any finite subset of those random variables follows a joint multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{K}$ [69]:

$$p(\mathbf{f}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \tag{3.1}$$

A Gaussian process is a fully probabilistic Bayesian model, where Bayesian inference can be used to make predictions from the data. However, whereas

a Bayesian model defines a prior over unknown model parameters, a Gaussian process model defines a prior directly over the functional relationships between the test inputs and training inputs and outputs. This Gaussian prior is defined by the mean and the covariance functions [67, 69]:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \tag{3.2}$$

$$k(\mathbf{x}, \mathbf{x}') = Cov[f(\mathbf{x}), f(\mathbf{x}')] \tag{3.3}$$

The mean and the covariance functions are sufficient to specify the joint distribution for any number of samples of a Gaussian process [102].

$$p(\mathbf{f}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{3.4}$$

where $(\mathbf{x}, \mathbf{x}')$ are a pair of input vectors (in a univariate data, $(\mathbf{x}, \mathbf{x}')$ are a pair of inputs and denoted as (x, x')). The definition above implies marginalisation, conditioning, and summation properties, which are the keys to Gaussian process regression models [67, 69, 103].

In cases where the mean function is assumed to be zero, which is often the case, the covariance function - on its own - relates one observation to another [104]. As it is often the case that there is no enough prior knowledge to assume a mean function, the GP prior is defined as

$$p(\mathbf{f}) = \mathcal{N}(0, \mathbf{K}) \tag{3.5}$$

Where $\mathbf{K}$ is a self covariance matrix of the training inputs computed from 3.3. Assuming that the observed output y is generated with Gaussian white noise ϵ around the underlying function f:

$$y = f(\mathbf{x}) + \epsilon, \tag{3.6}$$

This noise assumption together with the model gives rise to the likelihood:

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_n^2 \mathbf{I}) \tag{3.7}$$

where $\mathbf{I}$ is the identity matrix and σ_n^2 is noise variance. Integrating over the function values $\mathbf{f}$ gives the marginal likelihood which is simply the integral of the

likelihood times the prior:

$$\begin{aligned} p(\mathbf{y}) &= \int d\mathbf{f} p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}) \\ &= \mathcal{N}(0, \mathbf{K} + \sigma^2 \mathbf{I}) \end{aligned} \tag{3.8}$$

3.2.1 Covariance Functions

As mentioned previously, to specify a Gaussian process prior, the mean and the covariance functions are needed. All Gaussian process models developed throughout the thesis assume zero mean priors. This leaves the model developer with a selection of the covariance function to construct the covariance matrix $\mathbf{K}$.

A covariance function, often referred to as a kernel function, is a parametric function that specifies the covariances among pairs of random variables [105].

It has a central role in Gaussian process models, as it conveys all the information on the kind of function generated by the process, determines the properties of the samples drawn from the Gaussian process, and (in regression problems) controls the smoothness of the function underlying the data. Thus, if the covariance function was not chosen appropriately, it may worsen the performance of the model [106].
The choice of covariance functions is one of the elements in Gaussian process model selection; however, any covariance function can be used, provided it is positive semi definite ($\mathbf{x}^\mathsf{T}\mathbf{K}\mathbf{x} \geq 0, \forall \mathbf{x}$). In other words, it generates a non-negative definite covariance matrix for any set of inputs [67–69].

There are different types of covariance functions that satisfy this requirement, and the next section describes those functions commonly used in regression problems.

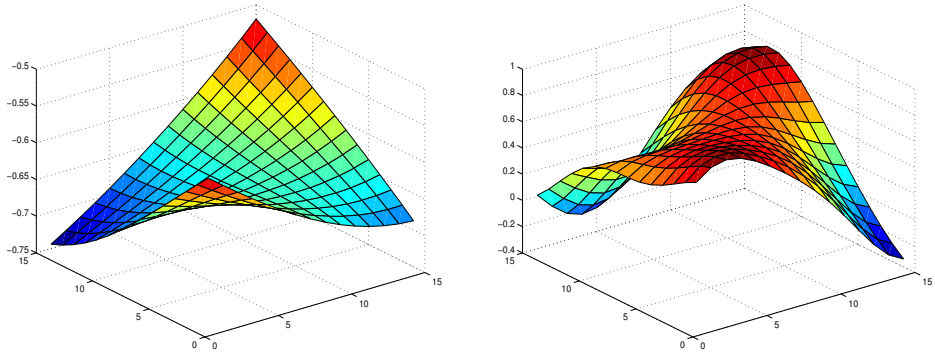
Stationary Covariance Functions

Stationary functions are functions that depend on the distance between the inputs $\mathbf{x} - \mathbf{x}'$ (Euclidean Distance), and are invariant to translations, meaning that they only depend on the distance between the inputs, not on their values. A stationary function can also be an isotropic function if it is invariant to translations and rotations. A widely used example of this type of function is the Squared Exponential (SE) covariance function given in the multivariate case (ARD) by

[107]:

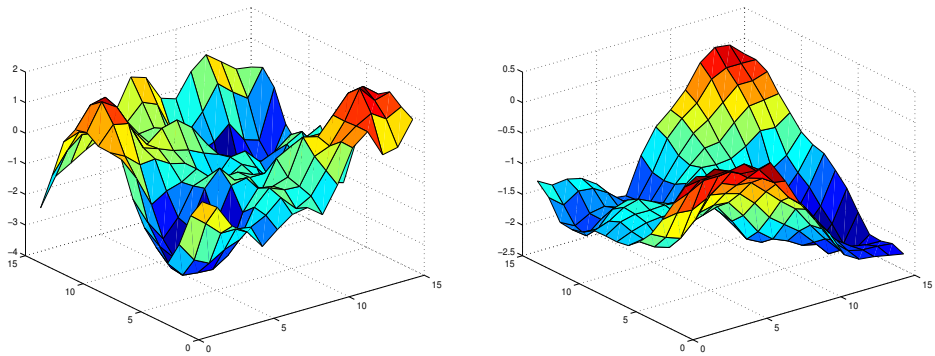
$$k(r) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{d=1}^{D} (\frac{r}{l_d})^2 \right) \tag{3.9}$$

where $r = |\mathbf{x_d} - \mathbf{x'_d}|$ the hyper parameters σ_f^2, and l_d are, respectively, signal variance and characteristic length scale in dimension d (how far to move the inputs space for the function values to be uncorrelated). The two hyper parameters, denoted by $\Theta = [\sigma_f^2, l_1, ..., l_d]$, govern the properties of sample functions drawn from the defined Gaussian process as illustrated in Figure 3.1, where two samples are drawn from a Gaussian process prior with a squared exponential covariance function with a fixed variance and two different length scales.



(a) $\sigma_f^2 = 0.2$, l = -1.5 (b) $\sigma_f^2 = 0.2$, l = -0.4

Fig. 3.1: Sample GP Prior with SE Function



(a) $\sigma_f^2 = 0.2$, l = -1.5 (b) $\sigma_f^2 = 0.2$, l = -0.4

Fig. 3.2: Sample GP Prior with MC Function

This function gives rise to smooth sample functions that are infinitely differentiable. Such smoothness stems from the structure of the function. Where l controls the influence of the distance between $\mathbf{x}, \mathbf{x}'$, when inputs are distant from

each other, the function tends to be zero and new inputs will have a negligible effect. As inputs get closer, the function reaches its maximum and the spectral density increases resulting in a very smooth function as illustrated in 3.1 [68, 69]. Although this smoothness is not realistic in practical problems, SE is widely accepted and used. In addition, all Gaussian process-based soft sensors (published so far) are developed using the SE.

Another example of a stationary covariance function is the Matérn Class (MC) covariance function given by 3.10 in the multivariate case (ARD):

$$k_{Mate'rn}(r) = \sigma_f^2 \frac{2^{1-v}}{\Gamma(v)} (\frac{\sqrt{2vr}}{l_d})^v k_v (\frac{\sqrt{2vr}}{l_d}) \tag{3.10}$$

where $r = |\mathbf{x_d} - \mathbf{x'_d}|$, $\Gamma(.)$ is a Gamma function, k_v is the modified Bessel function, and the hyper parameters l and σ_f^2 are the signal variance and characteristic length scale, and v control the roughness of the sample function as illustrated in Figure 3.2.

As per Rasmussen in [69], the two interesting cases in machine learning are when $v = \frac{3}{2}$ or $\frac{5}{2}$, and as v moves toward infinity the function converges to the SE covariance function. This function produces samples that are significantly rougher than those drawn from the SE as depicted in Figure 3.2 and Figure 3.1. The figures also illustrate the effects of the length scale on the shape of the functions.

The comparative study this research conducted considers the above two stationary covariance functions. The squared exponential was chosen as a base line for being used in all previously published GP-based soft sensors, while the Matérn Class (MC) was chosen because it is declared in [69] that it has the two most interesting cases in machine learning as pointed out previously. In addition to the comparative study, the neural network covariance function was considered as it is a non-stationary covariance function and investigating its performance in comparison to the stationary functions reveals the differences in the properties of both type of functions.

3.2.2 Advantages and Limitations of Gaussian process

Gaussian processes have attractive features over all machine-learning techniques, particularly in the field of dynamic nonlinear modelling. These include: [70, 108]

- Ability to measure prediction confidence through the measured variance.
- Model has few training hyper-parameters.

- Model structure can be determined relatively easily.
- Possibility to include prior knowledge in the model.

In contrast, Gaussian processes are not appropriate solutions for all problems, in that they are frequently not the appropriate prior for the modelled problem and cannot be used, for instance, to express functions with discontinuities. This can be addressed by using non-stationary covariance functions [100].

In addition to those modelling limitations, Gaussian process computational cost scales up to N^3 where N is the number of samples in the training data. This drawback may hinder Gaussian process soft sensor from online implementation in general, and in particular in the application of rapidly changing processes and high sampling rates. This issue is addressed by various approximation techniques, though they trade off prediction accuracy.

Another practical limitation of Gaussian process models is the inefficiency and the poor prediction accuracy that arises from the data dimensionality [67]. Hybrid methods (e.g., PCA and GP) are proposed to tackle this deficiency.

3.2.3 Gaussian process Regression

As Gaussian processes define a prior distribution on the underlying function being modelled, this Gaussian prior $p(\mathbf{f})$ is used to make inferences from the data, where the objective is to get the posterior distribution over the most likely functions from the data $p(\mathbf{f}|\mathcal{D})$. By defining a noise model that links the data to the most likely functions $p(D|\mathbf{f})$ the regression is simply a Bayesian inference as in 3.11:

$$p(\mathbf{f}|\mathcal{D}) = \frac{p(D|\mathbf{f})p(\mathbf{f})}{p(D)} \tag{3.11}$$

This means that only the conditional distribution is modelled, whereas the distribution of the inputs themselves is not specified.

Consider a data set of n observations and d dimension $\mathcal{D} = \{(\mathbf{x}_i, y_i)|i = 1, \cdots n\}$, where $\mathbf{x}$ denotes a training input vector, and y denotes its associated training output (target variable). All training inputs can be aggregated in n by d matrix $\mathbf{X}$, and the scalar targets in an n by 1 vector $\mathbf{y}$. The regression problem is to predict the target $y^\star$ that corresponds to the inputs vector $\mathbf{x}^\star$.

Introducing the noise term above gives the joint prior distribution of the

training and test targets:

$$(\mathbf{y}, y^{\star}) = \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K} + \sigma_n^2\mathbf{I} & \mathbf{k}^{\star} \\ \mathbf{k}_{\star} & k_{\star\star} \end{bmatrix}\right) \tag{3.12}$$

The expression $\mathbf{K} + \sigma_n^2\mathbf{I}$ denotes a covariance matrix between training inputs, where σ_n^2 is the noise variance and $\mathbf{I}$ is the identity matrix. While $\mathbf{k}_{\star}$ is a vector of covariances between the training inputs and a 'single' test point, $k_{\star\star}$ is the variance of the test point [68, 69, 100, 109]. Based on the above declared prior, the predictive distribution is computed by conditioning on the training outputs as in 3.13:

$$(y^{\star}|\mathbf{y}) = \mathcal{N}(\mu, \sigma^2) \tag{3.13}$$

$$\mu = \mathbf{k}_{\star}[\mathbf{K} + \sigma_n^2\mathbf{I}]^{-1}\mathbf{y} \tag{3.14}$$

$$\sigma^2 = k_{\star\star} - \mathbf{k}_{\star}[\mathbf{K} + \sigma^2\mathbf{I}]^{-1}\mathbf{k}^{\star} + \sigma^2\mathbf{I} \tag{3.15}$$

The computations in 3.13 become problematic as the number of training inputs exceeds a few thousand. This is due to the cost of computing the inverse of $[\mathbf{K} + \sigma_n^2\mathbf{I}]$, which requires $O(N^3)$ computation cost, and $O(N^2)$ memory cost for training, and at least $O(N)$ for testing where N is the number of training inputs [110].

These computational and memory requirements can be a major obstacle to implementing Gaussian process-based adaptive soft sensors online. To circumvent this limitation, which is a known disadvantage of Gaussian processes, many sparse methods have been proposed. A common strategy to most of these sparse methods is the construction of M data points variously referred to as inducing points, active set, support variables, or pseudo inputs [111]. The inducing points are exact whereas the remaining data is given some approximation which yields cheaper costs but less accuracy [111].

Although these approximation techniques allowed a reduction of the time complexity from $O(N^3)$ to $O(NM^2)$ and further to $O(M^3)$ as detailed in [111], and [112], respectively, a common drawback of the approximation techniques is finding the optimal size and the location indices of inducing points. The latter limitation is addressed in [113], where the locations are treated as extra hyper parameters of the Gaussian process model that are also optimised by maximising the marginal likelihood as detailed in Section 3.2.5.

Another approach to handling the computational cost issue is the use of local learning methods. It is argued in [114] that computational costs can be alleviated using multiple model methodologies such as mixtures of Gaussian processes.

However, the applicability of the approximation techniques in the soft sensors domain is still limited, particularly for soft sensors developed for predicting high sampling rate target variables or those employed in closed loop control. This is due to the reduced time complexity by approximation techniques of $O(NM^2)$ and $O(M^3)$, which are still not practical in the mentioned application, particularly if N is large or if the application demands a large number of inducing points M to get to the desired prediction accuracy.

3.2.4 Gaussian process Model Selection

As Gaussian processes are entirely determined by the mean and the covariance functions, model selection solely involves selection of the covariance function and the associated hyper parameters. However, it is often feasible to determine some properties of the covariance function, such as stationarity, from the context of the data, but it is more difficult to determine hyper parameters with confidence, such as the length scale [104].

A major virtue of Gaussian processes is the adequacy of selecting hyper parameters from the training data. Ideally this is done by placing a prior and computing the posterior $p(\Theta|D)$ on hyper parameters. However, the computations involved in the integrals over the hyper parameter space are not analytically tractable [103]. Instead, the hyper parameters can be optimised using an appropriate cost function, such as the marginal likelihood given in 3.8. More specifically, the negative log marginal likelihood given in 3.16 is minimised with respect to each of the hyper parameters in the vector Θ [69]. GP hyper parameter optimisation is sometimes referred to as type II maximum likelihood [103].

The marginal likelihood given in 3.16 distinguishes Bayesian inference scheme from other optimisation based schemes [69]. Its ability to automatically trade-off between model fit and model complexity makes it valuable [69].

$$L(\Theta) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log(2\pi) \tag{3.16}$$

The $-\frac{1}{2}y^T\mathbf{K}^{-1}y$ part is a data-fit, $\frac{1}{2}\log|\mathbf{K}|$ part is a complexity penalty, and $\frac{n}{2}\log(2\pi)$ is a normalisation constant.

Minimising the negative log likelihood $L(\Theta)$ requires the computation of $L(\Theta)$ derivative with respect to each hyper parameter in the vector Θ given in 3.17 where Tr denotes the trace:

$$\frac{\vartheta L(\Theta)}{\vartheta \Theta_i} = -\frac{1}{2} Tr\left[\mathbf{K}^{-1}\frac{\vartheta \mathbf{K}}{\vartheta \Theta_i}\right] + \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\frac{\vartheta \mathbf{K}}{\vartheta \Theta_i}\mathbf{K}^{-1}\mathbf{y}. \tag{3.17}$$

Cross-validation can also be used to select Gaussian process models. Cross-validation is a statistical method for evaluating and comparing models by dividing data into two segments: one used to train the model and the other to validate it. In typical cross-validation, the training and validation sets must crossover in sequential rounds in a method that ensures each data point has a chance of being validated. Cross-validation can be used with any loss function. In machine learning, the most widely used function is the squared error loss. However, as Gaussian process is a probabilistic model, it is argued in [67] that it is natural to use the predictive log probability loss function given by:

$$LD = \frac{1}{2}\log 2\pi + \frac{1}{2N}\sum_{i=1}^{N}(\log \sigma_i^2 + \frac{e_i^2}{\sigma_i^2}) \tag{3.18}$$

where σ_i^2 is the (i-th) prediction variance and e_i^2 is the error between the $(i-th)$ actual output and its corresponding prediction.

3.2.5 Gaussian processes Approximation

One of the fundamental problems in Gaussian processes is their computational complexity in large data sets when implemented in their full form as mentioned in Section 3.2.3.

Different approximation techniques are proposed to cure this limitation, most of which are based on reduced rank approximation to the covariance matrix (equivelent to $\mathbf{Q_N}$ in 3.23):

$$\mathbf{K} = \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{NM} \tag{3.19}$$

where $\mathbf{K}_{NM}$ is the covariance between the training data and a subset of the data variously referred to as inducing points, pseudo inputs, support points, or active set, while $\mathbf{K}_M$ is a self-covariance between the inducing points.

An example of these approximation techniques is the Sparse Gaussian Processes using Pseudo-inputs which is considered in this research and its performance is further improved by rationally selecting the pseudo inputs.

Sparse Gaussian Processes using Pseudo-inputs

This approximation technique considers the model with the a likelihood given by the GP predictive distribution given in 3.13 and parametrized by a subset of the data referred to as pseudo set $\bar{\mathrm{D}}$. The sparsity in the model arises from the fact that only pseudo data set of size $M \ll N$ is used with $\bar{\mathbf{X}}$ pseudo inputs and pseudo outputs $\bar{\mathbf{f}}$. As such the signle data point likelihood is given by [103]:

$$p(y|x, \bar{\mathbf{X}}, \bar{\mathbf{f}}) = \mathcal{N}(\mu_{spgp}, \sigma^2_{spgp}) \tag{3.20}$$

$$\mu_{spgp} = \mathbf{k}_{x\bar{\mathbf{X}}}[\mathbf{K}_{\bar{\mathbf{X}}}]^{-1}\bar{\mathbf{f}} \tag{3.21}$$

$$\sigma^2_{spgp} = k_{xx} - \mathbf{k}_{x\bar{\mathbf{X}}}[\mathbf{K}_{\bar{\mathbf{X}}}]^{-1}\mathbf{k}_{\bar{\mathbf{X}}x} + \sigma^2 \tag{3.22}$$

Where $\mathbf{k}_{x\bar{\mathbf{X}}}$ is the covariance between the single input x and the pseudo inputs $\bar{\mathbf{X}}$, $\mathbf{K}_{\bar{\mathbf{X}}}$ is the self covariance between the pseudo inputs, and k_{xx} is the self-covariance between the single input point x which is simply its variance.

It worth noticing here that this likelihood is the GP predictive distribution, however, the real data set replaced by the pseudo-data set and with no noise on the pseudo-outputs and thus there were denoted by $\bar{\mathbf{f}}$ rather than $\bar{y}$ to emphasis that they are not the real observation with noise variance [103].

Assuming that the output data is i.i.d generated given the inputs, the complete model likelihood is then given as:

$$p(\mathbf{y}|X, \bar{\mathbf{X}}, \bar{\mathbf{f}}) = \mathcal{N}(\mathbf{K}_{\mathbf{X}\bar{\mathbf{X}}}[\mathbf{K}_{\bar{\mathbf{X}}}]^{-1}\bar{\mathbf{f}}, diag(\mathbf{K} - \mathbf{Q}_N) + \sigma^2\mathbf{I}) \tag{3.23}$$

Where $\mathbf{Q_N} = \mathbf{K}_{\mathbf{X}\bar{\mathbf{X}}}[\mathbf{K}_{\bar{\mathbf{X}}}]^{-1}\mathbf{K}_{\bar{\mathbf{X}}\mathbf{X}}$ is a low rank covariance matrix between the training inputs and the pseudo inputs.

Learning in SPGP involves finding a suitable pseudo data that explains the real data, which is accomplished by integrating out the pseudo outputs $\bar{\mathbf{f}}$ instead of maximizing the likelihood with respect to $\bar{\mathbf{X}}$ and $\bar{\mathbf{f}}$. This is done by first placing a Gaussian prior on the pseudo output such that [103]:

$$p(\bar{\mathbf{f}}) = \mathcal{N}(0, \mathbf{K}_{\bar{\mathbf{X}}}) \tag{3.24}$$

This prior reflects the assumption that the pseudo data is similarly distributed as real data, and by integrating over the pseudo-outputs, the SPGP marginal likelihood (equivalent to the full GP marginal likelihood given in 3.8) is found

[103]:

$$p(\mathbf{y}) = \int d\bar{\mathbf{f}} p(\mathbf{y}|\bar{\mathbf{f}}) p(\bar{\mathbf{f}}) \tag{3.25}$$

$$\mathbf{y} = \mathcal{N}(0, \mathbf{K}_{\mathbf{X}\bar{\mathbf{X}}}[\mathbf{K}_{\bar{\mathbf{X}}}]^{-1}\bar{\mathbf{f}} + diag(\mathbf{K} - \mathbf{Q}_{\mathbf{N}} + \sigma^2\mathbf{I})) \tag{3.26}$$

It is noted that the SPGP marginal likelihood can be obtained by replacing the covariance matrix $\mathbf{K}$ in the full GP case by the low rank covariance matrix $\mathbf{K}_{\bar{\mathbf{X}}}$. The hyper parameters and the locations of the pseudo data can be found by maximizing the above derived SPGP marginal likelihood given in 3.26

To obtain the predictive distribution the joint distribution $p(y_{\star,\mathbf{y}})$ is first found, which is exactly the marginal likelihood given in 3.26 extended one new test point. Then by conditioning on the observed targets, the predictive distribution is computed whose mean and variance are given by [103]:

$$\mu_\star = \mathbf{K}_{x\bar{\mathbf{X}}}[\mathbf{Q}_N + diag(\mathbf{K} - \mathbf{Q}_N) + \sigma^2\mathbf{I}]^{-1}\mathbf{y} \tag{3.27}$$

$$\sigma_\star^2 = k_{xx} - \mathbf{K}_{x\bar{\mathbf{X}}}[\mathbf{Q}_N + diag(\mathbf{K} - \mathbf{Q}_N) + \sigma^2\mathbf{I}]^{-1}\mathbf{K}_{\bar{\mathbf{X}}x} + \sigma^2 \tag{3.28}$$

Where $\mathbf{K}_{\bar{\mathbf{X}}x}$ is the covariance between the test input and the pseudo inputs. The predictive distribution form is written such that it can be compared to the full GP predictive distribution given by 3.13, however, the difference is that the matrix being inverted is low rank + diagonal. This brings down the computational complexity from $O(N^3)$ to $O(M^2N)$ where N and M are the number of data points and the number of pseudo inputs respectively.

3.2.6 Gaussian processes in Soft Sensor Domain

In recent years, Gaussian process has been successfully employed in various applications in machine learning including soft sensors. Gaussian process is a relatively new option compared to other soft sensor building techniques; however, they are now more widely employed with promising results.

In [66] a Gaussian process-based soft sensor was proposed to estimate the freezing point of light diesel in a refinery. The soft sensor model employs the squared exponential covariance function, and the associated hyper parameters are optimised by maximising the marginal likelihood. The proposed sensor is benchmarked against ANN and SVM soft sensors, where experiments demon-

strated that the Gaussian process soft sensor showed a satisfactory performance. It was outperformed by the SVM soft sensor. The authors recommend using a maximum likelihood framework to optimise the hyper parameters, in contrast to this, the author in [115] criticised this framework and argued that it is sensitive to the initial choice of the hyper parameters. For this reason, it is recommended to complete multiple random restarts of the algorithm to avoid local minima problems. The proposed soft sensor has other limitations: i) it lacks adaptivity and cannot cope with the process drifts, ii) the choice of the covariance function is based on the process smoothness assumption, and iii) the target variable is often dependent on past process status, and thus dynamic modelling should be considered as an alternative to static.

To address the shortcomings of static soft-sensor modelling methods, [108] presents a dynamic Gaussian process soft sensor that captures process dynamicity. The Gaussian process model exploits the squared exponential covariance function. Fuzzy curve is utilised to determine the past most critical data points to the corresponding output, and the new data is then used to build the Gaussian process soft sensor. The dynamic characteristics of the process are incorporated into the model. The developed sensor is tasked to predict 4-CBA in a PTA oxidation process, and simulation results showed precise generalisation and an accurate prediction capability. However, the suggested model was not benchmarked against other methods, where a comparison between static and dynamic versions is crucial to pinpoint the virtue of capturing process dynamicity.

The authors in [72] address a common problem in industrial processes, where a process often encounters different operation modes. Modelling such a process by collecting a large number of data points to reflect all process operation modes and using a single model gives rise to poor prediction accuracy and over-fitting problems. Therefore, the authors proposed a soft sensor based on Affinity Propagation, Gaussian process and Bayesian Committee Machine (BCM).

The Affinity Propagation algorithm groups input data into clusters according to the process operation mode. Gaussian process is used to develop multiple local models that adopt the squared exponential covariance function; a mode for each cluster and BCM is used to combine predictions from the local models. BCM addresses traditional methods of forming global predictions from local models, such as the use of weighted averaging suggested in [116]. In such methods if a local model over-fits the data subset, it will be assigned a larger weight. Consequently, over-fitting is propagated and amplified and the prediction accuracy is compromised.

The soft sensor is employed in a real application and is tasked to estimate the light naphtha endpoint in hydro-cracker fractionators. The authors argued that the sensor has contributed to improving production. However, the reported results and findings are based on a qualitative validation. In addition, the sensor lacks an adaptivity mechanism to preserve the high prediction accuracy over time.

Because single model Gaussian process often perform poorly and demand more computational power, [117] emphasise a multimodal-based approach in order to enhance the predictive capabilities and computational cost. The authors set out bagging for Gaussian process soft sensors, where a number of Gaussian process models are built for bagged data. The local models are then combined to give a global prediction. The merits of this novel approach are the utilisation of the following two advantages of Gaussian processes: i) the predictive variance associated with Gaussian process model predictions, where the inverse of the variance is used to assign weights for the local models such that the higher the variance of the model prediction, the lower the weight assigned and ii) the product of two Gaussian distributions is still Gaussian. Based on this, the algorithm combines the weighted local models using a product rule that results in a global predictive distribution with a global mean and variance. The novel suggested soft sensor is validated to predict the melt flow rate in a polypropylene polymerisation process, and the results demonstrate its superior performance compared to other models, including the single-based model.

Striving to retain soft sensor prediction accuracy during online operation, [118] presented a Gaussian process adaptive soft sensor based on a moving window. The window slides as a new data point becomes available such that the oldest data point is excluded. The authors assert that this method allows the window to contain new and updated information about the process. The new data point is added by updating the covariance matrix such that an appropriate row and column is added. The virtue of the proposed method is that it exploits the previous matrix inverse and does not call for inverting the matrix each time the window is updated. Conversely, discarding old points to include new ones in the moving window might not be the optimal approach in some processes, as an old point may contain valuable predictive information. Similarly to previously reported and reviewed Gaussian process soft sensors, the authors used the squared exponential covariance function for developing the Gaussian process model.

The same window updating method and the adopted covariance function are applied in building the soft sensor presented in [119]. However, the proposed soft sensor model is a dynamic model that includes past input and output measure-

ments in an attempt to simultaneously capture process dynamics and nonlinearities. To enhance the predictive accuracy of the dynamic model, the bias between the observed and predicted targets is updated online taking into account the latest previous bias weighted factor. Additionally, online de-noising of the inputs and the target variables is performed to enhance the prediction accuracy of the dynamic model. The authors stress that this has contributed to enhancing overall sensor performance. Nonetheless, the suggested algorithm re-optimises the hyper parameters of the Gaussian process model at every new test point. This procedure is unnecessary, as these hyper parameters often do not significantly differ from each other. Thus employing a threshold-based updating method serves to preserve the computational demand in retraining and re-optimising the model as detailed in [120].

Adaptivity calls for adding data points online, and hyper parameter re-optimisation, which in Gaussian process based models entails expensive computational time. Striving to address the computational burden of Gaussian process, [120] presents an online dynamic model to predict ozone concentration in the air, multiple steps ahead. To overcome the computational burden of implementing Gaussian process-based models online, a sparse Gaussian process learning method is implemented. The sparse method is a combination of the Bayesian online approach and a sequential construction of the optimal subset of the data. The proposed model gave sufficiently accurate predictions and performed the task successfully. The employed Gaussian process model uses the squared exponential covariance function as widely reported in the literature [72, 108, 115, 118–120].

3.3 Summary

Gaussian processes are emerging as a promising alternative to other well-established soft sensor building techniques. This chapter gives the accounts and reasons for rendering and adopting Gaussian processes in the soft sensor domain, outlines the adequacy of Gaussian process model structure, and highlights Gaussian process model identification and selection.

Gaussian processes have the advantage over other soft sensors building techniques by being moderately simple to implement, being non-parametric models, and having a simpler model structure that is relatively easy to identify in comparison to other parametric techniques. Moreover, uncertainty of Gaussian process model predictions can be straightforwardly computed as a Bayesian model. This

lays a solid base for implementing soft sensors in industrial processes in closed loop applications (inferential control systems).

This chapter, on the other hand, pinpoints the fact that (so far) in all published Gaussian process-based soft sensors, developers assume that input points near each other result in similar predictions, and for this reason, the squared exponential covariance function is utilised in building the Gaussian process model. Other publications rationalise the choice of the squared exponential covariance function arguing that it has been proven to work well in a variety of applications and has been widely used in practice. In fact, the squared exponential is used blindly and as a default function in the soft sensor domain without a rigid reasoning in most of the publications reviewed. In contrast, the squared exponential covariance function implies smoothness, stationarity, and continuity that are often not practical.

This chapter attempted to review the few Gaussian process based soft sensors in chemical processes, and an important and obvious finding from the review is the lack of adaptivity in most of the soft sensors reported. This is attributed to the undesired computational power entailed by the online retraining and re-optimisation of the Gaussian process model that might not be feasible in some applications. Although different sparse techniques are proposed to address the issue, they have not been reported in soft sensor building. This could be ascribed to the poor prediction accuracy traded-off for the lower computational expense or to the unsatisfactory computational cost that sparse techniques offer.

Chapter 4

Model Structure Selection in Gaussian Process-based Soft Sensors

4.1 Introduction

Gaussian process models have a comparatively easy-to-determine model structure facilitated by the fact that Gaussian process models are completely determined by their mean and covariance functions. Thus, Gaussian process structure selection solely calls for selecting three elements: i) selection of input variables, ii) selection of a mean function, and (most importantly) iii) selection of a covariance function [67]. While variable selection is beyond the scope of this thesis, from a practical viewpoint and due to the lack of sufficient expert knowledge, the mean function is virtually always assumed to be zero. This is a valid practical and non-restrictive assumption; in addition, the posterior arising from the Gaussian process regression model is not a zero mean as was indicated in Section 3.2.3.

The last element in selecting Gaussian process model structure is the covariance function, which is the core theme of this chapter. The choice of the covariance function is crucial to the soft sensor predictive capability, as it reflects the assumptions made about the underlying function of the data.

This chapter investigates five Gaussian process model structures. The rationale is to make a practical recommendation on the choice for the covariance function in the case of insufficient expert knowledge. As it is often the case in industrial plants, a diligent recommendation that stems from a practical perspective is expected to facilitate the soft sensor development phase and improve soft sensor predictive accuracy.

The remainder of the chapter is organised as follows: as this chapter is the first practical chapter of the thesis, the general methodology adopted in developing all

soft sensors throughout the thesis is first introduced. All datasets and evaluation criteria are described before detailing the comparative study between the five Gaussian process model structures.

4.2 General Methodology

This section describes the methodology adopted in developing all soft sensors investigated in this chapter and throughout the thesis. It primarily embraces the steps detailed in the general block diagram shown in Figure 2.4. All data sets are pre-processed for the development process in the order shown in the procedure illustrated in Figure 4.1.

Time lags

Time lags between input process variables and target variables are identified by maximising the mutual information between them. Input variables are shifted backward by a one to twelve hour period, and the time shift that maximises the Mutual Information (MI) between the variables is identified as the adequate input variable lag.

Input Variable Selection

The optimal and most influential input variables are selected based on computing the Maximal Information Coefficient (MIC) between the input variables and the target variable. This was facilitated by the Maximal Information Nonparametric Exploration (MINE) algorithm detailed in [33]. Depending on the data and based on the MIC score between inputs and target variables, the top n input variables are selected. The selection decision of the top n variables is a model selection and identification step that trades off between performance and computational cost.

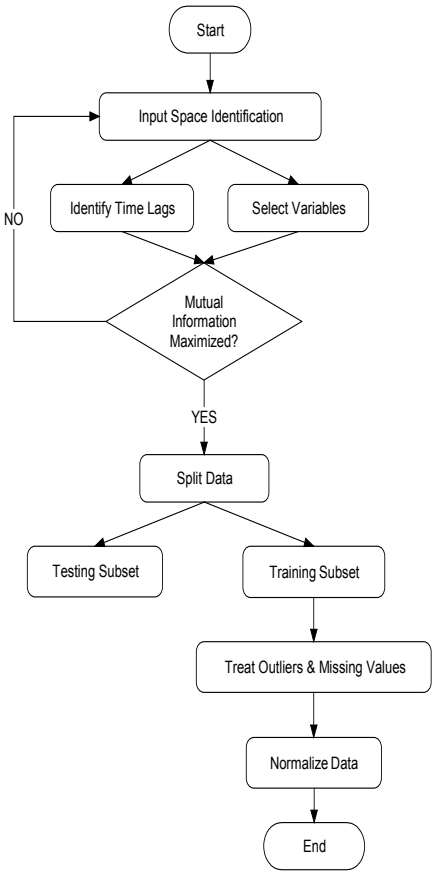


Fig. 4.1: Data Pre-processing Procedure

Data Splitting

In order to simulate unseen data, the data set is split into two subsets: 75% for training and 25% for testing. The testing subset, which was not pre-processed, represents every $4th$ point of the data to interpolate the training subset. The training subset was fully pre-processed and used for training and validation.

Outliers and Missing Values

As data preprocessing is not the focus of the research, sophisticated outlier detectors were not used, rather, outliers are identified using the 3σ rule and the Hampel identifier. 3σ rule is used because it is a widely adopted strategy [53], in addition to being the best-known criterion to detect an outlier [54]. Missing values are treated using the case deletion strategy. Case deletion was adopted, as the number of missing values was small, and some data did not contain any.

Data Standardisation

To overcome the influence of the dimensional effect of the input variables, all data is normalised to have a zero mean and unit variance to have normalised model parameters. For a given data set $X \in \mathbb{R}^{n \times m}$, where n are observations (rows), and m are variables (columns), it is scaled as in 4.1:

$$X_{Scaled} = \frac{\mathbf{x}_j - \bar{x}}{\sigma} \tag{4.1}$$

where $\bar{x}$ and σ are the mean and the standard deviation of the j^{th} variable ($\mathbf{x}_j$)

4.3 Data Sets Description

This section provides a brief description for all the data sets explored and utilised to develop all soft sensors throughout the thesis, where the soft sensors' tasks were to predict the target variables summarised in Table 4.1:

4.3.1 Sulphur Recovery Unit (SRU)

In order to prevent air pollution, a sulphur recovery unit degases acid gases, such as H_2S and NH_3 from acid gas streams before they are released into the atmosphere. The inlets of the unit stream from two plants that produce acid

Table 4.1: Description of Data Sets used

Data Set	Abbreviation	Data Type	Target Variable	Type of Target
Sulphur Recovery Unit	SRU	Measured	Hydrogen Sulphide	On-line Analysers
Debutanizer Column	DC	Measured	Butane Concentration	Measuring Device
Industrial Drier	ID	Measured	Residual Humidity	Lab Measurement
Thermal Oxidiser	TO	Measured	NOx concentrations	Not Known
Refinery Process	RP	Measured	Kerosene Freezing Point	Lab Measurements
Polymer Batch Process	PBP	Measured	Polymer Quality Index	Online Analyser
Catalyst Activation	CAT	Simulated	Catalyst Activity	Simulated Measurements
Polymerisation Process	PP	Simulated	Concentration	Simulated Measurements

gases (MEA and SWS), which are rich in H_2S and NH_3. Acid gases are burnt in a reactor, deoxidised, and transformed into pure sulphur. Other gaseous by-products are cooled causing the generation of liquid sulphur, which is then converted through high temperature converters into water vapour and sulphur vapour. The final tail gas stream of the SRU contains H_2S and SO_2 residuals.

Online analysers are used to measure the volume of the two gas residuals to monitor the conversion process and the performance of the SRU. Hydrogen sulphide (H_2S) and sulphur dioxide (SO_2) frequently damage the online analysers. Hence, a soft sensor that predicts the quantity of those two gases is of great importance to enhance and monitor the performance of the SRU process [3]. As the soft sensors considered in this research are multiple input, single output based models, the soft sensors developed from this data set are tasked to predict only the hydrogen sulphide shown in Figure 4.2. The size of the data is 10081 observations of 7 variables listed below:

- Gas flow MEA-GAS
- Air flow AIR-MEA
- Secondary air flow
- Gas flow in SWS zone
- Air flow in SWS zone
- $H2S$
- SO_2

4.3.2 Debutanizer Column

The debutanizer is a column in a de-sulphuring and naphtha splitter plant. It extracts the butane (C4) and the propane (C3) from the naphtha stream. The debutanizer maximises the C5 (stabilized gasoline) content in the debutanizer overheads and minimises the C4 (butane) content in the debutanizer bottoms

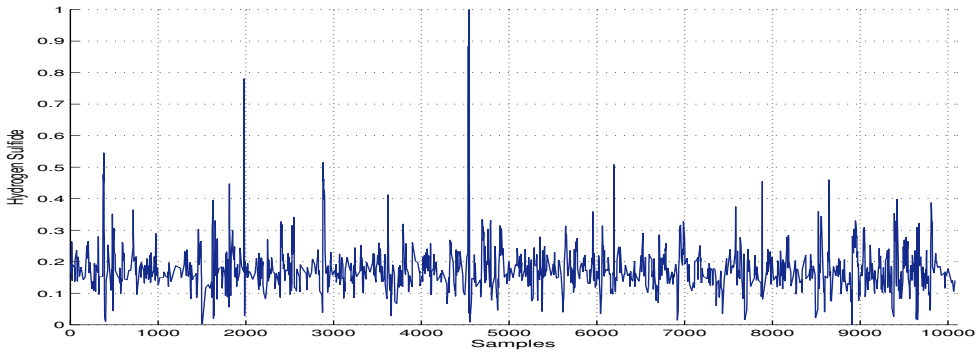


Fig. 4.2: Target Variable of the Sulphur Recovery Unit Data Set

measured by gas chromatographs. Because of the associated delays with gas chromatograph measurements and the cost of these measurements, a soft sensor is required to replace the gas chromatographs and predict the C4 content shown in Figure 4.3 [3]. To design the soft sensor, data was collected for about a three month period, which is represented in the 2394 observations by 8 data matrix. The process variables are listed below:

- Top temperature
- Top pressure
- Reflux flow
- Flow to next process
- 6th tray temperature
- Bottom temperature
- Bottom temperature
- C4 Content

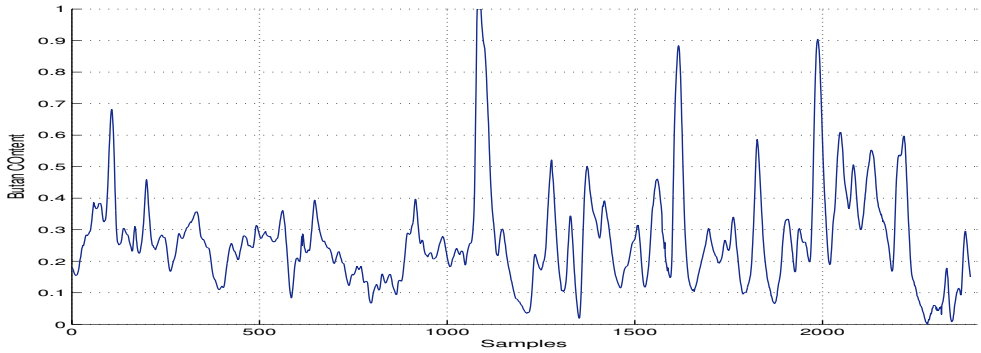


Fig. 4.3: Target Variable of the Debutanizer Column Data Set

4.3.3 Industrial Drier

The target variable of this data set is the residual humidity depicted in Figure 4.4 of the industrial drier processes. The humidity is measured using manual lab analysis, and because of the high associated costs of the lab measurements, it is analysed once every four hours. Data is collected for a seven month period, which is represented by the 1219 observations by 19 inputs data matrix to design the soft sensor [121].

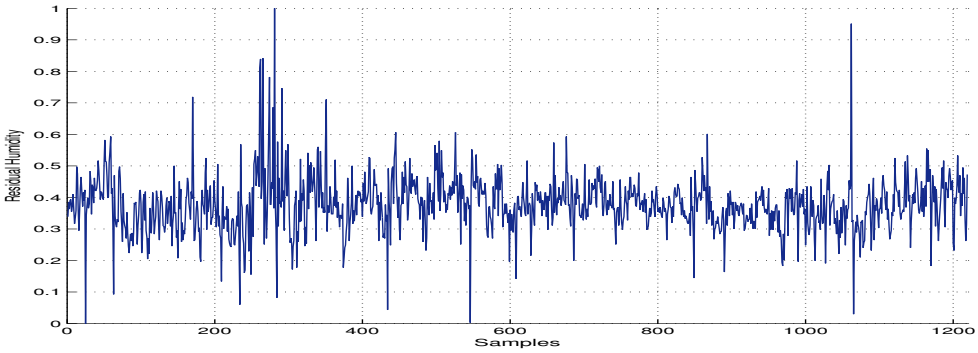


Fig. 4.4: Target Variable of the Industrial Drier Data Set

4.3.4 Thermal Oxidiser

In order to minimise the emissions of nitric oxide (NO) and nitrogen dioxide (NO_2), collectively referred to as NO_x shown in Figure 4.5, the resulting fumes of a large number of refineries and industrial plants are conveyed to a refinery's big chimneys. An online analyser installed in the top of the chimneys is used to measure the emissions rate of the NO_x. Due to harsh environmental conditions, the analyser is frequently offline for scheduled maintenance. For this reason a soft sensor that can continuously predict the emissions rate, particularly during a maintenance period is crucial to monitoring (NO_x) [121]. Data collected for a six month period is represented in the 2053 observations by 40 inputs data matrix.

4.3.5 Catalyst Activity in a Multi Tube Reactor

The reactor consists of 1000 tubes filled with a catalyst used for gas oxidisation. The reaction speed depends on the reactor temperature, which is cooled with a coolant to keep control over the reactor temperature. The cooling process causes the exothermal reaction to counteract to reach a maximum temperature somewhere along the tube. As the catalyst activity decays, this counteraction

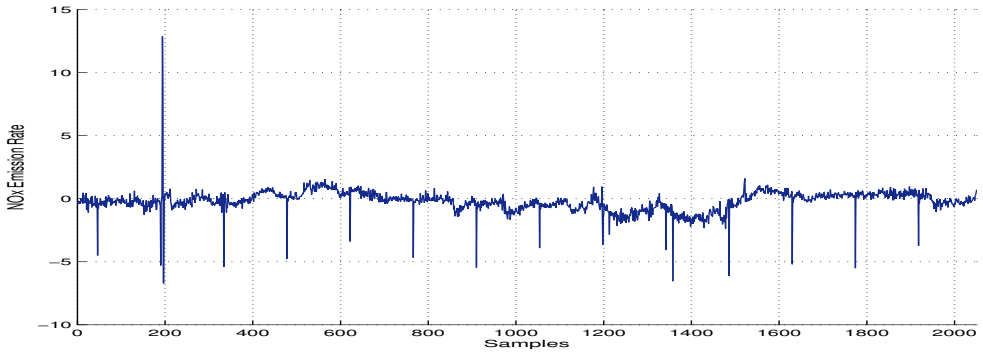


Fig. 4.5: Target Variable of the Thermal Oxidizer Data Set

gradually degrades along the tube. The input variables of this process are taken from another external large process, which means that the inputs are time variant and the feed varies over time; however, the catalyst activity decays more slowly than those effects. Several measurements, such as temperatures, flows, and concentrations are measured along the tubes to identify the process state. Fifteen inputs are chosen as input variables, while the target variable is a simulated catalyst activity inside the reactor as shown in Figure 4.6. One year of observations is considered in this data set, which is represented in the 5808 observations by 16 process variables data matrix [121]. The process variables are listed below:

- Time
- Measured flow of air
- Measured flow of combustible gas
- Measured concentration of combustible component in the combustible gass
- Total feed temperature
- Cooling temperature
- Temperature at length 1/20 of reactor length
- Temperature at length 2/20 of reactor length
- Temperature at length 4/20 of reactor length
- Temperature at length 7/20 of reactor length
- Temperature at length 11/20 of reactor length
- Temperature at length 16/20 of reactor length
- Temperature at length 20/20 of reactor length
- Product concentration of Oxgyne
- Product concentration of combustible component
- Catalyst Activity

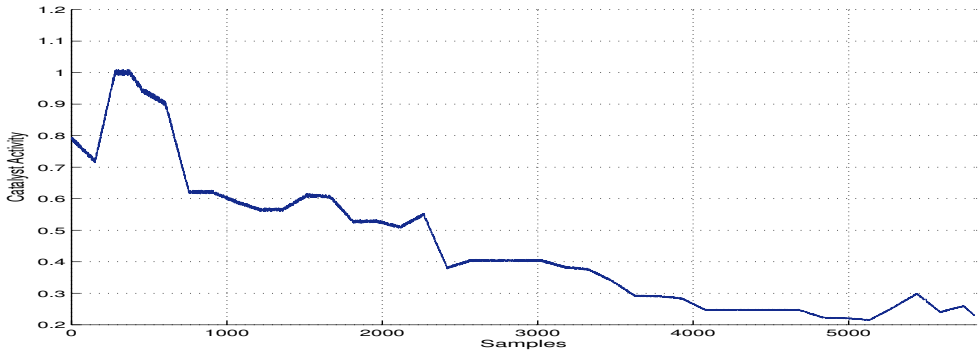


Fig. 4.6: Target Variable of the Multi Tube Reactor Data Set

4.3.6 Refinery Process

In the refinery process, crude oil is distilled into a number of fractions in a distillation column where light fractions rise up the column and heavy ones cascade in a liquid stream down the column. The freezing point of the kerosene streaming from the column is a quality index of plant production. The freezing point is measured by lab analysis once a day and adequate actions are taken accordingly to meet product specifications. A soft sensor that overcomes the lab measurement time delay and continuously predicts the kerosene freezing point is of significant importance to enhancing productivity and minimising off-specification products [122]. One year of observations is considered, which is represented in the 336 observations by 26 input variables data matrix. A plot of the target variable is shown in Figure 4.7.

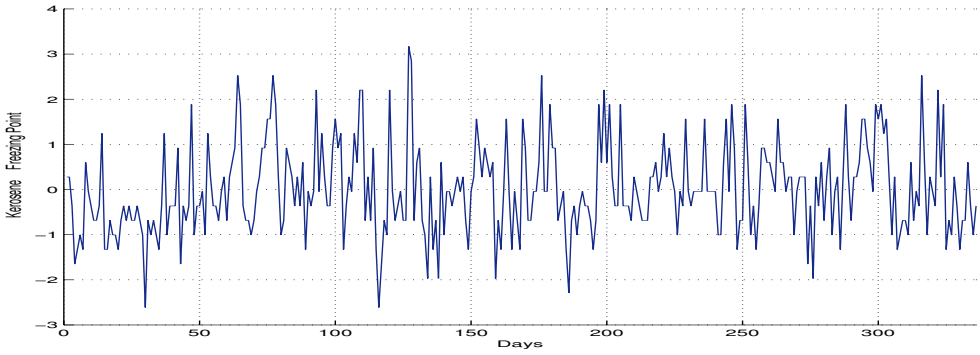


Fig. 4.7: Target Variable of the Refinery Process Data Set

4.3.7 Polymer Batch Process

This data set resembles a typical example of a lack of expert knowledge, which is a major challenge that soft sensor developers confront during development and

deployment phases of soft sensors. There is no operational description of the process from which this data set was retrieved, and thus the entire soft sensor development process is data-driven dependent. The size of the data set is 1485 observations by 52 input variables, and the task is to predict the size of the polymer balls illustrated in Figure 4.8 before the completion of the batch process. This is to take timely corrective actions to meet customer product specifications.

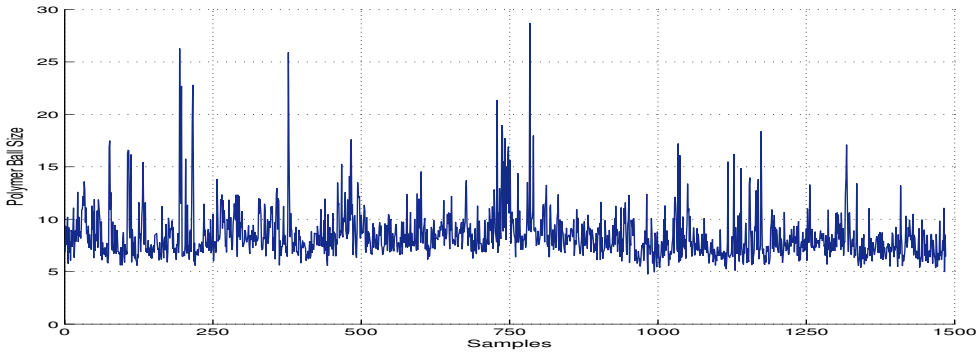


Fig. 4.8: Target Variable of the Polymer Batch Process Data Set

4.3.8 Polymerisation Process

The polymerisation process is a simulated chemical process that converts monomer into polymer through a dilution technique. The technique utilities a solvent flow that reduces the concentration of the monomer solution. The process is exothermic in that it requires an initiator to commence the reaction and then runs without the need for any external factors. The reaction temperature is controlled by adjusting the coolant temperature. In addition to intermediate by-products, the main product is converted from monomer into polymer [123]. To evaluate the effectiveness of the conversion process, a soft sensor is required to predict the volume of the converted polymer. The size of the data set generated is 9878 observations by 19 variables. A plot of the target variable is shown in Figure 4.9.

4.4 Evaluation Criteria

The plausibility, purposefulness, falseness, and overall performance of all soft sensors developed are validated and evaluated using ten different evaluation criteria. The criteria are described below and are used throughout the thesis:

- Mean Squared Error (MSE) - the lower, the better.

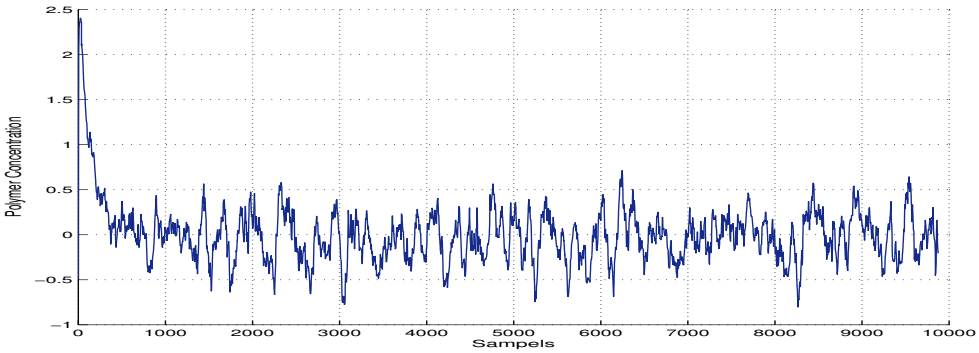


Fig. 4.9: Target Variable of the Polymerisation Process Data Set

- Correlation (R) - the higher, the better.
- Relevance Variance Tracking Precision (RVTP) - the higher, the better.
- Theils Inequality Coefficient (TIC) - the lower, the better.
- Fitness Function (FF) - the higher, the better.
- Log Predictive-Density Error (LPD) - the lower, the better.
- The Mean Standardised Log Loss (MSLL) - the lower, the better.
- Condition Number (CN) - the lower, the better.
- Rising Time (RT)
- Settling Time (ST)
- Overshoot Percentage (OP)
- Steady State Error (SSE)

MSE is used as a regular method to measure how the model predictions deviate from the true targets (prediction error). The RVTP is used to assess model prediction accuracy and consistency when input variables change. TIC is used to assess and compare two models rather than as an indication of model performance. R is used to measure the Pearson correlation between predicted and real targets, and FF scaled measure is used to capture goodness-of-fit in time series data. In addition, CN is used to evaluate covariance matrices used to make predictions in Gaussian process models. It measures how close a matrix is to singularity. LPD is used for evaluating GP models taking into account not only the mean of the model output, but the entire distribution.

MSLL is obtained by subtracting the loss from the model LPD. The MSLL is approximately zero for simple models and negative for better ones.

The last four measures are utilised to evaluate the performances and to assess the robustness and reliability of the control systems developed in the last chapter of the thesis.

These evaluation measures are used appropriately and according to the objective of the chapter, whereas some chapters use all ten criteria, others use only a selection.

4.5 Covariance Function Selection

The covariance function selected determines the quality of the soft sensor. As such, the selection of the covariance function should: i) be based on experts' prior knowledge, ii) reflect the soft sensor developer's assumptions about the characteristics of the functional relationship between input and target variables, and iii) adapt to the modelled data. However, the lack of sufficient prior expert knowledge, particularly in complex industrial plants, partially hinders selecting a covariance function phenomenologically. Therefore, especially in multivariate data sets, it is often ambiguous as to how to tune the covariance function structure to the data set being modelled.

As a result, often in the soft sensor domain the squared exponential covariance function is selected as a default function. This frequently yields a misspecified soft sensor model that does not ultimately fit the modelled data. So far (to the best of our knowledge), all previously published Gaussian process-based soft sensors used the squared exponential covariance function given in (3.2.1) despite the fact that it has mean squared derivatives of all orders (infinitely differentiable), which implies strong impractical and unrealistic smoothness.

In contrast to most covariance functions, the Matérn class covariance functions given in (6.4) are $v-1$ times differentiable [124] where v is a hyper parameter that controls the smoothness. The principal argument for selecting the squared exponential in previously published Gaussian process-based soft sensor papers is either the dominant use of the function or the smoothness assumption underlying the data.

This ambiguity of covariance function selection, specifically in complex multivariate data sets, and the scarcity of expert knowledge motivates devising a more practical heuristic recommendation to aid covariance function selection. This is proposed according to the detailed comparative empirical study reported in section 4.6. The conducted study is between the most widely used covariance function Squared Exponential (SE), the SEard (SE with Automatic Relevance Determination), the Matérn Class covariance functions (MC), the MCard (MC with Automatic Relevance Determination), and the ANN covariance function.

4.6 Case Study

Adopting the general methodology described in section 4.2, five Gaussian process-based soft sensors are developed for each of the data sets described in Table 4.1. The five soft sensors are built using the five covraince functions listed above. This sums up to 40 soft sensor models built to conduct the comparative study between the five covariance function. The sensors' task is to predict the target variables associated with each data set listed in Table 4.1.

Objectives

The primary objective behind conducting the comparative case study is to formulate empirical evidence that supports a particular covariance function choice in the context of Gaussian process-based soft sensors. This formulation is drawn from:

- Evaluating and comparing the predictive capabilities of the soft sensors built using the five aforementioned covariance functions.
- Investigating the numerical invertibility of the covariance matrices computed from both covariance functions.
- Analysing the influence of the noise on the accuracy of matrices' inversion and the consequent effect on the prediction performance.

4.6.1 Prediction Accuracy

Adopting the evaluation criteria summarised in Section 4.4, the prediction accuracy and performance stability of all soft sensors are evaluated quantitatively as reported in Table 4.2. Because the objective of the experiments is comparative-based, the soft sensors considered here are non-adaptive; they were trained offline where the hyper parameters were optimised only once [1] by minimising the negative log marginal likelihood as given in (3.16) described in Section 3.2.4. The hyper parameter v of the Matérn class covariance function is fixed to $\frac{3}{2}$ for all MC soft sensors. The motivation behind the choice of this value is as argued in [69] that $\frac{3}{2}$ and $\frac{5}{2}$ are the two most interesting values in machine learning.

The evaluated measures reported in Table 4.2 reveal that the best performing soft sensors are either MC or MCard based soft sensors. Precisely, MC and MCard based soft sensors slightly outperformed the SE, SEard, and ANN based

[1] There is no further training carried out during the testing phase.

Table 4.2: Quantitative Performance of the Five Soft Sensors

Data	Soft Sensor	MSE	R	RVTP	TIC	FF	LPD	MSLL
Sulphur Recovery Unit	SE	0.4859	0.6868	0.4628	0.0922	0.5836	1.0943	-0.2942
	MC	0.4312	0.7269	0.5228	0.0591	0.8884	**0.9932**	**-0.3953**
	SEard	0.4453	0.7344	0.5354	0.0673	0.7383	1.0893	-0.3083
	MCard	**0.4224**	**0.7497**	**0.5592**	**0.0383**	**0.8902**	1.0938	-0.3038
	ANN	0.6119	0.7268	0.5208	0.0546	0.0207	1.1180	-0.2796
Debutanizer Column	SE	0.0885	0.9461	0.8946	0.0553	0.9585	0.2593	-1.0720
	MC	**0.0782**	**0.9523**	**0.9068**	**0.0445**	**0.9815**	0.3836	-0.9478
	SEard	0.1700	0.8930	0.7976	0.0636	0.9336	0.2691	-1.0622
	MCard	0.1325	0.9177	0.8421	0.0502	0.9711	**0.2005**	**-1.0308**
	ANN	0.2050	0.9495	0.8495	0.0693	0.9676	0.1929	-1.0385
Industrial Drier	SE	0.7876	0.6064	0.3602	0.01172	0.6833	1.2633	-0.2587
	MC	0.7844	0.6111	0.3621	0.0967	0.8287	**1.2624**	-0.2595
	SEard	0.9476	0.4847	0.3296	0.1132	0.7684	1.3178	-0.2042
	MCard	**0.8359**	**0.5742**	**0.3872**	**0.0522**	**0.9850**	1.2814	**-0.2605**
	ANN	0.8593	0.5666	0.3798	0.1141	0. 7486	1.3276	-0.1943
Thermal Oxidiser	SE	0.1386	0.8756	0.7682	0.1449	0.4991	0.7772	-0.3777
	MC	0.1361	0.8790	0.7725	0.1445	0.5028	0.7721	-0.3828
	SEard	0.1272	0.8870	0.7864	0.1313	0.6082	0.7726	-0.3823
	MCard	0.1265	**0.8874**	**0.7874**	**0.1289**	**0.6220**	**0.7568**	**-0.3981**
	ANN	**0.0009**	0.8838	0.7810	0.1310	0.0757	0.7751	-0.3798
Refinery Process	SE	0.6265	0.6085	0.3685	0.0552	0.9948	1.456	-0.1405
	MC	0.6190	0.6153	0.3760	**0.0459**	0.9963	1.1692	-0.2457
	SEard	0.8329	0.4411	0.1603	0.1205	0.9986	1.3765	-0.0384
	MCard	0.6209	**0.6241**	**0.3953**	0.1093	**0.8727**	**1.1637**	**-0.2513**
	ANN	**0.3042**	0.6195	0.3836	0.1653	0.8506	1.1752	-0.2397
Polymer Batch Process	SE	0.6123	0.5653	0.3194	0.1104	0.7845	1.1721	-0.1933
	MC	**0.4895**	**0.6794**	**0.4556**	**0.0903**	**0.8844**	1.0825	-0.2829
	SEard	0.5064	0.6609	0.4365	0.1686	0.9537	1.0856	-0.2799
	MCard	0.5184	0.6777	0.4230	0.1223	0.9932	**1.0779**	-0.2675
	ANN	0.5630	0.6755	0.4518	0.3107	0.3129	1.0807	**-0.2847**
Catalyst Activation	SE	0.0024	0.9734	0.9475	0.0906	0.9572	-1.5209	-1.3746
	MC	**0.0011**	**0.9868**	**0.9738**	**0.0383**	**0.9985**	**-2.0973**	**-1.9510**
	SEard	0.0034	0.9984	0.9968	0.0596	0.7727	3.8234	2.4045
	MCard	0.0034	0.9984	0.9968	0.0593	0.7757	4.6688	3.2499
	ANN	0.0070	0.9854	0.9692	0.0000	0.7141	-0.4421	-1.8610
Polymerisation Process	SE	0.0552	0.9447	0.9720	0.0872	0.3270	-0.3680	-1.7781
	MC	0.0444	0.9556	0.9775	0.0842	0.3505	-0.8909	-2.3010
	SEard	0.0468	0.9762	0.9529	0.0816	0.3821	-2.7817	-4.1919
	MCard	**0.0276**	**0.9860**	**0.9722**	**0.0712**	**0.5107**	**-3.1950**	**-4.6052**
	ANN	0.02351	0.9807	0.9815	0.2365	0.3782	2.6016	2.6202

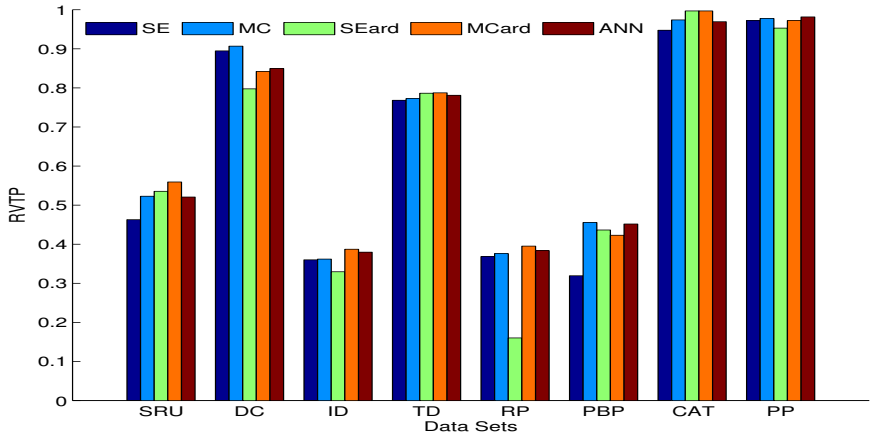


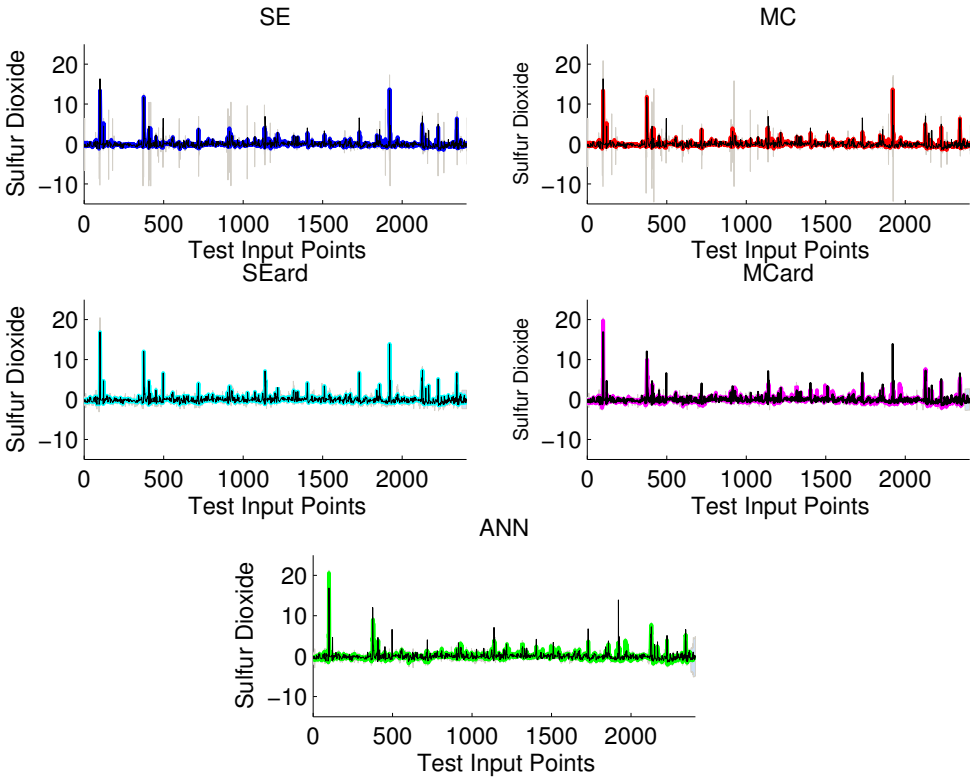
Fig. 4.10: RVTP of SE and MC Soft Sensors

soft sensors in all of the data sets investigated and at most of the evaluative measures used as indicated by the figures set in bold and qualitatively depicted in Figures 4.10 - 4.14.

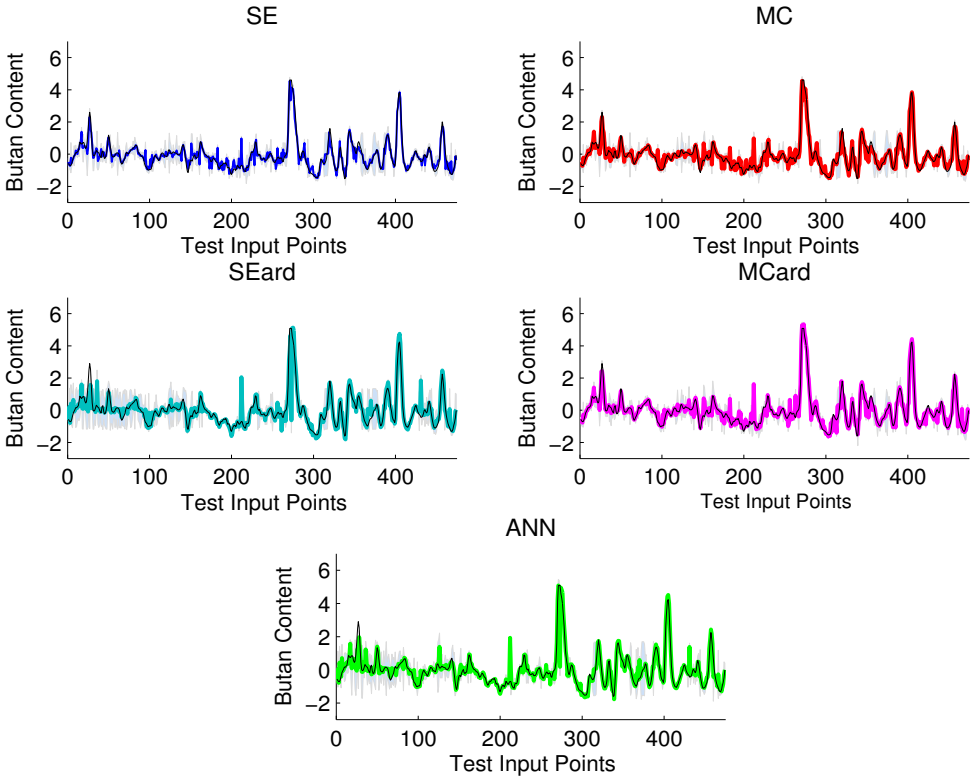
Looking at Figure 4.10, it can be noticed that the less performing models are those based either on SE or SEard depending on the data set. That is down to the smoothness property of the squared exponential functions. Such smoothness can be captured by the Mateŕn class functions as clearly indicated by the performance of the functions in CAT data set where the SEard is performing comparatively as accurate as the MCard.

The quantitative results also reveal that the use of simple covariance functions to model some of the data sets (e.g the refinery process data set) is not appropriate, and for this reason, composite covariance functions that models different structures like linearity, seasonality, and periodicity in the data are more appropriate and results to more accurate predictions.

Slight improvement in performance is of significant importance, particularly if the soft sensor is to be employed in critical closed-loop control applications. In such applications the soft sensor prediction is not used just as indication for control room operators (open loop) but is fed back to a controller to generate a control signal that targets tracking a process set point. The control signal is usually applied to the most influential independent variables used as in the soft sensor input space. In which case, enhancement in prediction accuracy plays a central role in fulfilling the control system targets and in achieving the application objectives.

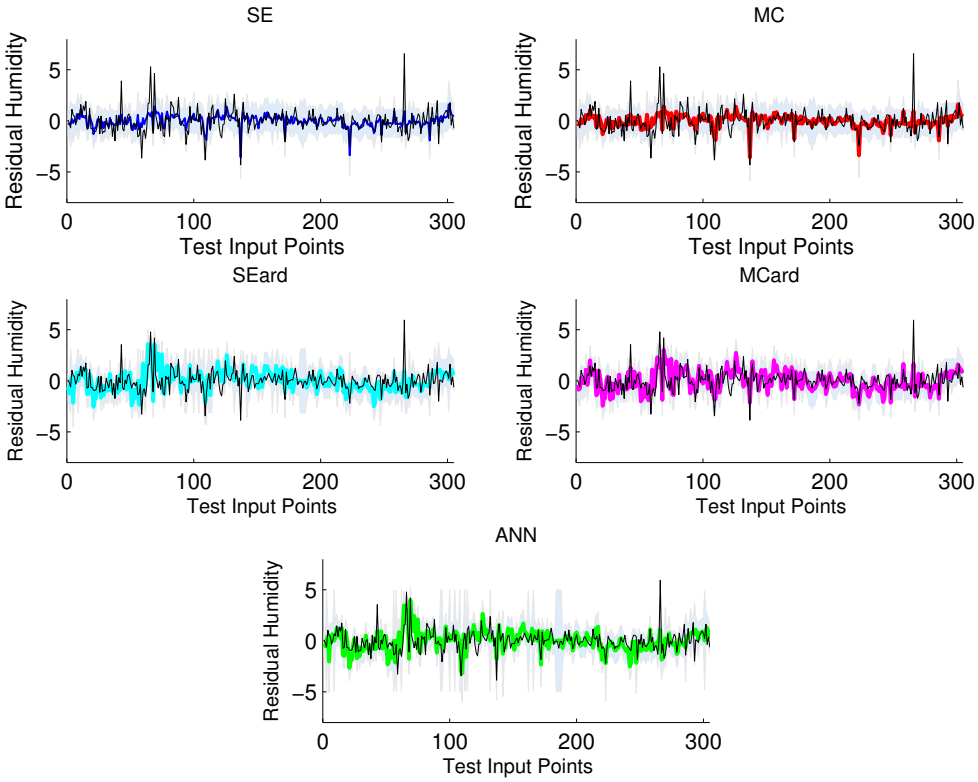


(a) Sulphur Recovery Unit

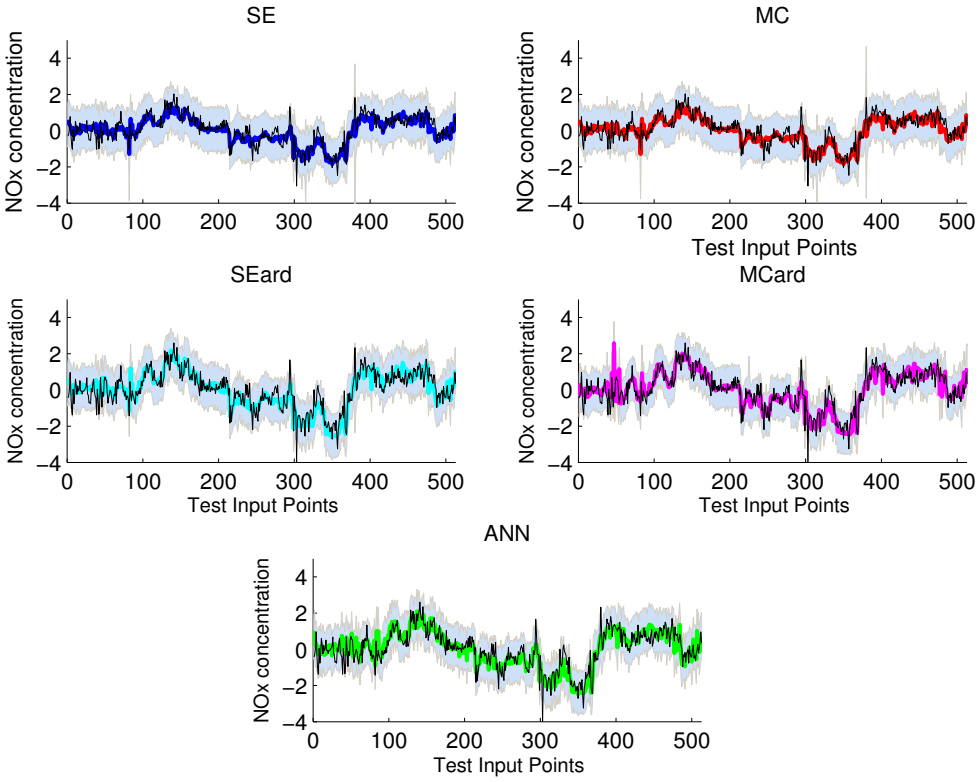


(b) Debutanizer Column

Fig. 4.11: Qualitative Evaluation of the Five Covariance Functions Studied

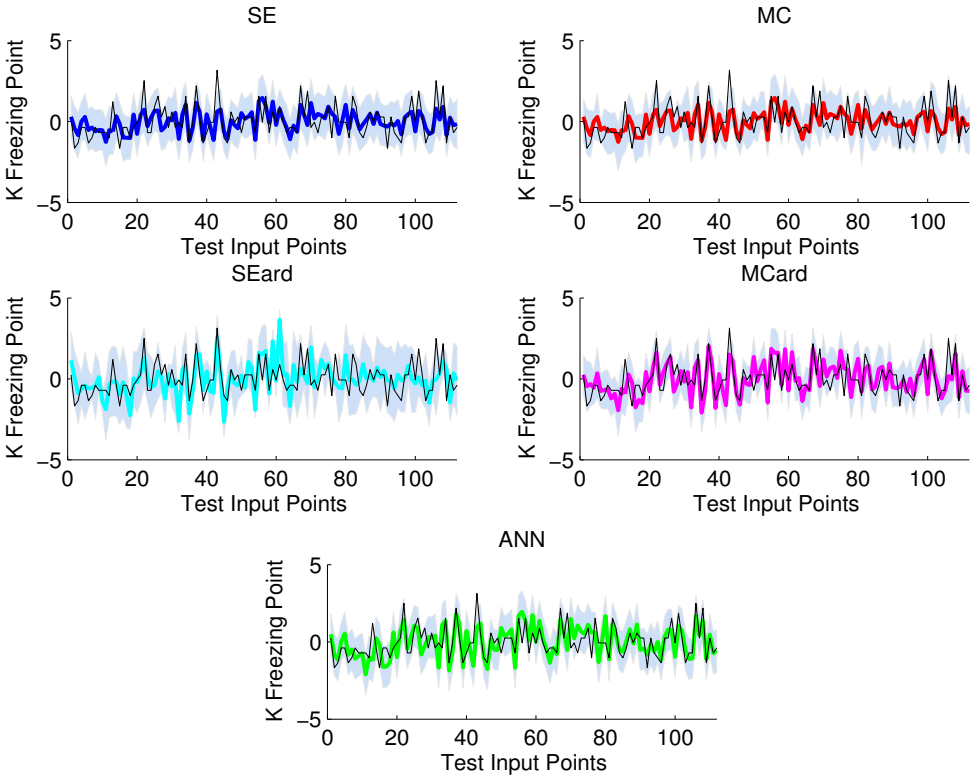


(a) Industrial Drier

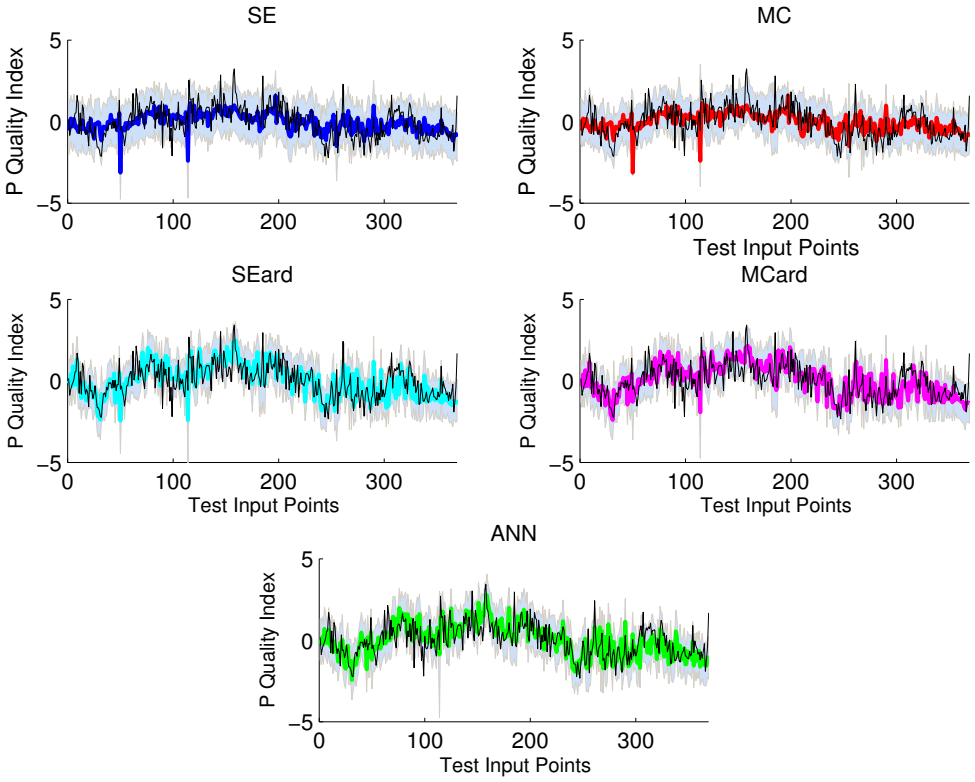


(b) Thermal Oxidiser

Fig. 4.12: Qualitative Evaluation of the Five Covariance Functions Studied

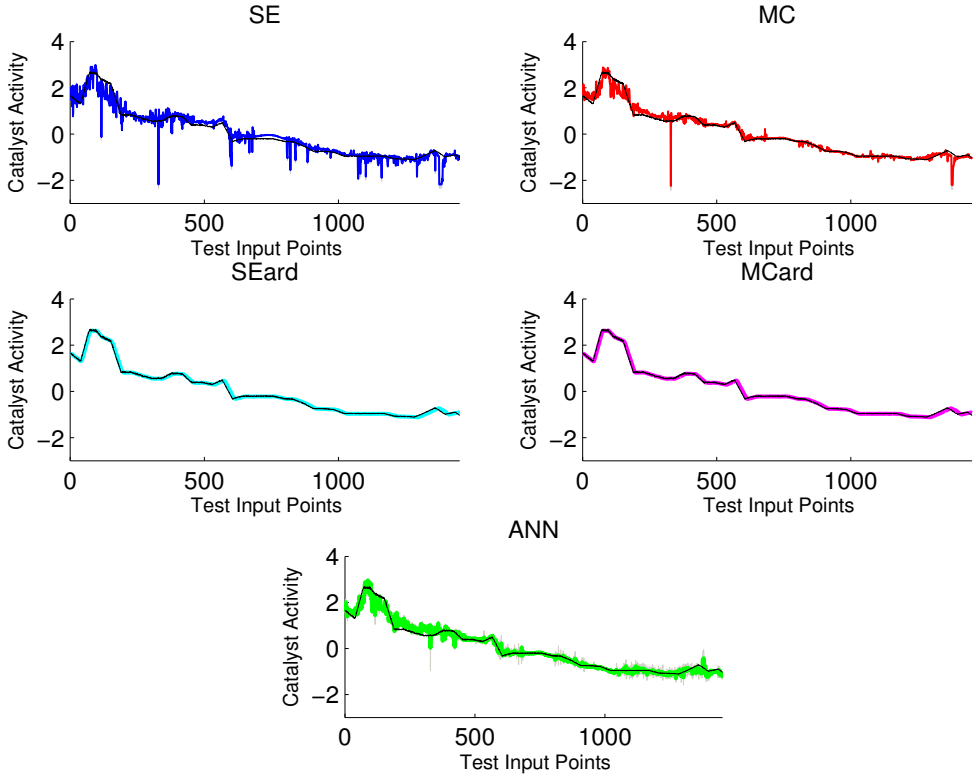


(a) Refinary Process

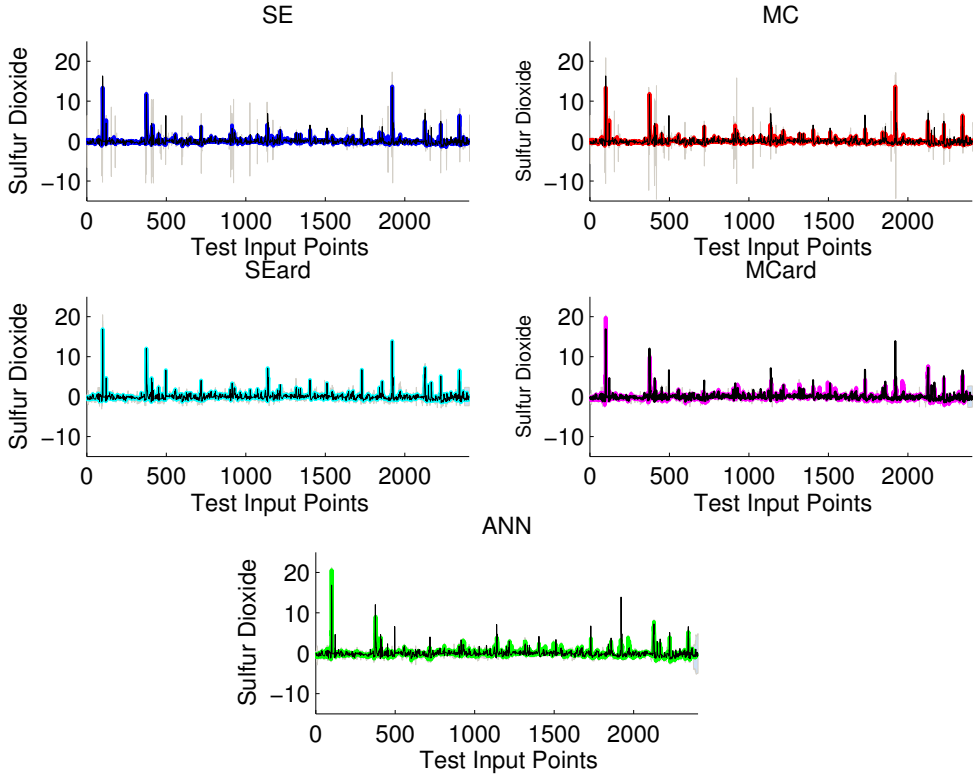


(b) Polymer Batch Process

Fig. 4.13: Qualitative Evaluation of the Five Covariance Functions Studied



(a) Catalyst Activation



(b) Polymerisation Process

Fig. 4.14: Qualitative Evaluation of the Five Covariance Functions Studied

Significance Test

To investigate the statistical significance of the above reported results, one way ANOVA (Analysis Of Variance) tests are conducted and the P-values are reported in Table 4.3 at $\alpha = 0.05$ (95% confidence level):

Data Set	P-Value
Sulphur Recovery Unit	0.8156
Debutanizer Column	0.0.9535
Industrial Drier	0.0.9567
Thermal Oxidiser	0.0.9912
Refinery Process	0.0.9234
Polymer Batch Process	0.0.9273
Catalyst Activation	0.0078
Polymerisation Process	0.0034

Table 4.3: P-Values of the ANOVA Statistical Significance Test

Data Set	P-Value
Sulphur Recovery Unit	0.9467
Debutanizer Column	0.7762
Industrial Drier	0.9324
Thermal Oxidiser	0.9861
Refinery Process	0.8169
Polymer Batch Process	0.8862
Catalyst Activation	0.0001
Polymerisation Process	0.0048

Table 4.4: P-Values of the Kruskal–Wallis Statistical Significance Test

The ANOVA test is based on assumptions that are likely to be violated, such as equality of variance, normality of distribution, level of measurement, and sample size. Therefore, to avoid these assumptions that might be restrictive and to perform a distribution-free statistical test, we resorted to the non-parametric Kruskal–Wallis test. It is an alternative to the ANOVA test in case of non-normality, different sample sizes, inhomogeneity of variances assumption [125]. The P-values of the test are reported in Table 4.4.

Noticeably, some of the data sets have high P-values that fail to accept that MC and MCard-based soft sensors are always outperforming the rest of the sensors; however, this (as explained in [114]) cannot be interpreted as support of the null hypothesis but as a lack of evidence against it that might be caused by error variability in the data. Thus, proving and accepting the null hypothesis does not mean failing to reject it.

Assuming the validity of these significance tests, it can be clearly observed that low P-values have been recorded for the two simulated data sets (catalyst activation and polymerisation process), which are characterised as noise free data. This motivates investigating the noise effect on the overall performance of the soft sensors in general and on the significance tests in particular.

4.6.2 Effect of Noise

To further investigate the above-formulated argument and study the effect of the errors and noise in the data on the overall performance of the soft sensors and on the significance tests, a random noise is added to the two clean simulated data sets (catalyst activation and polymerisation process). The performance of the soft sensors built from these data sets is observed along with the significance tests as reported in Tables 4.5 and 4.6.

Table 4.5: Performance of Simulated Noised Catalyst Activation Data.

Model	MSE	R	RVTP	TIC	FF	LPD	MSLL	ANOVA	Wallis
SE	0.1035	0.5571	0.3088	0.1241	0.6834	1.4855	-0.1825		
MC	0.1030	0.5611	0.3125	0.1271	0.6623	1.4839	-0.1841		
SEard	1.004	0.1172	0.3450	0.1104	0.7974	1.4334	0.0128	0.9879	0.9562
MCard	**0.1282**	**0.5974**	**0.3553**	**0.1065**	**0.8063**	**1.3370**	**-0.0836**		
ANN	0.1404	0.3833	0.2465	0.2733	0.1016	1.3419	-0.0787		

Comparing the results of the clean simulated catalyst activation and the polymerisation process data sets reported in Table 4.4 and the results of the same data sets after adding the noise reported in Tables 4.5 and 4.6, it can be observed that the noise added to the data affected the P-values of the significance tests and the overall performance of the sensors. The added noise precluded MC and MCard based soft sensors to be statistically distinguishable from those based on the rest of the three covariance functions (where it was statistically distinguishable in Table 4.4 before adding the noise).

This can be explained by the fact that an SE, SEard, and ANN covariance functions accumulates more rounding errors when computing the covariance matrix, and if the data is already noisy, the rounding errors accumulated from the inversion process are indiscernible and have negligible effects in the presence of noise. Consequently, MC and MCard models appear almost as the rest of the models; therefore, the significance test fails at a 95% confidence interval. If the covariance matrix is computed from clean data, the errors caused by the use of the SE covariance function for instance will be more evident, as there is no another

Table 4.6: Performance of Simulated Noised Polymerisation Process

Model	MSE	R	RVTP	TIC	FF	LPD	MSLL	ANOVA	Wallis
SE	0.1160	0.9403	0.8820	0.0551	0.7451	**0.1063**	-1.1609		
MC	**0.1089**	0.9448	0.8894	0.0697	0.5267	0.1306	-1.2502		
SEard	0.0926	0.9521	0.9052	0.0387	0.9000	0.1577	-1.2494	0.230	0.968
MCard	**0.0873**	**0.9547**	**0.9106**	**0.0313**	**0.9238**	0.1428	**-1.2643**		
ANN	0.1873	0.8919	0.7401	0.1013	0.8631	0.2147	-1.1924		

source of errors, and thus they propagate to affect the accuracy of the solution. As a result, MC and MCard appears statistically distinguishable from SE at 95% confidence intervals and higher.

In order to validate the above interpretation, the study investigates, evaluates, and compares covariance matrices computed from all covariance functions via the matrix condition number.

Condition Number

The condition number for an n by n data matrix $\mathbf{X}$, given by 4.2, is defined as the measure of relative errors in the inverse of $\mathbf{X}$ and can be used to measure the sensitivity of a system to numerical errors in the data [126]. It is a means of estimating the accuracy of a result when solving linear systems. Specifically, it indicates how far a matrix is from singularity [90, 127].

$$cond(X) = \|\mathbf{X}\|.\|\mathbf{X}^{-1}\| \tag{4.2}$$

where $\|\mathbf{X}\|$ is any norm of the matrix $\mathbf{X}$. However, the Euclidean norm is used to conduct the study in this chapter. In this context, the condition number is an indication of the accuracy of the predictions resulting from a matrix inversion process. Based on this, the condition number is utilised in this study to compare covariance matrices computed from all covariance functions for all data sets used as reported in Table 4.7.

Looking at Table 4.7, it is easily observed that in seven of the data sets MC and MCard covariance function has always resulted in covariance matrices with a lower condition number than those computed from the rest covariance function. It is only in the Thermal Oxidiser data set where the ANN function resulted to a covariance matrix with a lower condition number. This explains why ANN based soft sensor had the lowest MSE score in that data set.

A matrix with a low condition number is said to be well conditioned, and the matrix tends to singularity and turns into a non-invertible matrix as the condition number tends to infinity. In other words, matrices with low condition numbers can be inverted with less rounding errors than those with higher condition numbers. Consequently, predictions resulting from inverting matrices computed from MC and MCard covariance functions are more accurate than those resulting from matrices computed from the rest of the functions, as the former is susceptible to fewer errors than the rest, particularly in clean data sets.

In the context of soft sensor in cases of a lack of prior knowledge regarding

the underlying function of the data, these quantitative and qualitative results reveal that the MC or MCard covariance function should be considered instead of the widely used SE covariance function, as the former will either perform, as accurately as or more accurately than the latter, particularly in clean data sets. In addition, MC should be considered a base kernel among the four kernels considered in the compositional kernel search algorithm proposed in [128] that attempts to automate covariance function selection.

Table 4.7: Condition Numbers of the Five Covaraince Matrices

Data Set	Model	Condition Number
Sulphur Recovery Unit	SE	1.1851e+13
	MC	7.7792e+06
	SEard	9.4429e+06
	MCard	**1.5488e+04**
	ANN	6.3893e+08
Debutanizer Column	SE	1.0217e+06
	MC	**1.8301e+04**
	SEard	1.8382e+06
	MCard	1.2885e+05
	ANN	3.9569e+05
Industrial Drier	SE	2.7402e+12
	MC	**6.1144e+06**
	SEard	7.3400e+12
	MCard	3.3141e+07
	ANN	3.1072e+06
Thermal Oxidiser	SE	1.4826e+16
	MC	4.3860e+08
	SEard	1.9520e+15
	MCard	1.8074e+06
	ANN	**7.3149e+04**
Refinery Process	SE	1.6185e+05
	MC	3.4760e+03
	SEard	82.0937
	MCard	**37.3836**
	ANN	6.7020e+03
Polymer Batch Process	SE	2.2091e+08
	MC	1.0936e+06
	SEard	4.4415e+06
	MCard	**2.0122e+02**
	ANN	4.5363e+03
Catalyst Activation	SE	2.6607e+18
	MC	2.2797e+14
	SEard	1.1073e+19
	MCard	**1.2964e+11**
	ANN	5.5246e+18
Polymerisation Process	SE	4.7205e+18
	MC	2.7637e+17
	SEard	7.7417e+17
	MCard	**3.0250e+11**
	ANN	3.7446e+13

4.7 Summary

This chapter investigates the structure of Gaussian process-based soft sensors applied in industrial plants. The investigation is conducted by an empirical comparison between 40 different soft sensors built from eight industrial data sets The soft sensors built are based on five different covariance functions. Striving towards achieving the ultimate performance from the sensor built, the objective of the study is to devise a practical recommendation on covariance function selection in the absence of prior phenomenological knowledge when building Gaussian process-based soft sensors.

The results demonstrate that the Mateŕn class functions (with and without ARD) marginally outperform the rest of the functions including the widely used squared exponential covariance function, particularly in cases of clean, de-noised, and properly pre-processed data sets. Empirical results reveal that matrices computed from the Mateŕn class functions have a lower condition number. This hints that covariance matrices computed from Mateŕn class functions are further from singularity than those computed from squared exponential function, and thus are less susceptible to rounding errors, thus improving the accuracy of the predictions resulting from those matrices.

The slight improvement in prediction accuracy is of a paramount importance if the soft sensor is developed for inferential control applications where accuracy directly influences the reliability of the control system.

Based on the empirical results reported, the chapter concludes that the Mateŕn class covariance functions should be favoured over the widely used squared exponential function when there is not sufficient prior knowledge regarding the underlying function that states otherwise.

Chapter 5

Adaptive Gaussian Process-based Soft Sensors

5.1 Introduction

Adaptation is modifying the behaviour to fit the environment [129], while in the context of soft sensors, adaptation is the capability of the sensor to change its model structure during online operation [6] to cope with process changes.

When soft sensors were first introduced, they were mainly based on offline modelling using historical data. As expected, historical data does not contain all process conditions, statuses, behaviours, dynamics, and (most importantly) cannot contain future data. Even if that was possible, it would be difficult to identify the model, select the structure, and optimise the parameters or hyper parameters [1] with a method that could comprehend different conditions. Moreover, most of the industrial processes exhibit time-variant behaviours, unpredictable fluctuations, and non-stationary operating phases, the most common causes of which are [6]:

- Process input materials.
- Production of different product quality grades.
- Changes in the external environment.

As a result, model performance deteriorates, which necessitates building adaptive models that are capable of handling these changes (concept drift). This can be mitigated by one of the following approaches [6]:

- Instance selection: moving window techniques
  This technique is based on adapting the model on a selected set of points.

[1] The terms "parameters" and "hyper parameters" are to distinguish the parametric and nonparametric models.

As new data points become available, the oldest point is excluded and the window slides to include the new data point. The model is retrained in a sample-wise or block-wise manner. The former refers to model retraining at every new observation, while the latter refers to the retraining process after collecting a set of observations. In the block-wise approach, the model is retrained at a fixed number of observations or at a threshold point of performance deterioration.

- Instance weighting: recursive adaptation techniques
  Recursive based techniques use the model in the form of a set of coefficients and covariance matrix, which is usually down weighted by means of a forgetting factor, while new data points are added to the window sample-wise or block-wise. The new data points are usually given a higher weight than the previous points through the forgetting factor, which determines the strength of the adaptation process [130]. In such techniques, there is no need to store all the data in the memory.
- Ensemble methods
  This type of framework contains local models deployed to give predictions based on local data clusters, and the global prediction is computed by merging all local predictions.

As Gaussian process is a nonparametric model, the training data creates part of the model and cannot be omitted, as is the case with adaptive models based on recursive methods, this chapter adopts windowing-based techniques to approach adaptability in Gaussian process-based soft sensors. Specifically, it proposes different moving Gaussian process window algorithms updated using different criteria, and the rationale is to maximise the predictive information of the window to enhance the prediction accuracy and the overall performance of the sensor.

The structure of the soft sensors developed in this and subsequent chapters is based on the findings and results achieved in the previous chapter. The Matérn class covariance function with ARD is utilised in building all soft sensors models. However, unlike the soft sensors developed in the previous chapter, the soft sensors developed here are adaptive, and the adaptivity is based on different instance-based selection mechanisms.

5.2 Moving Window Mechanism

In concept drift, the moving window mechanism is used to adapt the model to the current process concept or state. The moving window (often referred to as the sliding window) selects a set of data observations that capture current process conditions and cope with its dynamics.

The moving window approach is based on two crucial and influential parameters: the window size and the step size. The former encodes the amount of data contained in the window, while the latter encodes the interval at which the window is updated.

Once the window is updated, the model can be retrained sample-wise (every new data point) or block-wise (every n accumulated number of data points) according to the specified step size that determines the updating frequency, whereas the window size parameter determines the learning/forgetting factor (often referred to as the stability-plasticity dilemma). Stability-plasticity dilemmas entail finding an optimal compromise between forgetting old process states and learning new process conditions.

Setting a plasticity factor too high corresponds to a step size that is too short and leads to catastrophic forgetting and an unstable model, where important data points that reflect the current process concepts are excluded, and thus the current process concept is forgotten. Consequently, the adaptive soft sensor tends to adapt to noise. In contrast to this is setting a plasticity factor too low, which leads to a model that is too stable and fails to adapt to abrupt and unexpected process changes.

Analogous to step size, inappropriately setting the window size leads to model performance deterioration; therefore, window size should reflect the current process states.

Striving to build a window that reflects the current process states and represent its current concept, the majority of moving window-based techniques assumes that the most recent data points are the most relevant to the current process concept, and based on this, all published moving windows utilise the time stamp of the data to update the window. However, this chapter assumes that old data points may contain valuable predictive information that contributes to reflecting the current process concept. As such, a third parameter for moving window techniques that comprehends the window-updating criterion is introduced. This aims to perform data selection more intelligently rather than using the data time

stamp.

Window Update Criterion

This section describes the general framework adopted for building moving window algorithms, which are distinctively updated using the criteria listed below:

1. Predictive Density Error.
2. Prediction Variance.
3. Mahalanobis Distance.
4. Time Stamp.

The first two criteria are based on the accuracy and uncertainty of the model predictions, whereas the last two are based on a specific characteristic of the input data. However, all algorithms share the same building procedure. The motivation behind excluding observations based on the above listed criteria is to investigate how to preserve the informativeness of the window and improve the accuracy of the predictions. The general procedure comprises three stages illustrated in Figure 5.1 and detailed below:

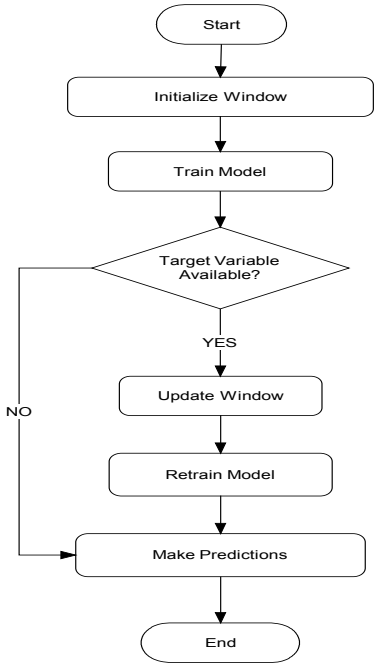


Fig. 5.1: Procedure of Moving Window Algorithms

1. Window initialisation: a) An initial window is constructed that contains inputs from the offline training data or when the algorithm is online from the streaming data associated with their observed targets as in 5.1. The size of the window is chosen based on the type of process and the data set retrieved.

$$X = \begin{bmatrix} x_{1,t-1} & \cdots & x_{j,t-1} \\ \vdots & \ddots & \vdots \\ x_{1,t-k} & \cdots & x_{j,t-k} \end{bmatrix}, Y = \begin{bmatrix} y_{t-1} \\ \vdots \\ y_{t-k} \end{bmatrix} \tag{5.1}$$

Where $(x_{j,t-k})$ is j^{th} input variable at time $t-k$. The time t is the time at which the target variable is predicted.

b) The model is trained and hyper parameters are optimised by minimising the negative log marginal likelihood given in(3.16). To update the window

on the basis of the predictive density error or the variance of the predictions, any output y_{t+1} predicted for a new vector of input points $\mathbf{x_{t+1}}$, the log predictive density error, and the variance of the i^{th} prediction at the i^{th} predicted target is computed, and is stored to be used as an updating criterion.

2. Normal prediction: when the target variable is not available, the algorithm predicts the target variable at the i^{th} point with a prediction confidence; however, the point is not added to the window as the target variable is not available.

3. Adaptation: when the real target variable is available at the i^{th} data point, the algorithm adapts by retraining the model and re-optimising the hyper parameters, and the window is updated according to one of the following updating criteria:

   - **Predictive Density Error**: the log predictive density error of the i^{th} prediction is computed, and the new data point is added to the window after deleting the one that gave rise to the highest predictive density error.
   - **Prediction Variance**: the variance of the i^{th} prediction is computed, and the new data point is added to the window after deleting the one with the highest variance.
   - **Mahalanobis Distance**: Mahalanobis distance between the i^{th} new data point and the data points in the window is first measured, and the new data point is added to the window after deleting its nearest neighbour.
   - **Time Stamp**: this is the conventional updating criteria that utilises the time stamp of the observation, where the most recent data point is added to the window after deleting the oldest data point.

   Finally, the model is retrained online as in (3.16).

As noted above, the window is updated according to the highest predictive density error. This step aims to delete the less informative data points and to maximise the information contained within the window. Deleting points with the highest predictive density error means deleting the inputs that give rise to an output with the highest uncertainty and least accuracy.

5.3 Experiments and Results

Following the soft sensor building procedure described in the previous chapter in Section 4.2, four adaptive soft sensors based on four different selective moving windows are developed. The four sensors are benchmarked against a soft sensor built using the widely accepted Partial Least Square moving window. The experiments were carried out on the eight data sets described in Section 4.3, where they were evaluated quantitatively by means of MSE, R, and RVTP. In addition, qualitative plots of the residual standard deviation are reported for each data set as detailed below:

5.3.1 Predictive Density Error Moving Window

The predictive density error is analogous to the mean squared error; however, the former is a more adequate measure for the evaluation of probabilistic models as argued in [67]. As such, this moving window is based on deleting inputs that give rise to less accurate predictions. The achieved results are reported in Table 5.1 and Figure 5.2:

Data Set	W.Size	MSE	R	RVTP	LPD	MSLL
SRU	**1200**	**0.6434**	**0.5969**	**0.3532**	**1.1052**	**-0.3108**
DC	500	0.3481	0.8092	0.6513	0.8474	-0.5707
ID	300	1.0219	0.3644	0.1328	1.4792	-0.0181
TD	300	0.2339	0.7908	0.6077	0.7941	-0.3609
CAT	20	0.0038	0.9982	0.9963	0.2275	-1.1901
RP	120	0.6758	0.5408	0.2898	1.2192	-0.1712
PBP	450	**0.8283**	**0.2882**	**0.0659**	**1.3494**	**-0.0091**
PP	200	0.0143	0.9930	0.9860	-0.7873	-2.2166

Table 5.1: Quantitative Performance of Predictive Density Error Window

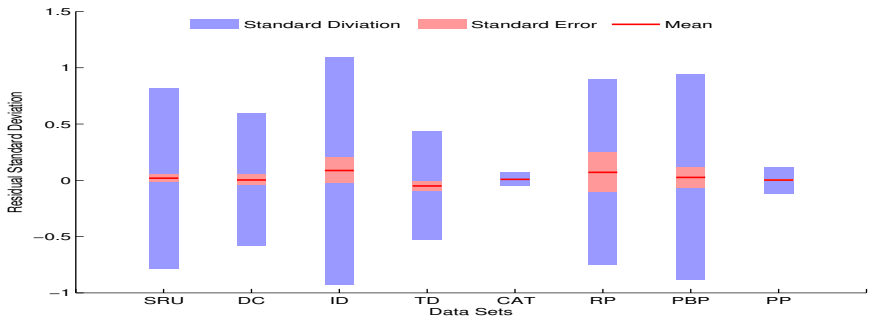


Fig. 5.2: Residual Standard Deviation of Predictive Density Error Window

5.3.2 Variance Moving Window

Aiming to fully utilise the Gaussian process model output that comprises the prediction and its associated uncertainty level (variance), the variance moving window is based on excluding the data points that give rise to a prediction with the highest uncertainty level. A summary of the quantitative and evaluative results is reported in Table 5.2 and Figure 5.3:

Data Set	W.Size	MSE	R	RVTP	LPD	MSLL
SRU	1200	0.6798	0.5628	0.3164	1.1167	-0.2993
DC	500	**0.3309**	**0.8178**	**0.6686**	**0.7873**	**-0.6307**
ID	400	0.9497	0.4363	0.1902	1.4548	-0.0424
TD	300	0.1436	0.8744	0.7624	0.4336	-0.7213
CAT	20	0.0310	0.9845	0.9689	-0.6209	-2.0385
RP	120	**0.6356**	**0.5740**	**0.3295**	1.1810	-0.2094
PBP	450	0.8325	0.2947	0.0607	1.2978	-0.0607
PP	200	0.0159	0.9923	0.9844	-0.8310	-2.2603

Table 5.2: Quantitative Performance of the Variance Window

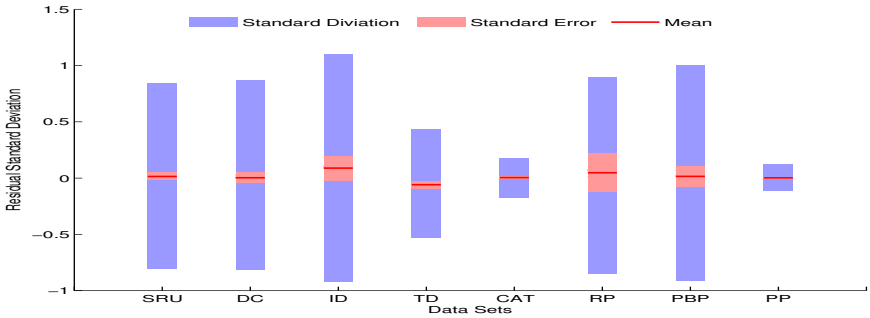


Fig. 5.3: Residual Standard Deviation of the Variance Window

5.3.3 Mahalanobis Distance Moving Window

Striving towards maximising the informativeness of the window and minimising redundancy and correlation among data points, Mahalanobis distance is adopted as a basis to update the window because Mahalanobis distance accounts for the scale and covariance of the data as declared in [131]. Evaluation results are detailed in Table 5.3 and Figure 5.4:

5.3.4 Time Stamp Moving Window

This is the dominant approach to updating windows, where it is assumed that the most recent points reflect current process conditions, while the oldest data

Data Set	W.Size	MSE	R	RVTP	LPD	MSLL
SRU	1200	0.6779	0.5645	0.3183	1.1186	-0.2974
DC	450	0.3548	0.8032	0.6446	0.8254	-0.5927
ID	400	**0.8799**	**0.5069**	**0.2568**	**1.3289**	**-0.1684**
TD	300	0.1369	0.8804	0.7724	0.4096	-0.7454
CAT	20	**2.7584e-04**	**0.9999**	**0.9997**	**-2.6181**	**-4.0356**
RP	120	0.6385	0.5739	0.3293	**1.1775**	**-0.2129**
PBP	450	0.8476	0.2795	0.0436	1.2996	-0.0588
PP	200	**0.0088**	**0.9957**	**0.9914**	**-1.0005**	**-2.4299**

Table 5.3: Quantitative Performance of the Mahalanobis Window

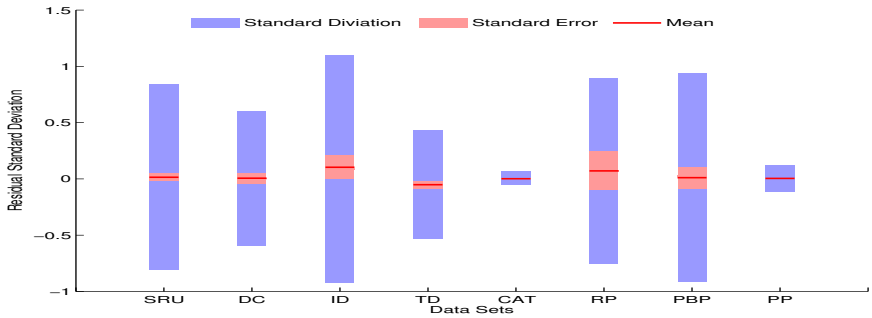


Fig. 5.4: Residual Standard Deviation of the Mahalanobis Window

points are deemed less informative. As such, the window excludes the oldest data point when a new one is available. Details of the results are given in Table 5.4 and illustrated in Figure 5.5:

5.3.5 PLS Moving Window

Like the preceding window, this moving window adopts the time stamp of the data point as updating criteria to update the window. However, it differs in the model structure, in that it is based on a multivariate statistical technique, PLS. This technique is utilised as a benchmark to comparatively evaluate the above reported Gaussian process moving windows.

Data Set	W.Size	MSE	R	RVTP	LPD	MSLL
SRU	1200	0.9589	0.2398	0.0354	1.3961	-0.0199
DC	500	0.7115	0.5385	0.2879	1.2112	-0.2069
ID	400	0.9174	0.4747	0.2249	1.3566	-0.1407
TD	300	**0.1377**	**0.8795**	**0.7706**	**0.4088**	**-0.7461**
CAT	20	2.7370e-04	0.9999	0.9997	-2.7815	-4.1991
RP	120	0.7283	0.4979	0.2291	1.2598	-0.1307
PBP	450	0.8793	0.2977	0.0132	1.3627	0.0042
PP	200	0.0092	0.9955	0.9910	-0.6475	-2.0768

Table 5.4: Quantitative Performance of the Time Stamp Window

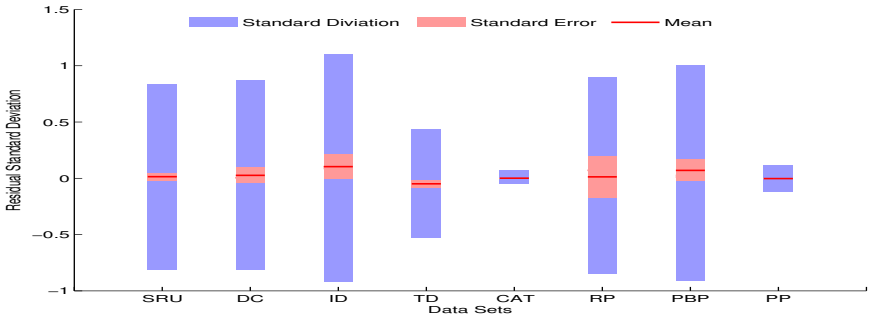


Fig. 5.5: Residual Standard Deviation of the Time Stamp Window

Data Set	W.Size	MSE	R	RVTP
SRU	1200	0.9639	0.1750	0.0304
DC	500	0.8563	0.3772	0.1423
ID	400	0.9876	0.4234	0.1556
TD	280	0.1547	0.8603	0.7400
CAT	20	0.1479	0.9256	0.8526
RP	120	0.7723	0.4608	0.1870
PBP	450	0.8699	0.2351	0.0201
PP	200	0.0118	0.7257	0.5160

Table 5.5: Quantitative Performance of the PLS Window

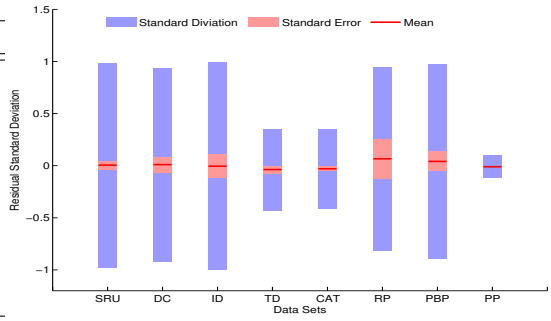


Fig. 5.6: Residual Standard Deviation of the PLS Window

The results are summarised in Table 5.5 and Figure 5.6:

5.4 Analysis

The above results clearly indicate the superior performance of the Gaussian process-based soft sensors over the benchmark PLS method. It is noteworthy; however, that the latter method represents 38% of the overall published soft sensor papers as per the survey conducted in [121].

Comparing the four criteria adopted to update the Gaussian processes windows, it is notable that the Mahalanobis distance outperformed the other criteria in three data sets out of eight, whereas the time moving window (the most widely accepted), has only outperformed the other windows in the thermal deoxidiser data set.

The outperformance shown by the Mahalanobis distance is due to the preserved informativeness of the moving window that is achieved by accounting for the covariance and scale among the data points. Empirical results place the window-updating criterion forward as a third critical parameter among window size and step size when adopting instance selection as an adaptation scheme.

On the other hand, moving window techniques suffer from a major drawback; they demand memory space to collect the data contained within the window. In large windows this drawback is prohibitive and restrictive, particularly in limited memory applications. In addition, in Gaussian process-based soft sensors, large windows demand considerable computational cost. This is primarily due to the known computation cost limitation of Gaussian processes. Being a nonparametric method, Gaussian process models require training data during the prediction phase. Therefore, addressing the computational cost and minimising the size of the window are among the research objectives detailed next.

5.5 Computational Complexity

Computational and memory requirements can be a major obstacle in the implementation of Gaussian process-based adaptive soft sensors online. To circumvent this limitation, which is a known disadvantage of Gaussian processes, many sparse methods have been proposed. A common strategy to most of these sparse methods is the construction of M data points variously referred to as inducing points, active set, support variables, or pseudo inputs [132], which are treated as exact whereas the remaining data is given some approximation, which yields a cheaper cost and less accuracy [132]. These approximation techniques allowed a reduction of the time complexity from $O(N^3)$ to $O(NM^2)$ as in [113] and to $O(M^3)$ as [112].

The active set M is selected based on different criteria, such as maximum information gain, matching pursuit, or greedy posterior maximisation. Another approach in addressing the selection of the active set is proposed in the Sparse Gaussian Processes using the Pseudo-inputs algorithm detailed in [113] where the authors proposed optimising the locations of the active set M by a gradient-based optimisation.

While [112] proposed an approximation technique that is independent on N in which the computational cost is $O(M^3)$, [133] approached the computational and memory problem in a completely different scheme. A sparse covariance function is proposed that can provide intrinsically sparse covariance matrices without the need for any approximation. The output of the suggested sparse covariance function vanishes outside some region R, as $k(\mathbf{x}, \mathbf{x}') = 0$ when $\mathbf{x}'$ is outside R. The region R can be specified automatically during model training through one of the function hyper parameters (the characteristic length scale) [133].

Although the above techniques have improved the computational time, they

are still limited in terms of their practicality in some applications of the soft sensor domain, particularly if the soft sensors are developed for predicting high sampling rate target variables or developed for inferential control tasks. This is due to the fact that the reduced time complexity by the approximation techniques of $O(NM^2)$ and $O(M^3)$ is still not practical in the mentioned applications, specifically, if N is large or if the application demands a large number of inducing points M to get to the desired prediction accuracy.

This section proposes a pre-processing technique that minimises the size of the data used to develop soft sensors and meanwhile preserve the informativeness of the original data. The newly extracted subset can then be utilised to develop a soft sensor based on either full Gaussian processes or based on a Gaussian process approximation technique.

The suggested technique is essentially based on clustering and filtering the data to significantly reduce the number of observations N by constructing an informative and representative subset of the original data. However, in the case of adopting Gaussian process approximation techniques, the constructed subset is further clustered to choose the M inducing points from this subset to base the approximation upon.

The proposed method exploits the results in the previous section, where the Mahalanobis distance is utilised as a similarity measure criterion to construct the informative subset as detailed in section 5.6. The adaptive soft sensors developed to investigate and demonstrate the effectiveness of the proposed method are all based on the Mahalanobis distance moving window described and investigated in the previous section. It is noteworthy that in order to simulate large window situations, all training data are stored in the moving window, hence window sizes are equal to the length of the training data.

5.6 Moving Clustering Filter

Data retrieved from industrial processes is recorded from sensors that are installed for process monitoring and controlling purposes, and therefore this data contains redundant and highly correlated points. As a result, billions of data points can be reduced to millions without any loss of information [134]. For this reason, eliminating such redundant, irrelevant, and misleading points from the data significantly saves time, while preserving the information contained. Com-

putation time is reduced by reducing N and M, the number of data points and the number of inducing points (described in the previous section), respectively.

In the case of using full Gaussian processes, the algorithm addresses the N in the original data (D) by retrieving an informative representative subset (D_{sub}) from the original data (D), while in the case of Gaussian processes approximation techniques, along with addressing the N in the original data, the algorithm improves the selected set of inducing points M from the retrieved subset (D_{sub}).

Retrieval of an informative representative subset is based on a clustering technique that starts by constructing a tree of agglomerative hierarchical clusters C_n from the data contained in the window.

Clusters are created using a single-linkage method, which is based on the minimum distance between points as a similarity measure as in (5.2). Single-linkage is used as it produces more elongated clusters than other methods, thus the data variability can be preserved (besides being considered the most versatile method [135]).

$$D_{mah}(C_i, C_j) = min_{x \in C_i, x' \in C_j} D_{mah}(\mathbf{x}, \mathbf{x}') \tag{5.2}$$

where $D_{mah}(C_i, C_j)$ is the Mahalanobis distance between the i^{th} and j^{th} clusters, and $(\mathbf{x}, \mathbf{x}')$ is a pair of input vectors. Once the input space is clustered, the data points are selected based on their first occurrence, and their associated outputs are selected accordingly. While the choice of the data point selection criterion is a performance dependent process, the idea behind adopting the clustering method in the proposed filter is to group the most comparable points based on the minimum Mahalanobis distance between them. This is to aid in choosing which points to include in subset D_{sub} to preserve the informativeness of the data and which points to exclude to reduce data size (N).

Measuring Mahalanobis distance between the data points and clustering them accordingly serves to screen the data by selecting one data point from each cluster to be representative of the cluster (this implies that the number of clusters determines the size N of the data subset D_{sub}), where choosing the number of clusters is a model selection process that is data and application dependent, and the number of clusters can be optimised accordingly. Once the data size is reduced and the representative subset is constructed, the same clustering and filtering method is used to choose the number of inducing points M from the constructed subset. Choosing M is also a model selection step that is dependent on the issue to solve.

The number of clusters (C_n) in the original data (D) and in the constructed

subset (D_{sub}) can be an arbitrary numbers or optimised numbers that trade off accuracy and computational time.

The moving window algorithm described in section 5.2 and illustrated in Figure 5.1 is adopted in this section such that it is combined with the clustering and filtering algorithm proposed in this section as outlined in the pseudo code below:

Algorithm 1 Mahalanobis Moving Window with Data Filter

1: $\mathcal{W} \leftarrow initialWindow(\mathcal{X}^{train}, \mathcal{Y}^{train})$ ▷ initialize training data window W
2: $\mathcal{D}_{Sub} \leftarrow ClusterFilter(\mathcal{W})$ ▷ cluster W and construct subset D
3: $\mathcal{M} \leftarrow ClusterFilter(\mathcal{D}_{Sub})$ ▷ cluster subset and get inducing points M
4: $\mathcal{GP} \leftarrow optFun(\mathcal{D}_{Sub}, \Theta, \mathcal{M})$ ▷ optimize hyper paramters
5: **if** $y^{online} available$ **then** ▷ in case the target is avialable
6: $\mathcal{M}_{dist} \leftarrow MahDist(\mathcal{X}^{new}, \mathcal{D}_{Sub})$ ▷ measure Mahalanobis between points
7: $\mathcal{V}^{short}, \mathcal{I}_{short} \leftarrow min(\mathcal{M}_{dist})$ ▷ find shortest distance index
8: $P^{old} \leftarrow (\mathcal{X}^{I_{short}}, \mathcal{Y}^{I_{short}})$ ▷ find point with that index
9: $\mathcal{P}^{new} \leftarrow (\mathcal{X}^{new}, \mathcal{Y}^{online})$ ▷ find new data point
10: $\mathcal{D}_{Sub} \leftarrow removePoint(\mathcal{D}_{Sub}, \mathcal{P}^{old})$ ▷ add new points and update window
11: $\mathcal{D}_{Sub} \leftarrow addPoint(\mathcal{D}_{Sub}, \mathcal{P}^{new})$ ▷ delete point with shortest distance
12: $\mathcal{D}_{Sub} \leftarrow ClusterFilter(\mathcal{D}_{Sub})$ ▷ cluster updated W and get subset D
13: $\mathcal{M} \leftarrow ClusterFilter(\mathcal{D}_{Sub})$ ▷ cluster subset and construct M
14: $\mathcal{GP} \leftarrow optFun(\mathcal{D}_{Sub}, \Theta, \mathcal{M})$ ▷ optimize hyper paramters
15: **else** ▷ in case the target is NOT avialable
16: $\mathcal{Y}^{p} \leftarrow predictTarget(\Theta, \mathcal{D}_{Sub}, \mathcal{X}^{test})$ ▷ make prediction
17: **end if**

5.7 Experiments and Results

The plausibility of the proposed method is validated on four data sets from those listed in Table 4.1. These are SRU, CAT, DC, and TD. The first two are comparatively large sized data sets whereas the last two are smaller sized sets, and for this reason, they were chosen to investigate the effectiveness of the proposed method on both data sizes.

The competency of the method proposed is demonstrated by combination with Gaussian process approximation techniques and with full Gaussian processes techniques that also. In the former, the developed soft sensors are adaptive soft sensors, whereas in the latter they are non-adaptive soft sensors that employ the Mater ń class covariance functions. The rationale behind this is to conduct

a thorough evaluation that looks into adaptive and non-adaptive soft sensors. The training time reported for the proposed filtering method includes the hyper parameter optimization time and the clustering time demanded by the moving filter.

5.7.1 Gaussian Process Approximation

The sparse Gaussian processes using Pseudo-Inputs Approximation Technique (SPGP) is used to develop the eight adaptive soft sensors from the four data sets mentioned above. Whereas four of these sensors adopt the filtering and clustering method to minimise the window of the data (referred to as SPGP-F soft sensors), the other four do not, and thus are referred to as SPGP soft sensors.

SPGP is chosen among other approximation techniques, as it provides a more accurate approximation of the full Gaussian processes than other techniques, in addition to the method used to learn the pseudo inputs, which allows improved accuracy [136].

Sulphur Recovery Unit

Tables 5.6, 5.7 and Figures 5.7(a), 5.7(b), and 5.7(c) show the effectiveness of the proposed filtering method. It can be easily deduced that the same performance accompanied with a paramount computational time reduction, illustrated in Figure 5.7(c) and reported in Table 5.7, is achieved by the SPGP-F soft sensor. Table 5.6 indicates that the proposed method efficiently constructed a subset of 39% of the data used by the SPGP soft sensor. While still preserving the informativeness of the window, the SPGP-F soft sensor performs as accurately as the SPGP soft sensor.

Reducing the data to 39% significantly saves computational time (training and testing time) by more than 75% of the time used by the SPGP before the proposed method is applied.
It is also noticeable that the SPGP-F soft sensor performs slightly better than the SPGP; this is clearly seen by TIC, which is usually used to compare different models rather than evaluating one particular model on its own [137].

A visual comparison of performance and time costs between the SPGP and the SPGP-F soft sensors is shown in Figure 5.7. While Figures 5.7(a) and 5.7(b) show that both soft sensors are performing similarly accurately, Figure 5.7(c) clearly shows the time saving achieved by the SPGP-F soft sensor.

Table 5.6: Soft Sensors Quantitative Performance - Sulphur Recovery Unit Data

Method	MSE	R	RVTP	TIC	Data Used
SPGP	0.5727	0.6268	0.3913	0.0163	100%
SPGP-F	0.5690	0.6315	0.3953	0.0093	39%

Table 5.7: Computational Times - Sulphur Recovery Unit Data

Method	Training Time			Testing Time		
	Min	Max	Avg	Min	Max	Avg
SPGP	119.25	170.53	146.24	0.0068	0.0226	0.0104
SPGP-F	24.96	35.15	30.98	0.0019	0.0109	0.0031

Catalyst Activation Process

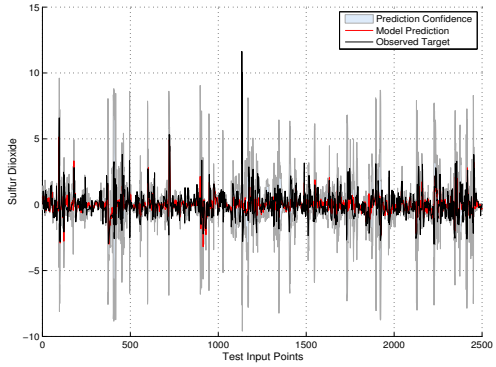
The results reported in Tables 5.8, 5.9 and Figures 5.8(a) and 5.8(b) confirm the findings outlined above, where the SPGP-F soft sensor uses a quarter (24%) of the data that the SPGP soft sensor uses, indicating that the proposed pre-processing method retrieved 24% of the data that can still provide information that 100% of the data provided. This significant data compression contributes towards time savings achieved in Figure 5.8(c) an reported in Table 5.9, which is of essential importance if the soft sensor is deployed in high sampling rate applications (e.g., inferential control systems).

In addition, this demonstrates the competency of the proposed filtering method as the data size gets larger. In applications where the computational time is not a bottleneck, this time saving shown in Figure 5.8(c) can be utilised to further improve the prediction accuracy by adding more informative data points N into the filtered data or increasing the number of the inducing points M in the Gaussian process model.

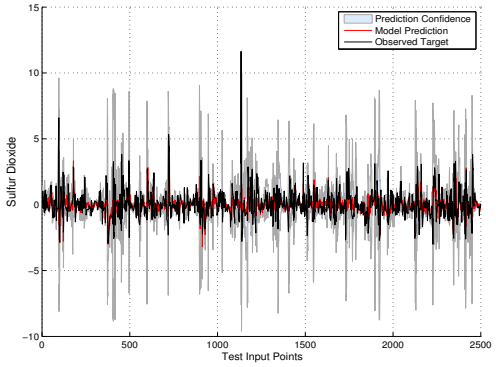
Table 5.8: Soft Sensors Quantitative Performance: Catalyst Activation Process Data

Method	MSE	R	RVTP	TIC	Data Used
SPGP	0.0065	0.9968	0.9936	0.0397	100%
SPGP-F	0.0048	0.9976	0.9952	0.0387	24%

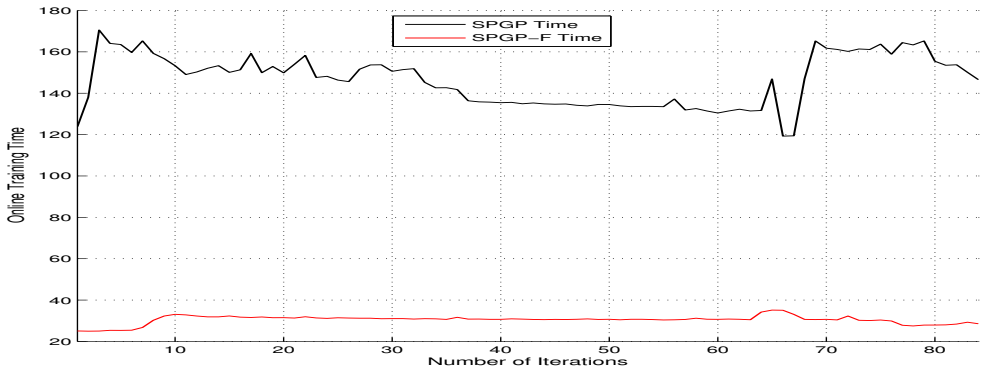
A visual comparison of performance and time cost between the SPGP soft sensor and the SPGP-F soft sensor is demonstrated in Figures 5.8(a), 5.8(b), and 5.8(c). The predictions are accurate, whereas the computational time of the SPGP-F soft sensor is significantly lower than the computational time of the



(a) Performance of SPGP-F Soft Sensors - SRU Data



(b) Performance of SPGP Soft Sensors - SRU Data



(c) Comparison of Online Training Time - SRU Data

Fig. 5.7: Performance of SRU Data Soft Sensors

Table 5.9: Computational Times - Catalyst Activation Process Data

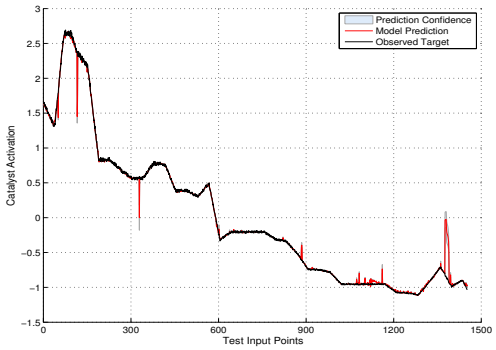
Method	Training Time			Testing Time		
	Min	Max	Avg	Min	Max	Avg
SPGP	103.88	139.06	124.39	0.0051	0.0090	0.0067
SPGP-F	7.53	10.51	9.1047	6.2e-04	0.0181	0.0011

SPGP soft sensor.

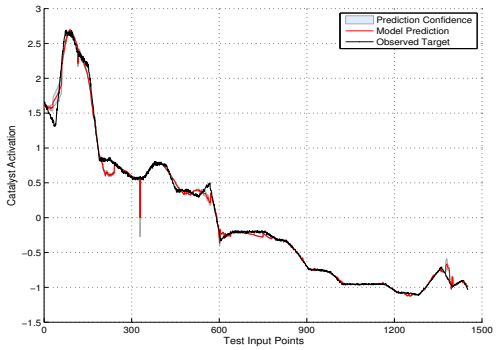
Debutanizer Column

This data set consists of 2394 observations. There was no need to use an approximation technique, but it was used to demonstrate the plausibility of the proposed method on small data sets. It is worth mentioning that in the larger sized data, the plausibility of the proposed filtering method becomes more apparent.

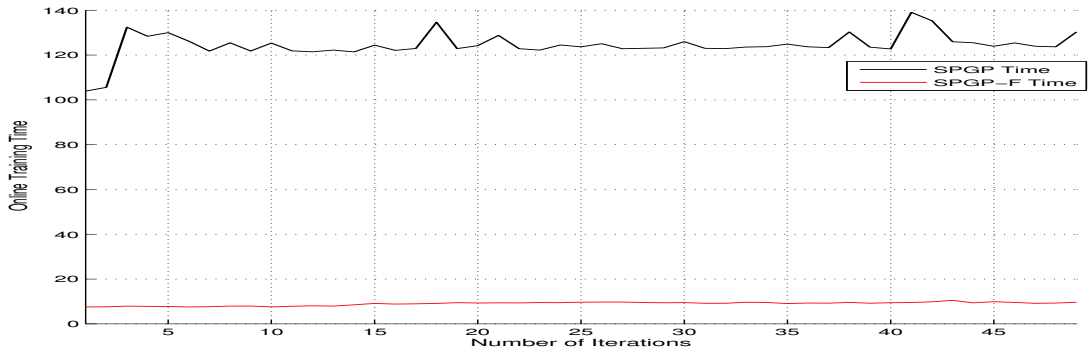
Looking at Tables 5.10, 5.11 and Figures 5.9(a), and 5.9(b), it can be clearly noted that the informativeness of the data can be preserved while the size is reduced to 60% of the original data size. This significantly reduces the compu-



(a) Performance of SPGP-F Soft Sensors - Catalyst Data



(b) Performance of SPGP Soft Sensors - Catalyst Data



(c) Comparison of Online Training Time - Catalyst Data

Fig. 5.8: Performance of Catalyst Data Soft Sensors

tational time by more than 50% as it is shown in Figure 5.9(c), which compares the SPGP and SPGP-F online training times.

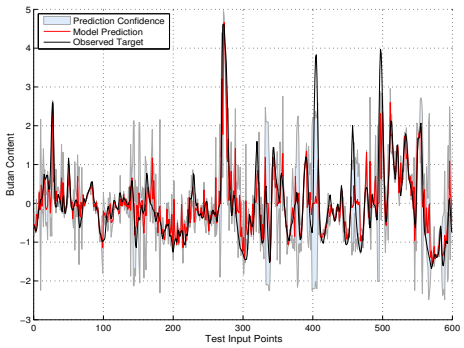
Table 5.10: Soft Sensors Quantitative Performance - Debutanizer Column Data

Method	MSE	R	RVTP	TIC	Data Used
SPGP	0.3257	0.8217	0.6739	0.0909	100%
SPGP-F	0.3263	0.8217	0.6739	0.0621	60%

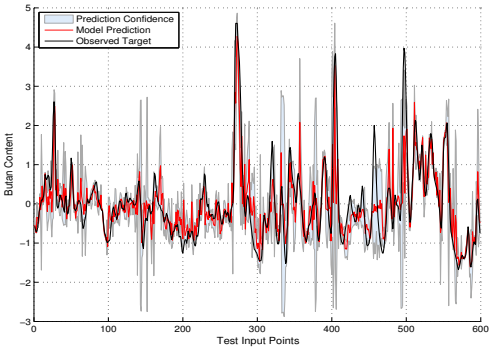
Table 5.11: Computational Times - Debutanizer Column Data

Method	Training Time			Testing Time		
	Min	Max	Avg	Min	Max	Avg
SPGP	41.149	47.788	44.98	0.0035	0.0050	0.0040
SPGP-F	18.91	27.19	21.76	0.0017	0.0038	0.0024

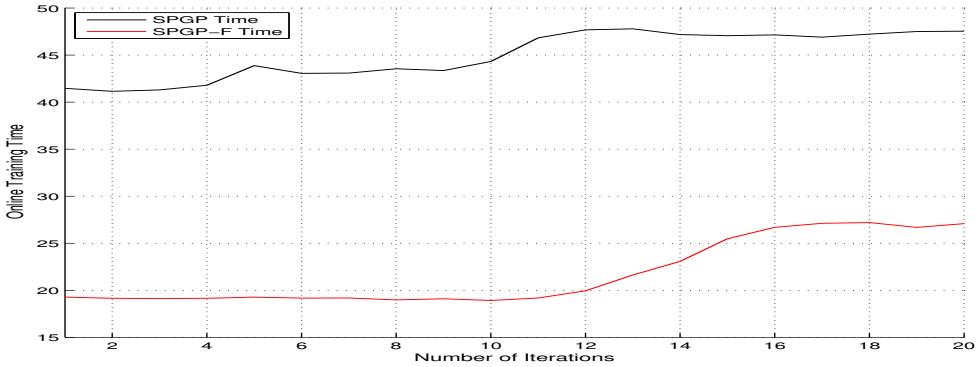
The visual comparison shown in Figure 5.9 also emphasises that the SPGP and the SPGP-F soft sensors are performing correspondingly in terms of accuracy as depicted in Figures 5.9(a), 5.9(b).



(a) Performance of SPGP-F Soft Sensor - Debutanizer Data



(b) Performance of SPGP Soft Sensors - Debutanizer Data



(c) Comparison of Online Training Time - Debutanizer Data

Fig. 5.9: Performance of Debutanizer Data Soft Sensors

Thermal Oxidiser

This is the second small sized data set. The results reported in Tables 5.12, 5.13 show that the soft sensor using the suggested filter. (SPGP-F) is performing relatively as accurately as the SPGP soft sensor, using only 76% of the data used by the SPGP sensor.

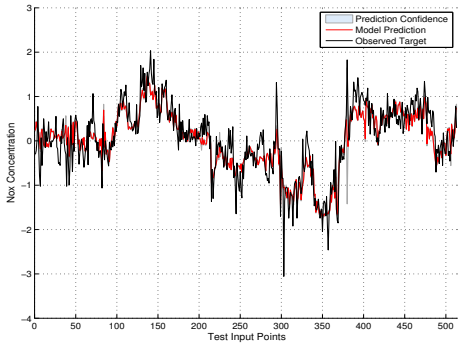
Table 5.12: Soft Sensors Quantitative Performance - Thermal Oxidiser Data

Method	MSE	R	RVTP	TIC	Data Used
SPGP	0.1284	0.8861	0.7844	0.1319	100%
SPGP-F	0.1306	0.8840	0.7802	0.1244	76%

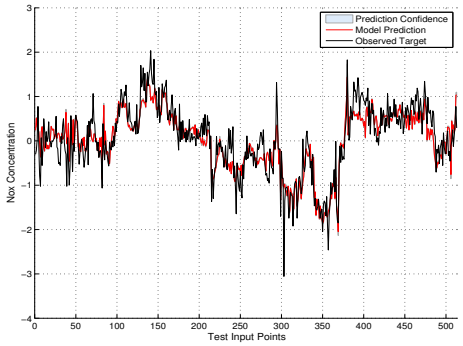
The visual comparison between the online training times of the both soft sensors depicted in Figure 5.10 noticeably illustrates the time reduction achieved by the suggested method.

Table 5.13: Computational Times - Thermal Oxidiser Data

Method	Training Time			Testing Time		
	Min	Max	Avg	Min	Max	Avg
SPGP	22.33	40.00	34.45	0.0021	0.0046	0.0030
SPGP-F	12.22	39.83	23.349	0.0012	0.0092	0.0023



(a) Performance of SPGP-F Soft Sensor - Thermal Oxidiser



(b) Performance of SPGP Soft Sensors - Thermal Oxidiser

(c) Comparison of Online Training Time - Thermal Oxidiser

Fig. 5.10: Performance of Thermal Oxidiser Data Soft Sensors

5.7.2 Full Gaussian Process Models (FGP)

To show the plausibility of the suggested method, and to further emphasise the method robustness of extracting informative subsets from industrial data sets, it is tested on Full Gaussian Process (FGP) non-adaptive soft sensors. The sensors are also developed from the above four outlined data sets. Full Gaussian Process models demand more computational time and give more accurate predictions than approximation methods.

The chapter reports in this section the results achieved from eight non-adaptive full Gaussian process soft sensors as summarised in Table 5.14. While four of the soft sensors employ the proposed clustering and filtering method during the de-

velopment stage (referred to as FGP-F), the other soft sensors do not (referred to as FGP).

The table details the quantitative performance of both soft sensors, where it can be observed that the sensors are similarly performing in terms of prediction accuracy; while in terms of computational training and testing time, the soft sensors employing the method proposed are demanding significantly less time than those referred to as FGP. This is due to the capability of the method to extract a representative subset of the original data that contributes the same informativeness of the original data.

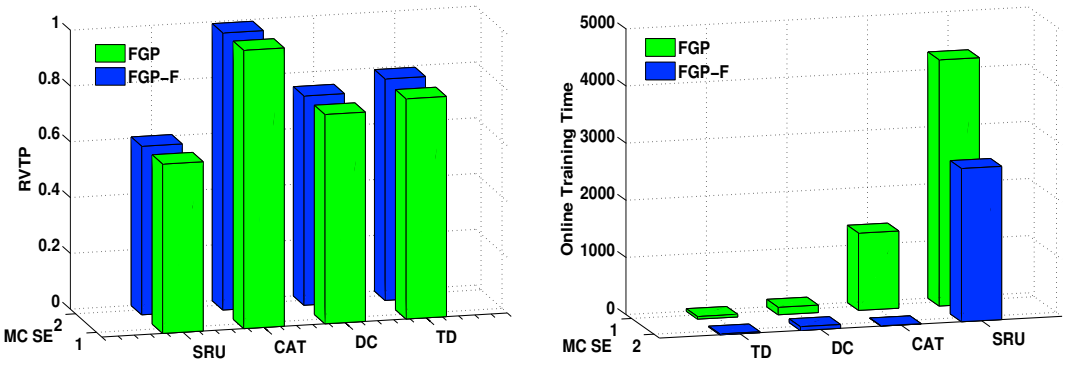
It is noteworthy that the times reported in Table 5.14 are the offline training and testing times (the times taken to train and to test the model on all data at once) as the soft sensors are non-adaptive. Being non-adaptive, the sensors were trained and tested only once, unlike the case with adaptive soft sensors where they were trained iteratively online whenever a new training point becomes available.

Table 5.14: Quantitative Performance of FGP and FGP-F Soft Sensors

Data	Method	MSE	R	RVTP	TIC	Data Used	T_{train}	T_{test}
SRU	FGP	0.3908	0.7804	0.6069	0.0362	100%	4.30e+03	19.997
	FGP-F	0.3934	0.7814	0.6042	0.0216	78%	2.68e+03	19.997
CAT	FGP	0.0028	0.9986	0.9972	0.0269	100%	1.35e+03	3.6392
	FGP-F	0.0068	0.9966	0.9933	0.0492	11%	4.7275	0.1672
DC	FGP	0.2664	0.8563	0.7333	0.0592	100%	132.07	0.6918
	FGP-F	0.2666	0.8563	0.7332	0.0720	90%	66.243	0.4244
TD	FGP	0.1255	0.8887	0.7895	0.1337	100%	45.266	0.2467
	FGP-F	0.1310	0.8845	0.7805	0.1416	59%	14.279	0.1216

A qualitative comparison between the computational time and prediction accuracy demanded by all the soft sensors for the four data sets is illustrated in Figure 5.11. The left plot shows that the FGP-F soft sensors have a comparable prediction accuracy to that achieved by the FGP sensors. This comparable performance is obtained in substantially less time than that demanded by the FGP soft sensor as depicted in the right plot 5.11(b).

The above results reported in Sections 5.7.1 and 5.7.2 empirically demonstrate the potentiality of the proposed method and its enhancement of the practicability of the Gaussian process-based soft sensors in the industrial process. This is attained by minimising the size of the moving window in adaptive soft sensors and minimising the size of the offline training data in non-adaptive soft sensors.



(a) Prediction Accuracy of FGP and FGP-F Soft Sensors

(b) Training Time of FGP and FGP-F Soft Sensors

Fig. 5.11: Comparison of Overall Performance of FGP and FGP-F Soft Sensors

The data reduction gained is of a great importance considering Gaussian process non-parametricity where training data is a part of the predicting model.

The virtue of the proposed data filter is its capability to preserve the informativeness of the original data and save computational time.

5.8 Summary

The chapter reviewed adaptive Gaussian process soft sensors and their practicality in industrial processes. The core adaptive schemes investigated were the windowing techniques, which are essentially based on an instance selection paradigm.

Empirical results indicated that in addition to the two main critical parameters of moving windows (window size and step size) window-updating criterion should be considered a third parameter. Conventionally, windows are updated by including new data points and excluding old ones; presumably they do not reflect the current process state and contribute less predictive information.

The chapter falsifies the above assumption and emphasises a more rationally selective window in which informativeness regarding the current process concept is maximised. Comparing different updating criteria that include predictive density error, Mahalanobis distance, prediction variance, and the time of the data point, it can be concluded that the updating criteria influences the informativeness of the window and deteriorates or enhances the model prediction accuracy. Mahalanobis distance has empirically shown that it resulted in a superior performance as an updating criterion in most of the data sets investigated.

The chapter addresses the computational time complexity of Gaussian pro-

cess-based soft sensors particularly when applied to online predictions using moving window techniques. It proposes a data filtering method that can be combined with full Gaussian process or approximation techniques. The objective is to reduce computational complexity by reducing the number of data points N and selectively choosing the number of inducing points M, while preserving the same prediction accuracy realised as when harnessing the full data, where results show that the developed sensors based on filtered and compressed data that can predict target variables significantly more quickly and accurately as those developed using the full data.

Time reduction, which does not trade off prediction accuracy, is a result of selectively constructing a representative and informative subset by ignoring redundant and correlated points in addition to appropriately choosing a number of inducing points on which the predictive distribution of the Gaussian process model is based.

In applications where the time can be traded off, time reduction can be utilised in improving prediction accuracy by creating more clusters and including more data points into the model. Alternatively, as full Gaussian process models give more accurate predictions than Gaussian process approximation techniques, the suggested method can be used with full Gaussian process models to save time and preserve the accuracy of the exact Gaussian process models.

The proposed method is appropriate for industrial data sets where redundancy, irrelevance, and multi-collinearity are expected, which are typical characteristics of industrial data sets. The method importance is more evident in applications where the target variable is acquired at high sampling rates, such as in inferential control applications.

Chapter 6

Gaussian Process Inferential Control Systems

6.1 Introduction

The problem with controlling inferentially measured process variables, which are often quality indices of particular products, has been an active research area since the 1970s. An adopted solution to address the problem is the implementation of a data-driven inferential control, which is a control scheme that controls inferred difficult-to-measure variables by utilising manipulated influential easy-to-measure variables [138].

Inferential control service as a solution to the problem of direct feedback control when expensive on-line sensors are not feasible [139]. Feedback control principle is based on corrective actions so as to minimize differences between desired set point and actual process output, and that is the key point of implementing feedback control in industry. Feedback control is also used for the purpose of changing the dynamics of a system from unstable to stabilized, from sluggish to responsive, and from drifting to constant. These behaviours can be altered accordingly to meet the application needs. In addition, it is used to reduce system sensitivity to disturbance, and to get rid of undesired effects of noise and distortion [140].

The majority of the publications in the area of inferential control are model driven. Conversely, this chapter approaches the problem from the data-driven prospective. This chapter proposes an inferential control system that employs a data-driven Gaussian process-based soft sensor. The task of the sensor is to infer the difficult-to-measure variable and use it as a feedback signal to an adequate controller. Hence it is termed Gaussian Process Inferential Control System (GP-ICS).

The GP-ICS is validated on controlling the concentration of a chemical prod-

uct in a chemical reactor process. The control system capability of controlling the concentration is further compared and benchmarked against an ANN and KPLS-based inferential control system.

This chapter, further, investigates the reliability of the suggested control system, and it enhances the robustness such that the system dependency on the process variables is minimised, and thus it can be reliably implemented in industrial application.

This chapter employs and utilises the results outlined in previous chapters such that: i) it adopts the Mateŕn class covariance function to develop the Gaussian process soft sensor, which is used as the feedback element in the inferential control system as recommended in Chapter 4, ii) it utilises Mahalanobis distance as a window-updating criteria to improve prediction accuracy, and iii) it employs the suggested filtering method to minimise the size of the moving window as concluded in chapter 5.

6.2 Gaussian Process Inferential Controller

The work presented in this section can be categorised as a model-based predictive control according to [90, 141]. However, instead of building an explicit process model to predict the difficult-to-measure variable, a Gaussian process-based soft sensor is used to provide continuous, reliable, and accurate predictions for the difficult-to-measure variables, which are then fed back to a PI controller.

The Gaussian process inferential control system consists of three elements: i) GP-soft sensor, ii) proportional integral PI controller, and iii) process as depicted in 6.1:

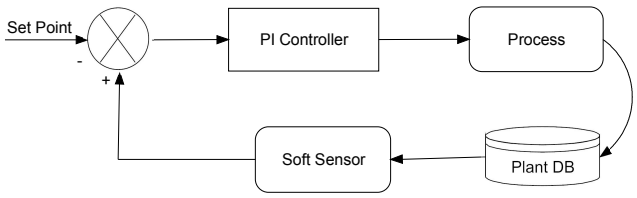


Fig. 6.1: Inferential Control System

For comparison and benchmarking purposes, an ANN Inferential Control System (ANN-ICS) and a KPLS Inferential Control System (KPLS-ICS) are developed. The former utilises an ANN-based soft sensor as a feedback element, whereas the latter utilises a Kernel PLS soft sensor. A description of the chemical

reactor process and the details of the GP, ANN, and KPLS soft sensors are given below:

6.2.1 Process Description

The suggested inferential control system is validated on a chemical reactor model to control chemical product concentration. Figure 6.2 depicts a simple sketch for the chemical reactor process. Seven easy-to-measure process variables are available: three flow measurements (F1, and F2 are the inlet flows, whereas F3 is the outlet flow), and three temperature measurements (T1 and T2 are the inlet flow temperatures, whereas T3 is the outlet flow temperature). In addition, a level measurement that is controlled via F1 using an existing PI controller whose proportional parameter (k_p) and integral parameter k_i are 0.0056 and 3600, respectively. These parameters are dealt with as a fixed part in the chemical reactor process as they belong to an existing PI controller, whose feedback signal is generated based on a real observed (non-inferential) flow measurement (F1).

The chemical reaction in the reactor is dependent on the concentration of a particular component (target variable) inside the reactor. The concentration is measured every 90 minutes using a lab analysis technique, and because of this delay, it cannot be appropriately controlled. The task is to continuously infer the concentration from the seven available measured process variables using the GP-soft sensor and feed the inferentially measured concentration to a PI controller so as to meet the product specified quality and take actions accordingly.

6.2.2 GP-Soft Sensor

As the soft sensor developed in this section is a part of the inferential control system, and the control system performance is entirely dependent on its prediction accuracy, the sensor predictions must be accurate, reliable, and continuous. These properties are obtained by harnessing the conclusions of the previous chapters as described below:

1. A Gaussian process model is first identified adopting the Mateŕn class covariance function as justified in 4.7.
2. A window of training or online streaming data is initialised.
3. The training window is clustered and filtered accordingly to minimise window size, reduce the computational cost, and retain the informativeness of the original data as detailed in 5.6.

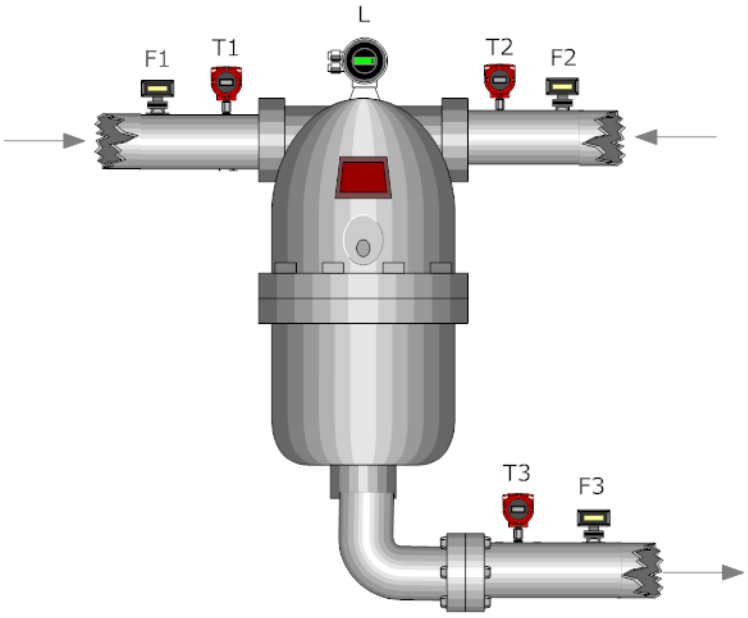


Fig. 6.2: Chemical Reactor Process

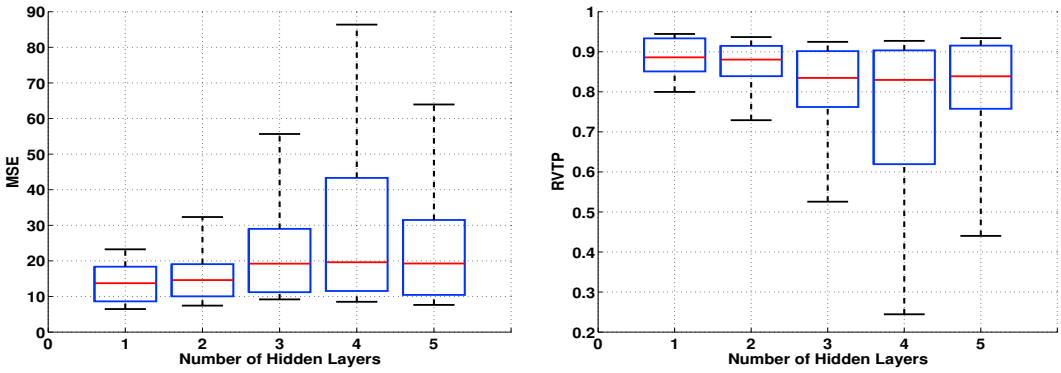
4. The model is then trained and the hyper parameters are optimised as explained in 3.2.4.
5. The model is deployed online and adapted when the target variable is available as suggested in 5.2.
6. The updated window is then clustered and accordingly filtered as in step three.
7. The model is finally retrained and hyper parameters are re-optimised as in step four.

The above steps enable soft sensor adaptability, enhance sensor prediction accuracy, and save computational time when delivering continuous predictions back to the PI controller.

6.2.3 ANN-Soft Sensor

As was previously outlined, ANN soft sensors are predominantly deployed in data-driven inferential control applications. Therefore, an ANN-based soft sensor is developed to be integrated with a PI controller and to build the ANN Inferential Control System used for benchmarking purposes. The ANN soft sensor structure is optimised, where the number of neurons and hidden layers were cross-validated ranging from one to 40 and one to five as demonstrated in Figure 6.3, respectively.

The box plots in panel 6.3(a) and 6.3(b) show that a one-hidden layer network is more accurately capable of predicting concentration than other multi-hidden



(a) MSE of Different ANN Network Structures (b) RVTP of Different ANN Network Structures

Fig. 6.3: Performance of Different ANN Network Structures

layer structures, where the former has the lowest prediction error MSE and the highest prediction accuracy RVTP. As such, the one-hidden layer structure is selected. The number of neurons in the one-hidden layer structure is then optimised. The optimisation criterion is based on the mean squared error and the relative variance tracking precision as shown in Figure 6.4.

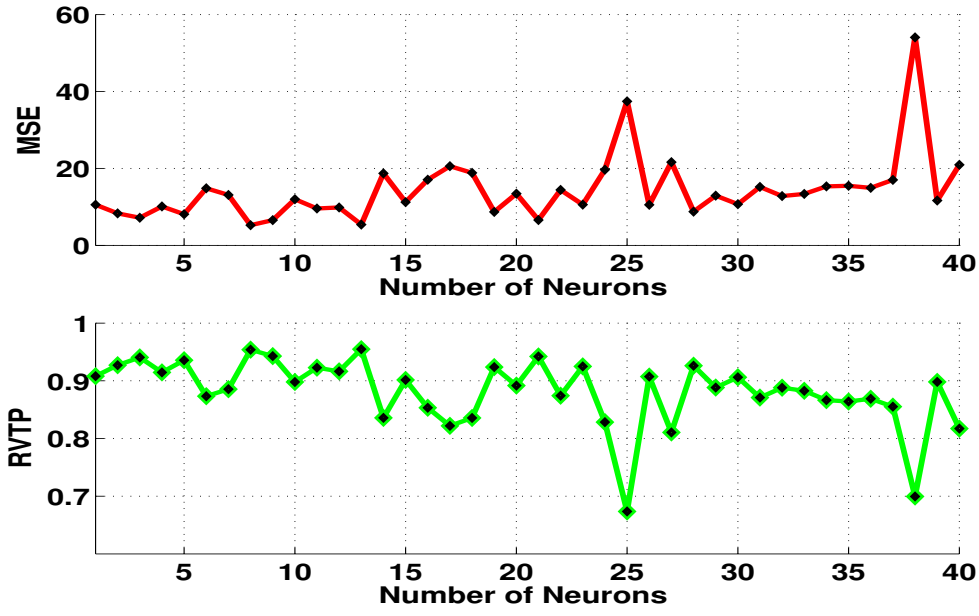


Fig. 6.4: Performance of Different ANN Structures

Based on the model selection procedure above, the ANN soft sensor network structure consists of one-hidden layer with eight neurons. The network is based on a feed-forward neural network trained using the Levenberg Marquardt algorithm.

6.2.4 KPLS-Soft Sensor

Multivariate statistical techniques have also been deployed in inferential control systems; therefore, kernel PLS is used as a second benchmark method to compare with the suggested GP-ICS. A PLS based technique is particularly selected as it was deployed in practical real applications as detailed in [142]. However, to capture the process non linearities, the non linear KPLS variant of the method is adopted. The main idea behind this approach is mapping the original data into a high dimensional feature space. Using the kernel trick, [1] the linear PLS can then be applied in the mapped space.

Identifying the model structure in a kernel PLS involves the selection of a kernel function, and a number of PLS components. The kernel function used to develop the soft sensor is the MSE function, whereas the number of components are cross-validated and selected based on their MSE and RVTP. The number of components explored ranged from one to 30 components as illustrated in Figure 6.5.

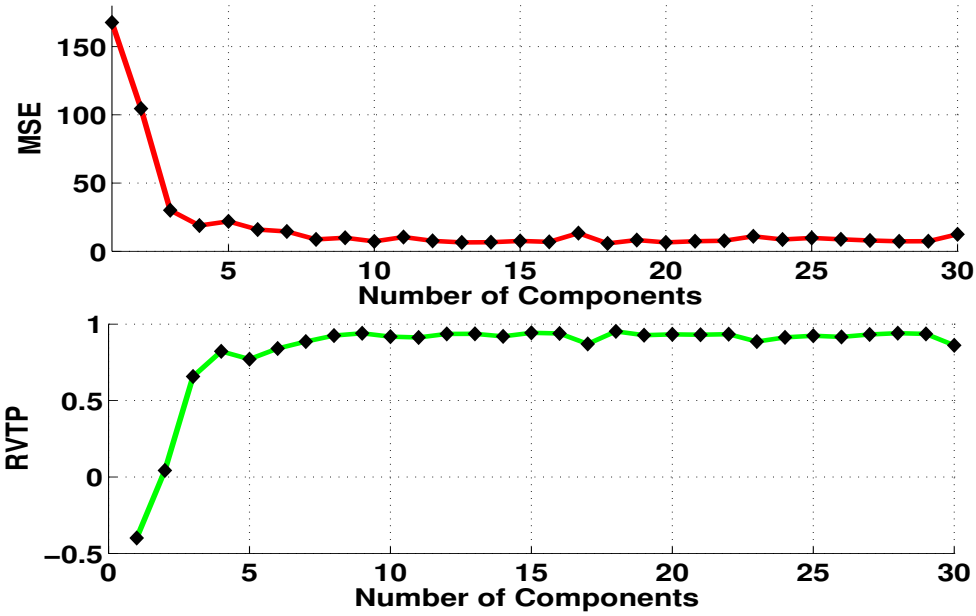


Fig. 6.5: Performance of KPLS Soft Sensor

According to the RVTP and the MSE as model evaluating criteria illustrated in Figure 6.5, 18 components are chosen as an optimal number. Eighteen components give rise to the highest prediction accuracy and lowest prediction error.

[1] The kernel trick is based on the fact that the dot product of two vectors in the feature space is evaluated by the kernel function.

6.2.5 PI Controller

A Proportional Integral Derivative (PID) controller is a process controlling tool that emerged in the 1940s and gained wide popularity ever since. The proposed inferential control methodology is suited to fixed-structure conventional controllers, however, PID controller is chosen as it is the appropriate controller for the chemical processes case study presented in the thesis, in addition to being the most common way of using feedback control in engineering systems [12].

It is also declared in [94] that in today's process control PID represents more than 95% of the control loops deployed, most of which are PI controllers whose formula is given by [12]:

$$u(t) = k_p e(t) + k_i \int_0^t e(t)dt \tag{6.1}$$

where $u(t)$ is the control signal and e is the control error (the difference between the set point and the actual process variable). While $k_p = 100$ and $k_i = 0.05$ are the proportional and the integral parameters, which are empirically tuned.

Control System Requirements

In order to implement the proposed inferential control system in the real chemical reactor plant, the control system controlling the simulated reactor should meet the following design requirements:

1. Closed Loop Stability
   (a) Minimize variability (minimize excessive variation in the manipulated variable T1)
   (b) robust stability (stable for the worst disturbance)
2. Dynamic Response
   (a) Minimum rise time
   (b) Minimum settling time
   (c) Minimum overshoot percentage (for saftey reasons)

6.3 Experiments & Results

In order to evaluate and validate the proposed inferential control system Gaussian Process Inferential Control System (GP-ICS), it is tasked to control the concentration of the target variable in the simulated chemical reactor process described

above. In order to mimic the real measurements retrieved from the seven sensors in the chemical reactor, a random white noise is added to each measurement and all evaluation in this chapter ere conducted in the presence of this noise.

The GP-ICS is benchmarked against two inferential control systems: ANN-ICS, and KPLS-ICS.

The three control systems are further compared to an Ideal Control System (I-CS). The I-CS omits the soft sensors and uses the actual concentration acquired every minute from the simulated chemical reactor process model. Although the target variable is available every 90 minutes in the real plant, in the I-CS case, it is simulated such that it is available every minute to simulate this ideal optimal and unrealistic situation.

MIC Analysis

The Maximal Information Coefficient (MIC) analysis is utilised to select the manipulated variable. The analysis, which is facilitated by the Maximal Information Nonparametric Exploration (MINE) algorithm described in [128], is conducted between the seven process variables and the concentration as illustrated in Figure 6.6. It suggests that the level (L) and (T1) are the most influential variables on the concentration. Because (L) is a controlled variable using an existing controller, T1 is chosen as a manipulated variable to control the concentration.

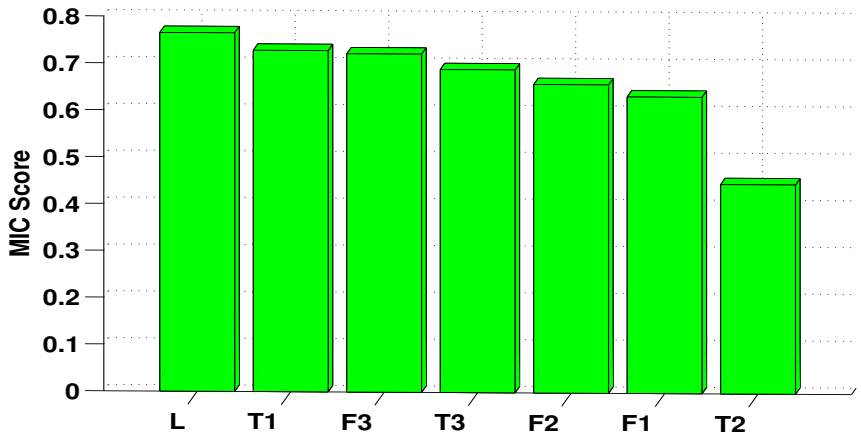
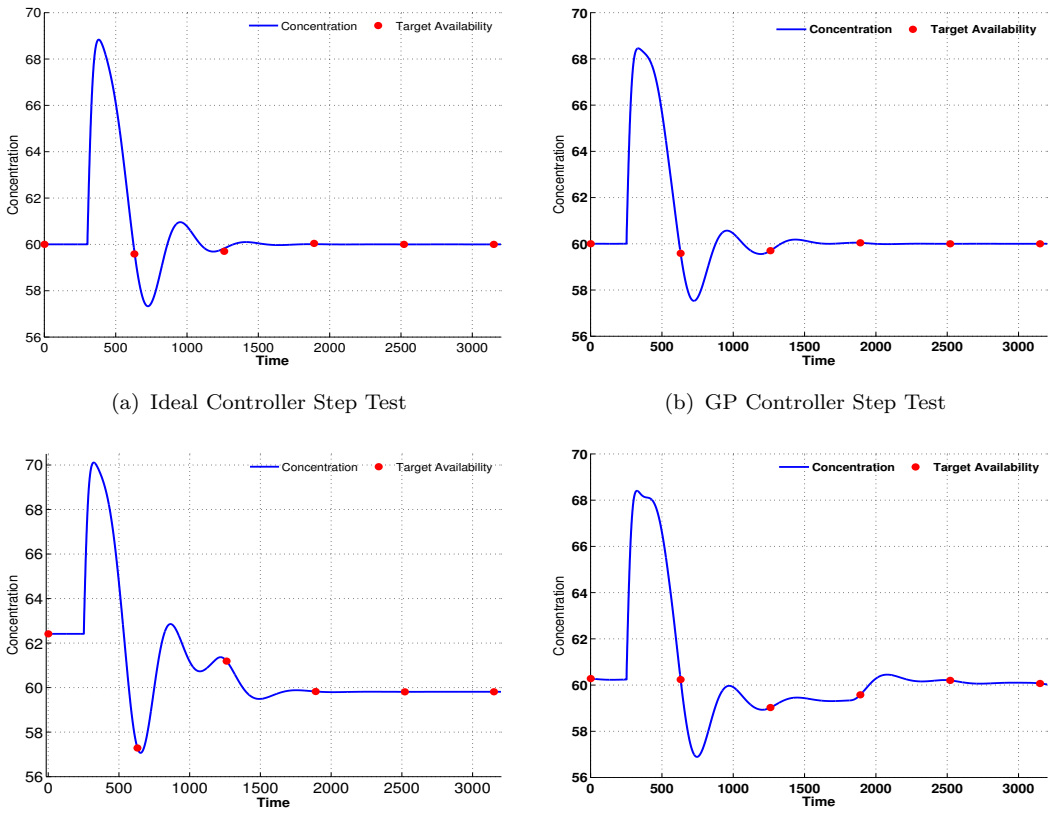


Fig. 6.6: MIC Analysis

6.3.1 Performance Evaluation

The performance of the control systems is assessed using seven evaluative measures. Rising Time (RT), Settling Time (ST), Overshoot Percentage (OP), and

Steady State Error (SSE) are used to evaluate the overall performance of the inferential control systems during the step tests shown in Figure 6.7. Mean Squared Error (MSE), Correlation (R), and Relevance Variance Tracking Precision (RVTP) are used to observe the performance and the prediction capability of the soft sensors. These quantitative results are summarised in Table 6.1.



(a) Ideal Controller Step Test

(b) GP Controller Step Test

(c) ANN Controller Step Test

(d) KPLS Controller Step Test

Fig. 6.7: Responses of ICSs to Step Tests

Table 6.1 shows two groups of evaluating criteria. The first three (MSE, R, and RVTP) demonstrate the performance of the soft sensors developed, and the prediction accuracy for the target variable. The prediction accuracy delivered by a soft sensor is reflected on the overall performance of the inferential control system employed. While the last four (RT, ST, OP, and SSE), which stand for rising time, settling time, overshoot percentage, and steady-state error, assess the control systems' performances, respectively.

It is clearly observed that the GP-based soft sensor outperforms the ANN and KPLS soft sensors, and it more accurately predicts the concentration of the product inside the reactor. This accords the GP-ICS close-loop stability of the

Table 6.1: Comparison between Different Inferential Control Systems

Controller	MSE	R	RVTP	RT	ST	OP	SSE
PI-CS	0	1	1	2.10e-7	115.96	14.725	6.43e-6
GP-ICS	0.9669	0.9863	0.9692	1.23e-6	120.24	14.093	3.78e-5
ANN-ICS	3.8718	0.9434	0.8842	0.5112	121.93	17.218	0.1877
KPLS-ICS	1.3249	0.9830	0.9599	0.0654	138.83	14.055	0.0244

process at a closer point to the set point and in a shorter time than the other two control systems, as indicated by the steady-state error (SSE) and the settling time (SE), respectively.

The SSE and the RT reveal that the GP-ICS performs virtually as accurately as the ideal 'unrealistic' control system and stabilises the process more rapidly and accurately than the ANN and KPLS ICSs.

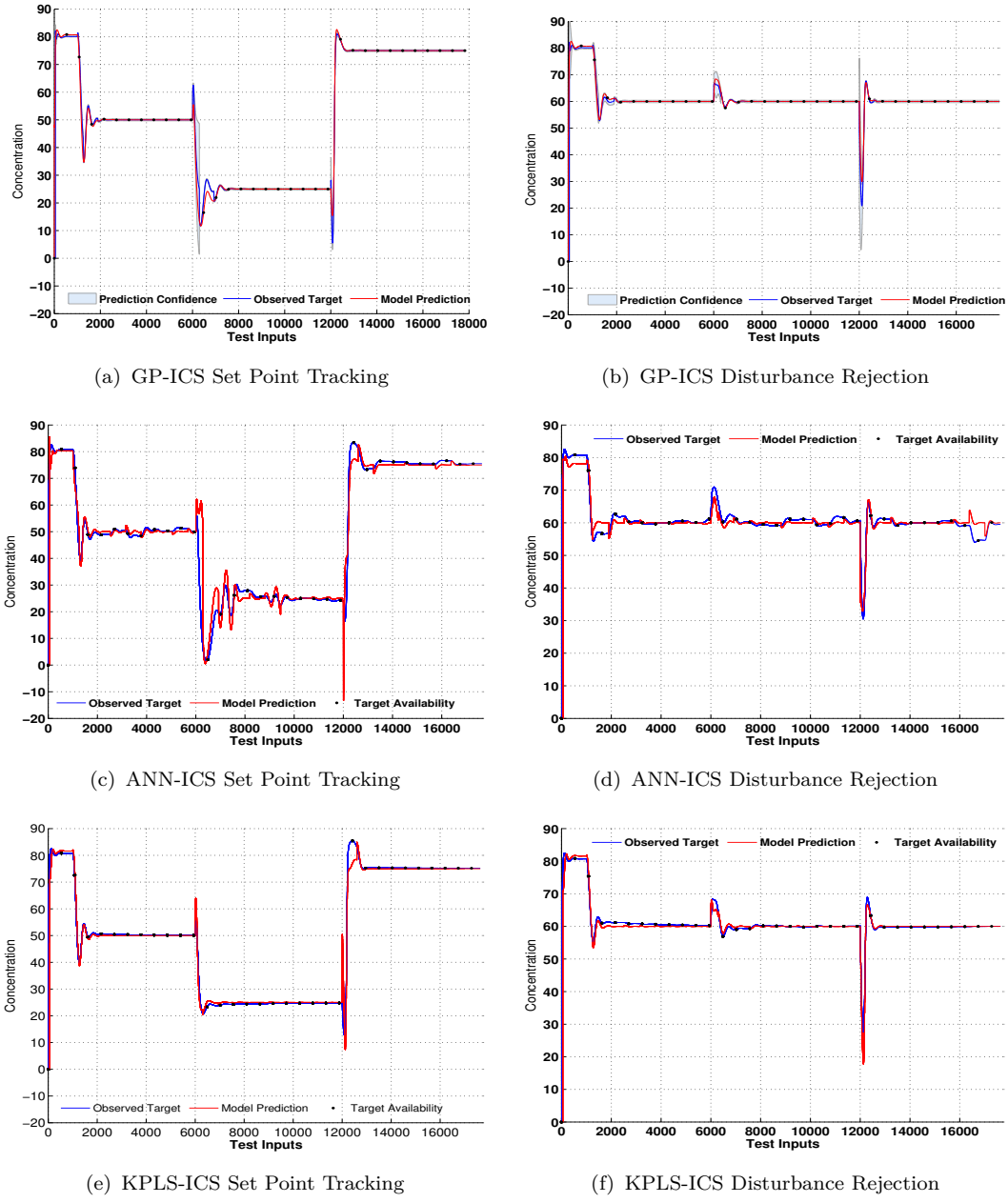
These findings emphasise that the prediction capability of the soft sensors plays a central role in determining the overall performance of the control systems. This was observed in the GP-ICS and can also be seen in the KPLS-ICS, where the KPLS soft sensor has a higher prediction accuracy than the ANN sensor, hence its steady-state error is lower than the ANN-ICS, despite the fact that the settling time is longer.

The remarks outlined above are confirmed by the step tests depicted in the four panels shown in Figure 6.7. The tests are carried out by exciting the process variables during the reactor steady-state condition. Panel 6.7(a) and 6.7(b) illustrate the comparable performance of the GP-ICS to the Ideal Control System where the step responses are virtually as smooth. Conversely, panels 6.7(c) and 6.7(d) indicate the oscillatory responses of the ANN and KPLS ICS systems, where both systems took a longer time to stabilise. In addition, the panels revealed that the steady-state condition of the reactor before exciting the process variables was not at the desired set point (60%) as a consequence of the controller performance that failed to minimize the error signal as accurately as the GP-ICS controller. This contributed to a longer rising time, higher steady-state error, and a longer settling time.

From an adaptation prospective, it can be observed that the KPLS soft sensor requires more data points to adapt than the rest of the sensors. In contrast is the GP soft sensor case, where two data points sufficed the online hyper parameter re-optimisation to adapt the model to the new process concept and preserve the soft sensor prediction accuracy. As a result, a quicker and more accurate process

stabilisation is achieved.

The control systems are also tested on disturbance rejection and set point tracking. The former is carried out by randomly introducing disturbances into the chemical reactor feeds at steady rate of every 5000th iteration, and the latter is carried out by changing the set point to 50%, 25%, and 75%. The qualitative results of these tests are reported in Figure 6.8.



(a) GP-ICS Set Point Tracking

(b) GP-ICS Disturbance Rejection

(c) ANN-ICS Set Point Tracking

(d) ANN-ICS Disturbance Rejection

(e) KPLS-ICS Set Point Tracking

(f) KPLS-ICS Disturbance Rejection

Fig. 6.8: Performance of the Inferential Control Systems at Different Process Operating Conditions

A visual assessment of the six panels shown in Figure 6.8 reveals that the GP-ICS system outperforms the other two systems during both cases: the disturbance rejection and the set point tracking. It can clearly be noted that the GP-ICS is closer to the Ideal Control System than the ANN-ICS and the KPLS-ICS. It can also be observed that the GP-ICS performs more satisfactorily, in particular in the regions where the disturbance is introduced or the set point is changed.

In cases where the disturbance lies outside the range of the training data, the prediction is accompanied by a higher confidence interval, which is advantageous to GP-ICS as the confidence interval is an indication of the reliability of the controller.

The figures also highlight the closed-loop stability achieved by the three systems, where the GP-ICS stabilises the process earlier and more accurately than the ANN-ICS as was empirically indicated. It is clearly noticeable that the KPLS-ICS performs comparably to the GP-ICS, where the two controllers successfully stabilised the process at the desired set point. Conversely, the ANN-ICS failed to perform as accurately.

The empirical results reported reveal that not only can the proposed GP-ICS system fulfil the task of predicting and controlling the difficult-to-measure variable (concentration), but it also outperforms the widely used ANN-based inferential controller and performs comparably to the ideal controller in all conducted tests, including disturbance rejection and set point tracking. Nevertheless, the system is susceptible to deterioration because it is process-variables dependent. Consequently, if one of the hardware sensors that measures the utilised easy-to-measure variable fails, the prediction accuracy degrades and the overall performance of the system is affected. This issue is the main theme of the next section.

6.4 Improving Robustness of the GP-ICS

Considering robustness in data-driven inferential control systems arises from the fact that the hardware sensors that measure input variables are susceptible to failures in any part of their components. These failures can be for various reasons, such as contamination, faulty installation, calibration, power supply, communication, environmental reasons, and/or improper service procedures.

The failure becomes more crucial if the failure occurs in the sensor measuring the variable being manipulated by the inferential control system. Such a failure directly disturbs the feedback signal, (i.e., the predictions of the soft sensor) and

consequently, the overall performance of the control system is disturbed.

To ensure proper reliability and robust functionality of the inferential control system and to preserve its performance during healthy and faulty conditions, this chapter proposes a remedy that minimises the failure effects on the performance of the inferential control system. It is noteworthy that the proposed remedy does not seek to detect, diagnose, accommodate, and /or isolate failure, instead aiming at coping with the failure and maintaining the soft sensor prediction accuracy under this new process concept.

This is accomplished by mapping the input-output data to the feature subspace and extracting the nonlinear scores S before it is fed to the Gaussian process soft sensor model. The nonlinear scores are extracted using the KPCA. As such, the Gaussian process model is fed by the KPCA scores rather than the original data that may contain faulty measurements. The control system employing this pre-processing step is referred to as GP-KPCA-ICS throughout this section.

To evaluate and compare the effectiveness of extracting the nonlinear scores by KPCA, Gaussian Process Latent Variable Model (GP-LVM) is used to build another inferential control system whose input data is the extracted scores via GP-LVM.

While the motivation behind the previous section focuses on integrating data-driven Gaussian process soft sensors with PI controllers to build an inferential control system, and benchmarking it with the state-of-the-art in the area, this section focuses on improving the robustness and the reliability of the overall control system, particularly during sensor input failures. This is accomplished by hybridising the kernel principal component analysis with the Gaussian process inferential control system to achieve the desired practical robustness during all process operating conditions (healthy and faulty statuses). The performance of the hybridized GP-KPCA-ICS system is compared to a GP inferential control system hybridized with GP-LVM. This control system is termed GP-LVM-ICS.

6.5 Kernel PCA

A crucial weakness in the conventional PCA methods is that they cannot generalise data that has complicated nonlinear structures, which cannot be simplified in a linear subspace [94]. To remedy this shortcoming, different nonlinear versions of PCA are proposed, an example of which is the kernel PCA .

KPCA implicitly transforms an input data vector $\mathbf{x}$ into a vector $\Phi(\mathbf{x})$ in a

d dimensional feature space $\mathcal{H}$ so linear PCA is performed in the feature space [143], where Φ is a nonlinear mapping function. However, such a computation in $\mathcal{H}$ has a high computational cost associated with diagonalising the covariance matrix given in [16]:

$$K = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x_i}\mathbf{x_i}^T \tag{6.2}$$

Using a kernel function, the dot product in the feature space can be computed directly in the input space, where it is possible to work implicitly in $\mathcal{H}$ while all computations are done in the input space (kernel trick). The kernel trick is used to map data points into an inner product space where linearity is introduced by means of the dot product. This eliminates the need to explicitly perform feature space mapping [144]:

$$\Phi(\mathbf{x_i}).\Phi(\mathbf{x_j}) = K(\mathbf{x_i}, \mathbf{x_j}) \tag{6.3}$$

Any kernel function that satisfies Mercerś theorem is a valid kernel [145, 146], which ensures that the kernel can act as a dot product in the feature space [16]. An example of which is the Mateŕn class kernel function given by:

$$k_{Mateŕn}(r) = \frac{2^{1-v}}{\Gamma(v)}(\frac{\sqrt{2vr}}{l})^v k_v(\frac{\sqrt{2vr}}{l}) \tag{6.4}$$

where $r = (\mathbf{x}_i, \mathbf{x}_j)$, Γ is the Gamma function, k_v is the modified Bessel function, and v and l are non-negative hyper parameters.

Using such a kernel, KPCA is performed in the original input space as summarised in the following steps [16, 143]:

1. Use 6.4 to construct the covariance matrix K.

2. Centre the covariance matrix K to get $K_{centred}$ as in 6.5.

$$K_{centred} = HKH \tag{6.5}$$

   where $H = I_n - \frac{J_n}{n}$ is the centring matrix, J_n is an $n \times n$ of ones matrix.

3. Use the eigen equation 6.6 to solve for the vectors α

$$K^2_{centred}\alpha^{\mathbf{k}} = n\lambda^k K\alpha^{\mathbf{k}} \tag{6.6}$$

where $\alpha^{\mathbf{k}} = (\alpha_1, ...\alpha_n)^T$ is an eigenvector, and λ^k is the k^{th} eigenvalue corresponding to k^{th} eigenvector $\alpha^{\mathbf{k}}$.

4. Normalise the eigenvectors:

$$\lambda^k(\alpha^{\mathbf{k}}, \alpha^{\mathbf{k}}) = 1 \tag{6.7}$$

5. Compute the k principle components:

$$\mathbf{v}^{\mathbf{k}}_{\mathbf{kpc}}(\mathbf{X}) = \sum_{i=1}^{n} \alpha_i^k K(\mathbf{x_i}, \mathbf{x_j}) \tag{6.8}$$

6. Project data observations onto the principle components to get the scores.

6.6 Robust GP Inferential Control

The suggested algorithm builds upon the soft sensor building procedure described in Section 6.2.2, where it is modified such that the scores of the data are extracted and fed into the soft sensor model instead of feeding the original data, which may contain faulty measurements of the faulty sensor. Details of the algorithm are described in the pseudo code 6.6.1 in which the data is mapped to a feature space using the KPCA. The data can also be analogously mapped to a feature space using GP-LVM:

6.6.1 Pseudo Code

The suggested algorithm is tested and validated on the simulated process described in Section 6.2.1 as discussed next.

6.7 Algorithm Evaluation

The evaluation of the suggested GP-KPCA-ICS is carried out on the simulated chemical reactor process described above. The experiments conducted build upon the results and conclusions found in Section 6.3 where T1 is utilised as the manipulated variable as suggested in Section 6.3, and the GP-ICS is adopted as the primary control system used for assessing the proposed robustness improving algorithm.

Algorithm 2 Robust GP Inferential Control Algorithm

$\mathcal{W} \leftarrow initialWindow(\mathcal{X}^{train}, \mathcal{Y}^{train})$ ▷ initialize training data window W
$\mathcal{S}_{pca} \leftarrow kernelPCA(\mathcal{W})$ ▷ extract data scores via KPCA
$\mathcal{GP} \leftarrow optFun(\mathcal{S}_{pca}, \Theta, \mathcal{M})$ ▷ optimize hyper paramters
if $y^{online} available$ **then** ▷ in case the target is avialable
  $\mathcal{M}_{dist} \leftarrow MahDist(\mathcal{X}^{new}, \mathcal{W})$ ▷ measure Mahalanobis between points
  $\mathcal{V}^{short}, I_{short} \leftarrow min(\mathcal{M}_{dist})$ ▷ find shortest distance index
  $\mathcal{P}^{old} \leftarrow (\mathcal{X}^{I_{short}}, \mathcal{Y}^{I_{short}})$ ▷ find point with that index
  $\mathcal{P}^{new} \leftarrow (\mathcal{X}^{new}, \mathcal{Y}^{online})$ ▷ find new data point
  $\mathcal{W} \leftarrow removePoint(\mathcal{W}, \mathcal{P}^{old})$ ▷ add new points and update window
  $\mathcal{W} \leftarrow addPoint(\mathcal{W}, \mathcal{P}^{new})$ ▷ delete point with shortest distance
  $\mathcal{S}_{PCA} \leftarrow kernelPCA(\mathcal{W})$ ▷ extract data scores via KPCA
  $\mathcal{GP} \leftarrow optFun(\mathcal{S}_{pca}, \Theta, \mathcal{M})$ ▷ optimize hyper paramters
else ▷ in case the target is NOT avialable
  $\mathcal{Y}^{p} \leftarrow predictTarget(\Theta, \mathcal{W}, \mathcal{X}^{test})$ ▷ make prediction
end if

As mentioned in Section 6.6 the proposed algorithm does not employ any fault detection mechanism, where monitoring, isolation, and diagnosis are dealt with accordingly. The algorithm presumes that it is enabled during all process operating conditions. As such, the pre-processing procedure performed using KPCA, and GP-LVM to extract the scores of the data is conducted in both statuses: normal (healthy) and faulty conditions of the process. For this reason Section 6.7.1 is dedicated to empirically show and demonstrate that during normal operating conditions of the process, the prediction accuracy and the controller performance achieved using the KPCA scores in the GP-KPCA-ICS are virtually the same as those achieved using the original data in GP-ICS. However, scores extracted using GP-LVM are not performing as accurately.

The behaviour of the process when the most influential variable (T1) is faulty is investigated at three different operation conditions that include steady state, set point tracking, and disturbance rejection as detailed in Section 6.7.2.

In the real process, the target variable is available every 90 minutes. However, for qualitative comparison purposes, it is plotted continuously so that it can be compared to the continuous soft sensor predictions.

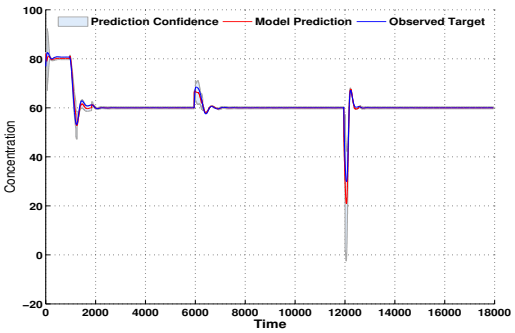
6.7.1 Normally Operating Process

This evaluation case is to establish that the use of the KPCA , and GP-LVM scores extracted from the original data results in the same performance as achieved using the original data. As a result, there is no need to perform any fault detection

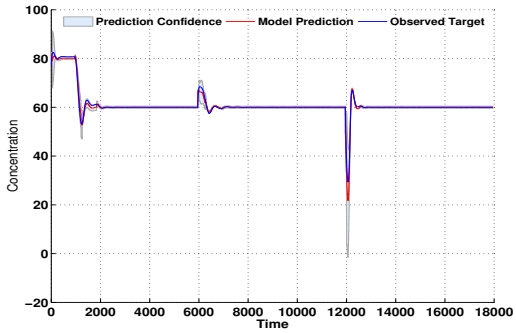
procedure that detects when to enable KPCA or GP-LVM, and when to disable it. This allows a continuous use of the KPCA or GP-LVM as a pre-processing during all process concepts. This precondition is explored and results are reported in Figures 6.9(a), 6.9(b), 6.9(c), and Table 6.2:

Table 6.2: Quantitative comparison between GP-ICS and GP-KPCA-ICS during normal process condition.

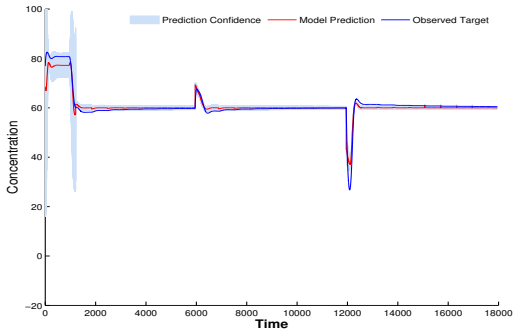
Controller	MSE	R	$RVTP$	RT	ST	OP	SSE
GP	0.7663	0.9892	0.9757	8.66e-8	117.63	11.49	4.07e-5
GP-KPCA	0.6341	0.9909	0.9801	1.77e-7	120.64	11.54	8.16e-5
GP-LVM	3.2298	0.8841	0.9087	4.5091e-04	126.0652	15.4625	0.0012



(a) GP- ICS without KPCA



(b) GP- ICS with KPCA



(c) GP- ICS with LVM

Fig. 6.9: Comparison between GP-ICS, GP-KPCA-ICS , and GP-LVM-ICS - Normal Conditions

The results reported in Figure 6.9 and in Table 6.2 validate and demonstrate that the use of the KPCA scores is as sufficient as the use of the original data. As a result, employing the GP-KPCA-ICS controller during the normal operating conditions of the process does not deteriorate controller performance, rather

it performs comparatively and slightly better than the GP-ICS. Consequently, the GP-ICS performance can be maintained, the need to detect system failures is eliminated, and the KPCA can be continuously deployed during all process operating conditions. Nonetheless, it is noticeable that the inferential control system employing GP-LVM is not performing as accurately, and the confidence intervals associated with the model predictions are higher. For this reason, GP-LVM-ICS is expected to fail to minimize the effect of hardware sensor faults as demonstrated next.

6.7.2 Fault Conditions

As described in Section 6.3, T1 is used for controlling the process. It was for this reason that T1 was chosen to be simulated as the faulty variable, as a failure in T1 depicts the worst-case scenario. Failure in T1 was simulated during three operating conditions of the process. These are steady state, set point tracking, and disturbance rejection. Qualitative and quantitative comparisons were conducted between the performance achieved by the GP-ICS controller, which utilises the raw data, and the performance achieved by GP-KPCA-ICS and GP-LVM-ICS, which utilise scores extracted from the raw data.

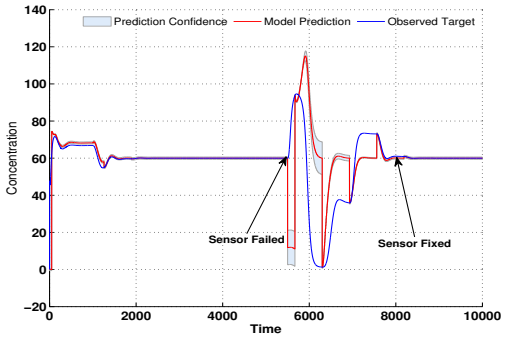
Steady State

Figures 6.10(a) and 6.10(b) show the behaviour of the processes when the failure first occurred and how the controller systems responded to the failure. It is generally expected - in all modes of operation - that the controller has the worst performance when a failure first occurs as the soft sensor is not yet adapted and trained on the failure condition, consequently the soft sensor prediction accuracy significantly deteriorates, which severely affects the controller performance.

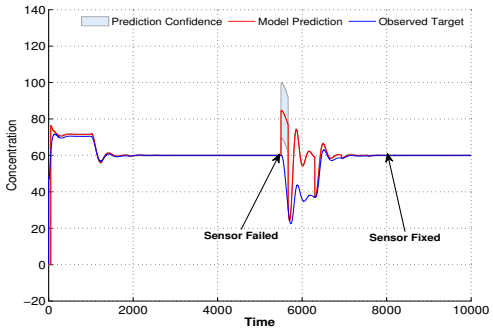
Table 6.3: Quantitative comparison between GP-ICS and GP-KPCA-ICS during the steady-state condition.

Controller	MSE	R	$RVTP$	RT	ST	OP	SSE
GP	830.8	0.2377	-0.2369	0.1238	128.13	53.22	0.3292
GP-KPCA	825.7	0.2983	-0.0816	0.1088	128.12	36.33	0.0569
GP-LVM	1.1694e+03	0.1823	-4.3909	0.0025	136.92	44.24	28.6677

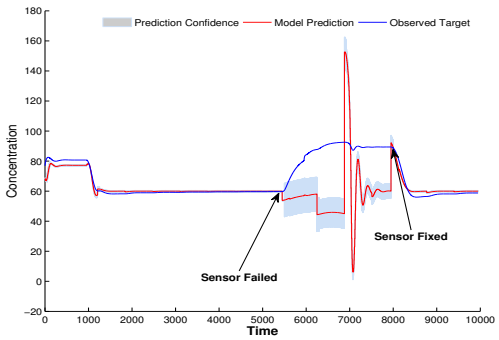
Comparing the figures, it is noticeable, as indicated in Figure 6.10(b), that



(a) GP- ICS without KPCA - Steady State



(b) GP- ICS with KPCA - Steady State



(c) GP- ICS with LVM - Steady State

Fig. 6.10: Comparison between GP-ICS, GP-KPCA-ICS, and GP-LVM-ICS during Steady State

the GP-KPCA-ICS minimised the effect of the failure such that the system has a shorter rising time and a lower overshoot percentage, which can be seen by the smoother response depicted in panel 6.10(b). This allowed the process to settle earlier and with a smaller steady-state error than the process controlled by the GP-ICS as shown in Figure 6.10(a) and empirically reported in Table 6.3.

The achieved performance could not be replicated by the GP-LVM-ICS as shown in Figure 6.10(c) where the predictions of the soft sensor are noticeable inaccurate and thus the controller failed to meet the design requirements in terms of closed loop stability and robustness against fluctuations as indicated by the steady state error.

Figure 6.10(b) and Table 6.3 reveal GP-ICS and GP-KPCA-ICS have virtually the same settling time; however, the soft sensor trained by the KPCA scores slightly outperforms the sensor trained using the raw data as indicated by the MSE, R, and RVTP. In addition, Figure 6.10(b) reveals that the process has settled even before the failure is fixed, which is not the case in Figure 6.10(a).

This is due to the more accurate predictions fed back to the GP-KPCA-ICS than those fed back to the GP-ICS. As a result GP-KPCA-ICS has a shorter rising time, a lower overshoot percentage, and a lower steady-state error as can also be observed in Table 6.3. Conversely, the GP-LVM-ICS has a higher overshoot percentage as a consequence of the simulated fault, and has a significantly higher steady state error that indicates the system's inability to stabilise the process at the desired set point.

Setpoint Tracking

This is to investigate how inferential control systems precisely track a set point change once a failure occurs. The results reported in Figures 6.11(a), 6.11(b), 6.11(c), and Table 6.4 detail the systems' behaviour at two different set points, 40% and 80%. The GP-KPCA-ICS settled the process more quickly (as indicated by the settling time) and more accurately (as indicated by the steady-state error) than the GP-ICS and the GP-LVM-ICS.

Table 6.4: Quantitative comparison between GP-ICS, GP-KPCA-ICS, and GP-LVM-ICS during the setpoint tracking condition.

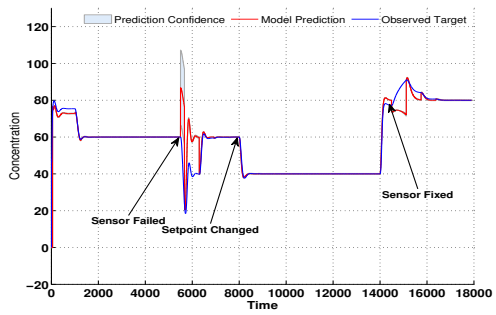
Controller	MSE	R	$RVTP$	RT	ST	OP	SSE
GP	2.0844	0.8977	0.7890	8.28e+4	6.21e+5	1.37e-5	0.0234
GP-KPCA	0.1444	0.9879	0.9686	4.86e+3	5.34e+5	0.52e-6	1.33e-6
GP-KPCA	1.9e+03	0.1541	-1.7822	1.5e+04	8.3e+05	40.19	23.83

Disturbance Rejection

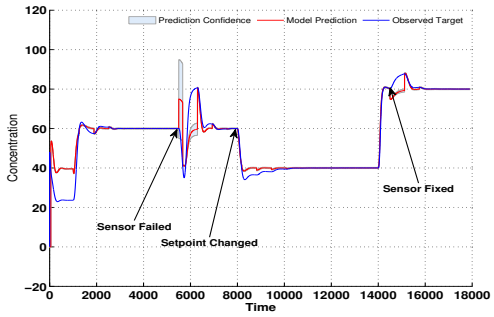
This section describes the robustness of both systems against an extreme scenario simulated as an input disturbance introduced immediately after a failure occurred and before the systems adapted to the failure.

It is clearly noticeable that the GP-KPCA-ICS alleviated the effect of the failure (as demonstrated by Figure 6.12(b)) such that it attenuated the magnitude of the disturbance and prevented the process from extreme overshoots as is the case with both GP-ICS and GP-LVM-ICS depicted in Figures 6.12(a), and 6.12(c) and indicated by the overshoot percentages reported in Table 6.5.

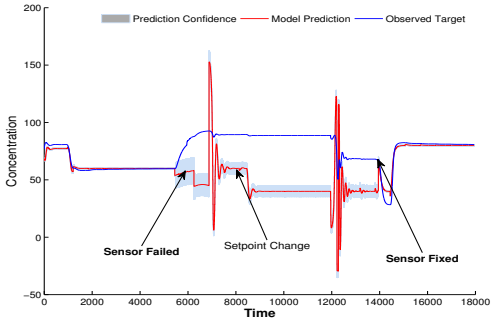
This robust behaviour of the GP-KPCA-ICS is a result of the prediction accuracy of the employed soft sensor used as the feedback element as reported by the MSE, R, and RVTP in Table 6.5.



(a) GP- ICS without KPCA - Setpoint Tracking



(b) GP- ICS with KPCA - Setpoint Tracking



(c) GP- ICS with LVM - Setpoint Tracking

Fig. 6.11: Comparison between GP-ICS, GP-KPCA-ICS, and GP-LVM-ICS during Setpoint Tracking

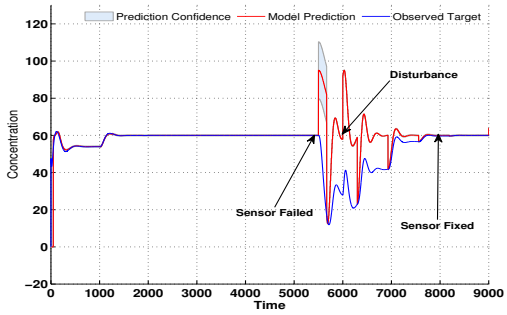
Table 6.5: Quantitative comparison between GP-ICS and GP-KPCA-ICS during the disturbance rejection condition.

Controller	MSE	R	$RVTP$	RT	ST	OP	SSE
GP	307.37	0.1802	-0.339	0.1532	159.10	61.31	0.3495
GP-KPCA	41.49	0.4763	0.0513	0.3578	160.57	26.68	4.52e-4
GP-LVM	1.05e+03	0.0867	-3.798	9.683	119.7	83.75	29.39

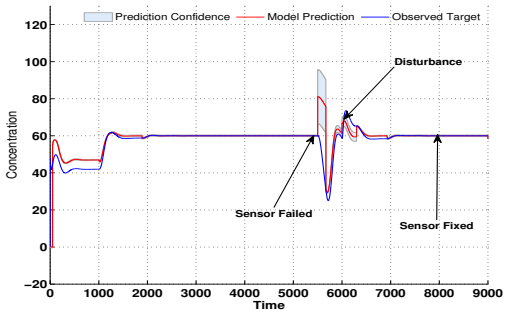
6.7.3 Analysis

The quantitative and qualitative results detailed above assert the validity, reliability, and robustness of performing model identification procedures in the feature subspace mapped using KPCA during both extreme error-in-variable (faulty) and standard process (healthy) conditions.

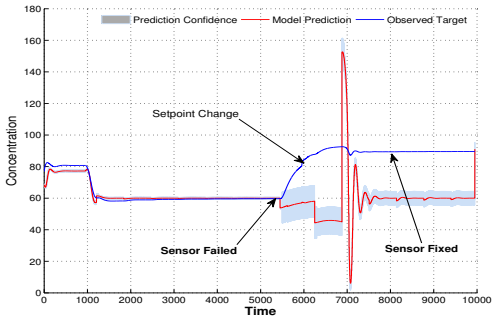
During normal operating conditions, performing model identification in the projected input-output data using KPCA preserved the system's overall perfor-



(a) GP- ICS without KPCA - Disturbance Rejection



(b) GP- ICS with KPCA - Disturbance Rejection



(c) GP- ICS with LVM - Disturbance Rejection

Fig. 6.12: Comparison between GP-ICS, GP-KPCA-ICS, and GP-LVM-ICS during Disturbance Rejection

mance achieved using the original input-output data. However, the empirical results reported in Table 6.2 indicate that the use of the projected data introduced longer rising and settling times; nevertheless, the process is stabilised with a lower steady-state error. It is noteworthy that mapping the data using GP-LMVM does not produce similar results as those achieved using KPCA.

The results also reveal the advantages of performing the model identification procedure in the projected data subspace where scores of the projected data summarised the informativeness of the raw data unaffected by the faulty measurements contained therein. This is due to the numerical reliability of the subspace model identification [147] performed by the KPCA. In addition, as argued in [148] properly extracting features can reduce the effect of noise and eliminate redundant information in the data that is irrelevant to the task the data is used for.

In contrast, scores retrieved using GP-LVM were strongly correlated that deteriorated the soft sensor prediction accuracy, and consequently the ICS employing the GP-LVM failed to meet the control system design requirements detailed in

6.2.5.

The Gaussian process soft sensor identification and validation in the KPCA feature subspace resulted in more accurate predictions, observed in the three failure conditions explored, which always caused the GP-KPLS-ICS control system to have a lower steady-state error than the GP-ICS despite the fact that this is often achieved at the expense of a longer settling time as noticed during the disturbance rejection validation case.

The GP-KPCA-ICS system has a considerably smoother response to the T1 simulated failure than the GP-ICS system. This is observed in the three failure conditions simulated as visually demonstrated in all plots reported. That can be explained by the reduced correlation and increased informativeness attained from the data extracted KPCA scores.

A fundamental issue with data generated under closed loop conditions is the high level of correlation between noise and process measurements [149]. Minimising the correlation in the input-output data matrix achieved by the subspace projection matrix (scores) allowed for a better-conditioned covariance matrix used by the Gaussian process soft sensor to predict the feedback signal. This, in turn, gave rise to more accurate predictions achieved by the hybridised model (GP and KPCA). Providing the controller more accurate predictions essentially influenced the control system's overall performance; therefore, GP-KPLS-ICS alleviated the effect of T1 failures, had a smoother response, and consequently settled the process more accurately and in a virtually similar time as the GP-ICS system.

6.8 Summary

This chapter introduces and explores the applicability of Gaussian process-based soft sensors in closed loop PI controllers and the reliability and robustness of the constructed control system, which is examined and enhanced using a hybrid method. The suggested algorithms are validated on a simulated chemical reactor process to control the concentration of a particular component inside the reactor, which is designated the reactor quality index.

The first part of the chapter looks into utilising the developed Gaussian process soft sensor in the previous chapters and explores its practicality in building inferential control systems. The proposed data-driven Gaussian process inferential control system is benchmarked and compared to ANN and KPLS-based inferential controllers, where the two systems employ ANN and KPLS soft sen-

sors, respectively. In addition, the suggested system is compared to an ideal optimal controller that mimics an unrealistic situation in which the target variable (concentration) was assumed to be continuously measured. As such, the optimal controller utilises continuously observed measurements acquired every minute from the process simulator.

Empirical results show that the proposed system accomplishes the task of predicting and controlling the target variable and outperforms the benchmarking systems, particularly the ANN inferential controller. The GP-ICS performs as accurately as the ideal controller in all tests including disturbance rejection and set point tracking tests.

The second part of the chapter explores the reliability of the GP-ICS, as it is data-driven and process variable dependent. Process variables are susceptible to failures for different reasons, and such failures can significantly affect the prediction accuracy of the Gaussian process sensor employed as the feedback element of the control system.

This scenario is mitigated by projecting the input-output raw data into a new subspace using Kernel PCA and performing model identification and validation in the new data subspace. In other words, using the KPCA scores as Gaussian process soft sensor inputs instead of using the raw data. For comparison and evaluation purposes, mapping the data to the feature space was performed using GP-LVM and another inferential control system was built that employ this pre-processing step, however, the GP-LVM mapping resulted to very strongly correlated scores that in consequence affected the soft sensor prediction capability. This in turn caused the control system failure to stabilize the process timely and accurately.

The proposed readability-enhancing techniques are validated on a worst-case scenario where the sensor of the manipulated variable (T1) is simulated as a faulty sensor during various process operation conditions. Empirical results show that not only can the GP-KPCA-ICS proposed system minimise the failure effects, but can also maintain and preserve controller performance during the normal operating conditions of the process. This eliminated the need to perform fault detection procedures and allowed the continuous deployment of the system during all process conditions.

Chapter 7

Conclusions

7.1 Research Summary

Data-driven soft sensors are gaining wide popularity as they offer valuable solutions to problems associated with their hardware counterparts. This is attained at low costs, less time, and with efficient use of resources. In addition, they are considered valuable alternatives to the first principle models due to the complexity and difficulty of development associated with the latter.

The Gaussian process is emerging as a promising soft sensor modelling technique that retains a relatively easy implementation and offers a comparatively simpler model identification. Despite this, the use of Gaussian processes in the soft sensor domain is hindered, its wide adoption is hampered, and its modelling advantages are not well recognised.

This research targets the applicability of Gaussian processes in the soft sensor domain. It explores their practicality in industrial applications and investigates their appropriateness in critical chemical processes.

This research first outlines the central role of data-driven soft sensors, particularly in chemical processes. It reviews and critically analyses conventional soft sensor building procedures. It pinpoints some current shortcomings and emphasises the importance of a systematic integration of expert knowledge into soft sensor building procedures.

The research steps through the stages of building soft sensors and sheds light on the data pre-processing phase where it brings into focus some of the drawbacks involved. Specifically, it underlines the overlooked importance of iteratively identifying time lags and influential input process variables rather than approaching the two steps separately. In extreme cases, identifying time lags is entirely omitted or is not reported.

Conclusions

Most importantly, the critical review synthesises the first stage of the research and reveals that Gaussian processes are not well investigated in the soft sensor domain, and are not thoroughly assessed in terms of their applicability, appropriateness, and practicality in developing adaptive soft sensors tasked with online predictions. In addition, the research points out the scarcity of publications in this area compared to multivariate statistics and neural networks as soft sensor modelling techniques when very recently Gaussian process-based soft sensor publications began to appear.

Being moderately simple, Gaussian processes are increasingly adopted within the machine-learning community as a powerful modelling technique. The simplicity of implementation, which does not sacrifice any prediction accuracy, is one of the virtues of Gaussian processes and among their strongest properties that distinguish them from other machine-learning modelling techniques. Despite this simplicity in model structure selection, soft sensor developers tend to use one widely used and extensively reported model structure: the squared exponential covariance function and its associated hyper parameters. The structure is utilised as a default model that presumes smoothness in the underlying function that is often not realistic in data retrieved from industrial processes.

Therefore, the research looks into Gaussian process model identification and structure selection. It conducts a thorough comparative study between the predominantly used squared exponential and the Mateŕn class covariance functions. The study utilises different data sets and uses various evaluation criteria that allow a rigorous assessment of the two functions. The study explored the properties of these two functions and assessed their effects on the prediction accuracy of the models, adopting each of the functions. Empirical results reveal that under the condition of clean de-noised and properly pre-processed data, the Mateŕn class outperforms the widely accepted squared exponential. In addition, the Mateŕn class performs at least as accurately as the squared exponential in noisy data. The study concluded that in the case of insufficient phenomenological knowledge that recommends otherwise, the Mateŕn class should be selected instead.

Adaptability is the property that distinguished soft sensors from their first generation and enhanced their central role in a variety of applications. For this, the research investigates building adaptive Gaussian process soft sensors based on instance selection techniques (windowing). The rationale behind this investigation is to suggest a strategy that maximises the informativeness of the block of instances selected, which reflects the predictive capability of the model. This is attained by introducing a new dimension (parameter) that determines the up-

dating criteria adopted to include new data points and exclude irrelevant points to the current process concept.

Three different updating strategies are compared to the conventional strategy. The latter assumes that the time stamp (age) of the data point determines the relevancy of the point to the process concept. Nevertheless, in most of the data sets reviewed, the Mahalanobis distance strategy is found to be more adequate than the rest of the strategies including the prediction variance of the GP model. As a result, the research summarises that the window-updating strategy should be considered along with the commonly optimised parameters: window and step sizes.

Striving towards minimising the window size, and preserving the informativeness obtained in the previous research stage, this research proposes a clustering and filtering technique capable of extracting a data subset that is predictively as informative as the original data. This significantly compresses the data size, considerably reduces computational time, and (most importantly) allows deploying the GP soft sensor in predicting high sampling rate target variables continuously, reliably, and accurately.

An example of such an application is a data-driven inferential control, where a data-driven soft sensor is utilised as a feedback element integrated with an appropriate controller. Harnessing the results in the previous stages of the research, the research explores the applicability of Gaussian process soft sensors in this control domain. It first proposes a GP soft sensor that feeds the predictions back to a PI controller to build a Gaussian process inferential control system (GP-ICS).

The control system is tested and validated on controlling the concentration of a chemical product inside a simulated chemical reactor process. The system is further evaluated by benchmarking it against ANN and KPLS inferential control systems. Empirical results show a superiority in the GP-ICS performance that primarily stems from the predictive capability of the GP soft sensor.

Because data-driven inferential control systems exploit hardware sensor measurements to make predictions and to manipulate process variables accordingly, the reliability of the inferential control system is entirely dependent on these measurements. Based on that, the research further investigates the reliability and robustness of the proposed control system. Specifically, it proposes a mitigation procedure that seeks to preserve control system performance in cases of hardware sensor failures, particularity, failures occurring in sensors measuring the manipulated variable.

This is attained through a subspace identification based method, where KPCA

is utilised to project the input-output data into a new subspace where the model identification and validation is performed. This allows minimisation of the effects of the failure and quick recovery of the functionality of the control system. This is achieved by training and validating the GP soft sensor using the KPCA scores instead of the raw data that contain faulty measurements. The suggested strategy is tested using the chemical reactor process previously introduced and is further compared to a control system that does not employ the proposed method. The results demonstrate the plausibility and the effectiveness of the suggested mitigation procedure where the effects of the failure is considerably minimised and the control system reliability is efficiently preserved.

7.2 Summary Research Contributions

This section describes the main research contributions and the achieved set of goals that answered the research questions raised during the research.

7.2.1 GP Model Structure Selection

Chapter 4 addressed Gaussian process soft sensor model identification and model structure selection. The maximal information coefficient (MIC) was employed to conduct various data analyses that included identifying influential input variables. The MIC criteria, which indicate linear and nonlinear complex relationships among the data being analysed, was not reported in the soft sensor literature previously. Previous analysis reported in the soft sensor literature mainly focused on utilising the Pearson correlation coefficient.

Gaussian process model structure was the central theme in chapter 4, where the research contributed by putting forward the use of the Matérn covariance function instead of the squared exponential. This, as per the research results, ensures that the soft sensor model adopting the Matérn covariance performs at least as accurately as adopting the squared exponential. This contributed towards enhancing soft sensor prediction accuracy particularly in clean data sets.

7.2.2 Adaptability and Computational Time

Chapter 5 investigated a critical attribute in data-driven soft sensor models: adaptivity based on instance selection. The contribution in this area is the introduction of various criteria (other than the time stamp of the data point) that can be

considered for updating the window containing the selected instances (observations). Window updating criteria is a third parameter that the research suggests for optimising soft sensor model identification procedures.

Being a non-parametric model, the training data of the Gaussian process soft sensor is a part of the model that is required during prediction time. The size of the data can be a bottleneck to employing the soft sensor for online predicting applications due to the required memory and the necessary computational time. Chapter 5 contributed in this area by proposing a data filter that retains the informativeness of the data, while simultaneously significantly reducing its size.

7.2.3 GP in Inferential Control Systems

Chapter 6 explored the application of predicting high sample rate target variables and controlling them accordingly. The main contribution of the chapter is the introduction of Gaussian process-based soft sensors in building inferential control systems, in addition to demonstrating their superior performance over ANN and KPLS inferential control systems. This facilitated a more accurate, stable control over the controlled process by using a model that is easier to identify, select, and implement, the Gaussian process soft sensor model.

This chapter further contributed by addressing critical scenarios, such as a failure in the controller-manipulated variable. This is obtained by considering subspace model identification. Such a contribution enables a more robust and reliable operation of the inferential control system than feeding raw data possessing faulty measurements.

7.3 Research Limitations

This research conducted five different studies that suffer from several limitations. The Gaussian process model selection study relied mainly on data sets retrieved from continuous processes where fewer data set were retrieved from batch processes. The latter type of processes as outlined in Section 2.2 are characteristically and dynamically different from continuous processes. Consequently, this may cast a minor doubt on the validity of data representativeness and the generalisation of the conclusions drawn.

More batch process data allow more generalisable results for both types of processes. The issue of data representativeness is due to the difficulty of retrieving batch process data sets for knowledge protection and confidentiality reasons.

Conclusions

The time constraints and lack of the immediate expert knowledge necessitated conducting all experiments based on empirical conclusions without the incorporation of phenomenological views, particularly those needed during the data pre-processing phase (e.g., time lag identification).

7.4 Open Issues

The research results demonstrated the effectiveness of Gaussian processes as soft sensor building techniques and have shown their promising practicality and applicability in industrial processes. However, there are some potential improvements in the soft sensor domain, in general, and in GP-based models in particular.

- **Data reconciliation for soft sensor development:**
  Chemical process data inherently contains random or/and systematic measurement errors, and the objective of data reconciliation is to correct and adjust these measurements to satisfy balance equations. It is argued in [24] that resolving these small measurement errors results in a considerable improvement in overall process performance and economy. Based on this, soft sensors could be developed using not only pre-processed data, but also reconciled data. Such a research direction may enhance the value of data-driven soft sensors.

- **Enhance model transparency and improving prediction accuracy:**
  Model transparency is of substantial importance in industrial processes, as it supports the decision making process. In the Gaussian process-based soft sensor context, such a research direction can be assisted by the automatic selection of composite components of covariance functions, where each component can capture a particular process concept or trend. These separate components convert the GP model into a more transparent one via different local covariance functions.

- **Model adaptivity and computation complexity:**
  The training data and the set of the covariance function hyper parameters are the two elements comprising a Gaussian process model. For this reason, GP model adaptation should be considered at different levels (local and global). While the local level involves adding data points into the model, the global level involves re-optimisation of the hyper parameters once a data point is added. Rather than re-optimising at every added point, a threshold

should be considered to trigger the adaptation. In addition, the threshold should be utilised such that the global adaption is carried out rigorously and based on the data and the process. This stems from the fact that a re-optimisation of a set of hyper parameters during steady-state status is not as important as a re-optimisation during a change in the process dynamics.

- **Exploiting GP Prior:**
  Because GP is a way of defining a prior over functions, expert knowledge can be incorporated into the GP prior. This property of Gaussian processes should be fully utilised in the soft sensor domain, where expert knowledge plays a central role at all levels in all soft sensor design phases. Although this research was conducted in an industrial environment, the unavailability of expert domain knowledge and the associated cost of achieving it were the two main limitations that burdened this research direction.

  This is a potential future direction for additional research, as the lack of an appropriate incorporation procedure is one of the shortcomings identified in the review conducted in this research.

- **Robust GP for outlying observations:**
  Outliers are a typical issue in most datasets retrieved from industrial processes, and they are usually addressed in the data pre-processing phase of soft sensor development. Because of the masking and swamping issues that an outlier detection mechanism may encounter, pre-processed data may still contain outlying observations. In GP modelling, a single outlying observation influences the function modelled by the rest of the observations due to the Gaussian assumption of the observation model (likelihood). This is referred to as outlier rejection and is dealt with by replacing the Gaussian assumption with a Student's-t distribution. However, it is another potential research direction as in the soft sensor domain it may eliminate the need to perform some data pre-processing steps, such as dealing with missing values and outlying observations. The result of this future work could be a soft sensor model with more accurate and more robust predictive capabilities.

Abbreviations

ADALINE Adaptive Linear Letwork. 34

ANN Artificial Neural Networks. 23, 25, 29, 34, 35, 37, 45, 91–93, 98–100, 111

ANN-ICS ANN Inferential Control System. 92, 93, 97, 99, 100

BCM Bayesian Committee Machine. 46

BIC Bayesian Information Criterion. 25

CDU Crude Distillation Unit. 29

CN Condition Number. 59

DCS Distributed Control System. 29

DDPC Data-Driven Direct Predictive Control. 32

DPLS Dynamic Partial Least Square. 33, 34

EKF Extended Kalman Filter. 34

EM Expectation-Maximization. 19

FF Fitness Function. 59

FFNN Feed Forward Neural Network. 34

FGP Full Gaussian Process. 87, 88

FNN False Nearest Neighbour. 24

FPM First Principle Models. 2, 11, 24

Abbreviations

GP Gaussian process. 2–6, 21, 25, 26, 28, 30, 33, 35, 38–49, 51, 52, 59–61, 68, 72, 76–80, 82–84, 87–93, 96, 98, 102, 105, 110, 111

GP-ICS Gaussian Process Inferential Control System. 91, 94, 97–100, 105–111

GP-KPLS-ICS GP-KPLS Inferential Control System. 102, 105–110

GPs Gaussian processes. 3, 4, 25, 28, 35, 38, 41–46, 48, 78–80, 83

I-CS Ideal Control System. 97, 99, 100

IANN Iteratively Adjusted Neural Network. 16

ICA Independent Component Analysis. 30, 37

IVDS Input Variable and Delay Selection. 17, 18

KPCA Kernel Principle Component Analysis. 37, 102, 103, 105, 106, 108, 110, 111

KPLS Kernel Partial Least Square. 91, 92, 94, 98, 99, 111

KPLS-ICS KPLS Inferential Control System. 92, 97, 98, 100

LM Levenberg Marquardt. 34, 93

LS-SVM Least Square-Support Vector Machine. 22

LV Latent Variable. 31

MC Matérn Class. 5, 40, 60, 61, 63–68

MCC Midcourse Correction. 33

MI Mutual Information. 18, 19, 52

MIC Maximal Information Coefficient. 17, 18, 52, 97

MICA Multivariate Independent Component Analysis. 30

MIMO Multiple Input Multiple Output. 38

MINE Maximal Information Nonparametric Exploration. 52, 97

MISO Multiple Input Single Output. 24, 38

MLP Multi-Layer Perceptron. 16, 17, 37, 38

MPC Model Predictive Control. 31–33

MPCA Multivariate Principle Component Analysis. 30

MSE Mean Squared Error. 16, 17, 26, 27, 59, 93, 95, 98, 109

OLS Ordinary Least Squares. 23

OP Overshoot Percentage. 59, 98

PCA Principle Component Analysis. 17, 23, 24, 30, 31, 37, 42, 102

PCR Principle Component Regression. 25, 35, 37

PI Proportional Integral. 33, 34, 91–93, 95, 96, 102, 111

PID Proportional Integral Derivative. 33, 34, 95

PIMS Process Information Management System. 18

PLS Partial Least Square. 17, 23–25, 29–32, 35, 37, 95

R Correlation. 59, 98, 109

RBF Radial Basis Functions. 31

RD-DPLS Reduced Order Partial Least Square. 34

RMS Randomly Missing Spectra. 20

RMSE Relative Mean Squared Error. 26, 27

RMV Randomly Missing Values. 20

RNN Recurrent Neural Networks. 34

RT Rising Time. 59, 98

RVTP Relevance Variance Tracking Precision. 17, 27, 59, 93, 95, 98, 109

SDD Steady-State Detection. 14

SE Squared Exponential. 40, 41, 60, 61, 63, 65–67

Abbreviations

SMV Systematically Missing Value. 20

SPC Statistical Processes Control. 33

SPE Squared Prediction Error. 22

SPGP Pseudo-Inputs Approximation Technique. 83–86

SSE Steady State Error. 59, 98

ST Settling Time. 59, 98

SVM Support Vector Machine. 29, 45

TIC Theils Inequality Coefficient. 27, 59, 83

Notations

$\mathbf{R}$	real numbers.
Θ	vector of Gaussian process model hyperparameters.
$\mathbf{x}^*$	row vector of test input data.
y^*	test instance of the target variable.
L	the marginal likelihood of the Gaussian process model.
n	number of elements in a vector/matrix.
$\mathbf{f}$	a set of random variables.
$\mathbf{x}$	mean function.
$k(\mathbf{x}, \mathbf{x}')$	covariance function of a pair of inputs.
x_t	a data point at time t.
$(\mathbf{x}, \mathbf{x}')$	pair input vectors.
$\mathbf{K}$	covariance matrix.
$\mathcal{GP}$	Gaussian process.
σ_f^2	variance of the modeled function.
d	dimension of a vector.
l	characteristic length scale.
l_d	characteristic length scale of dimension d.
$\Gamma(.)$	Gamma function.
k_v	Modified Bessel function.
σ_{error}^2	variance of prediction error.
$\|K\|$	determinant of the covariance matrix K.
K^{-1}	inverse of the covariance matrix K.
$\mathbf{y}^T$	transpose of the target variable $\mathbf{y}$.
LD	predictive density error.
$\mathbf{I}$	identity matrix.
$\mathbf{k}_\star$	test and training covariance matrix.

Notations

$k_{\star\star}$	test inputs covariance matrix (variance).
σ_n^2	noise variance.
$\bar{x}$	mean of the vector $\mathbf{x}$.
$\sigma^2_{measurement}$	variance of observed target variable.
σ	standard deviation.
D_{sub}	subset of data D.
D_{mah}	mahalanobis distance.
$min_{x \in C_i, x' \in C_j}$	minimum distance between the point x and x' in clusters C_i and C_j.
N	data size.
$u(t)$	control signal.
k_p, k_i	proportional and the integral parameters.
$\alpha^{\mathbf{k}} = (\alpha_1, ...\alpha_n)^T$	vector of eigenvectors.
λ^k	eigenvalue of the kth eigenvectors.
y_i	The ith data instance of the observed target variable.
$\hat{y}_i$	The ith data instance of the predicted target variable.
$\mathbf{x}_i$	Row vector of the ith data instance of the input data.
$\mathbf{X}$	Matrix of input data.
$\mathbf{y}$	Column vector of the target variable.
$\mathbf{D}$	data set.

Bibliography

[1] R. A. F. M. Ronny A.Caytano Teran, "Soft-sensor based on artificial neuronal network for the prediction of physico-chemical variables in suspension polymerization reactions," *Chemical Engineering Transactions*, vol. 24, no. 4, pp. 529–539, 2011.

[2] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven soft sensors in the process industry," *Computers & Chemical Engineering*, vol. 33, no. 4, pp. 795–814, 2009.

[3] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia, *Soft sensors for monitoring and control of industrial processes*. London: Springer, 2007.

[4] H. Vezvaei, S. Ordibeheshti, and M. Ardjmand, "Soft-sensor for estimation of gasoline octane number in platforming processes with adaptive neuro-fuzzy inference systems (anfis)," vol. 5, pp. 163 – 167, 2011.

[5] D. Slišković, R. Grbić, and Ž. Hocenski, "Methods for plant data-based process modeling in soft-sensor development," *Automatika–Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 52, no. 4, pp. 306–318, 2012.

[6] P. Kadlec, R. Grbic, and B. Gabrys, "Review of adaptation mechanisms for data-driven soft sensors," *Computers & chemical engineering*, vol. 35, no. 1, pp. 1–24, 2011.

[7] S. Park and C. Han, "A nonlinear soft sensor based on multivariate smoothing procedure for quality estimation in distillation columns," *Computers & Chemical Engineering*, vol. 24, no. 2-7, pp. 871–877, 2000.

[8] D. Solomatine, L. See, and R. Abrahart, "Data-driven modelling: concepts, approaches and experiences," in *Practical hydroinformatics*. Springer, 2008, pp. 17–30.

[9] K. A. Janes and M. B. Yaffe, "Data-driven modelling of signal-transduction networks," *Nature Reviews Molecular Cell Biology*, vol. 7, no. 11, pp. 820–828, 2006.

[10] L.-Y. Wei, "A ga-weighted anfis model based on multiple stock market volatility causality for taiex forecasting," *Applied Soft Computing*, vol. 13, no. 2, pp. 911–920, 2013.

Bibliography

[11] N. M. Khalfe, "Soft sensor development and optimization of the commercial petrochemical plant integrating support vector regression and genetic algorithm," *Chemical Industry & Chemical Engineering Quarterly*, vol. 15, no. 3, pp. 175–187, 2009.

[12] R. Smith, *Chemical Process Design and Integration*. New York: Wiley, 2005.

[13] J. D. Kelly and J. D. Hedengren, "A steady-state detection (ssd) algorithm to detect non-stationary drifts in processes," *Journal of Process Control*, vol. 23, no. 3, pp. 326–331, 2013.

[14] J. M. Modak, "Haber process for ammonia synthesis," *Resonance*, vol. 7, no. 9, pp. 69–77, 2002.

[15] R. M. Felder and R. W. Rousseau, *Elementry Principles of Chemical Processes*. MA: John Wiley & Sons, 2008.

[16] G. F. Stewart, B. S. Schweigert, J. Hawthorn, and J. C. Bauernfeind, *Carotenoids as Colorants and Vitamin A Precursors: Technological and Nutritional Applications*. New York: Elsevier, 2012.

[17] S. Yin, S. X. Ding, A. H. Abandan Sari, and H. Hao, "Data-driven monitoring for stochastic systems and its application on batch process," *International Journal of Systems Science*, vol. 44, no. 7, pp. 1366–1376, 2013.

[18] M. Champagne and M. Dudzic, "Industrial use of multivariate statistical analysis for process monitoring and control," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 1. IEEE, 2002, pp. 594–599.

[19] H. Li, D. Yu, and J. Braun, "A review of virtual sensing technology and application in building systems," *HVAC&R Research*, vol. 17, no. 5, pp. 619–645, 2011.

[20] X. Wang, "Soft sensor modeling method for freezing point of diesel fuel based on pca and ls-svm," in *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*. IEEE, 2008, pp. 9157–9161.

[21] S. Khatibisepehr, B. Huang, and S. Khare, "Design of inferential sensors in the process industry: A review of bayesian methods," *Journal of Process Control*, vol. 23, no. 10, pp. 1575–1596, 2013.

[22] B. Lin, B. Recke, J. Knudsen, and S. Jørgensen, "A systematic approach for soft sensor development," *Computers & chemical engineering*, vol. 31, no. 5, pp. 419–425, 2007.

[23] P. Bangert, *Optimization for Industrial Problems*. Springer, 2012.

[24] J. A. Romagnoli and M. C. Sanchez, *Data Processing and Reconciliation for Chemical Process Operations*. CA: Academic Press, 1999.

[25] F. Souza, P. Santos, and R. Araújo, "Variable and delay selection using neural networks and mutual information for data-driven soft sensors," in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*. IEEE, 2010, pp. 1–8.

[26] K. Ghosh, M. Ramteke, and R. Srinivasan, "Optimal variable selection for effective statistical process monitoring," *Computers & Chemical Engineering*, vol. 60, pp. 260–276, 2014.

[27] A. Pani and H. Mohanta, "A survey of data treatment techniques for soft sensor design," *Chemical Product and Process Modeling*, vol. 6, no. 1, pp. 1–21, 2011.

[28] G. Gregorcic and G. Lightbody, "Gaussian process approach for modelling of nonlinear systems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 4-5, pp. 522–533, 2009.

[29] F. A. Souza, R. Araújo, T. Matias, and J. Mendes, "A multilayer-perceptron based method for variable selection in soft sensor design," *Journal of Process Control*, vol. 23, no. 10, pp. 1371–1378, 2013.

[30] F. Souza, R. Araújo, S. Soares, and J. Mendes, "Variable selection based on mutual information for soft sensors applications," in *9 th Portuguese Conference on Automatic Control, Coimbra*, 2010.

[31] B. Lu, I. Castillo, L. Chiang, and T. F. Edgar, "Industrial pls model variable selection using moving window variable importance in projection," *Chemometrics and Intelligent Laboratory Systems*, vol. 135, pp. 90–109, 2014.

[32] M. Ma, J. Ko, S. Wang, M. Wu, S. Jang, S. Shieh, and D. Wong, "Development of adaptive soft sensor based on statistical identification of key variables," *Control Engineering Practice*, vol. 17, no. 9, pp. 1026–1034, 2009.

[33] D. Reshef, Y. Reshef, H. Finucane, S. Grossman, G. McVean, P. Turnbaugh, E. Lander, M. Mitzenmacher, and P. Sabeti, "Detecting novel associations in large data sets," *science*, vol. 334, no. 6062, pp. 1518–1524, 2011.

[34] Y. Chen, P. Du, and Y. Wang, "Variable selection in linear models," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 6, no. 1, pp. 1–9, 2014.

[35] R. May, G. Dandy, and H. Maier, "Review of input variable selection methods for artificial neural networks," *Artificial neural networks?methodological advances and biomedical applications*, pp. 19–44, 2011.

[36] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, no. 1, pp. 175–186, 2014.

[37] F. Souza and R. Araujo, "Variable and time-lag selection using empirical data," in *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on.* IEEE, 2011, pp. 1–8.

[38] L. Xie, H. Yang, and B. Huang, "Fir model identification of multirate processes with random delays using em algorithm," *AIChE Journal*, vol. 59, no. 11, pp. 4124–4132, 2013.

Bibliography

[39] M. Delgado, E. Nagai, and L. de Arruda, "A neuro-coevolutionary genetic fuzzy system to design soft sensors," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 5, pp. 481–495, 2009.

[40] X. Pan, G. Metzner, N. Selby, C. Visser, T. Letwaba, and K. Coleman, "Development of weightometer soft sensor," *Application of Computers and Operations Research in the Minerals Industries (APCOM)*, vol. 104, pp. 261–166, 2003.

[41] J. G. Webster, *The Measurement, Instrumentation, and Sensors: Handbook*. NY: CRC Press, January 29, 2014.

[42] G. Tomasi and R. Bro, "Parafac and missing values," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 2, pp. 163 – 180, 2005.

[43] E. Eirola, A. Lendasse, V. Vandewalle, and C. Biernacki, "Mixture of gaussians for distance estimation with missing data," *Neurocomputing*, vol. 131, pp. 32–42, 2014.

[44] A. K. Pani, V. K. Vadlamudi, and H. K. Mohanta, "Development and comparison of neural network based soft sensors for online estimation of cement clinker quality," *ISA transactions*, vol. 52, no. 1, pp. 19–29, 2013.

[45] M. N. Norazian, A. B. A. Mohd Mustafa, Y. Ahmad Shukri, R. Nor Azam *et al.*, "Comparison of linear interpolation method and mean method to replace the missing values in environmental data set." Universiti Malaysia Perlis (UniMAP), 2007.

[46] I. Žliobaitė and J. Hollmén, "Fault tolerant regression for sensor data," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 449–464.

[47] S. Cateni, V. Colla, and M. Vannucci, "Outlier detection methods for industrial applications," *Advances in Robotics, Automation and Control*, pp. 265–282, 2008.

[48] K. Warne, G. Prasad, S. Rezvani, and L. Maguire, "Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion," *Engineering applications of artificial intelligence*, vol. 17, no. 8, pp. 871–885, 2004.

[49] M. Gupta, J. Gao, C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 9, pp. 2250–2267, 2014.

[50] H. J. Galicia, Q. P. He, and J. Wang, "Adaptive outlier detection and classification for online soft sensor update," in *International Symposium on Advanced Control of Chemical Processes ADCHEM*, 2012, pp. 402–407.

[51] B. Berger and F. Rauscher, "Robust gaussian process modelling for engine calibration," in *Proceedings of the 7th Vienna International Conference on Mathematical Modelling (MATHMOD)*, 2012, pp. 159–164.

[52] P. Jylänki, J. Vanhatalo, and A. Vehtari, "Robust gaussian process regression with a student-t likelihood," *The Journal of Machine Learning Research*, vol. 12, pp. 3227–3257, 2011.

[53] R. Lehmann, "3 σ-rule for outlier detection from the viewpoint of geodetic adjustment," *Journal of Surveying Engineering*, vol. 139, no. 4, pp. 157–165, 2013.

[54] Y. Zhao, B. Lehman, R. Ball, J. Mosesian, and J.-F. de Palma, "Outlier detection rules for fault detection in solar photovoltaic arrays," in *Applied Power Electronics Conference and Exposition (APEC), 2013 Twenty-Eighth Annual IEEE*. IEEE, 2013, pp. 2913–2920.

[55] W. Xiong, W. Zhang, D. Liu, and B. Xu, "Fuzzy pruning based ls-svm modeling development for a fermentation process," in *Abstract and Applied Analysis*, vol. 2014. Hindawi Publishing Corporation, 2014, pp. 1–7.

[56] C. F. Dormann, J. Elith, S. Bacher, C. Buchmann, G. Carl, G. Carré, J. R. G. Marquéz, B. Gruber, B. Lafourcade, P. J. Leitão *et al.*, "Collinearity: a review of methods to deal with it and a simulation study evaluating their performance," *Ecography*, vol. 36, no. 1, pp. 027–046, 2013.

[57] L. Anselin and B. Baltagi, *A companion to Theoretical Econometrics*. Cornwall, UK: Blackwell Oxford, 2001.

[58] A. H. El-Shaarawi and W. W. Piegorsch, *Encyclopedia of Environmetrics*. New Jersey: John Wiley & Sons, 2001.

[59] H. K. Alaei and K. Salahshoor, "The design of new soft sensors based on pca and a neural network for parameters estimation of a petroleum reservoir," *Petroleum Science and Technology*, vol. 30, no. 22, pp. 2294–2305, 2012.

[60] T. Li, J. Yi, Y. Su, W. Hu, and C. Yu, "Reduction of secondary variables on soft-sensing model based on similarity in feature subspace," in *Intelligent Control and Automation (WCICA), 2011 9th World Congress on*. IEEE, 2011, pp. 162–165.

[61] F. A. Souza and R. Araujo, "Mixture of partial least squares experts and application in prediction settings with multiple operating modes," *Chemometrics and Intelligent Laboratory Systems*, vol. 130, no. 0, pp. 192 – 202, 2014.

[62] I. T. Cameron and K. Hangos, *Process Modelling and Model Analysis*. London: Academic Press, 2001, vol. 4.

[63] M. Matzopoulos, *Dynamic Process Modeling: Combining Models and Experimental Data to Solve Industrial Problems*. Wiley-VCH Verlag GmbH & Co. KGaA, 2010, vol. 7, pp. 1–33.

[64] C. Shang, F. Yang, D. Huang, and W. Lyu, "Data-driven soft sensor development based on deep learning technique," *Journal of Process Control*, vol. 24, no. 3, pp. 223–233, 2014.

[65] K.-L. Du and M. Swamy, "Introduction to fuzzy sets and logic," in *Neural Networks and Statistical Learning*. Springer, 2014, pp. 645–676.

[66] Z.-h. XIONG, J.-c. ZHANG, and H.-h. SHAO, "Gp-based soft sensor modeling," *Acta Simulata Systematica Sinica*, vol. 4, p. 006, 2005.

[67] J. Q. Shi and T. Choi, *Gaussian Process Regression Analysis for Functional Data*. Florida: CRC Press, 2011.

[68] A. G. Wilson, "Covariance kernels for fast automatic pattern discovery and extrapolation with gaussian processes," Ph.D. dissertation, PhD thesis, University of Cambridge, 2014. URL http://www. cs. cmu. edu/andrewgw/andrewgwthesis. pdf, 2014.

[69] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press,, 2006.

[70] J. Kocijan, "Gaussian process models for systems identification," in *Proc. 9th Int. PhD Workshop on Sys. and Cont*, 2008, pp. 8–15. [Online]. Available: http://dsc.ijs.si/jus.kocijan/GPdyn/

[71] B. Gabrys, "Learning hybrid neuro-fuzzy classifier models from data: To combine or not to combine?" *Fuzzy Sets and Systems*, vol. 147, no. 1, pp. 39–56, 2004.

[72] X. Li, H. Su, and J. Chu, "Multiple model soft sensor based on affinity propagation, gaussian process and bayesian committee machine," *Chinese Journal of Chemical Engineering*, vol. 17, no. 1, pp. 95–99, 2009.

[73] D. Murray smith, "Methods for the external validation of continuous system simulation models: a review," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 4, no. 1, pp. 5–31, 1998.

[74] K. Azman and J. Kocijan, "Application of gaussian processes for black-box modelling of biosystems," *ISA transactions*, vol. 46, no. 4, pp. 443–457, 2007.

[75] C. K. Amos, "Soft sensors win hard jobs," *Chemical Processing*, vol. 68, no. 12, pp. 31–33, 2005.

[76] B. O. Teixeira, W. S. Castro, A. F. Teixeira, and L. A. Aguirre, "Data-driven soft sensor of downhole pressure for a gas-lift oil well," *Control Engineering Practice*, vol. 22, pp. 34–43, 2014.

[77] L. L. T. Chan, Y. Liu, and J. Chen, "Nonlinear system identification with selective recursive gaussian process models," *Industrial & Engineering Chemistry Research*, vol. 52, no. 51, pp. 18 276–18 286, 2013.

[78] Z. Ge, B. Huang, and Z. Song, "Mixture semisupervised principal component regression model and soft sensor application," *AIChE Journal*, vol. 60, no. 2, pp. 533–545, 2014.

[79] Z. Ge, Z. Song, and F. Gao, "Review of recent research on data-based process monitoring," *Industrial & Engineering Chemistry Research*, vol. 52, no. 10, pp. 3543–3562, 2013.

[80] C. K. Yoo, J.-M. Lee, P. A. Vanrolleghem, and I.-B. Lee, "On-line monitoring of batch processes using multiway independent component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 71, no. 2, pp. 151 – 163, 2004.

[81] R. Isermann, "Model-based fault-detection and diagnosis–status and applications," *Annual Reviews in control*, vol. 29, no. 1, pp. 71–85, 2005.

[82] R. Grbić, D. Slišković, and P. Kadlec, "Adaptive soft sensor for online prediction and process monitoring based on a mixture of gaussian process models," *Computers & chemical engineering*, vol. 58, pp. 84–97, 2013.

[83] S. A. Iliyas, M. Elshafei, M. A. Habib, and A. A. Adeniran, "Rbf neural network inferential sensor for process emission monitoring," *Control Engineering Practice*, vol. 21, no. 7, pp. 962–970, 2013.

[84] F. J. Doyle III, "Nonlinear inferential control for process applications," *Journal of Process Control*, vol. 8, no. 5, pp. 339–353, 1998.

[85] E. F. Camacho and C. B. Alba, *Model Predictive Control*, 2nd ed. London: Springer, 2013.

[86] R. Kwadzogah, M. Zhou, and S. Li, "Model predictive control for hvac systems 2014; a review," in *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*, 2013, pp. 442–447.

[87] D. Laurí, J. Rossiter, J. Sanchis, and M. Martínez, "Data-driven latent-variable model-based predictive control for continuous processes," *Journal of Process Control*, vol. 20, no. 10, pp. 1207–1219, 2010.

[88] M.-J. Bruwer, M. Golshan, and J. MacGregor, "System and method for the model predictive control of batch processes using latent variable dynamic models," May 13 2009, uS Patent App. 12/465,352.

[89] X. Wu, J. Shen, Y. Li, and K. Y. Lee, "Data-driven modeling and predictive control for boiler–turbine unit," *Energy Conversion, IEEE Transactions on*, vol. 28, no. 3, pp. 470–481, 2013.

[90] J. Kocijan, "Control algorithms based on gaussian process models: A state-of-the-art survey," *Special International Conference on Complex Systems: Synergy of Control, Communications and Computing*, vol. 16, pp. 273–280, 2011.

[91] D. Laurí, B. Lennox, and J. Camacho, "Model predictive control for batch processes: Ensuring validity of predictions," *Journal of Process Control*, vol. 24, no. 1, pp. 239–249, 2014.

[92] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and B. Likar, *Predictive Control with Gaussian Process Models*, MA, USA, 2003, vol. 1.

[93] S. Aumi, B. Corbett, T. Clarke-Pringle, and P. Mhaskar, "Data-driven model predictive quality control of batch processes," *AIChE Journal*, vol. 59, no. 8, pp. 2852–2861, 2013.

[94] K. J. Åström, "Control system design lecture notes," *Department of Mechanical and Environmental Engineering University of California Santa Barbara*, 2002. [Online]. Available: http://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom.html

[95] H. J. Galicia, Q. P. He, and J. Wang, "A reduced order soft sensor approach and its application to a continuous digester," *Journal of Process Control*, vol. 21, no. 4, pp. 489–500, 2011.

[96] H. J. Galicia, Q. Peter He, and J. Wang, "Comparison of the performance of a reduced-order dynamic pls soft sensor with different updating schemes for digester control," *Control Engineering Practice*, vol. 20, no. 8, pp. 747–760, 2012.

[97] S. Vijaya Raghavan, T. Radhakrishnan, and K. Srinivasan, "Soft sensor based composition estimation and controller design for an ideal reactive distillation column," *ISA transactions*, vol. 50, no. 1, pp. 61–70, 2011.

[98] A. Rani, V. Singh, and J. Gupta, "Development of soft sensor for neural network based control of distillation column," *ISA transactions*, vol. 52, no. 3, pp. 438–449, 2013.

[99] S. Geethalakshmi and N. Pappa, "Artificial neural network based soft sensor for fermentation of recombinant pichia pastoris," in *Advances in Computer Engineering (ACE), 2010 International Conference on.* IEEE, 2010, pp. 148–152.

[100] J. Bernardo, J. Berger, A. Dawid, A. Smith *et al.*, "Regression and classification using gaussian process priors," in *Bayesian Statistics 6: Proceedings of the sixth Valencia international meeting*, vol. 6, 1998, p. 475.

[101] R. Neal, "Bayesian learning for neural networks," Ph.D. dissertation, University of Toronoto, Toronot, 1996.

[102] M. K. Simon, *Probability Distributions Involving Gaussian Random Variables: A Handbook for Engineers and Scientists*, NY, 2007.

[103] E. L. Snelson, "Flexible and efficient gaussian process models for machine learning," Ph.D. dissertation, Citeseer, 2007.

[104] M. Ebden, "Gaussian processes for regression: A quick introduction," *Available on Internet httpwww robots ox ac uk*, 2008.

[105] A. Girard, "Approximate methods for propagation of uncertainty with gaussian process models," Ph.D. dissertation, Mémoire de these, Glasgow University, Glasgow, Glasgow, UK, 2004.

[106] K. Ghiasi-Shirazi, R. Safabakhsh, and M. Shamsi, "Learning translation invariant kernels for classification," *The Journal of Machine Learning Research*, vol. 11, pp. 1353–1390, 2010.

[107] M. Genton, "Classes of kernels for machine learning: a statistics perspective," *The Journal of Machine Learning Research*, vol. 2, pp. 299–312, 2002.

[108] F. Yongfeng, "A dynamic soft-sensor modeling method based on fc-gp for 4-cba content," in *Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE*. IEEE, 2011, pp. 1–4.

[109] M. Osborne and S. Roberts, "Gaussian processes for prediction," Technical Report PARG-07-01. Available at www. robots. ox. ac. uk/parg/publications. html, University of Oxford, Tech. Rep., 2007.

[110] M. E. Harmon and S. S. Harmon, "Reinforcement learning: a tutorial," *WL/AAFC, WPAFB Ohio*, vol. 45433, 1996.

[111] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998.

[112] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," *CoRR*, vol. abs/1309.6835, 2013.

[113] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," pp. 1257–1264, 2006.

[114] Z.-h. Xiong, G.-h. Huang, and H.-h. Shao, "On-line estimation of concentration parameters in fermentation processes," *Journal of Zhejiang University. Science. B*, vol. 6, no. 6, p. 530, 2005.

[115] K. Tiwari, K. Mehta, N. Jain, R. Tiwari, G. Kanda, and N. New Providence, "Selecting the appropriate outlier treatment for common industry applications," in *NESUG Conference Proceedings on Statistics and Data Analysis, Baltimore, Maryland, USA*, 2007, pp. 1–5.

[116] Z. Ge, T. Chen, and Z. Song, "Quality prediction for polypropylene production process based on clgpr model," *Control Engineering Practice*, vol. 19, no. 5, pp. 423–432, 2011.

[117] T. Chen and J. Ren, "Bagging for gaussian process regression," *Neurocomputing*, vol. 72, no. 7, pp. 1605–1610, 2009.

[118] R. Grbic, D. Slikovic, and P. Kadlec, "Adaptive soft sensor for online prediction based on moving window gaussian process regression," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, vol. 2. IEEE, 2012, pp. 428–433.

[119] W. Ni, S. K. Tan, W. J. Ng, and S. D. Brown, "Moving-window gpr for nonlinear dynamic system modeling with dual updating and dual preprocessing," *Industrial & Engineering Chemistry Research*, vol. 51, no. 18, pp. 6416–6428, 2012.

[120] D. Petelin, J. Kocijan, and A. Grancharova, "On-line gaussian process model for the prediction of the ozone concentration in the air," *Comptes Rendus de l Academie Bulgare des Sciences*, vol. 64, no. 1, pp. 117–124, 2011.

[121] P. Kadlec, "On robust and adaptive soft sensors," Ph.D. dissertation, Bournemouth University, Bournemouth, UK, 2009.

Bibliography

[122] R. Caponetto, G. Dongola, A. Gallo, and M. G. Xibilia, "Fpga based soft sensor for the estimation of the kerosene freezing point," in *Industrial Embedded Systems, 2009. SIES'09. IEEE International Symposium on.* IEEE, 2009, pp. 228–236.

[123] D.-H. Lee and C. Hsu, "Polymerization of butadiene in toluene with nickel (ii) stearate–diethyl aluminum chloride catalyst. ii. kinetic study," *Journal of Applied Polymer Science*, vol. 26, no. 2, pp. 653–666, 1981.

[124] C. J. Paciorek and M. J. Schervish, "Nonstationary covariance functions for gaussian process regression," *Advances in neural information processing systems*, vol. 16, pp. 273–280, 2004.

[125]

[126] L. El Ghaoui, "Inversion error, condition number, and approximate inverses of uncertain matrices," *Linear algebra and its applications*, vol. 343, pp. 171–193, 2002.

[127] M. T. Tham, G. A. Montague, A. Julian Morris, and P. A. Lant, "Soft-sensors for process estimation and inferential control," *Journal of Process Control*, vol. 1, no. 1, pp. 3–14, 1991.

[128] D. K. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, "Structure discovery in nonparametric regression through compositional kernel search," in *ICML (3)'13*, 2013, pp. 1166–1174.

[129] C. Doom, "Get smart: How intelligent technology will enhance our world," Computer Sciences Corporation, Brussels, Tech. Rep., 2001.

[130] M. Chen, S. Khare, and B. Huang, "A unified recursive just-in-time approach with industrial near infrared spectroscopy application," *Chemometrics and Intelligent Laboratory Systems*, vol. 135, pp. 133–140, 2014.

[131] G. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, 1999.

[132] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[133] A. Melkumyan and F. Ramos, "A sparse covariance function for exact gaussian process inference in large datasets," in *IJCAI*, vol. 9, 2009, pp. 1936–1942.

[134] P. Bangert, *Optimization for industrial problems.* Bremen, Germany: Springer, 2012.

[135] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[136] E. Snelson and Z. Ghahramani, "Variable noise and dimensionality reduction for sparse gaussian processes," *arXiv preprint arXiv:1206.6873*, 2012.

[137] M. Osborne, "Bayesian gaussian processes for sequential prediction, optimisation and quadrature," Ph.D. dissertation, University of Oxford, 2010.

[138] A. Bahar and C. Özgen, "State estimation and inferential control for a reactive batch distillation column," *Engineering applications of artificial intelligence*, vol. 23, no. 2, pp. 262–270, 2010.

[139] L. Wang, C. Chessari, and E. Karpiel, "Inferential control of product quality attributes application to food cooking extrusion process," *Journal of Process Control*, vol. 11, no. 6, pp. 621–636, 2001.

[140] K. J. Aström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

[141] P. Kadlec and B. Gabrys, "Adaptive local learning soft sensor for inferential control support," in *Computational Intelligence for Modelling Control & Automation, 2008 International Conference on*. IEEE, 2008, pp. 243–248.

[142] S. Kim, M. Kano, S. Hasebe, A. Takinami, and T. Seki, "Long-term industrial applications of inferential control based on just-in-time soft-sensors: Economical impact and challenges," *Industrial & Engineering Chemistry Research*, vol. 52, no. 35, pp. 12 346–12 356, 2013.

[143] E. Corchado, M. Wozniak, A. Abraham, A. C. P. L. F. de Carvalho, and V. Snaśel, "Recent trends in intelligent data analysis." *Neurocomputing*, vol. 126, pp. 1–2, 2014. [Online]. Available: http://dblp.uni-trier.de/db/journals/ijon/ijon126.html#CorchadoWACS14

[144] S. M. Dhokai, V. B. Vala, and V. H. Shah, "Pca and kpca algorithms for face recognition–a survey," *International Journal of Advance Engineering and Research Development*, vol. 1, no. 4, 2014.

[145] H. Q. Minh, P. Niyogi, and Y. Yao, "Mercers theorem, feature maps, and smoothing," in *Learning theory*. Springer, 2006, pp. 154–168.

[146] A. J. Izenman, *Modern multivariate statistical techniques: regression, classification, and manifold learning*. Springer, 2009.

[147] J. Wang and S. J. Qin, "A new subspace identification approach based on principal component analysis," *Journal of Process Control*, vol. 12, no. 8, pp. 841–855, 2002.

[148] M. Wu and J. D. Farquhar, "A subspace kernel for nonlinear feature extraction." in *IJCAI*, 2007, pp. 1125–1130.

[149] S. J. Qin, "An overview of subspace identification," *Computers & chemical engineering*, vol. 30, no. 10, pp. 1502–1513, 2006.