University of Alberta

A BAYESIAN APPROACH FOR SOFT SENSOR DEVELOPMENT

by

©

Shima Khtibisepehr

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

in

Process Control

Department of Chemical and Materials Engineering

Edmonton, Alberta
Fall 2007

Your file Votre référence
ISBN: 978-0-494-33275-7
Our file Notre référence
ISBN: 978-0-494-33275-7

Abstract

On-line measurements of key process variables are restricted by the availability of measurement techniques or by the reliability and high installation cost of instruments. Even if appropriate instrumentation exists, some measurable variables are only available through off-line analysis in a laboratory. Soft sensors are inferential models that provide continuous on-line estimates of quality variables from readily available process measurements. This thesis deals with practical issues associated with the development of gray box models for soft sensor applications. Development and implementation of soft sensors entail many challenges due to the quality of collected data. Some of these problems, including measurement noises, missing measurements, and outliers, are addressed in this thesis. Different classical treatments are discussed and evaluated.

Bayesian models are a compact way to represent probability distributions. They can also be extended to dynamic Bayesian models to represent dynamic processes. A Bayesian network is flexible in model structure, easy to expand, capable of dealing with irregularly sampled data or missing data, and powerful in many applications. In this thesis, soft sensor problem is formulated in a Bayesian framework in order to overcome the typical limitations of conventional methods to deal with the existing challenges. In addition, the efficiency and effectiveness of the Bayesian approach is demonstrated on numerical simulations, a pilot-scale experiment, and an industrial case study.

Acknowledgments

I would like to express my gratitude to my supervisor, Dr. Biao Huang, whose expertise, understanding, and patience, added considerably to my graduate experience. Our weekly meetings have proved to be one of my best learning experiences and have been an invaluable asset throughout my research. It was a great pleasure for me to work under his supervision.

Dr. Sirish Shah deserves special appreciation as an outstanding instructor throughout my time at the University of Alberta. Special Thanks also go to Dr. Amos Ben-Zvi and Dr. Jong Min Lee. Working closely with them as a TA was one of the most rewarding and enjoyable experiences of my academic career. I would also like to thank the faculty, staff and colleagues in the Department of Chemical and Materials Engineering with whom I had the pleasure of working in a rich and active environment.

This work has been financially supported by the Syncrude Canada Ltd. On the Syncrue side, I would first like to tank Aris Espejo for sponsoring NSERC projects, his leadership in the University of Alberta and Syncrude collaborative work. Many thanks also go to Dr. Bo Li for his motivation and constructive guidance.

I would like to convey my gratitude to my family. I am deeply and forever indebted to my parents for their love and support. Thanks for encouraging me to be an independent thinker, and having confidence in my abilities to go after new things that inspired me. Thanks for letting me pursue my dreams for so long and so far away from home. Special thanks also go to Shiva whose friendship is an invaluable treasure to me. Words cannot express my deepest gratitude to Kasra for his love, patience and encouragement. Thanks for giving me new dreams to pursue.

Lastly, I thank my God whose blessing has made it all possible in the first place.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Models that are entirely based on physical principles including mass, components, and energy balances are called first principles models [7,8,9]. Since complete knowledge of process behavior is required to build first principles models, they can often be expensive and time-consuming. Black box modeling is proposed for situations in which physical understanding of the process is absent [14,18]. Black box models do not use any information about the process and their application is limited. In chemical engineering applications, gray box modeling is the most commonly used approach; it combines both of the above two approaches in developing an industrial model. Gray box models [2,15,17], also called hybrid models, are a useful alternative in situations where insight into a system is required but a complete first principles model is difficult to construct.

One of the key applications of gray box modeling is in soft sensor development [1,5]. Soft sensors are mathematical models that provide on-line estimates of difficult-to-measure variables from readily available variables. These sensors are often needed in chemical processes, because some important process variables or modes are difficult or expensive to measure on-line. An overview of the soft sensors applied in seven different fields of process engineering is given in [16]. These fields are: 1. Diagnosis of process operations, 2. Monitoring and analysis of process trends, 3. Intelligent control, 4. Heuristics and logic in planning and scheduling of process operations, 5. Modeling languages, simulation and reasoning, 6. Intelligence in scientific computing, and 7. Knowledge based engineering design.

With respect to the application, development of a soft sensor requires following four distinct steps:

1. *Problem definition*: Based on the requirements, the goal of the soft sensor model is precisely formulated.
2. *Design*: Having formulated the modeling problem, the structure of the model is determined and the model parameters are identified.
3. *Evaluation*: The structure and parameters of the model need to be validated with the real data. If the model error is outside the acceptable boundaries, we need to revisit design phase and improve the developed model.
4. *Application*: When the model performance is acceptable, the soft sensor will be put in operation.

In gray box modeling, the data quality is crucial for the development of reliable models; however, real-world data are commonly contaminated with measurement noises, outliers, and missing measurements. As a result, the satisfactory performance of soft sensors can be achieved only if we are able to cope with these issues.

As we shall see in Chapter 3, in classical least squares methods it is assumed that the independent process variables have been exactly observed and that the only dependent variable is noisy. Consequently, these approaches yield biased parameter estimates for most industrial applications. The total least squares (TLS) [12], also referred to as classical Errors-in-Variables (EIV), is an alternative approach to compensate for the measurement noises. TLS differs distinctly from the classical least squares methods, for the reason that measurement errors in both dependent and independent variables are taken into account. The key problem with this approach, however, is that the underlying assumption in TLS is not necessarily true for real world data and, again, the resulting parameter estimates may also be biased [10].

Another aspect to be considered in the soft sensor development is related to outliers and missing measurements. There are a variety of methods for handling incomplete and inconsistent data [4,6], but many of them are problem specific and problematic. In general, the existing treatments of missing values can be classified into two main categories: 1. deletion, and 2. imputation methods. The most common approach is to simply exclude the cases with missing values from the analysis. However, if we do not want to lose data and, perhaps, information, we may try to predict missing items. There are many imputation methods available, such as mean substitution, LOCF method, regression imputation, NIPALS algorithm, and EM, each with their own advantages and disadvantage; they will be further discussed in Chapter 4.

Outliers are observations far from most others in a set of data. They are almost the same kind of problem as missing values, but could be worse if not detected and deleted. Box plot is a helpful graphical tool that provides criteria for detection of outlying observations. Once outliers are detected, one of the missing values treatment approaches is applied. Robust regression [6] is an alternative approach to handle outliers. In this approach, a weight is assigned to each observation so that outliers are given reduced weight. Many methods have been developed for robust regression; M-estimation [11] is the most commonly used one.

Representation of multimodal processes is another issue that may arise in the development and implementation of soft sensors [3]. As one might expect, some systems have multiple modes or regimes of behaviors, hence multiple models are used to cover all operating conditions. To produce valid results, all that remains is to choose the model that best fits the current observations. In addition, to perform many other tasks (e.g. fault diagnosis), we need to determine which operation mode all components[1] in the system are in. To do so, we need to develop dynamic models, which are capable of describing the transition of variables with respect to time. Static models are often used in design and optimization, while dynamic models are widely applied in process control.

This thesis employs a Bayesian approach in addressing the challenges associated with soft sensor development and implementation. A Bayesian network [13] is flexible in model structure, easy to expand, capable of dealing with irregularly sampled data or missing data, and powerful in many applications.

Before explaining the contributions and scope of this thesis in detail, let us take a closer look at the Bayesian concept.

1.2 Introduction to the Bayesian method

1.2.1 Notation

A Bayesian network is a graphical model for representing probabilistic relationships between a set of random variables in a system. We denote random variables by upper case letters (e.g. X, Y), and signify the actual value of these variables by the corresponding lower case letters

[1] These components can for example be actuators, pumps, and any other physical objects.

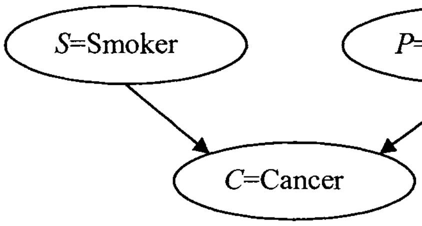

Figure 1.1. A simple Bayesian network example

(e.g. x, y). Sets of variables are symbolized by bold-face upper case letters (e.g. $\boldsymbol{X}, \boldsymbol{Y}$) and their values are represented by the corresponding lower case letters (e.g. ***x***, ***y***). We use the notations $P(X)$ and $P(X = x)$ to denote the probability distribution for X and the probability that X takes the value x, respectively. Commonly, $P(X \mid Y)$ refers to the conditional distribution of X given Y and $P(X \mid Y = y)$ refers to the conditional distribution of X given Y = y. Finally, $E(X)$ stands for the expectation of X when the context has made the corresponding distribution clear.

1.2.2 Representation

Bayesian networks are directed graphical models, in which nodes represent random variables and the arcs represent conditional dependence/independence assumptions. An example of a Bayesian network is presented in Figure 1.1. This network shows the effects of smoking and pollution on lung cancer.

Let $\boldsymbol{X} = \{X_1, ..., X_N\}$ be a set of random variables, where we topologically order the nodes (parents before children) as $1, ..., N$ in the network. Each node, X_i, directly depends on its parents $Par(X_i)$, and a set of *Conditional Probability Distributions* (CPD*s*) parameterizes this dependency. In the case of discrete variables, this distribution is often stored as the *Conditional Probability Table* (CPT), i.e. a table where the probabilities are given for all the combinations of values that the variable and its parents can take. The CPTs of the Bayesian network shown in Figure 1.1 are given in Table 1.1 and Table 1.2.

According to an alternative definition of independency for Bayesian networks known as the *Directed Local Markov Property*, $Par(X_i)$ provides a set of parents of X_i that render X_i independent of all its other parents. After giving these specifications, the joint probability distribution can be calculated as follows:

$$\begin{aligned}P(X_1,...,X_N) &= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1,X_2)...P(X_N \mid X_1,...,X_{N-1}) \\ &= \prod_{i=1}^{N} P(X_i \mid X_{1:i-1}) \\ &= \prod_{i=1}^{N} P(X_i \mid Par(X_i)) \qquad \textbf{(1.1)}\end{aligned}$$

The first line is defined via the *Chain Rule* of probability and rewritten in the form of the second line. Finally, the third line follows because node X_i is independent of all its ancestors, $X_{1:i-1}$, given its parents. The last equation holds only if the network is arranged according to Pearl's algorithm, which is discussed next.

Table 1.1. The CPTs for the parent nodes in the Bayesian network presented in Figure 1.1

S	*P*(*S*)
True	0.3
False	0.7

P	*P*(*P*)
True	0.6
False	0.4

Table 1.2. The CPT for the node *C* in the Bayesian network presented in Figure 1.1

S	*P*	*C*	$P(C \mid S,P)$
True	True	True	0.8
True	True	False	0.2
True	False	True	0.7
True	False	False	0.3
False	True	True	0.4
False	True	False	0.6
False	False	True	0.1
False	False	False	0.9

1.2.3 Network construction

We are now ready to formally define how to construct a Bayesian model. The condition that $Par(X_i) \subseteq \{X_1,...,X_{i-1}\}$ allows us to construct a network from a given ordering of nodes using *Pearl's Network Construction Algorithm* [13]:

Algorithm 1.1.

1. Choose the set of relevant variables, $\boldsymbol{X}$, that describe the domain.
2. Choose an ordering for the variables, $\boldsymbol{X} = \{X_1,...,X_N\}$.
3. While there are variables left:
    a. add the next variable, X_i, to the network;
    b. add arcs to the X_i node from some minimal sets of nodes already in the network, $Par(X_i)$, such that the following conditional independence property is satisfied: $P(X_i \mid X_1,...,X_{i-1}) = P(X_i \mid Par(X_i))$, where $X_1,...,X_{i-1}$ are all the variables preceding X_i, including $Par(X_i)$;
    c. define the CPT for X_i.

Pearl's network-construction algorithm satisfies the Markov property and expresses conditional dependencies in probability distributions.

1.2.4 Bayes' theorem

Bayesian philosophy originated from an interpretation of Bayes' theorem, which adjusts the probability of a hypothesis, *h*, conditioned upon some evidence, *e*, in the light of new information:

$$P(h \mid e) = \frac{P(e \mid h)P(h)}{P(e)} \tag{1.2}$$

Each term in Bayes' Theorem has a conventional name as given below:

- $P(h)$ is the *prior probability*, because it does not take into account any information about evidence, *e*.

- $P(h|e)$ is the *posterior probability*, because it is derived from or depends upon the specified value of evidence, e.
- $P(e|h)$ is the *conditional probability* of seeing the evidence (e) given that the hypothesis (h) is true; as a function of h given e, it is also called the *likelihood function.*
- $P(e)$ is the prior or marginal probability of e, and acts as a *normalizing constant.* It can be calculated as $P(e) = \sum_i P(e \mid h_i)P(h_i)$.

In addition, the ratio $\frac{P(e \mid h)}{P(e)}$ is sometimes called the standardized likelihood.

As a simple example, Bayes' theorem helps to show that, for rare conditions, the majority of positive results may be false positives, even if the test for that condition is (otherwise) reasonably accurate. For instance, a drug test is performed on athletes in Olympic Games. Suppose that the test has a reasonable success rate:

- if a tested athlete has taken the drug, the test accurately reports this a "positive" 80% of the time, and
- if a tested athlete has not taken the drug, the test accurately reports this a "negative" 99% of the time.

According to the database, however, only 0.1% of the athletes have taken drugs (*i.e.* with probability 0.001). Now, these probabilities are sufficient to define joint probability distribution. Let h be the event that the athlete has taken the drug, and e be the event that the test returns a positive result. Using Bayes' theorem, we are able to calculate the probability that he has taken the drug:

$$\begin{aligned} P(D = Yes \mid T = Pos) &= \frac{P(T = Pos \mid D = Yes)P(D = Yes)}{P(T = Pos \mid D = Y)P(D = Y) + P(T = Pos \mid D = N)P(D = N)} \\ &= \frac{0.8 \times 0.001}{0.8 \times 0.001 + 0.01 \times 0.999} \\ &= 0.074 \end{aligned}$$

Hence the probability that a positive result is a false positive approximately (1-0.074) = 0.926.

The application of Bayes' theorem to obtaining posterior distributions will be investigated further when we discuss Bayesian learning and inference.

1.3 Contributions

The main contributions of this thesis are listed below.

1. It applies path analysis for optimal selection of soft sensor variables.
2. It develops a hybrid soft sensor for on-line estimation of a quality variable in the froth treatment process.
3. It develops an EM-based Bayesian framework that is robust to data contaminated with measurement noises.
4. It proposes using the Bayesian method as a novel approach for soft sensor development.
5. It formulates a Bayesian EIV framework which detects and handles outliers.
6. It represents temporal models using hybrid dynamic Bayesian networks to estimate model parameters and to represent multimodal processes.

1.4 Thesis outline

A hybrid soft sensor for on-line estimation of a quality variable in the froth treatment process is developed in Chapter 2. Although special attention is given to one industrial case study, this chapter will serve as an introduction into soft sensor development in general.

The rest of this thesis is concerned with formulating the soft sensor problem in a Bayesian framework in order to overcome the typical limitations of conventional soft sensor development methods. These limitations will be addressed in Chapters 3 and 4. Chapter 3 discusses the problem of parameter estimation biased by noisy measurements. We provide a brief overview of Bayesian learning, and then present an EM-based Bayesian framework which is robust to noise-contaminated data sets. This approach is applied to estimating the model parameters of a three-tank system, which will serve as an experimental example throughout this thesis. Results are compared to the estimates obtained from the classical Least Square Regression and Total Least Squares methods. Existence of outliers and missing measurements in data sets are discussed in Chapter 4. In that chapter, first we will review some of the existing treatments of outliers and missing values. Next, we will define the concept of Bayesian inference and give a quick introduction to inference algorithms. The main contribution of this chapter, however, is an explanation of how to deal with outliers and missing values using the Bayesian approach. We compare the Bayesian approach with others through experimental and industrial evaluations.

We next turn our attention to Bayesian models which represent dynamic processes. Chapter 5 provides an overview of learning and inference in dynamic Bayesian models. Later sections of this chapter focus on Switching Kalman Filters as an example of a hybrid dynamic Bayesian model, i.e. a model that contains both discrete and continuous variables. Further, the problem of multimodal processes is solved using switching Kalman filters.

The thesis will conclude in Chapter 6 with a discussion of the most important results and with some suggestions for future research.

1.5 A note on software

Most of the algorithms and examples in this thesis have been implemented using Bayes Net Toolbox (BNT), which is an open source MATLAB package available at http://bnt.sourceforge.net/. Moreover, the probabilistic frameworks for nonlinear systems and non-Gaussian variables have been developed in WinBUGS. A free educational version of WinBUGS can be downloaded from http://www.mrc-bsu.cam.ac.uk/bugs/.

1.6 Bibliography

1. Aaron L. Z. and T. Y. Shan, *Development of an Industrial Soft Sensor*, Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore, 2004
2. Abony J., J. Madar, and F. Szeifert, *Combining First Principles Models and Neural Networks for Generic Model Control*, 6th On-line World Conference on Soft Computing in Industrial Applications (WSC6), Internet, September 2001
3. Azimzadeh F., H. A. Palizban, J. A. Romagnoli, *Online Optimal Control of a Batch fermentation Process Using Multiple Model Approach*, Proceedings of the 37th IEEE Conference on Decision & Control, Tampa, Florida, December 1998
4. Brandel, J., *Empirical Bayes Methods for Missing Data Analysis*, Technical Report 2004:11, Department of Mathematics, Uppsala University, June 2004
5. Champagne M., M. Amazouz, and R. Platon, *the Application of Soft Sensors in the Pulp and Paper and Cement Manufacturing Sectors for Process and Energy Performance Improvement*, Technical Report. CETC 2005-100(TR), Effective Assets Inc., June 2005
6. Chen C., *Robust Regression and Outlier Detection with the ROBUSTREG Procedure*, Proceedings of the 27th Annual Users group International Conference; SAS Institute, Cary, NC, 2002
7. Esmaily Radvar G., *Practical issues in Non-linear System Identification*, M.Sc. Thesis, Department of Chemical and Materials Engineering, University of Alberta, Spring 2002
8. Friedman Y. Z., E. A. Neto, and C. R. Porfirio, *First Principles Distillation Inference Models for Product Quality Prediction*, Hydrocarbon Processing Journal, February 2002
9. Grantham, S. D., L.H. Ungar, *A First Principles Approach to Automated Troubleshooting of Chemical Plants*, , Computers and Chemical Engineering, Vol. 14, No. 7, pp. 783-798, 1990
10. Huang B., *Detection of Abrupt Changes of Total Least Squares Models and Application in Fault Detection*, IEEE Transactions on Control Systems Technology, Vol. 9, No. 2, March 2001
11. Huber, P. J., *Robust Estimation of a Location Parameter*, Annals of Mathematical Statistics, Vol. 101, pp. 35-73, 1964
12. Huffel S. V. and J. Vandewalle, *The Total Least Squares Problem : Computational Aspects and Analysis*, Society of Industrial and Applied Mathematics, Philadelphia, 1991

13. Korb K. B. and A. E. Nicholson, *Bayesian artificial intelligence*, Chapman & Hall/CRC, 2004

14. Lakshminarayanan L., A. K. Tangirala, and S. L. Shah, *Soft Sensor Design Using Partial Least Squares and Neural Networks: Comparison and Industrial Applications*, AIChE Annual Meeting, Miami Beach, FL, November 1998

15. Li B., *Project Review: Soft Sensors*, Technical Report., University of Alberta, February 2005

16. Stephanopoulos, G. and C. Han, *Intelligent Systems in Process Engineering: a Review,* Computers and Chemical Engineering, Vol. 20, No. 6/7, pp. 743–791, 1996

17. Van Lith P., *Hybrid Fuzzy-First Principles Modeling*, Twente University Press, Netherlands, 2002

18. Zamprogna E., M. Barolo, and D. E. Seborg, *Optimal Selection of Soft Sensor Inputs for Batch Distillation Columns Using Principal Component Analysis*, Journal of Process Control, Vol. 15, pp. 39-52, 2005

Chapter 2

Soft Sensor Development for the Froth Treatment Process

This chapter discusses the issue of the hybrid modeling in soft sensor development. The theoretical aspects and steps in soft sensing are illustrated through an industrial case study. The ultimate goal of this effort is to develop an inferential model for the froth treatment process and to enhance on-line estimation of a quality variable through this application. Due to the lack of data measurements or some other required information, we are not able to develop a complete first principles model. Instead, we use our basic understanding of the process to form the model structure for developing a reliable soft sensor. Having selected an appropriate model structure, parameter estimation techniques are applied to it to obtain a hybrid inferential model. The reliability of developed model is then verified by evaluating its prediction performance on both off-line and on-line data of froth treatment process.

2.1 Introduction

The difficulty in monitoring key quality variables is that they are often not available on demand due to limitations such as the low reliability or high installation and maintenance cost of the equipment. There is a need to develop inferential models to provide continuous on-line estimations of the variables that can otherwise be measured only after several hours of lab analysis. A soft sensor [1,3] can infer process state and product quality variables from readily available process measurements. There are several advantages of soft sensors in comparison with traditional instrumentation.

1. They give more insight into process through capturing the information hidden in data.
2. They are emerging technology which allows industrial users to:

a) improve productivity [5],
b) become more energy efficient [13],
c) reduce environmental impact [14], and
d) improve business profitability by reducing the production cost associated with off-spec products [10].

3. They are easy to maintain.
4. They involve no capital cost.

Since the mid 80's, there has been growing interest in the use of soft sensor technology for industrial applications [15]. The overview of soft computing provided in [15] distinguishes seven areas of soft sensor application in process engineering:

- Diagnosis of process operations
- Monitoring and analysis of process trends
- Intelligent control
- Heuristics and logic in planning and scheduling of process operations
- Modeling languages, simulation and reasoning
- Intelligence in scientific computing
- Knowledge based engineering design

Regardless of the field of application, there are mainly three types of soft sensors available:

1. *White box or first principles models*: First principles modeling [4,6,8] is based on formulating and solving mass or energy balances on chemical process systems. Although first principles models have many advantages, they are the least common soft sensors in existence today. Since the large number of measurements needed are not often available in the industrial applications, not all the terms in the macroscopic balances are exactly or even partially known. Even when critical information about a process is available, the computation time may become excessive for complex chemical systems.
2. *Black box models*: Black box modeling [12,18] is a purely data-driven approach, in which process knowledge is not included in the development of soft sensors. These models are useful if a physical understanding of the system is absent or not relevant.
3. *Gray box models*: If the process knowledge is incomplete, the gray box models [2,13,16] combine both first principles and black box modeling. In this approach, also known as hybrid

modeling, black box models are used only to represent the unknown terms of mass or energy balances or other first principles equations.

A soft sensor development project is an iterative process consisting of collecting process data, cleaning them up, developing a soft sensor model, and evaluating the model performance. If the performance is acceptable, the sensor will be placed on-line. A less satisfactory verification may require starting over, or at least requires a close examination of each development step.

To illustrate the discussed methodology in on-line estimation of a quality variable using the gray box approach, we provide one industrial case study. This chapter will be divided into three main sections with the first part containing background on the process. This will include the summary of the froth treatment process, an introduction to existing Naphtha to Bitumen ratio (N:B) measurements, and evaluation of their performances. The second section will be the development of soft sensor including first principles and path analysis. Finally, the prediction performance of the developed soft sensor will be compared with the performance of the current measurements.

2.2 Process description

2.2.1 The froth treatment process

Hot water is added to oil sands, and the resulting slurry is piped to an extraction plant where it is agitated and oil is skimmed from the top. The combination of hot water and agitation releases bitumen from the sands, and allows small air bubbles to attach to the bitumen droplets. The bitumen froth floats to the top of separation vessels, and is further treated to remove residual water and fine solids. Bitumen is much thicker than conventional crude oil, so it must be either mixed with lighter petroleum (either liquid or gas) or chemically split before it can be transported by pipeline for upgrading into synthetic crude oil [19]. In the froth treatment process the bitumen froth is diluted by adding a gasoline-like product called naphtha. Diluted bitumen froth is then fed into a combination of Inclined Plate Settlers (IPS) and centrifuges to remove contaminants. Generally, the undiluted feed is divided into two parts; one goes through two stages of centrifuges, while the other one goes through IPS. The IPS underflows join the froth from the froth tank to form the feed to the first centrifuge stages. A schematic figure of the froth treatment process is given in Figure 2.1.

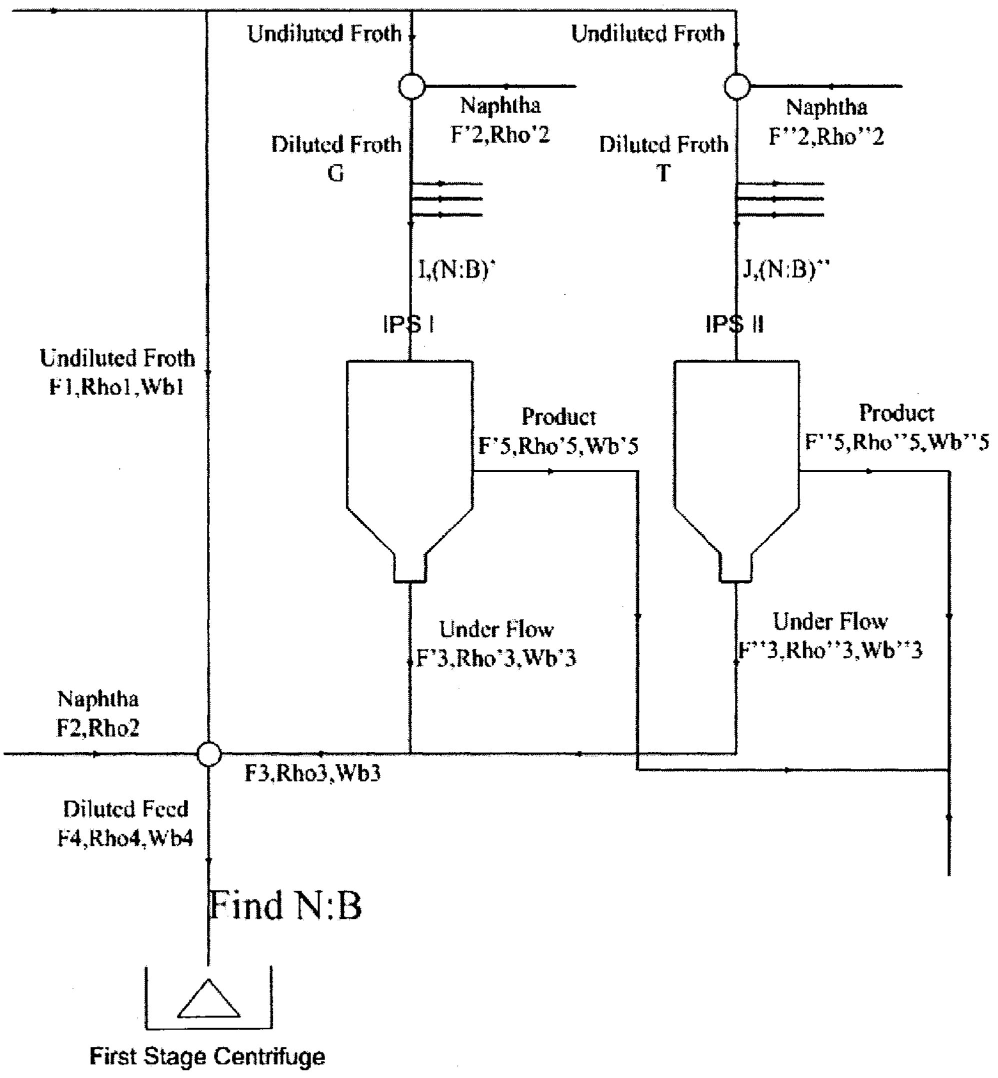

Figure 2.1. Part of a process flow diagram of a froth treatment plant

2.2.2 Existing measurements

It is very important to maintain the Naphtha to Bitumen ratio (N:B) in feed streams at certain levels so as to achieve effective and efficient separation at an affordable cost. There exist three kinds of Naphtha to Bitumen ratio measurements in diluted froth fed to the first centrifuge stage:

- Refractometer: On-line physical instrumentations
- Calculated Tags: On-line calculation algorithms
- Lab Data: References for N:B estimation

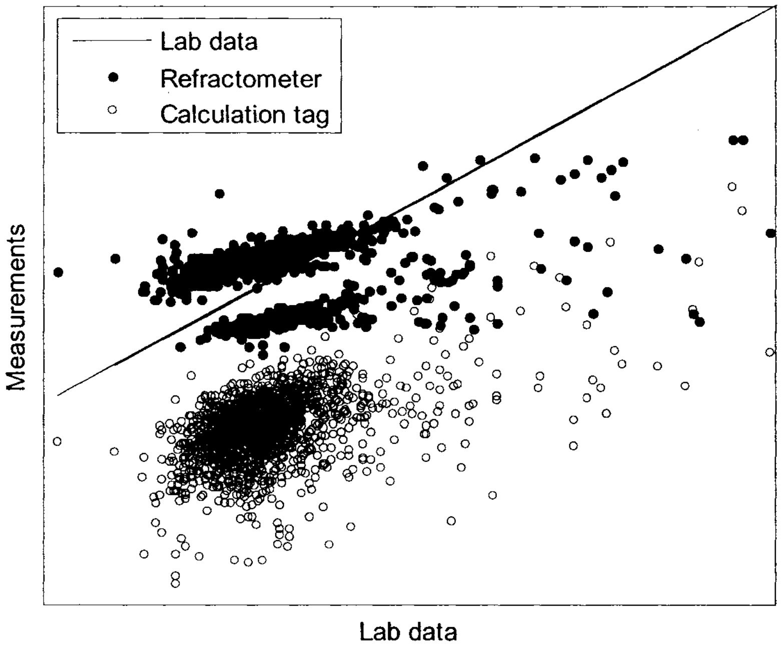

Figure 2.2. Scatter plot comparison of existing N:B measurements

N:B is a key quality variable that is not available "on demand" and is available only after several hours of lab analysis. For this reason, refractometer and calculation tags have been used to provide on-line N:B estimation and increase control efficiency. Lab data is used to evaluate the accuracy of N:B ratios given by refractometers and existing calculation algorithms.

Outputs of refractometer and calculation tags are compared with lab data in Figure 2.2. It is clear that existing measurements are not accurate enough. Specifically, the calculation results are lower than the lab data, and smaller N:B ratios than true values were used in N:B ratio controllers. This under-estimation resulted in the addition of more naphtha than required, and consequently increased the operating cost for this plant. As a result, there is a need to improve the accuracy of on-line N:B estimation.

2.3 Soft sensor development

Our overall purpose is to increase bitumen production while reducing naphtha usage via better N:B measurements. In order to achieve this objective, we are interested in deploying soft sensor technology to our problem.

Presently, however, due to the complexity of the system and the interrelation of centrifuge stages with IPS units, we do not have a fundamental and complete understanding of the process. Thus, we are interested in the gray models that use first principles to search for an appropriate model structure, while historical data reveal the relationship between N:B ratio and on-line measurable process variables to develop soft sensors with improved accuracy. The data collection process is important because data quality is crucial for the development of reliable models. Our data was collected by automated data historians. First, our efforts were focused on cleaning and preprocessing data in order to achieve a data set which adequately represents the process for the normal operation, so that reliable data would be used to develop and validate the sensor.

2.3.1 First principles

To estimate the N:B in the diluted feed to the first centrifuge stage, mass balance equations for naphtha and dry bitumen are formulated as follows:

$$F_4\rho_4 Wb_4 = F_1\rho_1 Wb_1 + F_3\rho_3 Wb_3 \qquad \textbf{Dry Bitumen} \qquad \textbf{(2.1)}$$

$$F_4\rho_4 Wn_4 = F_2\rho_2 + F_3\rho 3Wn_3 \qquad \textbf{Naphtha} \qquad \textbf{(2.2)}$$

where:

- F_1: Undiluted feed flow rate
- F_2: Under-flow rate from IPS
- F_3: Naphtha flow rate
- F_4: Diluted feed flow rate
- ρ_i: Density
- Wb_i: Weight ratio of dry bitumen
- Wn_i: Weight ratio of Naphtha.

Combining balance equations we have:

$$NB = \frac{F_2\rho_2 + F_3\rho 3Wn_3}{F_1\rho_1 Wb_1 + F_3\rho_3 Wb_3} = \frac{F_2\rho_2 + F_3\rho_3 Wn_3}{F_4\rho_4 Wb_4} \qquad \textbf{(2.3)}$$

In this equation, F_2 and F_4 are the only variables for which we have measurements. Based on some understanding of the physics of the process, we need to make four initial assumptions in order to estimate the remaining unknown variables.

1. The Naphtha to Bitumen ratio of the product and underflow streams of the IPS I and IPS II are almost the same as the measured N:B's in the corresponding diluted froth feeds to them.
2. Although we know that the diluted froth density, naphtha density, product density and underflow density vary from time to time, we first consider them as being constants. We will come back to this assumption and adjust the effects of density changes using correction factors.
3. We can estimate the composition of the product streams as follows:

$$(1-Ww'_5)=(1+{}^{1}\!/_{(NB)'})Wn'_5 \qquad \textbf{(2.4)}$$

$$(1-Ww''_5)=(1+{}^{1}\!/_{(NB)''})Wn''_5 \qquad \textbf{(2.5)}$$

4. Considering dynamic equations around IPS vessels, the flow rate of the product streams can be estimated :

$$I_1\rho'_{dil}-F'_5\rho'_5-F'_3\rho'_3=A'\frac{dh'}{dt}\rho'_{in}\Rightarrow$$

$$F'_5\rho'_5=A'\frac{dh'}{dt}\rho'_{in}+F'_3\rho'_3-I_1\rho'_{dil} \qquad \textbf{(2.6)}$$

$$J_1\rho''_{dil}-F''_5\rho''_5-F''_3\rho''_3=A''\frac{dh''}{dt}\rho''_{in}\Rightarrow$$

$$F''_5\rho''_5=A''\frac{dh''}{dt}\rho''_{in}+F''_3\rho''_3-J_1\rho''_{dil} \qquad \textbf{(2.7)}$$

Based on these initial assumptions, the amount of Naphtha in the IPS underflow stream is then estimated from Equation 2.8:

$$F_3\rho_3Wn_3=\underbrace{\left[\frac{I_1}{G}F'_2\rho'_2-F'_5\rho'_5Wn'_5\right]}_{IPS\ I}+\underbrace{\left[\frac{J_1}{T}F''_2\rho''_2-F''_5\rho''_5Wn''_5\right]}_{IPS\ II} \qquad \textbf{(2.8)}$$

where:

- $G=\sum_{j=1}^{4}I_j$
- $T=\sum_{i=1}^{4}J_i$

Prediction of N:B based on the first-principles model requires the measurement of the densities and the exact estimation of Naphtha content in the IPS underflows. In the absence of these measurements, gray box modeling of the process is considered where a new function is defined to represent the black box part of the model. Combining initial assumptions with Equation 2.3, the N:B estimation model is formulated as follows:

$$NB = \frac{F_2\rho_2 + F_3\rho_3 Wn_3}{F_4\rho_4 Wb_4} \Rightarrow$$

$$NB = \frac{F_2 + (1/\rho_2)F_3\rho_3 Wn_3}{F_4(\rho_4/\rho_2)Wb_4} \Rightarrow$$

$$NB = \frac{F_2 + \theta_1 F_3\rho_3 Wn_3}{g(F_2, F'_3, F''_3, F_4, (NB)', (NB)'', \rho_i)} \tag{2.9}$$

It is assumed that $g = F_4(\rho_4/\rho_2)Wb_4$ is a function of some of the known process variables. We cannot determine how much each variable influences the predicted g, based only on our understanding of the process. As a result, the sensitivity of g to each input variable has been tested by the use of "*path analysis*" to identify the known variables having the largest influence on the accuracy and robustness of the prediction.

2.3.2 Path analysis

The basic idea of path analysis

There is an entire methodological/philosophical area of science devoted to causal theory. This area of study concentrates on theories regarding how a researcher can conclude whether X *actually* causes Y. Given an output variable, Y, and a list of several input variables, X, correlations between variables are calculated by regression analysis. It is known, however, that the existence of a causal relationship cannot be concluded from a significant correlation coefficient. Path analysis, which is an extension of regression analysis, provides a framework for the researcher to think more carefully about how the X and Y variables are related, as well as how the X variables are related to each other. Generally, path analysis is the combination of assumed causal theory with empirical evidence that can:

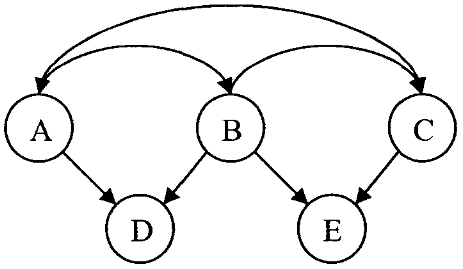

Figure 2.3. An example of path model

1. provide a graphical way to represent the assumed theory;
2. provide a way to empirically estimate whether the assumed relationships are positive, negative, and importantly to test whether the relationship is zero and hence not supported by the data;
3. provide a way to estimate the assumed causal effect that one variable has on another through its assumed causal effect on other variables; and
4. prove whether the experimentally changed input actually causes the output changes.

As far as non-experimental data is concerned, however, path analysis has some limitations such as:

1. uncertainty whether one input variable actually causes the output,
2. difficulty in determining the direction of causal order between variables, and
3. inability to distinguish between models that result in identical correlation patterns.

Key terms and symbols

In this section, we first introduce the main concepts which are important to the understanding of the path analysis. Key terms and symbols of path analysis are defined as follows:

Path model: A diagram relating independent, intermediary, and dependent variables as shown in Figure 2.3.
Causation: A straight, single-headed arrow represents the assumption that the variable at the base of the arrow is a cause of the variable at the head.
Correlation: A curved, double-headed arrow represents an unanalyzed (spurious) association between two variables. This association is correlation, a result of causal variables that are not part of the model of interest.

Causal paths: Causal paths to a given variable include the direct paths from arrows leading to it and correlated paths from endogenous variables correlated with others which have arrows leading to the given variable.

Exogenous variable: A variable whose variability is assumed to be determined by causes outside of the model. No attempt is made to explain the variability of exogenous variables or their relations with other exogenous variables.

Endogenous variable: A variable whose variability is explained by exogenous or endogenous variables in the model.

Path coefficient: A standardized regression coefficient showing the direct effect of an independent variable on a dependent variable in the path model. The symbol, p_{ji}, is the path coefficient for X_i a X_j. Thus when the model has two or more causal variables, path coefficients are partial regression coefficients which measure the extent of the effect of one variable on another in the path model, using standardized data or the correlation matrix as input.

Effect decomposition: Based on assumed causal relationships, any bivariate correlation between two variables can be broken down into a series of effects. Path coefficients may be used to decompose correlations in the model into four pieces.

1. **Direct Effect** is the influence of one variable on another that is unmediated by any other variable, i.e. each single-headed arrow represents a direct effect.
2. **Indirect Effect** is an effect that is mediated by at least one intervening variable.
3. **Unanalyzed** is a correlation involving unanalyzed associations among predetermined variables.
4. **Spurious** is a correlation due to joint dependence on common or correlated variables.

In general,

- Total effect = Total causal effect + Non-causal effect
- Total causal effect = Direct effect + Indirect effect

An illustrative example

To illustrate our discussion of path analysis, we consider the following regression model:

$$Y = p_{Y1}X_1 + p_{Y2}X_2 + \ldots + p_{Yn}X_n + p_{Ye}e \qquad \textbf{(2.10)}$$

where X_i's and Y are standardized variables.

The correlation coefficient between Y and X_i can be constructed from the path diagram:

$$r_{Yi} = Cov(Y, X_i) = Cov(\sum_{j=1}^{n} p_{Yj} X_j, X_i) = \sum_{j=1}^{n} p_{Yj} r_{ji} \qquad \textbf{(2.11)}$$

We also need to study variance decomposition to gain more information about effects.

$$\begin{aligned} 1 = Var(Y) &= Var\left(\sum_{i=1}^{n} p_{Yi} X_i + p_{Ye} e\right) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{n} p_{Yi} r_{ik} p_{Yk} + p_{Ye}^2 \\ &= \sum_{i=1}^{n} p_{Yi}^2 + 2\sum_{i=1}^{n}\sum_{k=i+1}^{n-1} p_{Yi} r_{ik} p_{Yk} + p_{Ye}^2 \\ &= v_d + v_i + v_u \end{aligned} \qquad \textbf{(2.12)}$$

where,

- v_d : Contribution from direct effects
- v_i : Contribution from indirect effects
- v_u : Contribution from unknown sources

Based on variance decomposition, we can define two useful indices:

1. *The completeness index* of the selected variables is formulated as $\gamma_c = v_d + v_i$ and bounded from 0 to 1. γ_c indicates the portion of output variance that can be explained by the selected input variables.
2. *The significance index* of the direct effect is formulated as $\gamma_d = 1 - \frac{|v_i|}{|v_d|}$. γ_d indicates the portion of the effects represented by the direct effects[9].

Application of path analysis to the identification of g function

Many measured variables should be investigated to identify the ones having the largest influence on accuracy and robustness to predict $g_1 = F_4(\rho_4/\rho_2)Wb_4$. The output variable g is estimated as follows:

$$NB = \frac{F_2 + (1/\rho_2)F_3\rho_3Wn_3}{F_4(\rho_4/\rho_2)Wb_4} \Rightarrow g = F_4(\rho_4/\rho_2)Wb_4 = \frac{F_2 + (1/\rho_2)F_3\rho_3Wn_3}{NB} \tag{2.13}$$

Using lab data, we calculate the direct path coefficients to find the main contributors to *g*. It can be difficult, however, to select the optimal set of variables to be used as inputs, because there are many possible measured variables affecting *g*. The direct effects on $g_1 = g(F_2, F_2', F_2'', F_3', F_3'', F_4, NB', NB'', Ww_5', Ww_5'', I_1, J_1, F_3\rho_3Wn_3)$ are reported in Table 2.1. According to the coefficients of Table 2.1, we may choose F_3', F''_3, F_4, F_2, and $F_3\rho_3Wn_3$ as main contributors. Direct path coefficients of the new path diagram, g_2, are presented in Table 2.2.

Table 2.1. Direct effects on g_1

Variable	**Path Coefficient**
F_2'	0.0112
F_3'	-0.1848
NB'	-0.0127
Ww'_5	0.0335
I_1	-0.0321
$F_3\rho_3Wn_3$	0.4827
F_4	0.8853
F_2''	0.0507
F''_3	-0.2237
NB''	-0.0477
Ww''_5	0.0040
J_1	-0.0875
F_2	0.1571

Table 2.2. Direct effects on g₂

Variable	**Path Coefficient**
F_3'	-0.2070
$F"_3$	-0.2642
$F_3\rho_3Wn_3$	0.5175
F_4	0.8373
F_2	0.1953

Table 2.3. Summary of indices

	γ_c	γ_d
g_1	0.7502	0.6577
g_2	0.7123	0.6362

In order to decide on the most suitable model for on-line implementation, the completeness and significance indices of the selected variables for g_1 and g_2 are calculated and given in Table 2.3. The first column of the table shows that the selected variables for both g_1 and g_2 can explain most of the variability of g function. The second column indicates that the source of variability is isolated and can be identified. In addition, the completeness index related to g_1 is not considerably greater than the one corresponding to g_2. These results indicate that the variables presented in Table 2.2 are sufficient to explain most of the variability in the g function. This way, a trade-off between a low modeling error and model complexity can be made.

2.3.3 The inferential model

We also reap the benefits of our process knowledge to explore the possible structures of g_2:

$$NB = \frac{F_2 + \theta_1 F_3 \rho_3 Wn_3}{\theta_2 F_2 + \theta_3 F_3 \rho_3 Wn_3 + \theta_4 F_4 + \theta_5 F'_3 + \theta_6 F"_3} \qquad \textbf{(2.14)}$$

Now, data fitting techniques such as direct non-linear regression can be used in Equation 2.14 to estimate model parameters.

2.4 Performance evaluation

2.4.1 Off-line results

Altogether 1751 data points have been used to train and evaluate our soft sensor. The proposed method was applied to the timed data sets and the model was developed using 1151 data points collected from July 2003 to March 2004. The model performance will be verified using a new set of records collected from April 2004 to June 2005. Figure 2.4 presents scatter plot comparisons of each N:B measurement in relation to the lab data. The ideal case would be for all the data points to lie exactly along the diagonal, indicating that the model and the lab data are exactly the same. It is obvious that estimated N:B values from the soft sensor fit the lab data much better than the ones from calculation tags. Figure 2.5 depicts a zoomed view of the soft sensor and refractometer outputs shown in Figure 2.4. This zoomed figure reveals that the soft sensor provides reasonably successful prediction followed by capturing changes in both measured and quality variables.

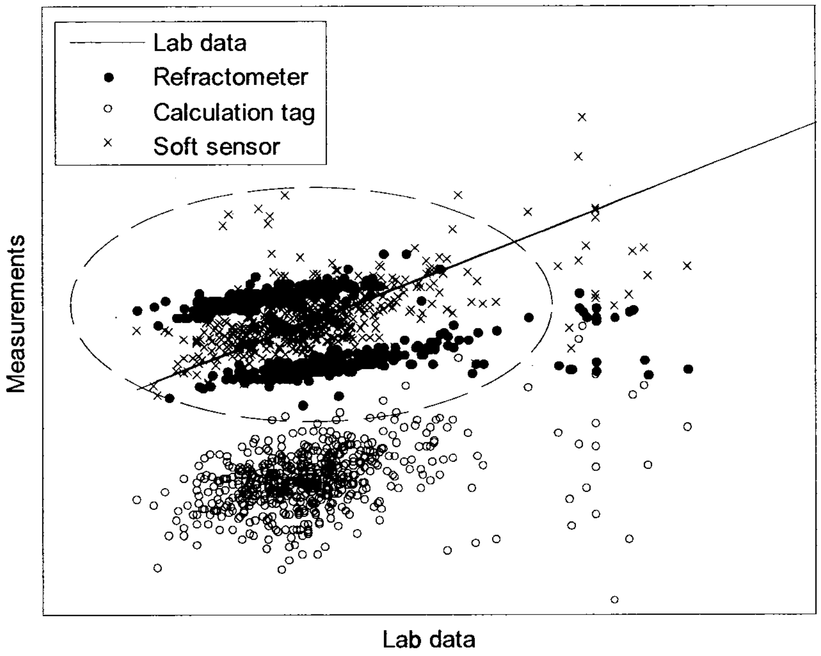

Figure 2.4. Scatter plot comparison of different N:B measurements in off-line verification

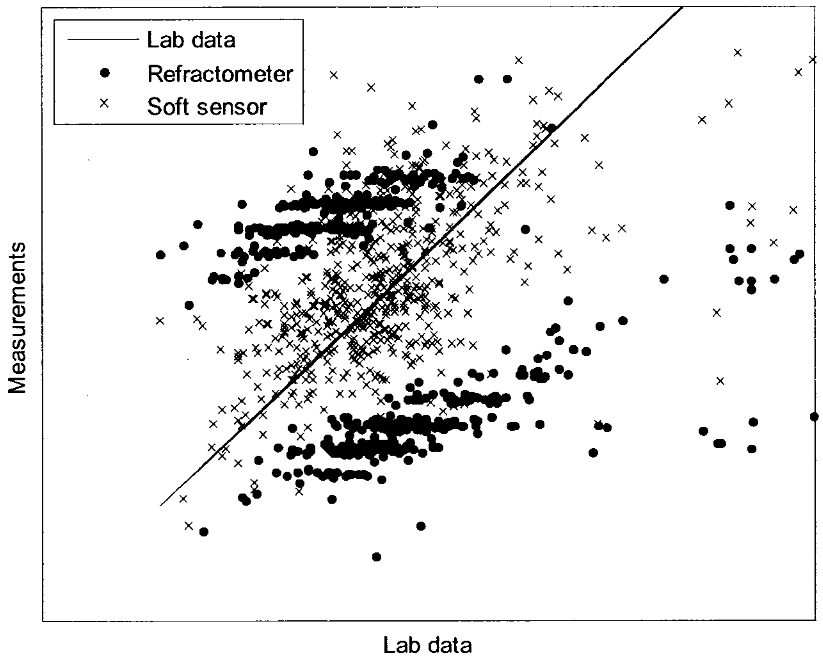

Figure 2.5. Zoomed Figure 2.4 around dashed circle

Table 2.4. Mathematical comparison of N:B measurements on off-line testing data

	Refractometer	**Calculation Tags**	**Soft Sensor**
Mean Absolute Error	0.0638	0.2032	0.0269
Standard Deviation	0.0504	0.0409	0.0367
Mean Squared Error	0.0057	0.0434	0.0014

There are two different criteria for evaluating the performance of an instrument: accuracy and precision. Accuracy is represented by the average of absolute errors, and precision is usually expressed through the standard deviation of errors. Mean absolute error is the extent of agreement between an observed variable and the reference value for the parameter being measured. Small values, corresponding to high accuracy, can be defined as a combination of high precision and low bias. Standard deviation measures how spread the values in a data set are. A large standard deviation indicates that the data points are far from the mean and a small standard deviation indicates that they are clustered closely around the mean. Besides, mean squared error is commonly used to indicate the integrated performance of accuracy and precision.

To analyze the prediction error for each approach one can see their mean absolute error, standard deviation, and mean squared errors presented in Table 2.4. Since the soft sensor is both more accurate and more precise, according to the results presented in this table, we conclude that the developed soft sensor provides better estimation with less mean squared prediction error.

2.4.2 Implementation results

The developed soft sensor has been undergoing on-line tests since June 9, 2006. Soft sensor and refractometer measurements from June 9, 2006 to January 15, 2007 are presented in Figure 2.6 and Figure 2.7, respectively. The error analyses for this period are presented in Table 2.5. Although refractometer and soft sensor are comparable in terms of mean absolute and mean square errors, trend comparisons in Figure 2.6 and Figure 2.7 reveal that the refractometer records essentially a straight line while the soft sensor does capture significant changes much better.

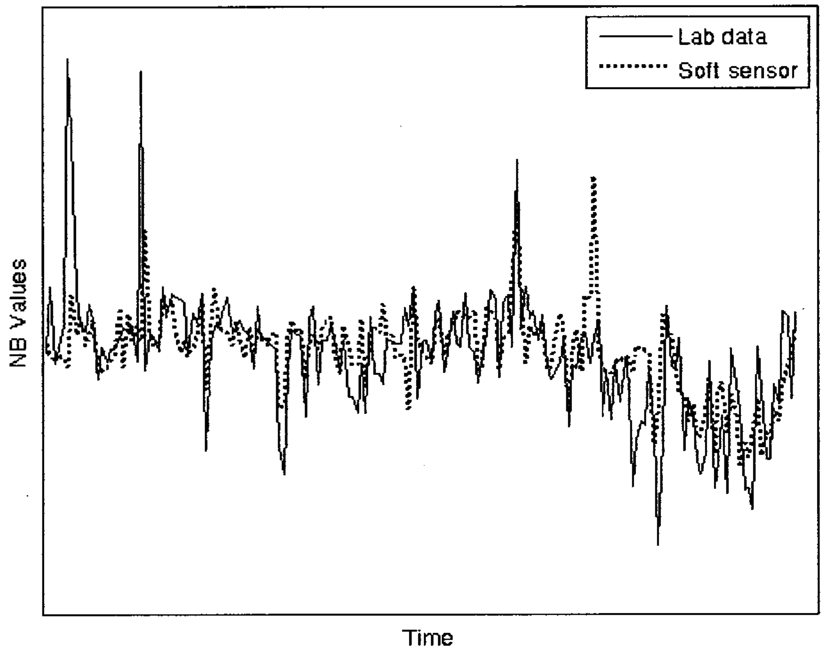

Figure 2.6. Trend comparison of lab data and soft sensor measurements

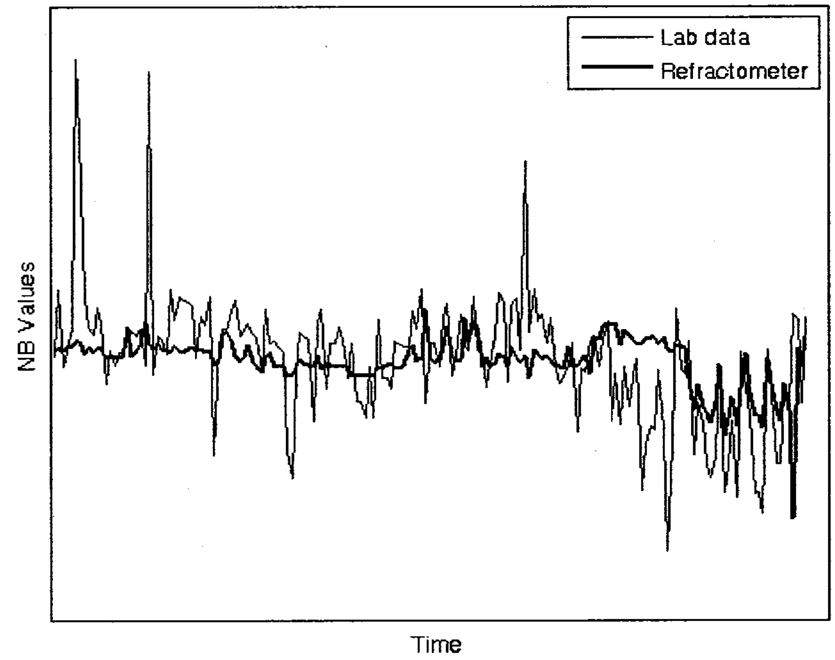

Figure 2.7. Trend comparison of lab data and refractometer measurements

Table 2.5. Mathematical comparison of N:B measurements on on-line data

	Refractometer	**Calculation Tags**	**Soft Sensor**
Mean Absolute Error	0.0453	0.1777	0.0394
Standard Deviation	0.0287	0.0603	0.0492
Mean Squared Error	0.0045	0.0366	0.0036

To conclude this section, we summarize the soft sensor development procedure described above in the one-step regression flow chart presented in Figure 2.8.

2.5 Conclusion

N:B is a key quality variable that is not available on demand but only after several hours of lab analysis. For effective and efficient control, on-line N:B estimation is required. In First Stage Centrifuge, Refractometer results are not reliable or accurate enough. In addition, existing calculation results are lower than those found in lab data, and smaller N:B ratios have been used in N:B ratio controllers. For these reasons, the accuracy of N:B measurements should be improved. In order to achieve this objective, we have deployed soft sensor technology to our problem. Our soft sensor development project is an iterative process consisting of collecting process data, cleaning them up, developing a soft sensor model, and reviewing the model's performance. By scatter plot comparison and error analysis of the developed soft sensor and existing measurements, we have concluded from our test data sets that our developed soft sensor does improve the estimation accuracy of N:B.

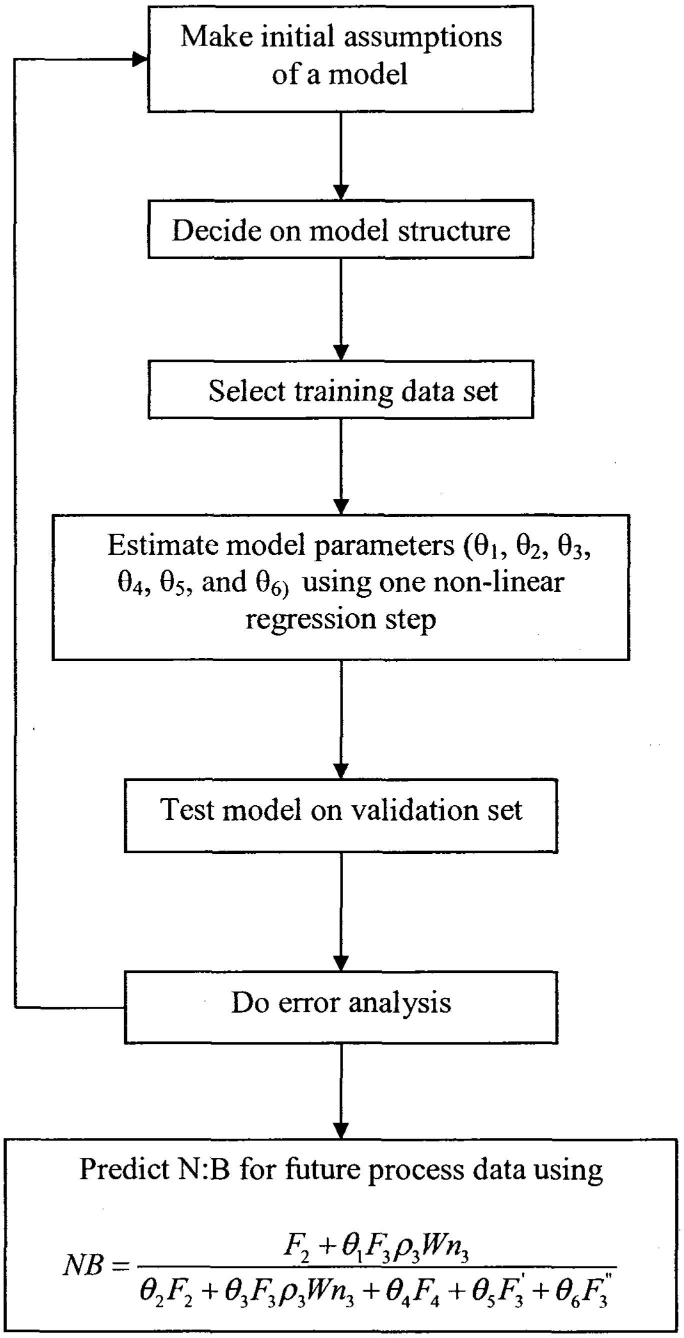
Make initial assumptions of a model
Decide on model structure
Select training data set
Estimate model parameters (θ_1, θ_2, θ_3, θ_4, θ_5, and θ_6) using one non-linear regression step
Test model on validation set
Do error analysis
Predict N:B for future process data using
$NB = \frac{F_2 + \theta_1 F_3 \rho_3 Wn_3}{\theta_2 F_2 + \theta_3 F_3 \rho_3 Wn_3 + \theta_4 F_4 + \theta_5 F_3^{'} + \theta_6 F_3^{''}}$

Figure 2.8. The flow diagram of the one-step method for soft sensor development

2.6 Bibliography

1. Aaron L. Z. and T. Y. Shan, *Development of an Industrial Soft Sensor*, Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore, 2004
2. Abony J., J. Madar, and F. Szeifert, *Combining First Principles Models and Neural Networks for Generic Model Control*, 6th On-line World Conference on Soft Computing in Industrial Applications (WSC6), Internet, September 2001
3. Champagne M., M. Amazouz, and R. Platon, *the Application of Soft Sensors in the Pulp and Paper and Cement Manufacturing Sectors for Process and Energy Performance Improvement*, Technical Report. CETC 2005-100(TR), Effective Assets Inc., June 2005
4. Esmaily Radvar G., *Practical issues in Non-linear System Identification*, M.Sc. Thesis, Department of Chemical and Materials Engineering, University of Alberta, Spring 2002
5. Feil B., J., Abonyi, P. Pach, S. Nemeth, P. Arva, M. Nemeth, and G. Nagy, *Semi-mechanistic Models for State-Estimation Soft Sensor for Polymer Melt Index Prediction*, in Lecture Notes in Computer Sciences, L. Rutkowski *et al.* (Editors), pp. 1111-1117, Springer-Verlag, Berlin, 2004
6. Friedman Y. Z., E. A. Neto, and C. R. Porfirio, *First Principles Distillation Inference Models for Product Quality Prediction*, Hydrocarbon Processing Journal, February 2002
7. Garson G. D., *Multivariate Analysis in Public Administration: Path Analysis,* Lec. Note, North Carolina State University, Fall 2001
8. Grantham, S. D., L.H. Ungar, *A First Principles Approach to Automated Troubleshooting of Chemical Plants*, , Computers and Chemical Engineering, Vol. 14, No. 7, pp. 783-798, 1990
9. Huang B., N. Thornhill, S. Shah, and D. Shook, *Path Analysis For Process Troubleshooting,* Proceedings of AdConIP, Kumamoto, Japan, pp. 149-154, June 2002
10. Karmer M. K., R. S. H. Mah, *Model-Based Monitoring,* 2nd Conference on Foundations of Computer Aided Process Operations, Colorado, July 1993
11. Korb K. B. and A. E., Nicholson, *Bayesian Artificial Intelligence*, Chapman & Hall/CRC, 2004
12. Lakshminarayanan L., A. K. Tangirala, and S. L. Shah, *Soft Sensor Design Using Partial Least Squares and Neural Networks: Comparison and Industrial Applications*, AIChE Annual Meeting, Miami Beach, FL, November 1998

13. Li B., *Project Review: Soft Sensors*, Technical Report., University of Alberta, February 2005

14. Oliveira-Esquerre K. P., D. E. Seborg, R. E. Bruns, M. Mori, *Application of Steady State and Dynamic Modeling for the Prediction of the BOD of an Aerated Lagoon at a Pulp and Paper Mill: Part I. Linear Approaches*, Chemical Engineering Journal, Vol. 104, pp. 73-81, 2004

15. Stephanopoulos, G. and C. Han, *Intelligent Systems in Process Engineering: a Review,* Computers and Chemical Engineering, Vol. 20, No. 6/7, pp. 743–791, 1996

16. Van Lith P., *Hybrid Fuzzy-First Principles Modeling,* Twente University Press, Netherlands, 2002

17. Wall M. M., *Latent variable modeling: Notes for path analysis,* Lec. Note, University of Minnesota, Fall 2004

18. Zamprogna E., M. Barolo, and D. E. Seborg, *Optimal Selection of Soft Sensor Inputs for Batch Distillation Columns Using Principal Component Analysis*, Journal of Process Control, Vol. 15, pp. 39-52, 2005

19. http://en.wikipedia.org/wiki/Tar_sands , accessed: February 2006

20. http://www.syncrude.ca/users/folder.asp , accessed: February 2006

Chapter 3

Bayesian Modeling for Error-in-Variable Problem

This chapter addresses the problem of parameter estimation from noise-contaminated input and output data. First, we focus on linear regression methods, since an extension to non-linear regression is straightforward using locally weighted learning approaches. Regular least squares regression and total least squares (TLS) are presented as classical approaches to estimating parameters. In these approaches, the accuracy of the estimated parameters suffers from noisy data sets and/or the inequality of the error variances among different variables. We develop a Bayesian algorithm that automatically detects measurement noises and improves parameter estimation. We then demonstrate the effectiveness of the proposed approach on simulation examples as well as through a pilot-scale experiment.

3.1 Introduction

The mathematical modeling of chemical processes is a core aspect of the simulation and optimization tools used for design and control purposes. Using appropriate models is necessary in optimizing process operating conditions, improving process analysis, and designing control strategies. A common problem in the development of process models is determining unknown parameters. Reliable data fitting techniques [1] such as regression methods are generally used to estimate model parameters on the basis of available laboratory or process data.

In classical least squares methods it is assumed that the independent process variables have been exactly observed and that the only dependent variable is noisy. However, the existing measurements, from which the parameters are estimated, are often contaminated with instrument error, process noise, and unmodeled process characteristics. As a result, the classical least squares approaches yield biased parameter estimates for most industrial applications. Alternative methods, such as the total least squares (TLS) [11] have been applied to compensate for the measurement noises. It has been proven that the TLS solution offers optimal parameter estimates in models with Gaussian measurement error, also referred to as classical Errors-in-Variables (EIV) models. Although TLS addresses input noise, it assumes that the measurement errors are independent random variables with zero mean and equal variances. Otherwise, the covariance of measurement noise needs to be known. In real world systems, this assumption is not necessarily true and, again, the resulting parameter estimates will be biased [10].

Bayesian modeling [3] is a statistical parameter estimation technique that improves the reliability of the estimation procedure. The most common approach to learning Bayesian models is to estimate the parameters using the Expectation Maximization (EM) algorithm [5,7,16]. The EM algorithm estimates parameters by iteratively finding the expectation of the parameters in the E step and then computing the maximum likelihood estimates of the parameters in the M step; this is done by maximizing the expected likelihood found in the E step. Our main contribution in this work is the development of a Bayesian procedure that improves parameter estimation in the presence of measurement noise in all variables (i.e. error in variables), and subsequent comparative studies using other classical methods. This framework automatically detects noise in the measured process data and identifies the model parameters. In addition, the Bayesian network is flexible in model structure, easy to expand, capable of dealing with irregularly sampled data or missing data, and powerful in many applications [14]. This is further discussed in Chapter 4 of this thesis.

In this chapter, we first review the concept of Bayesian learning and then provide an overview of the learning algorithms. We next develop an algorithm to bring in Bayesian modeling as a new approach for coping with measurement noises in model parameter estimation. Finally, we evaluate our approach on numerical simulations and a pilot-scale experiment.

3.2 Bayesian learning

A Bayesian approach to building a model can be decomposed into two basic problems. The first concerns learning the model structure, $\mathcal{M}$, while the second concerns learning the parameters, Θ, once the structure of the network has been selected. To define a learning problem, prior distributions over model structures $P(\mathcal{M})$ and over the parameters for each model structure $P(\Theta \mid \mathcal{M})$ are assumed. Assume a training data set of independent and identically distributed observations $\mathcal{D} = \{X_1, ..., X_N\}$. We can now use Bayes' theorem to express the posterior distribution for Θ as follows:

$$P(\Theta \mid \mathcal{M}, \mathcal{D}) = \frac{P(\mathcal{D} \mid \Theta, \mathcal{M}) P(\Theta \mid \mathcal{M})}{P(\mathcal{D} \mid \mathcal{M})} \tag{3.1}$$

For notational convenience, we drop the implicit conditioning on the model structure, which will not be used from now on. Therefore, Equation 3.1 can be written in the simplified form:

$$P(\Theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \Theta) P(\Theta)}{P(\mathcal{D})} \tag{3.2}$$

where $P(\Theta)$ is the prior distribution of Θ and $P(\mathcal{D} \mid \Theta)$ is the likelihood function.

To learn the parameters of a Bayesian model, we start off by defining the prior distribution, $P(\Theta)$, of the parameters taking into account all information aside from the data itself. The effect of data is then investigated in order to make some values of Θ less likely than others by multiplying the prior distribution with the likelihood function, $P(\mathcal{D} \mid \Theta)$. The posterior distribution of parameters, $P(\Theta \mid \mathcal{D})$, is thus more concentrated than the prior. The parameter values at the maximum of the posterior distribution are known as the *Maximum A Posteriori* (MAP), and are obtained from the following expression:

$$\Theta_{MAP} = \underset{\Theta}{\arg\max}\, P(\Theta \mid \mathcal{D}) \tag{3.3}$$

Within the limits of the large data set and non-informative priors (e.g. uniform), the MAP estimates are identical to the *Maximum Likelihood* (ML) estimators which can be expressed as follows:

$$\Theta_{ML} = \underset{\Theta}{\arg\max}\, P(\mathcal{D} \mid \Theta) \tag{3.4}$$

The following learning problem demonstrates the advantage of MAP based on Bayesian approach. Suppose that we want to compute the probability distribution of $\Theta = \{\theta_1, \theta_2\}$, which is a set of independent model parameters. Assume our data set is in fact independent of θ_2, so that $P(\mathcal{D} \mid \theta_2) = P(\mathcal{D})$. The form of Bayes' theorem convenient to our purpose of parameter learning is introduced as:

$$\begin{aligned} P(\theta_1, \theta_2 \mid \mathcal{D}) &= \alpha\, P(\mathcal{D} \mid \theta_1, \theta_2) P(\theta_1) P(\theta_2) \\ &= \alpha\, P(\mathcal{D} \mid \theta_1) P(\theta_1) P(\theta_2) \end{aligned} \tag{3.5}$$

As discussed above, the ML and MAP estimates are, respectively, calculated as:

$$\begin{aligned} \{\theta_1, \theta_2\}_{ML} &= \underset{\theta_1, \theta_2}{\arg\max}\, P(\mathcal{D} \mid \theta_1) \\ &= \underset{\theta_1}{\arg\max}\, P(\mathcal{D} \mid \theta_1) \end{aligned} \tag{3.6}$$

and,

$$\begin{aligned} \{\theta_1, \theta_2\}_{MAP} &= \underset{\theta_1, \theta_2}{\arg\max}\, P(\theta_1, \theta_2 \mid \mathcal{D}) \\ &= \underset{\theta_1, \theta_2}{\arg\max}\, P(\mathcal{D} \mid \theta_1) P(\theta_1) P(\theta_2) \end{aligned} \tag{3.7}$$

Given the prior distribution of parameters, the generic solution of ML and MAP estimators is as plotted in Figure 3.1. Since a bound on the prior is computed as well in MAP estimation, a feasible region for both θ_1 and θ_2 is obtained by MAP estimators. But, θ_2 is inconclusive from ML estimator.

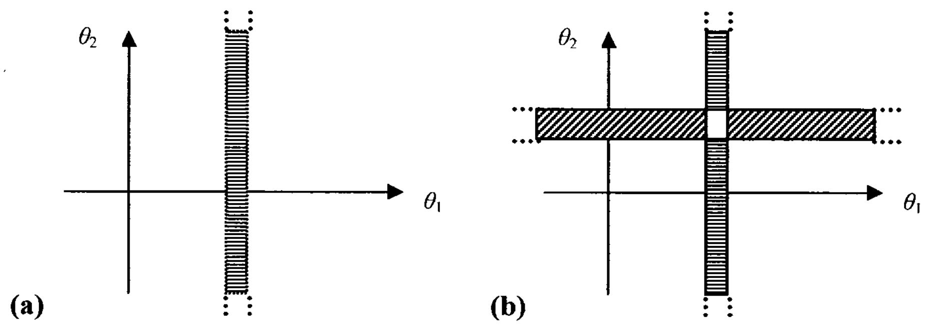

Figure 3.1. (a) ML estimates, (b) MAP estimates

3.3 Parameter learning algorithms

This section will focus on the problem of estimating model parameters when the structure of the model is given a priori. When learning parameters for a given model structure, two cases are usually considered:

1. Training data sets are complete, or
2. Training data sets are incomplete.

3.3.1 Known structure with complete data

Consider $\mathcal{D} = \{X_1, ..., X_N\}$ as a set of complete observed variables, each of which can be a vector. Assume $\mathcal{D}$ is independently and identically distributed (iid). To compute the posterior distribution, $P(\Theta \mid \mathcal{D})$, for a given data set, $\mathcal{D}$, the likelihood of the data set needs to be formulated as:

$$P(\mathcal{D} \mid \Theta) = \prod_{i=1}^{N} P(X_i \mid \Theta) \tag{3.8}$$

The goal of learning in this case is to find the maximum likelihood estimates (MLEs) of the parameters. These are obtained by maximizing the likelihood or the log likelihood:

$$L(\Theta) = \sum_{i=1}^{N} \log P(X_i \mid \Theta) \tag{3.9}$$

If the observations include all the variables in the Bayesian network, the log likelihood scoring function decomposes into a series of terms for each node:

$$\begin{aligned} \log P(X_i \mid \Theta) &= \log \prod_{j} P(X_i^{(j)} \mid Par(X_i^{(j)}), \theta_j) \\ &= \sum_{j} \log P(X_i^{(j)} \mid Par(X_i^{(j)}), \theta_j) \end{aligned} \tag{3.10}$$

where θ_j are the parameters that define the conditional probability of $X^{(j)}$ given its parents, $Par(X^{(j)})$. Now we need only to specify how to estimate the parameters of each type of CPD given their local data, $\{\theta_j(X_i^{(j)}, Par(X_i^{(j)}))\}$.

3.3.2 Known structure with incomplete data

Dealing with incomplete data samples is a very common case for machine learning from real world. Two kinds of incompleteness need to be considered in Bayesian modeling:

1. *missing* values: when the values for some attribute are simply absent from some of the joint observations; and
2. *hidden* variables: when all the measurements for some relevant variables are completely missing.

Missing values in real life data sets are one of the challenges, that pose a serious problem in building models. Let $\boldsymbol{D}_o \subset \mathfrak{D}$ and $\boldsymbol{D}_m \subset \mathfrak{D}$ denote the observed and unobserved variables in the data set, respectively. To deal with missing data, the optimal method for learning probabilities is to compute the conditional density, $P(\Theta \mid \boldsymbol{D}_0)$, where the observed variables, $\boldsymbol{D}_o \subset \mathfrak{D}$, are incomplete. There is an exact Bayesian solution which is computationally intractable: the mixed densities of multinomial networks, $P(\Theta \mid \boldsymbol{D}_o)$, must be computed over every possible completion of the set of unobserved attributes, $\boldsymbol{D}_m$, across all joint samples which are incomplete.

Making the strong simplifying assumption that the missing data are independent of the observed data, three useful approximation techniques for parameter estimation can be considered:

1. *Expectation Maximization (EM) algorithm*, an iterative and deterministic algorithm;
2. *Gradient Ascent Training*; and
3. *Gibbs sampling algorithm*, a stochastic sampling technique.

Here, we concentrate on the EM algorithm because of its several advantages listed below [16]:

- It automatically takes care of parameter constraints.
- It can handle priors easily.
- It can handle deterministic constraints.
- It is simple to implement.

However, some limiting factors still remain for the EM algorithm.

- The estimated parameters may be the best locally, but may not be the best globally.
- Point estimates of the parameters are obtained rather than their probability distributions.

For incomplete data, the posterior distribution over the parameters of a model is no longer a product of independent posteriors. This implies that the log likelihood cannot be decomposed as in Equation 3.10. Through marginalization, the log likelihood can be written as:

$$L(\Theta) = \log P(\boldsymbol{D}_o \mid \Theta) = \log \sum_{\boldsymbol{D}_m} P(\boldsymbol{D}_o, \boldsymbol{D}_m \mid \Theta) \tag{3.11}$$

The basic idea behind EM is to apply Jensen's inequality[1] to Equation 3.11 to get a lower bound on the log likelihood, and then to iteratively maximize this lower bound [16]:

$$\begin{aligned} \log \sum_{\boldsymbol{D}_m} P(\boldsymbol{D}_o, \boldsymbol{D}_m \mid \Theta) &= \log \sum_{\boldsymbol{D}_m} Q(\boldsymbol{D}_m) \frac{P(\boldsymbol{D}_o, \boldsymbol{D}_m \mid \Theta)}{Q(\boldsymbol{D}_m)} \\ &\geq \sum_{\boldsymbol{D}_m} Q(\boldsymbol{D}_m) \log \frac{P(\boldsymbol{D}_o, \boldsymbol{D}_m \mid \Theta)}{Q(\boldsymbol{D}_m)} \\ &= \sum_{\boldsymbol{D}_m} Q(\boldsymbol{D}_m) P(\boldsymbol{D}_o, \boldsymbol{D}_m \mid \Theta) - \sum_{\boldsymbol{D}_m} Q(\boldsymbol{D}_m) \log Q(\boldsymbol{D}_m) \\ &= F(Q, \Theta) \end{aligned} \tag{3.12}$$

where Q is an arbitrary distribution over $\boldsymbol{D}_m$. Starting from some initial parameters, Θ_0, the EM algorithm alternates between maximizing F with respect to Q and Θ, respectively.

Algorithm 3.1. Maximum Likelihood EM

0. Set Θ_0 arbitrarily; select a desired degree of precision, ε, for $\hat{\Theta}_{k+1}$; while $|\hat{\Theta}_{k+1} - \hat{\Theta}_k| > \varepsilon$ complete the following two steps:
1. **Expectation step:** compute the probability distribution over missing values:

$$\underset{Q}{\arg\max}\ F(\hat{Q}, \hat{\Theta}_k) \Rightarrow \hat{Q}_{k+1}(\boldsymbol{D}_m) = P(\boldsymbol{D}_m \mid \boldsymbol{D}_o, \hat{\Theta}_k) \tag{3.13}$$

2. **Maximization step:** compute the new ML estimate, $\hat{\Theta}_{k+1}$, given $P(\boldsymbol{D}_m \mid \boldsymbol{D}_o, \hat{\Theta}_k)$;

$$\underset{\Theta}{\arg\max}\ F(\hat{Q}_{k+1}, \Theta) = \underset{\Theta}{\arg\max} \sum_{\boldsymbol{D}_m} P(\boldsymbol{D}_m \mid \boldsymbol{D}_o, \hat{\Theta}_k) \log P(\boldsymbol{D}_m, \boldsymbol{D}_o \mid \hat{\Theta}_{k+1}) \rightarrow \hat{\Theta}_{k+1} \tag{3.14}$$

[1] $f(\sum_j \lambda_j y_j) \geq \sum_j \lambda_j f(y_j)$

3.4 An illustrative example of the EM algorithm

In this section, a simple numerical example is presented to give more insight into the EM algorithm.[1] Suppose X reports the outcomes of a soft sensor which is used to classify the quality of a chemical product. The output of the sensor is divided multinomially into four categories, so that the observed data consist of:

$$Y = (y_1, y_2, y_3, y_4) = (40, 340, 85, 35) \tag{3.15}$$

To simplify the problem, we also assume that the following model represents the probabilities of each zone:

$$\{P_1 = \frac{1}{4}(1-\theta), P_2 = \frac{1}{2}+\frac{1}{4}\theta, P_3 = \frac{1}{4}\theta, P_4 = \frac{1}{4}(1-\theta)\} \text{ for } 0 \le \theta \le 1; \tag{3.16}$$

Therefore, the ML parameter of this model is obtained by maximizing the likelihood function of Equation 3.17:

$$P(Y \mid \theta) = \frac{(y_1+y_2+y_3+y_4)!}{y_1!y_2!y_3!y_4!}(\frac{1}{4}(1-\theta))^{y_1}(\frac{1}{2}+\frac{1}{4}\theta)^{y_2}(\frac{1}{4}\theta)^{y_3}(\frac{1}{4}(1-\theta))^{y_4} \tag{3.17}$$

If we split the second zone into two different quality zones, ***Y*** represents an incomplete data set from a five-category multinomial distribution. The new data set is specified as:

$$X = (x_1, x_2, x_3, x_4, x_5) \tag{3.18}$$

where,

- (x_1, x_4, x_5) are known to be (y_1, y_3, y_4), and
- (x_2, x_3) need to be estimated from $y_2 = x_2 + x_3$

The posterior distribution over the parameters of the new model, represented by Equation 3.19, cannot be calculated as before, because:

$$P(X \mid \theta) = \frac{(x_1+x_2+x_3+x_4+x_5)!}{x_1!x_2!x_3!x_4!x_5!}(\frac{1}{4}(1-\theta))^{x_1}(\frac{1}{2})^{x_2}(\frac{1}{4}\theta)^{x_3}(\frac{1}{4}\theta)^{x_4}(\frac{1}{4}(1-\theta))^{x_5} \tag{3.19}$$

[1] The idea of this example is taken from [5]

Table 3.1. Parameter estimation in a numerical example using EM algorithm

k	θ^{k+1}
0	0.500000
1	0.746622
2	0.782512
3	0.786467
4	0.786887
5	0.786932
6	0.786937
7	0.786937

This model is parameterized from incomplete data via the EM algorithm by alternating between the expectation and maximization steps:

0. Start from $\theta_0 = 0.5$ as an initial guess.
1. **Expectation step:** compute the probability distribution over missing values:

$$x_2^{k+1} = 340\left(\frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{4}\theta^k}\right) \text{ and } x_3^{k+1} = 340\left(\frac{\frac{1}{4}\theta^k}{\frac{1}{2}+\frac{1}{4}\theta^k}\right) \quad \textbf{(3.20)}$$

2. **Maximization step:** compute the new ML estimate, θ^{k+1}, given the estimated complete data $(40, x_2^{k+1}, x_3^{k+1}, 85, 35)$:

$$\theta^{K+1} = \frac{x_3^k + 85}{x_3^k + 40 + 85 + 35} \quad \textbf{(3.21)}$$

Table 3.1 gives the values for θ after each k iteration, for k=0,1,...,7.

3.5 Bayesian modeling for error in variable

The principle and applications of the EM algorithm have been explained in detail. In this section, we propose an EM-based Bayesian framework for estimating the variances of measurement noise and capturing measurement errors to advance parameter estimation. To do this, we will put forward some ideas for improvement of the EM algorithm, based on iteratively freezing some parameters. To start with, we will partition the variables into $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{X}^{\circ})$ to distinguish between measured and hidden variables of a model. Assume that both input and output measured vectors are arranged in the rows of the matrix, $\boldsymbol{X}=\{X_1,...,X_N\}$. The hidden or noise-free variables are denoted as $\boldsymbol{X}^{\circ}=\{X_1^{\circ},...,X_N^{\circ}\}$, and the measurement noises are given by $\boldsymbol{E}=\{e_1,...,e_N\}$. Thus, we have:

$$X_i = X_i^{\circ} + e_i \qquad \text{for } i = 1,...,N \tag{3.22}$$

The condition that $Par(X_i^{\circ}) \subseteq \{X_1^{\circ},...,X_{i-1}^{\circ}\}$ allows us to construct a Bayesian model for the error-in-variable application using Pearl's network-construction algorithm discussed in Chapter 1. While there are variables left:

1. Add the next variables, X_i and X_i°, to the network;
2. $Par(X_i) = X_i^{\circ}$; and then
3. Add arcs to the X_i° node from some minimal sets of nodes already in the network, $Par(X_i^{\circ})$, such that the following conditional independence property is satisfied: $P(X_i^{\circ} \mid X_1',...,X_{i-1}') = P(X_i^{\circ} \mid Par(X_i^{\circ}))$, where $X_1',...,X_{i-1}'$ are all the variables preceding X_i°, including $Par(X_i^{\circ})$.

To complete the definition of the model, we now need to define the conditional probability distributions (CPDs) for each of the nodes. A priori, the following standard assumptions about the underlying probability distributions are made:

$$e_i \sim N(0,\varepsilon_i) \tag{3.23}$$

$$Z_i \sim N(\mu_i + W_i Par(Z_i), \sigma_i^2) \tag{3.24}$$

where W is the regression matrix.

Combining Equations 3.23 and 3.24, the general form of CPDs for measured and hidden variables can be expressed as follows:

$$\begin{cases} X_i \sim N(x_i^{\circ}, \varepsilon_i) \\ X_i^{\circ} \sim N(\mu_i^{\circ} + W_i Par(X_i^{\circ}), \sigma_i^{\circ}) \end{cases} \qquad \textbf{(3.25)}$$

Taking the following iterative steps, the Bayesian framework for error-in-variable applications is formulated:

Algorithm 3.2.

0. Start from initial guesses over regression matrices $,W_1^{(0)}, W_2^{(0)}, ..., W_N^{(0)}$, and conditional mean values $\mu_1^{(0)}, \mu_2^{(0)}, ..., \mu_N^{(0)}$; Select a desired degree of precision, ε, for W_i and μ_i. While $|W_i^{(k+1)} - W_i^{(k)}| > \varepsilon$ and $|\mu_i^{(k+1)} - \mu_i^{(k)}| > \varepsilon$ do:
1. Freeze the regression matrices and conditional mean values; learn the updated model, using the EM algorithm to estimate the conditional variances of the noisy measured variables, $\varepsilon_i^{(k)}$.
2. Freeze the conditional variances of the noisy measured variables $\varepsilon_i^{(k)}$; learn the updated model, using the EM algorithm, to estimate the regression matrices, $W_i^{(k+1)}$, and conditional mean values, $\mu_i^{(k+1)}$.

3.6 Simulation example: linear and non-linear models

As an illustration, let's set out a simple network in which the model equation for noise-contaminated input and output variables is expressed as follows:

$$\begin{aligned} &X_i = X_i^{\circ} + e_i \qquad i = 1,2,3 \\ &X_3^{\circ} = a_0 + a_1 X_1^{\circ} + a_2 X_2^{\circ} \end{aligned} \qquad \textbf{(3.26)}$$

where,

- X_1 and X_2 are the measured input variables
- X_3 is the output variable
- X_1° , X_2°, and X_3° are the noise-free variables
- e_1, e_2, and e_3 are the measurement errors.

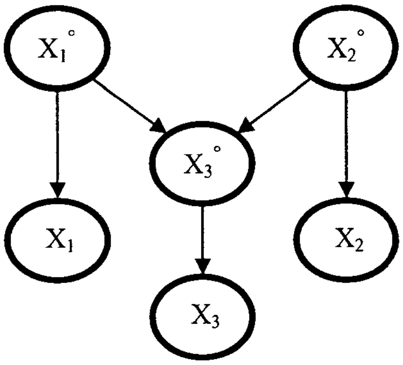

Figure 3.2. Graphical model of Equation 3.26

Table 3.2. Conditional probability parameters for the simulated linear model

Variable	Parents	Weight(W)	Mean	Variance
X_1^o *node*	-	-	$E(X_1)$	σ_1^o
X_2^o *node*	-	-	$E(X_2)$	σ_2^o
X_3^o *node*	X_1^o, X_2^o	$\{a_1, a_2\}$	a_0	~0
X_1 *node*	X_1^o	1	0	ε_1
X_2 *node*	X_2^o	1	0	ε_2
X_3 *node*	X_3^o	1	0	ε_3

This structure is captured in the graphical model of Figure 3.2. Conditional probability parameters for this model are discussed next and summarized in Table 3.2:

1. X_1^o *node*: The distribution on X_1^o is $N\left(E(X_1), \sigma_1^o\right)$.
2. X_1 *node*: With a continuous parent, the conditional distribution of X_1 on X_1^o is $X_1 \mid X_1^o = x_1^o \sim N\left(x_1^o, Var(e_1)\right)$.
3. X_2^o *node*: The distribution on X_2^o is $N\left(E(X_2), \sigma_2^o\right)$.
4. X_2 *node*: With a continuous parent, the conditional distribution of X_2 on X_2^o is $X_2 \mid X_2^o = x_2^o \sim N\left(x_2^o, Var(e_2)\right)$.

5. X_3^o *node*: With continuous parents, the conditional distribution of X_3^o on X_1^o and X_2^o is $X_3^o \mid X_1^o = x_1^o, X_2^o = x_2^o \sim N(\mu_3^o + w_{31}x_1^o + w_{32}x_2^o, \sigma_3^o)$.

6. X_3 *node*: With a continuous parent, the conditional distribution of X_3 on X_3^o is $X_3 \mid X_3^o = x_3^o \sim N\left(x_3^o, Var(e_3)\right)$.

To provide a numerical example, the discussed model was created for the following specified values:

$$\begin{array}{ll} X_1^o \sim N(0,1) & e_1 \sim N(0,(0.1)^2) \\ X_2^o \sim N(0,2) & e_2 \sim N(0,(0.15)^2) \\ X_3^o = 3 + 2X_1^o + X_2^o & e_3 \sim N(0,(0.2)^2) \end{array} \quad \textbf{(3.27)}$$

To evaluate the effect of the noise variances, simulation was repeated with $e_1 \sim N(0,(0.31)^2)$, $e_3 \sim N(0,(0.5)^2)$, and $e_2 \sim N(0,(0.63)^2)$. The deviations of the estimates from the true values are compared among the different estimation techniques:

1. classical regression,
2. error-in-variable (EIV), and
3. Bayesian inference.

The estimated model parameters and error variances are presented in Table 3.3 and Table 3.4. Mean squared error (MSE) is commonly used to indicate the integrated performance of accuracy and precision; therefore, the MSE of the estimates was evaluated and compared to assess the performance of the estimation techniques under consideration. Results from the numerical example that summarize the MSEs of the estimates are compared in Figure 3.3. As expected, the MSE of the estimated parameters from the Bayesian approach is lower than those from regression and EIV. Under low noise levels, the MSEs are comparable, and the advantage of the Bayesian algorithm diminishes; however, under high noise levels, the MSEs of Bayesian estimates are significantly lower than those of the others. For low noise data, estimatess can be obtained reliably from the measurements alone, and recovering the noise-free variables does not improve the estimates further; however, for high noise data, detecting the measurement noises can improve the estimates more significantly.

Table 3.3. Parameters of the simulated linear model

Noise level	*Method*	*Coefficients*			*Mean Squared Error*
		a_0	a_1	a_2	
Low	**True**	**3**	**2**	**1**	-
	Regression	2.9864	1.9906	0.9748	3.0261e-004
	EIV	3.0086	2.0074	0.9775	2.1163e-004
	Bayesian	2.9867	2.0068	0.9884	1.1923e-004
High	**True**	**3**	**2**	**1**	-
	Regression	3.0018	1.7915	0.8708	0.0200
	EIV	3.1972	1.8904	0.8954	0.0206
	Bayesian	2.9937	1.9034	0.9590	0.0037

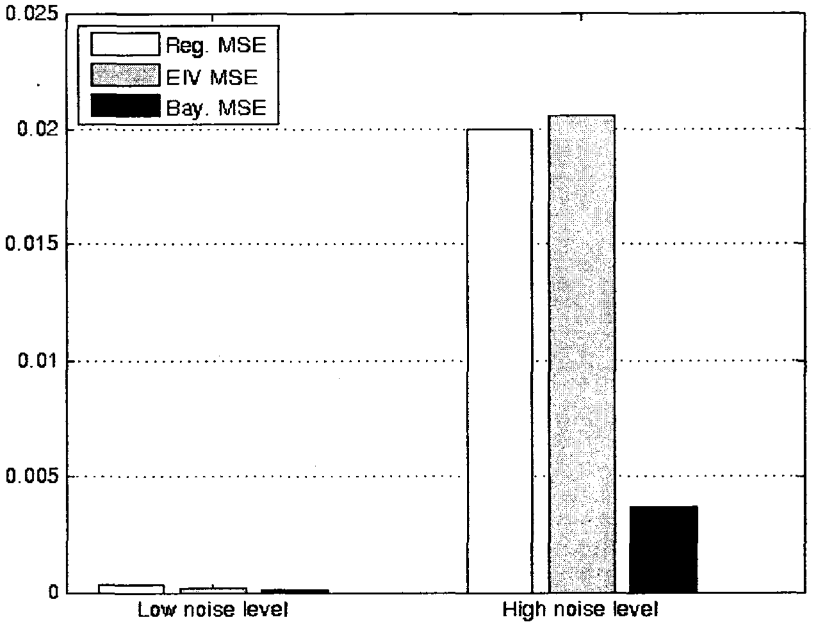

Figure 3.3. The MSEs of the estimates of the simulated linear model

Table 3.4. Conditional variances of each node in the simulated linear model

Noise level	*Method*	*Variance of* e_1	*Variance of* e_2	*Variance of* e_3
Low	**True**	0.01	0.025	0.04
	Bayesian	0.0096	0.0384	0.0406
High	**True**	0.1	0.25	0.4
	Bayesian	0.0776	0.2397	0.4667

Next, the non-linear parameter estimation problem is presented. We show how to modify this procedure to recover noise-free measured variables and estimate the parameters in a model of form 3.28:

$$\begin{cases} Y = a_0 + a_1 X^{a_2} \\ Z = b_0 + b_1 Y^{b_2} \end{cases} \qquad \textbf{(3.28)}$$

The Bayesian network representing these equations is plotted in Figure 3.4 Selecting parameter vectors as $a = \{a_0, a_1, a_2\} = \{5,1,2\}$ and $b = \{b_0, b_1, b_2\} = \{6,2,0.5\}$, our approach is evaluated through a simulated non-linear model. Since the EIV approach has been formulated only for linear relations between variables, we compare the Bayesian algorithm only with the classical least squares regression. Note that a novel EIV approach for non-linear parameter estimation using interval analysis is presented in [6], in which noise variances need to be assigned beforehand. An additional advantage of the presented Bayesian framework is that noise variances are also estimated from the measured variables.

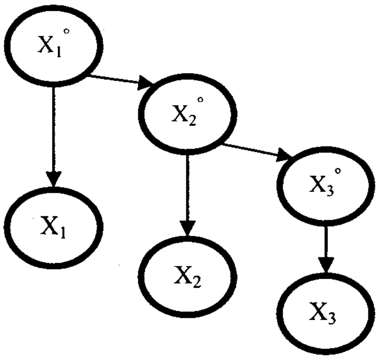

Figure 3.4. Graphical model of Equation 3.28

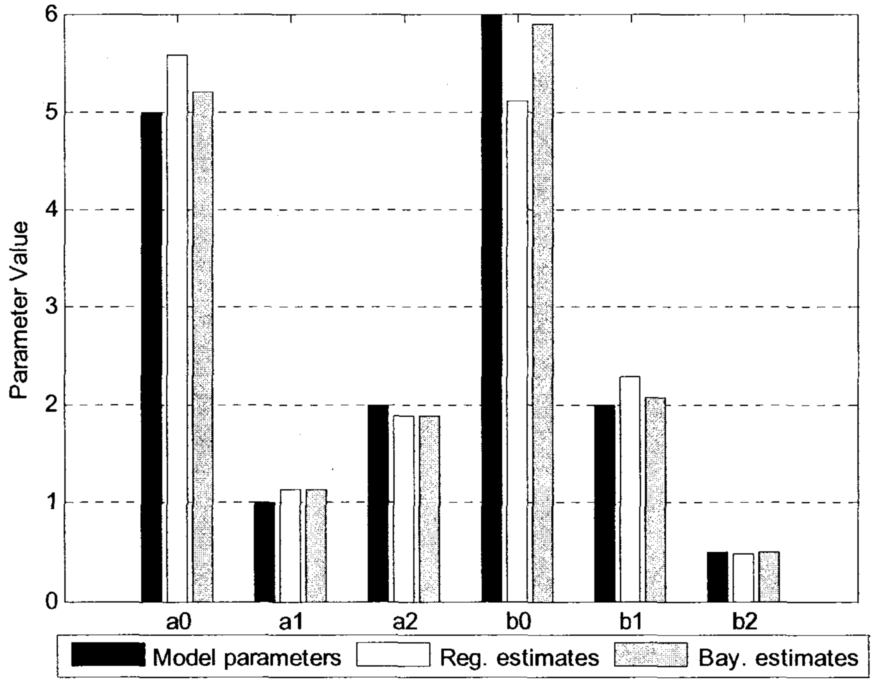

Figure 3.5. The estimates of the simulated non-linear model

As shown in Figure 3.5, the estimates using the Bayesian approach are more accurate than those using regression. This visual judgment is confirmed by the statistical comparison of MSEs presented in Table 3.5. This table indicates that the MSEs of the classical least squares regression are considerably larger than those of the Bayesian algorithm.

Table 3.5. Parameters of the simulated non-linear model

Method	*Coefficients*						*Mean Squared Error*
	a_0	a_1	a_2	b_0	b_1	b_2	
True	**5**	**1**	**2**	**6**	**2**	**0.5**	-
Regression	5.5836	1.1336	1.8855	5.1068	2.2799	0.4872	0.2080
Bayesian	5.2053	1.1415	1.8966	5.8888	2.0803	0.4910	0.01529

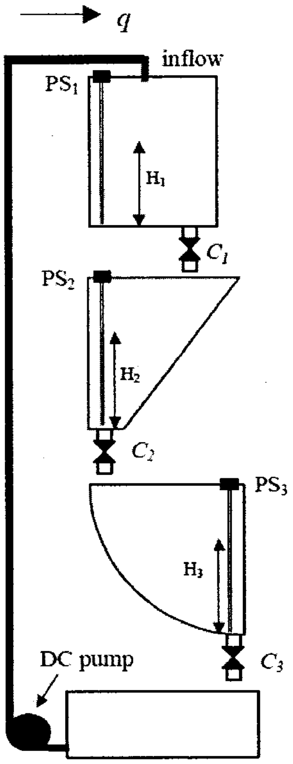

Figure 3.6. The configuration of the three-tank system

3.7 Experimental evaluation: three-tank system

Consider the multi-tank system shown in Figure 3.6 which consists of a number of tanks placed above each other. Some of the tanks have a constant cross section, while others are spherical or prismatic, therefore having a variable cross section. Liquid is pumped into the upper tank from the supply tank by the pump driven by a DC motor. The liquid outflows the tanks only as a result of gravity. The output orifices act as flow resistors.

The multi-tank system is related to liquid level control problems which commonly occur in industrial storage tanks. For example, steel-producing companies around the world have repeatedly confirmed that substantial benefits are gained from accurate mould level control in continuous bloom casting. Mould level oscillations tend to stir foreign particles and flux powder into molten metal, resulting in surface defects in the final product [8]. In order to control the liquid levels, it is very important to have accurate measurements of the variables in the system.

Assuming the laminar outflow of an "ideal fluid" for this three-tank system, the system of equations describing the process can be obtained as follows:

$$\begin{aligned} \frac{dV_1}{dt} &= q - C_1\sqrt{H_1} \\ \frac{dV_2}{dt} &= C_1\sqrt{H_1} - C_2\sqrt{H_2} \\ \frac{dV_3}{dt} &= C_2\sqrt{H_2} - C_3\sqrt{H_3} \end{aligned} \tag{3.29}$$

where,

- V_i: Fluid volume in the i^{th} tank, $i = 1,2,3$
- q: Inflow to the upper tank
- H_i: Fluid level in the i^{th} tank, $i = 1,2,3$
- C_i: Resistance of the output orifice of i^{th} tank

Liquid levels in the tanks are the state variables of the system. For the tank system there are four inputs: liquid inflow, q, and valve settings, C_1, C_2, C_3. Therefore, several models of the tanks system can be analyzed. Let's consider the steady state condition. The basic equation in this study is the mass balance equation:

$$q = C_1H_1^{\alpha_1} = C_2H_2^{\alpha_2} = C_3H_3^{\alpha_3} \tag{3.30}$$

Sometimes, turbulence and acceleration of the liquid in the tube should be taken into account; therefore the general flow coefficients (α_i's) are applied in Equation 3.30. In the case of laminar flows, as mentioned above, we assume $\alpha_i = 0.5$ according to Bernoulli law. It is, therefore, possible to transform the non-linear system to a linear one:

$$C_1^2H_1 = C_2^2H_2 = C_3^2H_3 \tag{3.31}$$

To formulate the relation between H_2 and the measured variables, Equation 3.31 can be rewritten as:

$$\begin{cases} H_{2,H_1} = a_1 + a_2H_1 \\ H_{2,H_3} = b_1 + b_2H_3 \end{cases} \tag{3.32}$$

The problem at hand involves parameter estimation of the simplified three-tank system, in which α_i 's have already been set to some constants. Our Bayesian framework is compared with the two other parameter estimation techniques on three-tank data as shown in Figure 3.7. Results that summarize the values and MSEs of the estimates of various estimation methods are presented in Table 3.6. As expected, the Bayesian approach shows better performance in estimating the parameters of model 3.32, which partly represents the three-tank system.

Finally, to demonstrate the efficiency of the EM-based Bayesian framework on a real world non-linear model, our algorithm is applied to Equation 3.30 for the task of estimating the parameters for the non-linear model. The whole set of parameters representing the three-tank setup are estimated using classical regression and the Bayesian approach. The estimates are compared with values provided by the vendor. In Table 3.7, the estimated parameters and mean squared relative error (MSRE) of each method are shown. Since the parameters of this system have different orders of magnitude, the MSREs of the estimates are calculated instead of their MSEs to provide a fair comparison. Table 3.7 indicates the Bayesian parameter estimation approach performed around 10% better than the classical regression, thus validating the effectiveness of our de-noising framework. These experimental results are consistent with previous simulation results.

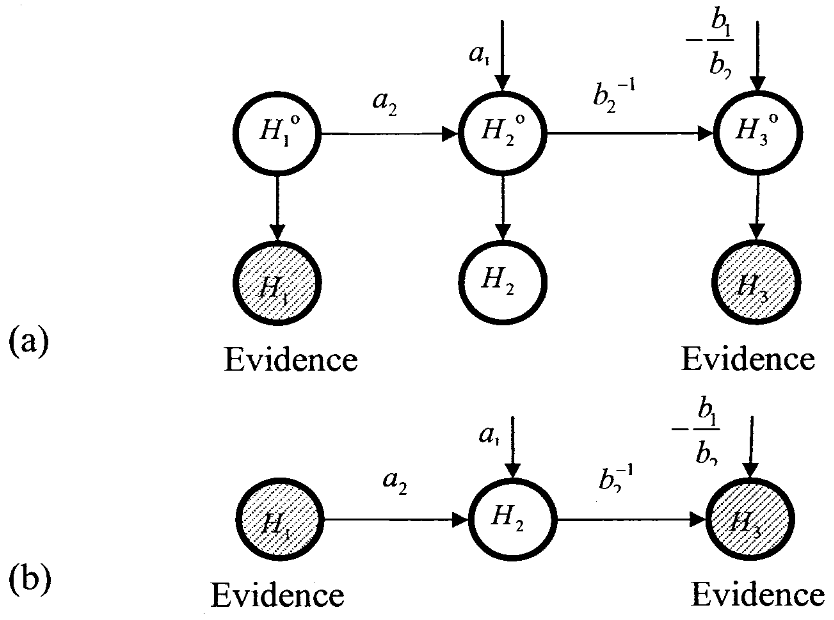

Figure 3.7. (a) Bayesian or EIV model of the three-tank system, (b) Regression model of the three-tank system

Table 3.6. Parameters of Equation 3.32

Method	*Coefficients*				*Mean Squared Error*
	a_1	a_2	b_1	b_2	
True	**0**	**0.7192**	**0**	**0.7049**	-
Regression	0.0529	0.6165	0.0537	0.4567	0.0195
EIV	0.0218	0.8807	0.0071	0.7537	0.0072
Bayesian	0.0358	0.7611	0.0048	0.7750	0.0020

Table 3.7. Parameters of the three-tank system

Method	*Parameters*						*Mean Squared Relative Error (%)*
	α_1	$C_1 \times 10^4$	α_2	$C_2 \times 10^4$	α_3	$C_3 \times 10^4$	
Vendor	**0.5**	**1.8307**	**0.5**	**2.1857**	**0.5**	**1.8125**	-
Regression	0.2637	1.3626	0.3101	1.4741	0.3329	1.4336	11.42
Bayesian	0.3438	1.6156	0.4896	2.2056	0.4156	1.7044	2.40

3.8 Conclusion

In this chapter, we have explored the problem of model parameter estimation from noisy input and output variables. Classical least squares regression and total least squares (TLS) have been discussed as traditional parameter estimation approaches. The principle and applications of the EM algorithm have been reviewed in detail. The EM-based Bayesian framework has been presented that is robust to measurement noise. This algorithm enables us to estimate the data noise variances and to recover noise-free variables. Numerical simulations and experimental examples have also been presented to demonstrate the efficiency of the techniques proposed. In the case of noisy data sets, results from the present study show that the Bayesian approach gives a better estimation of a model's parameters than the regression and classical EIV methods.

3.9 Bibliography

1. Bard Y., *Non-linear parameter estimation*, Academic Press, New York, 1974
2. Bishop C. M. and M. E. Tipping, *Bayesian Regression and Classification*, in Advances in Learning Theory: Methods, Models and Applications, J.A.K. Suykens *et al.* (Editors), IOS Press, NATO Science Series III: Computer and Systems Sciences, Vol. 190
3. Charniak E., *Bayesian Networks without Tears*, AI Magazine, pp. 50-63, Winter 1991
4. Chen B., S. C. Huang, R. A. Hawkins, and M. E. Phelps, *An Evaluation of Bayesian Regression for Estimating Cerebral Oxygen Utilization with Oxygen-15 and Dynamic PET*, IEEE Transaction on Medical Imaging, Vol. 7, No. 4, December 1988
5. Dempster A. P., N. M. Laird, and D. B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society Series B, Vol. 39, pp. 1-38, 1977
6. Gau C. Y. and M. A. Stadtherr, *Reliable Nonlinear Parameter Estimation Using Interval Analysis: Error-in-Variable Approach*, PSE 2000, Keystone, CO, July 2000
7. Ghahramani Z., *Learning Bayesian Networks*, Lecture Notes on Artificial Intelligence, Department of Computer Science, University of Toronto, October 1997
8. Graebe S. F. and G.C. Goodwin., *Control Design and Implementation in Continuous Steel Casting*, IEEE Control Systems, pp. 64-71, August 1995
9. Helsper E. L. and C. van der Gaag., *Building Bayesian Networks through Ontologies*, Proceedings of the 15th Eureopean Conference on Artificial Intelligence (ECAI 2002), pp. 680-684, 2002
10. Huang B., *Detection of Abrupt Changes of Total Least Squares Models and Application in Fault Detection*, IEEE Transactions on Control Systems Technology, Vol. 9, No. 2, March 2001
11. Huffel S. V. and J. Vandewalle, *The Total Least Squares Problem : Computational Aspects and Analysis*, Society of Industrial and Applied Mathematics, Philadelphia, 1991
12. Jo S. and S. W. Kim, *Consistent Normalized Least Mean Square Filtering with Noisy Data Matrix*, IEEE Transactions on Signal Processing, Vol. 53, No. 6, June 2005
13. Koller D., N. Friedman, *Inference with Bayesian Networks*, Lecture Notes, Stanford University, November 2000

14. Korb K. B. and A. E. Nicholson, *Bayesian artificial intelligence*, Chapman & Hall/CRC, 2004
15. Lerner U. N., *Hybrid Bayesian Networks for Reasoning about Complex Systems*, Ph.D. Thesis, Department of Computer Science, University of Stanford, October 2002
16. Murphy K. P., *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. Thesis, Department of Computer Science, University of California, Berkely, Fall 2002
17. Murphy K. P., *How to Use Bayes Net Toolbox,* Tutorial of BNT Toolbox, Department of Computer Science, University of California, Berkley, June 2004
18. Schuermans S., I. Markovsky, P. D. Wentzell, and S. V. Huffel, *On the Equivalence between Total Least Squares and Maximum Likelihood PCA*, Analytica Chimica Acta, pp. 254-267, January 2005
19. Ting J. A., A. D'Souza, and S. Schaal, *Bayesian Regression with Input Noise for High Dimensional Data*, Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, 2006

Chapter 4

Bayesian Inference for Outliers and Incomplete Data

The objective of this chapter is to study the practical aspects of building and testing soft sensors, such as dealing with outliers and incomplete data. The concept of the Bayesian inference is introduced as a novel approach for soft sensor development, which can cope with these issues. The ordinary least squares (OLS), partial least squares (PLS), and robust regression are presented as other soft sensor models, ones which are commonly used in industrial applications.

Reviewing *deletion* and *imputation* methods, we show how OLS and PLS regression techniques deal with incomplete data sets. A brief introduction to existing outliers detection methods such as robust regression and box plot construction criteria are provided as well. Through comparison of the parameter estimates, the Bayesian and regression approaches are evaluated for their treatment of outliers, missing values, and non-normal measurement errors. The feasibility and performance of Bayesian approach is proven by demonstrating how well it can handle practical issues associated with various simulation, experimental and industrial cases.

4.1 Introduction

First Principle Modeling (FPM) is the preferred methodology for deriving a soft sensor; however, the main drawback of this approach is that it requires an exact knowledge of the process as well as accurate process data in order to calibrate the model's coefficients. As a result, the most commonly used strategy for soft sensor development is to combine process knowledge and data fitting techniques to obtain empirical models. Statistical methods applied in soft sensor development can be broken into two main categories:

1. OLS regression, and

2. Multivariate statistical techniques.

OLS regression is a simple and straightforward approach to explaining the relationship between variables. If the model's error term is normally, independently and identically distributed, OLS yields the most efficient unbiased estimators for the model's coefficients. It suffers from numerical problems, however when the input variables are strongly correlated. As a result, multivariate statistical methods such as partial least squares (PLS) [12,23] have attracted wide interest. PLS is a regression technique applied to linear polynomial models to simultaneously explain variations in both input and output variables and to maximize their covariance. This approach has been used for many industrial applications in chemical engineering, such as estimating distillation compositions and polymer quality variables [18,22].

Typically, a soft sensor model is built to make predictions for future cases in which only the inputs are known. Usually, the results of conventional model training techniques are condensed into a single set of weights that can be used for making further predictions. For this reason, most of the data-driven soft sensor approaches require complete data samples in order to work or produce valid results; however, it is common to have blanks in industrial data [17,22], because of sensor failure or different sensor acquisition rates. Another issue related to soft sensors is outliers [15,22]. Outliers, which can be simply regarded as the data points that are located far from the rest of the data, are almost the same kind of problem as missing values and could be worse if not detected and deleted. Outliers occur frequently in industrial data sets and are harmful to the soft sensor models derived by regressions.

There are a variety of methods for handling incomplete and inconsistent data [2,3,15], but many of them are problem specific. Applying the Bayesian approach to soft sensors enables us to improve prediction results and overcome the problem of outliers and missing values in real world data. In contrast to classical statistical techniques, Bayesian modeling results in a posterior distribution over network weights [1,21]. If the input variables are set at the values for some new cases, the posterior distributions of model weights will give rise to a distribution over the output of the model which is known as the predictive distribution for this new case. Although the mean of the predictive distribution could be used for a single-valued prediction, the full predictive distribution reveals how uncertain this prediction would be. Additionally, Bayesian models handle incomplete data sets without difficulty because the dependencies among all variables are discovered through modeling [5,7,12,16].

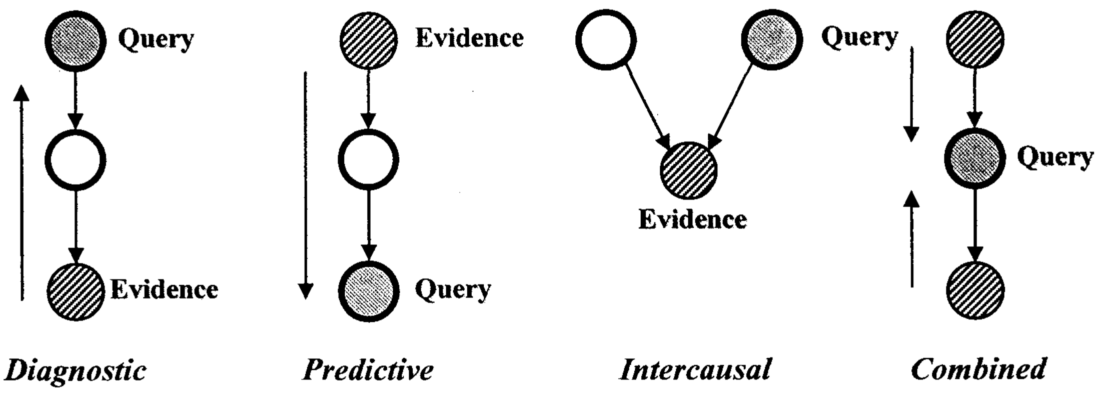

Figure 4.1. Types of reasoning

This chapter begins with an introduction to Bayesian inference theory and algorithms. Next, existing treatments for missing values and outliers are reviewed briefly. We then propose the Bayesian method as a novel approach for general problems of modeling and prediction in soft sensors. Following this, practical issues in soft sensor development are considered, and treatments of outliers and missing values are illustrated through simulations. Finally, the effectiveness of Bayesian techniques in comparison with OLS and PLS is verified for the industrial and experimental case studies presented in the previous chapters.

4.2 Bayesian inference

The most common task using Bayesian networks is probabilistic inference followed by computing the posterior probability distribution for a set of query nodes, given values for some evidence nodes. The different types of reasoning that can be performed by Bayesian inference are shown in Figure 4.1 and outlined below.

1. *Diagnostic reasoning:* Moving in the opposite direction of the network arcs, this typically infers causes of problems from past evidences and symptoms.
2. *Predictive reasoning:* Following the direction of the network arcs, this forecasts future beliefs about effects based on new information about causes.
3. *Intercausal reasoning:* When there are exactly two possible causes for a particular effect, represented by a v-structure in the BN, with knowledge of the effect the presence of one explanatory cause renders an alternative cause less likely.

4. *Combined reasoning*: Performing diagnostic and predictive reasonings simultaneously, this predicts a query node whenever both its parents and children are observed.

Inference, or model evaluation, is the process of updating probabilities of outcomes based on model parameters and the values for the measured variables. Assume a training data set of independent and identically-distributed observations vectors, $\mathcal{D} = \{\boldsymbol{X}_1, ..., \boldsymbol{X}_N\}$. Once we have observed a set of evidence variables, X_E, Bayesian inference uses the predictive distribution to predict the variables of interest, X_Q:

$$P(X_Q \mid X_E, \mathcal{D}) = \int P\left(X_Q \mid \Theta, X_E\right) P\left(\Theta \mid X_E, \mathcal{D}\right) d\Theta \tag{4.1}$$

To illustrate this inference problem, we consider the relatively simple, but widely studied, problems of linear regression for independent and identically-distributed data. For simplicity, we shall consider a single output system as follows:

$$\boldsymbol{X}_N = f\left(\boldsymbol{X}_n; \Theta\right) + \varepsilon \qquad \text{for } n = 1, ..., N-1 \tag{4.2}$$

$$f\left(X; \Theta\right) = \sum_{i=1}^{M} \theta_i \, \phi_i\left(X\right) = \Theta^T \Phi\left(X\right) \tag{4.3}$$

where $\varepsilon : N\left(0, \sigma^2\right)$ is the independent and identically-distributed noise, $\Theta = \left(\theta_1, ..., \theta_M\right)$ is the set of parameters, and $\Phi\left(X\right) = \left(\phi_1\left(X\right), ..., \phi_M\left(X\right)\right)$ are the fixed basic non-linear functions.

Least square regression is the classical approach to obtaining single-valued estimates for the parameters that minimize the error function defined by:

$$E\left(\Theta\right) = \frac{1}{2} \sum_{n=1}^{N-1} \left| f\left(\boldsymbol{X}_n; \Theta\right) - \boldsymbol{X}_N \right|^2 \tag{4.4}$$

To predict new values for $\boldsymbol{X}^* = \left\{X_1^{new}, ..., X_{N-1}^{new}\right\}$, the minimizer of error function, Θ_{LS}, is used to evaluate $f\left(\boldsymbol{X}^*; \Theta_{LS}\right)$.

The maximum likelihood estimation is another parameter estimation technique which maximizes $P\left(\boldsymbol{X}_N \mid \mathcal{D}, \Theta, \sigma^2\right)$. If we assume a normal distribution for the noise term, $\varepsilon_n : N(0, \sigma^2)$, the likelihood of all the data is given as:

$$P\left(X_N \mid \mathcal{D},\Theta,\sigma^2\right)=\prod_{n=1}^{N-1}P\left(X_N \mid X_n,\Theta,\sigma^2\right)$$

$$=\prod_{n=1}^{N-1}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{\{f(X_n;\Theta)-X_N\}^2}{2\sigma^2}\right] \quad \textbf{(4.5)}$$

Thus, the negative logarithm of the likelihood is:

$$-\log P\left(X_N \mid \mathcal{D},\Theta,\sigma^2\right)=\frac{N-1}{2}\log\left(2\pi\sigma^2\right)+\frac{1}{2\sigma^2}\sum_{n=1}^{N-1}\{f(X_n;\Theta)-X_N\}^2 \quad \textbf{(4.6)}$$

Since the first term on the right hand side of Equation 4.6 is independent of Θ, minimizing the negative logarithm of the likelihood function is equivalent to minimizing the squared error function in the case of a normally distributed error.

Taking the prior distribution of parameters into account, MAP Bayesian estimates the most probable values for Θ under the posterior distribution $P\left(\Theta \mid \mathcal{D},\sigma^2\right)$. In terms of prediction, MAP Bayesian makes use of Θ_{MAP} to determine $P\left(X_N^{new} \mid X^*,\Theta_{MAP},\sigma^2\right)$. Although this method gives a probability distribution for X_N^{new}, it has been shown in [21] that the mean of this distribution is identical to the solutions of the penalized least square approach. The classical regression and MAP Bayesian predictions do not take into consideration the uncertainty over the parameters; however, in the predictive distribution for X_N^{new} obtained from the true MAP Bayesian method the uncertain variables, Θ, are integrated out:

$$P(X_N^{new} \mid X^*,\mathcal{D},\sigma^2)=\int P\left(X_N^{new} \mid \Theta,X^*,\sigma^2\right)P\left(\Theta \mid X^*,\mathcal{D},\sigma^2\right)d\Theta \quad \textbf{(4.7)}$$

To conclude this section, we summarize the steps considered necessary in Bayesian inference in order to make a prediction:

1. Specifying the prior distribution of parameters,
2. Computing the data likelihood,
3. Computing the posterior distribution of parameters having observed the training data set, and
4. Computing the predictive distribution which will marginalize any uncertain parameters.

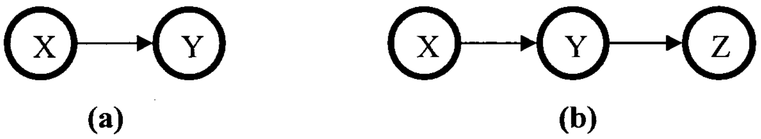

Figure 4.2. (a) A two-node chain, (b) A three-node chain

4.3 Inference algorithms

Regardless of the role of an inference system, two types of inference algorithms are applied:

- Exact inference algorithms: By now, few algorithms have been designed to give an exact answer to a probabilistic query.
- Approximate inference algorithms: For complex networks in which exact inference becomes computationally infeasible, approximate inference algorithms must be used.

Both exact and approximate inference algorithms are computationally complex and their computation time depends on different factors such as the structure of the network including how highly connected it is, how many undirected loops there are, and the locations of evidence and query nodes.

This section goes on to present brief descriptions of the main idea behind some of the viable exact and approximate inference algorithms; however, complete discussion of these is beyond the scope of this chapter.

4.3.1 Exact inference

We begin by considering a very simple inference task in two-node chains. Predictive and diagnostic reasoning is represented by the following computation:

1. If we have evidence about the parent node, $X = x$, $P(Y \mid X = x)$ is obtained straight from the value in CPT.
2. If we have evidence about the child node, $Y = y$, a simple application of the Bayes' theorem can be used:

$$P(X = x \mid Y = y) = \frac{P(Y = y \mid X = x)P(X = x)}{P(Y = y)} = \alpha P(x)\lambda(x) \qquad \textbf{(4.8)}$$

where, $\alpha = \dfrac{1}{P(Y=y)}$ is a normalizing constant and $\lambda(x) = P(Y=y \mid X=x)$ is the likelihood.

Applying the same method, the inference task in three-node chains is performed as follows:

1. If we have evidence about the parent node, $X = x$, the simple chain rule based on independencies implied in the network can be applied:

$$P(Z \mid X=x) = \sum_{Y=y} P(Z \mid Y)P(Y \mid X=x) \tag{4.9}$$

2. If we have evidence about the child node, $Z = z$, the combination of Bayes' theorem and the chain rule is applied to obtain:

$$\begin{aligned} P(X=x \mid Z=z) &= \frac{P(Z=z \mid X=x)P(X=x)}{P(Z=z)} \\ &= \frac{\sum_{Y=y} P(Z=z \mid Y=y, X=x)P(Y=y \mid X=x)P(X=x)}{P(Z=z)} \\ &= \frac{\sum_{Y=y} P(Z=z \mid Y=y)P(Y=y \mid X=x)P(X=x)}{P(Z=z)} \\ &= \alpha P(x)\lambda(x) \end{aligned} \tag{4.10}$$

Now, let us take a look at the slightly more complex model demonstrated in Figure 4.3. This network represents a polytree in which we have at most one path between any pair of nodes. Assuming X is the query node and $\mathbf{E}$ is some set of evidence nodes, the task is to update the conditional probability of X by computing $P(X \mid \mathbf{E})$.

Kim and Pearl's message passing algorithm [12] is one of the appropriate ones applied to these types of singly-connected networks. This algorithm requires different types of parameters to be maintained and then used to do local belief updating in the following three steps:

1. Belief updating,
2. Bottom-up propagation, and
3. Top-down propagation.

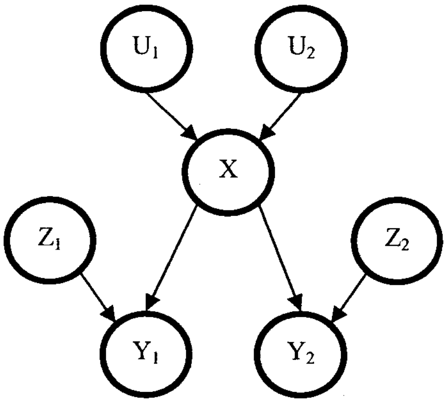

Figure 4.3. A generic polytree

One of the biggest advantages of Bayesian networks is that they have a bidirectional message passing architecture. Since Bayesian networks pass data between nodes and note the expectations from the world model, they can be considered bi-directional systems [9].

In the most general case, the Bayesian network has a multiply-connected structure. In multiply-connected networks, at least two nodes are connected by more than one path in the underlying graph; therefore, the message passing algorithm for polytrees does not work. The clustering method [12,16] is one of the most famous methods for dealing with this kind of problem. Clustering inference algorithms perform inference in two stages:

1. Transforming the Bayesian network into a probabilistically equivalent polytree by merging nodes and removing the multiple paths between two nodes as shown in Figure 4.4, and
2. Performing belief updating on that polytree.

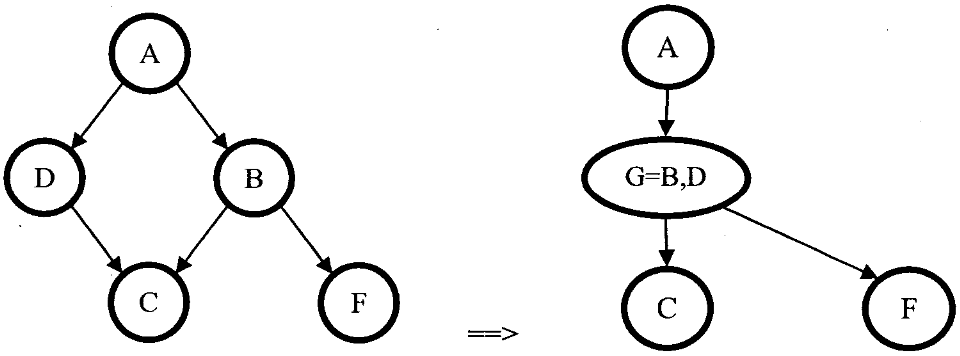

Figure 4.4. Ad hoc clustering of a multiply-connected Bayesian network

4.3.2 Approximate inference

Despite its benefits, the exact Bayesian inference also has drawbacks. One drawback is the difficulty of obtaining accurate conditional probabilities for large or densely connected networks. Because of this, approximate algorithms must be used. Stochastic simulation is one of the approaches developed to approximate inference for multiply-connected networks. In order to estimate the posterior probability of a query node, stochastic simulation uses a network to generate a large number of cases from its distribution. Logic sampling (LS), likelihood weighting (LW), and Markov Chain Monte Carlo (MCMC) are different types of sampling algorithms. The detailed discussion of exact and approximate inference can be found in [12,16].

4.4 Dealing with missing values

4.4.1 Nature of missing data

$\mathcal{D} = \{X_1, ..., X_N\}$ denotes a data set in which some of the values are missing. The observed and missing parts of $\mathcal{D}$ are represented by $\boldsymbol{D}_o$ and $\boldsymbol{D}_m$, respectively. For any data set, a matrix, $\mathcal{M} = \{M_{il}\}$, indicates whether D_i^l is observed, $M_{il} = 1$, or missing, $M_{il} = 0$. The missing data mechanism is described by the conditional distribution, $P(\mathcal{M} \mid \mathcal{D}, \Theta)$, where Θ denotes a set of parameters. Based on different conditionality, several mechanisms of missing data have been defined by [19]:

1. *Missing Completely At Random* (MCAR): In this case, the probability that data is missing does not depend on any part of $\mathcal{D}$:

$$P(\mathcal{M} \mid \mathcal{D}, \Theta) = P(\mathcal{M} \mid \Theta) \quad \textbf{(4.11)}$$

   For example, sensor failure results in missing data, yet values of observed variables do not fall into a discernible pattern.

2. *Missing At Random* (MAR): Often data are not missing completely at random, but they may be classified as missing at random. In MAR, the probability that data is missing depends on observed data:

$$P(\mathcal{M} \mid \mathcal{D}, \Theta) = P(\mathcal{M} \mid \boldsymbol{D}_o, \Theta) \quad \textbf{(4.12)}$$

In the process industries, sometimes measuring quality variables is costly or time-consuming. When this is the case, process variables are monitored through the regularly-measured condition variables. That is, quality variables are measured when condition variables indicate the process is drifting away from the sphere of normal operations. Thus, the missingness of the quality variables is determined by the observed values for condition variables.

3. *Not Missing At Random* (NMAR): In this case, the missingness depends on both observed and missing data. This happens, for instance, when we collect data with a sensor which is not able to detect values over a particular threshold.

Since observed data include all information necessary in order to estimate missing data distribution in MCAR and MAR cases, the missingness is ignorable.

4.4.2 Existing treatments for missing data

There are two different ways of handling missing data. The most common approach is to simply exclude the cases with missing values from the analysis. If we do not want to lose data and, perhaps, information, we may try to guess at missing items. The second approach is generally called *imputation*. Six techniques belonging to these classes are presented below.

1. *Casewise deletion*: We select only cases that do not contain any missing values for any of the variables. Under the MCAR assumption, this method leads to unbiased estimates of parameters. Nevertheless, a lot of non-missing data will be thrown out resulting in losing pieces of informative data.
2. *Mean substitution*: A natural method of imputation is to replace all missing values in a variable by the mean of that variable. Although mean substitution preserves the sample size, it may considerably change the values of variance, correlations and regression coefficients. This method assumes MCAR missingness mechanism.
3. *The LOCF method*: In the last observation carried forward (LOCF) method, the last measured observation before the missing one is imputed. This approach is applicable only to situations in which measurements are expected to be constant over time.
4. *Regression imputation*: A regression model is built to predict the missing value based on complete cases. This approach is less likely to produces bias, but may still underestimate variance. The regression imputation assumes the data are MAR.

5. *NIPALS algorithm*: Consider a data matrix, **X**, having the structure $\mathbf{X} = \mathbf{TP}'$, where $\mathbf{T}$ is a matrix of scores and **P** is a matrix of loadings. Treatment of missing values with PLS-NIPALS can be implicitly considered as a simple imputation method, in which PLS loadings and components are iteratively calculated as slopes of least squares lines passing through the origin of the available data. It is equivalent to setting the residuals for all missing elements in the least square objective function to zero in each iteration step.
6. *The EM algorithm*: The EM is a well-known algorithm for estimating the parameters of the probability density function of an incomplete sample. At each iteration step, the missing values are replaced by the expected values from the conditional normal distribution given the present data and the current estimates of the means and covariances. To illustrate: consider having to estimate $Y = aX + b$ and then use X to estimate Y wherever it is missing. We would first take estimates of the variance, covariance and mean, perhaps from casewise deletion. We would then use these estimated parameters to solve for the regression coefficients a and b. Having filled in missing data with these estimates, we would then use the complete data to recalculate the regression coefficients. We would continue this process until the parameters converged.

4.5 Dealing with outliers

Outliers are observations far from most others in a set of data. Typically, these observations represent a random error that we would like to be able to control to obviate the possible harmful effects of outliers. That is why the detection of outliers in pre-processing, or applying robust methods of parameter estimation which are not sensitive to the presence of outliers, are practical issues in soft sensor development. Some possible approaches to dealing with outliers are listed below.

1. *Box plot*: The box plot is a helpful graphical tool for determining how severe any outlying observations are. A box plot is constructed by drawing a box between the upper and lower quartiles with a solid line drawn across the box to locate the median. Defining $Q_{0.25}$ and $Q_{0.75}$ to be the first and third quartiles, outlier detection criteria are characterized by the following fences:

- Lower inner fence: $Q_{0.25} - 1.5 \times (Q_{0.75} - Q_{0.25})$
- Upper inner fence: $Q_{0.75} - 1.5 \times (Q_{0.75} - Q_{0.25})$
- Lower outer fence: $Q_{0.25} - 3 \times (Q_{0.75} - Q_{0.25})$
- Upper outer fence: $Q_{0.75} - 3 \times (Q_{0.75} - Q_{0.25})$

By definition, a *mild outlier* is a point beyond an inner fence on either side, while an extreme outlier is a point beyond an outer fence. Having detected the outliers, data transformation or deletion is applied to soften their impact.

2. *Robust regression*: In the OLS regression, it is assumed that the noise term is i.i.d. normal (identically and independently distributed normally). This assumption about the noise term is not correct, because the presence of outliers causes longer and fatter tails. Robust regression is a regression technique designed to limit the effects of outliers. The idea is to assign a weight to each observation so that outliers are given reduced weight. The most common method of robust regression is M-estimation, introduced by [10]. This solution is called the iteratively reweighted least squares (IRLS), because the estimated parameters depend upon the weights, while the weights depend upon the residuals, and the residuals depend upon the estimated parameters.
3. *Bayesian inference*: A Bayesian framework for error in variables is constructed and applied iteratively to predict the noise-free variables. Algorithm 4.1 shows in detail how the Bayesian approach deals with outliers in the training data set.

Algorithm 4.1.

0. Start with a set of measured variables, $\mathcal{D} = \{X_1, ..., X_N\}$, to learn the parameter of the Bayesian model.
1. **Bayesian diagnostic inference**: Predict noise-free variables from the identified model.
2. **Bayesian learning:** Determine the parameters of the Bayesian model from the new training data set $\mathcal{D} = \{\hat{X}_I^{\circ}, ..., \hat{X}_N^{\circ}\}$.

Steps 1 and 2 are repeated until parameters of the model converge.

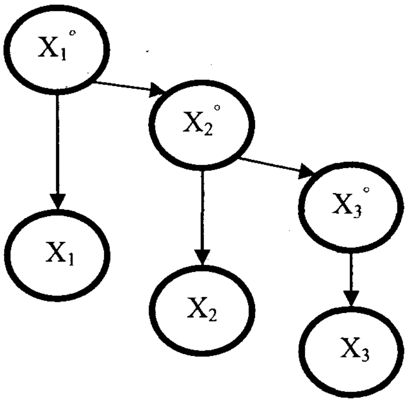

Figure 4.5. The Bayesian EIV framework for a sequential model

4.6 Bayesian approach for soft sensor development

The main objective in soft sensor development is to find a model that gives the best prediction in the application in which it will be implemented. Since the success of an empirical model depends totally on the quality of the process data, the pretreatment of data is crucial. Thus, noise reduction, outliers detection, and missing values treatment need to be considered in modeling and implementing soft sensors.

Chapter 3 discussed the problem of estimating model parameters for noisy data and presented an EM-based Bayesian framework for error-in-variable problem. In this chapter, Bayesian learning and Bayesian inference are combined to outline a new way of deriving and employing soft sensor models. This approach will be illustrated through its application to simulated sequential and multiple linear models.

4.7 Simulation example 1: sequential model

Assume that we want to develop a soft sensor corresponding to the model in Figure 4.5 in order to predict X_2 whenever a set of measured inputs, $\{X_1, X_3\}$, is available. As we have already noted in Chapter 3, X_i, X_i^{o} are measured and noise-free variables. For simulation purposes, the associated parameters are specified as follows:

$$\begin{aligned} X_i &= X_i^{\circ} + e_i & e_1 &\sim N(0,(0.31)^2) \\ X_2^{\circ} &= a_1 X_1^{\circ} + a_2 & e_2 &\sim N(0,(0.31)^2) \\ X_3^{\circ} &= b_1 X_2^{\circ} + b_2 & e_3 &\sim N(0,(0.5)^2) \end{aligned} \quad \textbf{(4.13)}$$

To compare the Bayesian approach with the OLS regression technique, we first investigate how well the former estimates the model parameters, and then we evaluate the prediction results. Here, four methods are applied to the soft sensor problem.

1. *Ordinary least square (OLS) regression*: The system is represented by two sequential models as follows:

   $$\hat{X}_{2,X_1} = \hat{a}_1 X_1 + \hat{a}_2 \quad \textbf{(4.14)}$$

   $$\hat{X}_{2,X_3} = \frac{1}{\hat{b}_1} X_3 - \frac{\hat{b}_2}{\hat{b}_1} \quad \textbf{(4.15)}$$

   Each of these models takes a portion of the available measurements into account to estimate the entire set of parameters and predict the output.

2. *Partial least square (PLS) regression*: In order to make use of all information on hand in predicting the output variable, the PLS approach is used to model the system in a linear polynomial way for prediction purposes:

   $$\bar{X}_{2,PLS} = c_1 \bar{X}_1 + c_2 \bar{X}_3 \quad \textbf{(4.16)}$$

   Both one-component and two-component PLS models are investigated in this example. Unless otherwise noted, it is assumed that data is either mean centered or autoscaled prior to analysis in PLS.

3. *The EIV technique*: Once again, parameters for Equations 4.14 and 4.15 are estimated using EIV. Since EIV models are useful only when the primary goal is model parameter estimation rather than prediction, we do not consider them in our prediction performance comparison.

4. *The Bayesian method*: The EM algorithm is applied to building a Bayesian network representing the model's structure in a flexible framework. Then Bayesian inference is used to infer about the output variables, given new observations for inputs.

Using these methods, the parameters of this model are estimated and presented in Table 4.1. The mean square errors of estimates are also given in Figure 4.6. Comparison of MSEs tells us that the Bayesian model fits the training data much better than the other models do.

Table 4.1. Estimated parameters for three-variable network example

Method	Coefficients				*Mean Square Error*
	a_1	a_2	b_1	b_2	
Actual	**1**	**1**	**2**	**0**	-
OLS	0.9095	1.0013	2.1076	-0.135	0.0095
EIV	0.9577	1.0681	2.0646	0.0977	0.0050
Bayesian Inference	0.9825	0.9988	1.9810	0.0003	1.6719e-004

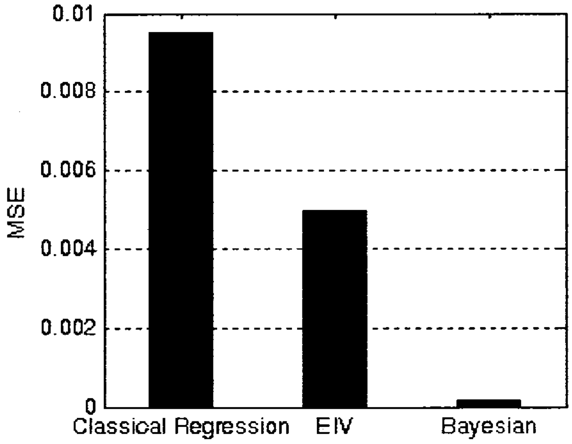

Figure 4.6. The MSEs of the estimates of the simulated sequential model

Although X_2 is assumed to have been observed during model identification, for soft sensor applications the derived model is used to predict X_2^o given only the measurements for X_1 and X_3. $\hat{X}_{2,X_1}$ and $\hat{X}_{2,X_3}$ result from OLS, $\hat{X}_{2,PLS1}$ from one-component PLS, $\hat{X}_{2,PLS2}$ from two-component PLS and $\hat{X}_2^o$ from Bayesian inference; all are presented in Figure 4.7 and compared in Table 4.2. Since the classical regresion models represent only parts of the system, they do not make use of all the available data. Hence, their prediction performances are poorer than those of the two-component PLS and Bayesian models which take into account all the variables. According to Table 4.2, the solution given by the two-component PLS is the same as the solution given by the Bayesian model. These results are not surprising,

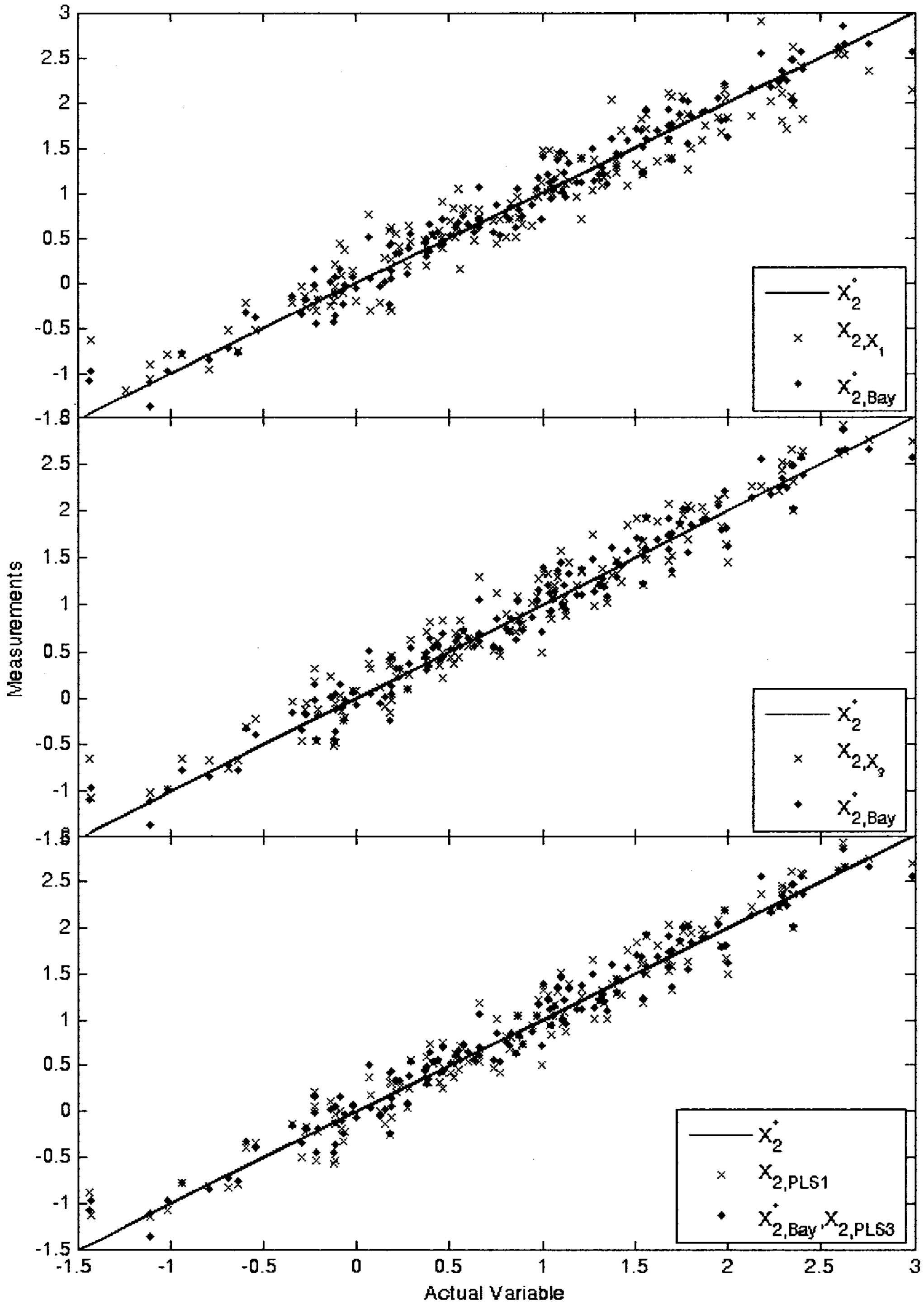

Figure 4.7. Scatter plot comparison of different estimated X_2^o in the simulated sequential model

because we did not quantify any prior knowledge about parameter values. As mentioned before, the MAP predictions are identical to the maximum likelihood predictions with non-informative prior. In addition, the maximum likelihood results are as same as least squares results in the case of normally distributed error. It is notable that the two-component PLS performs better than one-component PLS; this is because it captures a greater percentage of the variances.

Since we have just seen that the Bayesian inference and PLS perform the same with regard to prediction, we might conclude that the Bayesian framework is just a probabilistic interpretation of regression methods. Its advantages will become apparent, however, in treatments of missing values, as well as the non-normal errors in the next example.

Table 4.2. Mathematical comparison of different predicted X_2^o in the simulated sequential model

	Absolute Error Average	*Standard Deviation*	*Mean Square Error*
OLS Regression (Based on X_1)	0.2184	0.9422	0.0779
OLS Regression (Based on X_3)	0.1962	1.0056	0.0599
PLS Regression (1 Component)	0.1758	0.9897	0.0485
PLS Regression (2 Components)	0.1461	1.0040	0.0342
Bayesian Inference	0.1461	1.0040	0.0342

Table 4.3. Mathematical comparison of measured and recovered noise-free variables in the simulated sequential model (Given X_1 and X_3)

	X_1	$\hat{X}_1^o$	X_3	$\hat{X}_3^o$
Absolute Error Average	0.2501	0.1531	0.3958	0.3029

Table 4.4. Mathematical comparison of measured and recovered noise-free variables in the simulated sequential model (given X_1, X_2, and X_3)

	X_1	$\hat{X}_1^o$	X_3	$\hat{X}_3^o$	X_2	$\hat{X}_2^o$
Absolute Error Average	0.2501	0.1332	0.3958	0.2627	0.2469	0.132

In this example, the information provided is sufficient to recover noise-free measured inputs, given as X_1^o and X_3^o; this is considered to be diagnostic reasoning. As presented in Figure 4.8 and Figure 4.9, if only the values for X_1 and X_3 are available, the Bayesian framework is able to capture the measurement noises. Taking X_1^o and X_3^o as references, we evaluate the accuracy of $\hat{X}_1^o$ and $\hat{X}_3^o$ in comparison with X_1 and X_3 in Table 4.3. As presented in Table 4.4, much more noise can be captured if X_2 has also been observed.

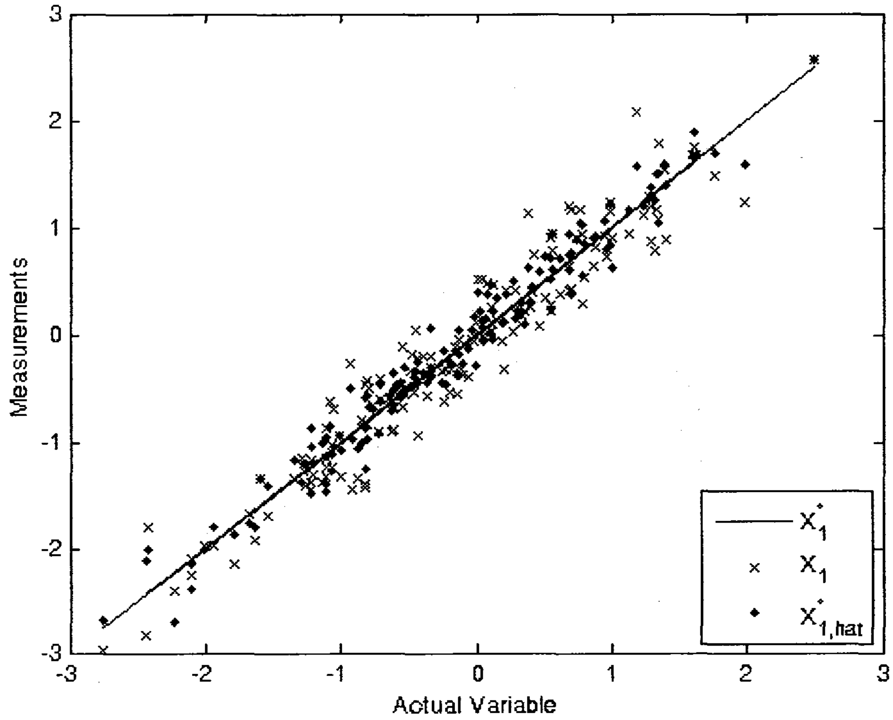

Figure 4.8. Scatter plot comparison of measured and recovered X_1^o in the simulated sequential model

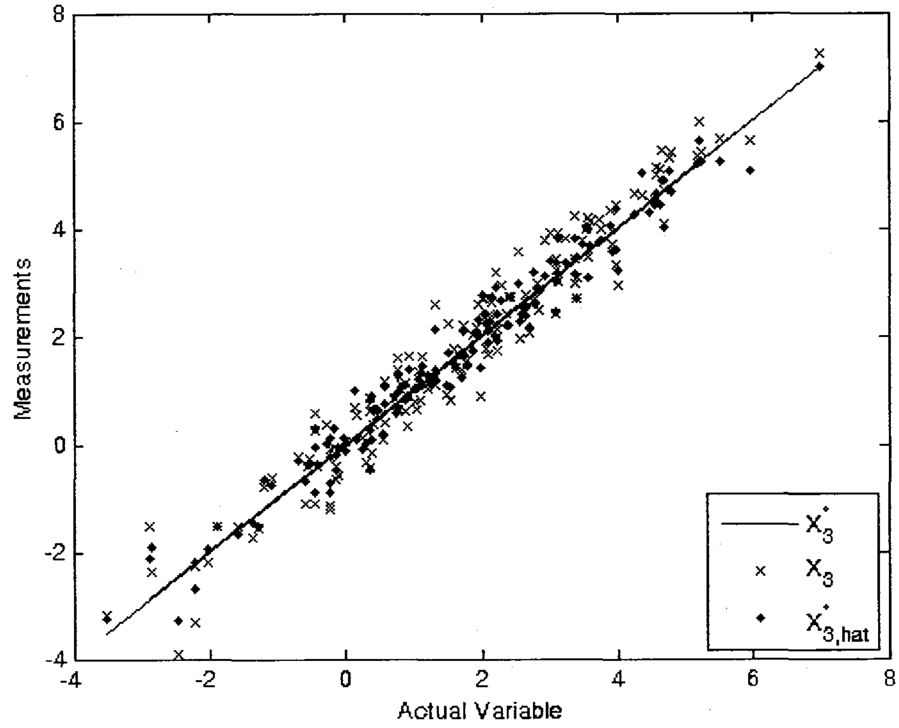

Figure 4.9. Scatter plot comparison of measured and recovered X_3^o in the simulated sequential model

4.8 Simulation example 2: A multiple linear model

4.8.1 Missing values

In soft sensor applications, it is common to represent a model in a multiple linear way as follows:

$$X_n = a_0 + a_1 X_2 + a_2 X_2 + \ldots + a_m X_m \qquad \textbf{(4.17)}$$

Here, we shall see how the Bayesian approach can be applied to handling the issues that come up in deriving a multiple linear soft sensor. As an example, consider the network presented in Figure 4.10, with the following model expressions:

$$\begin{array}{ll} X_1^o \sim N(0,1) & e_1 \sim N(0,(0.31)^2) \\ X_2^o \sim N(0,2) & e_2 \sim N(0,(0.5)^2) \\ X_3^o = 3 + 2X_1^o + X_2^o & e_3 \sim N(0,(0.63)^2) \end{array} \qquad \textbf{(4.18)}$$

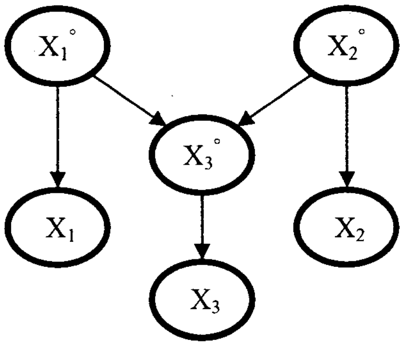

Figure 4.10. The Bayesian EIV framework for a multiple linear model

First, we generate a complete data set and then randomly hide a pre-determined percentage of values to simulate partially observed measurements. In keeping with the MAR mechanism, data are deleted randomly with missing value probabilities of 5%, 10%, 20%, 30% and 50% applied equally to the input variables. This procedure is repeated 5 times for each level of missing data. Applying Bayesian learning, we can fit the specified model and estimate its parameters in a maximum likelihood sense, using the EM algorithm as explained before.

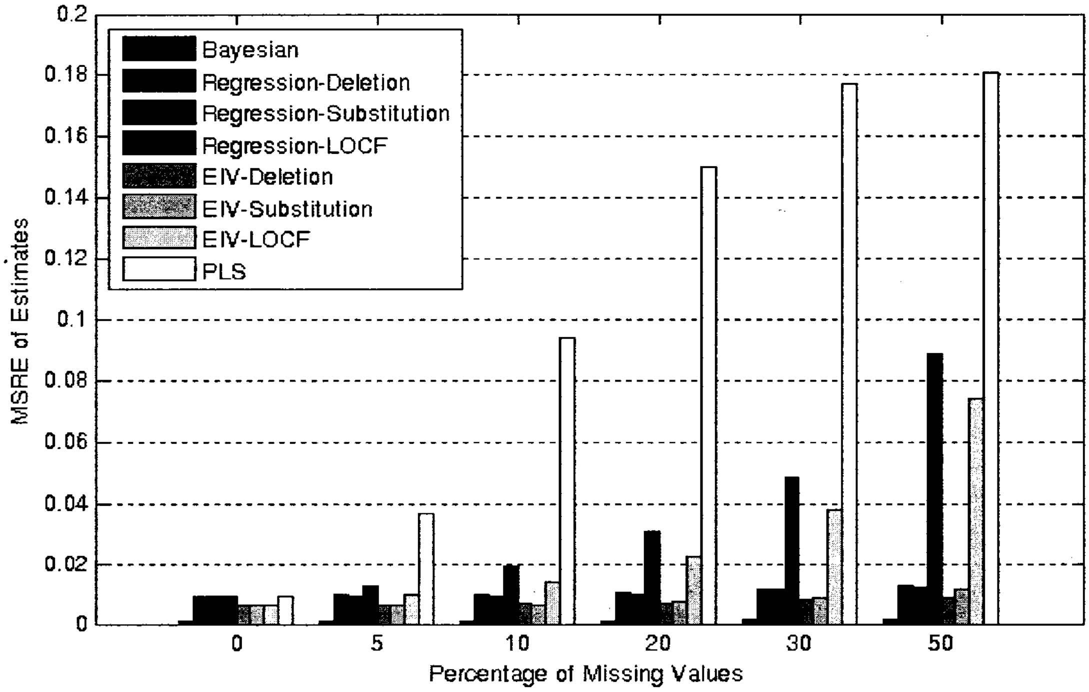

Figure 4.11. The MSREs of the estimates of the multiple linear model for different levels of missing values

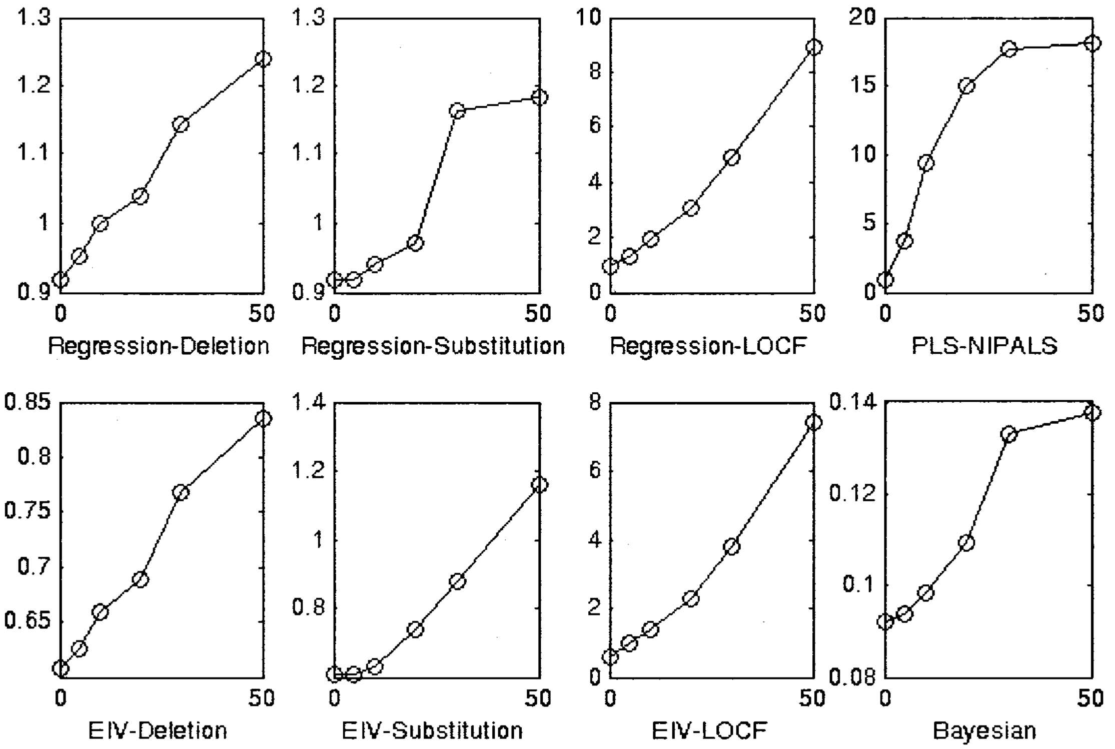

Figure 4.12. $100\times MSRE$ **of estimates by various techniques under different levels of missing data**

The Bayesian approach is compared with the following methods:

1. OLS regression ,
2. EIV,
3. The partial least square regression-NIPALS algorithm.

Since incomplete data are not allowed in the OLS regression and EIV, these methods handle missing values by using casewise deletion, mean substitution, and LOFC technique. Nevertheless, PLS regression in its standard form involving the use of the NIPALS algorithm can deal with missing values. Figure 4.11 and Figure 4.12 show the performance of all approaches based on mean square relative error of the estimated parameters. The Bayesian approach performs best for all levels of missing data. OLS, EIV, and PLS are comparable for complete data sets; however, the performance of PLS deteriorates significantly as the percentage of missing values increase. Clearly, the NIPALS algorithm is sensitive to missing values. If NIPALS algorithm be replaced with casewise deletion or mean substitution, PLS estimates converge to the OLS regression solution. LOCF also performs poorly in both OLS and EIV. The performance of casewise deletion and mean substitution are acceptable and comparable. This is not surprising, because the measured variables are distributed normally. As a result, it is highly likely that the values for missing observations will equal to the population mean. The effect of the prior distributions will be studied in detail later in this chapter. The main drawback of casewise deletion is that, with a high number of variables, the probability that a case $\left[X_1^j,\ldots,X_N^j\right]$ will be completely measured is low, and therefore $\boldsymbol{X_o}$ may be empty. One of the interesting features of Figure 4.12 is the analogous trend of MSREs when parameter estimates are obtained by OLS and EIV. It is notable, however, that the overall performance of EIV is better than that of the OLS regression for each of the presented missing values techniques.

4.8.2 Outliers

Let us now compare the conventional methods for outlier detection using the Bayesian framework. We first generate a complete data set representing the model described by Equation 4.17. The outliers' samples are also simulated according to the six-sigma rule:

$$\left|x_{ji}^{\mathrm{o}}-\mu_i^{\mathrm{o}}\right|>3\sqrt{\sigma_i^{\mathrm{o}}} \tag{4.19}$$

where μ_i^{o} and σ_i^{o} are the mean and variance of the X_i^{o} variable so that X_i^{o} is distributed as $X_i^{\mathrm{o}}\sim N\left(\mu_i^{\mathrm{o}},\sigma_i^{\mathrm{o}}\right)$.

We evaluate the performance of the following parameter estimation methods in the presence of outliers:

1. OLS regression,
2. PLS regression,
3. EIV,
4. OLS regression and Box plot technique[1],

[1] We first remove the outliers from training data set followed by the box plot outlier detection criteria. Next, the OLS regression is performed on the new set of data. The fences used to investigate the presence of outliers in our data set are presented in Table 4.5.

5. Robust regression,
6. Bayesian learning/inference (algorithm 4.1).

Table 4.6 reports the numerical values of estimated parameters and MSREs of the estimates. The MSREs are also plotted in Figure 4.13.

Table 4.5. Criteria for outlier detection

Outlier Detection Criteria	*Specification*		
	X_1	X_2	X_3
Median	0.0850	-0.0268	3.1158
Lower quartile	-0.6511	-1.0214	1.4284
Upper quartile	0.7788	1.0608	4.9519
Lower inner fence	-2.7961	-4.1447	-3.8568
Upper inner fence	2.9238	4.1840	10.2371
Lower outer fence	-4.9411	-7.2680	-9.1420
Upper outer fence	5.0688	7.3073	15.5223

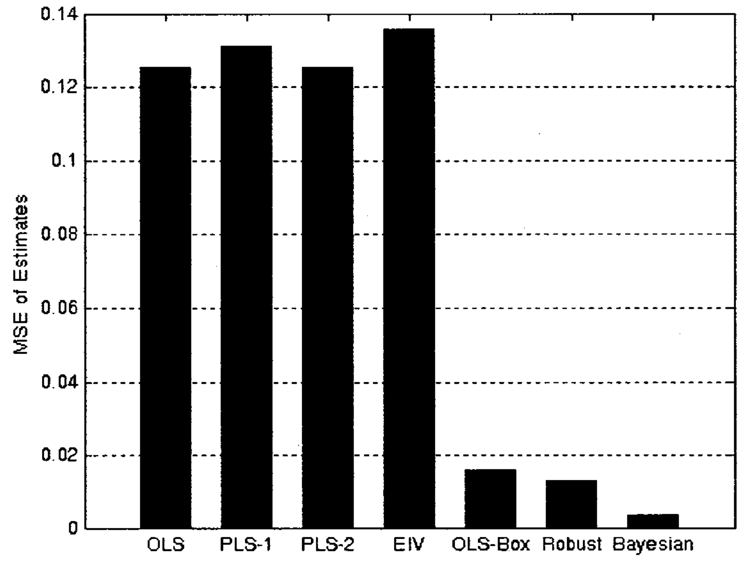

Figure 4.13. The MSREs of the estimates of the multiple linear model in the presence of outliers

Table 4.6. Parameters of the simulated multiple linear model in the presence of outliers

Method	*Coefficients*			*Mean Squared Relative Error*
	a_0	a_1	a_2	
True	**3**	**2**	**1**	-
OLS	3.0142	1.2019	0.5347	0.1253
PLS-1 Comp.	3.0137	1.2202	0.5091	0.1310
PLS-2 Comp.	3.0142	1.2019	0.5347	0.1253
EIV	4.2064	1.4261	0.596	0.1358
OLS-Box Plot	3.0883	1.7491	0.8258	0.0157
Robust	3.0023	1.7822	0.8366	0.0129
Bayesian	2.9731	1.8695	1.0754	0.0033

It is obvious that OLS and PLS are not efficient parameter estimation techniques in the presence of outliers; however, if we first use Box plot criteria to detect outliers and then remove them from the training data set, the performance of OLS improves. Since robust regression automatically detects outliers and downweighs them, the associated MSRE of estimates is also reasonable. If outliers represent the random measurement errors, EIV and Bayesian inference are applied to compensate for the noise. Yet, Figure 4.13 shows that EIV fails in parameter estimation, if a data set contains any outliers. Applying the Bayesian EIV framework not only captures the relatively large errors corresponding to outliers, but also preserves the size of the training data set. Comparing MSREs of the methods included in this study, it confirms that the Bayesian learning/inference algorithm is able to deal with outliers much better than other techniques can.

4.8.3 Lognormal errors

So far, we have considered only the case where measurement errors are normally distributed. The normal distribution assumption may not always be realistic for industrial data. To argue against unconditional reliance on the assumption of error term's normality in OLS and PLS regressions, we assume that measurement errors have a lognormal distribution. It is well known that the lognormal distribution has the probability density function:

$$f(x\,|\,\mu,\sigma)=\frac{1}{x\sigma\sqrt{2\pi}}\exp\left(-\frac{[\ln x-\mu]^2}{2\sigma^2}\right) \qquad \text{for } x>0 \qquad \textbf{(4.20)}$$

Given Equation 4.20, it follows that the expected values and variance are obtained from:

$$E(X)=\exp\left(\mu+\frac{\sigma^2}{2}\right) \qquad \textbf{(4.21)}$$

$$Var(X)=\left[\exp\left(\sigma^2\right)-1\right]\times\exp\left(2\mu+\sigma^2\right) \qquad \textbf{(4.22)}$$

Suppose the model structure of Equation 4.23 is considered to be:

$$\begin{aligned} &X_1^{\circ}\sim N(10,1)\\ &X_2^{\circ}\sim N(10,2)\\ &X_3^{\circ}=3+2X_1^{\circ}+X_2^{\circ} \end{aligned} \qquad \textbf{(4.23)}$$

The prior probabilities used to simulate lognormal distributions for measurement errors are given in Table 4.7. Due to the limitations of the available Bayesian software, we investigate only the case in which the error distribution is well-known. The estimated model parameters from classical regression and Bayesian method are compared in Table 4.8. The MSRE of estimates provides an illustration of the potential of the Bayesian method to improve parameter estimation when the measurement error is not normally distributed.

Table 4.7. Parameters of lognormal measurement errors

Measurement Error	μ	σ
e_1	0.001	0.35
e_2	0.001	0.35
e_3	0.001	0.4

Table 4.8. Parameters of the simulated multiple linear model with lognormal errors

Method	*Coefficients*			*Mean Squared Relative Error*
	a_0	a_1	a_2	
True	**3**	**2**	**1**	-
Regression	6.6191	1.7262	0.9132	0.4839
Bayesian	3.6118	2.0795	1.0166	0.0145

In conclusion, Bayesian inference helps us to attain increased performance by improving the efficiency in the treatment of outliers, missing values, and non-normal errors. It is noteworthy that the performance of the Bayesian approach can be further improved if we consider the following issues:

1. Increasing the size of the training data set,
2. Decreasing the number of hidden variables,
3. Increasing the number of iterations in EM algorithm, and
4. Using informative priors, e.g. identifying the variances of measurement errors [16].

4.9 Experimental evaluation: the three-tank system

In this section, we perform an experimental evaluation of Bayesian inference for a sequential model and then briefly discuss the results.

Consider the linear model representing the liquid level in the middle tank of the three-tank system presented in Chapter 3. Once the model has been identified, the aim is to predict H_2^o from a new set of observations. To investigate the prediction performance of the Bayesian approach on real data, the Bayesian technique is compared with PLS for both complete and incomplete samples. A training data set consisting of 851 steady state points is used to derive the Bayesian and PLS models. The PLS formulation is presented in Equation 4.24 as:

$$\overline{H}_{2,PLS} = 0.6061 \times \overline{H}_1 + 0.0887 \times \overline{H}_3 \quad \textbf{(4.24)}$$

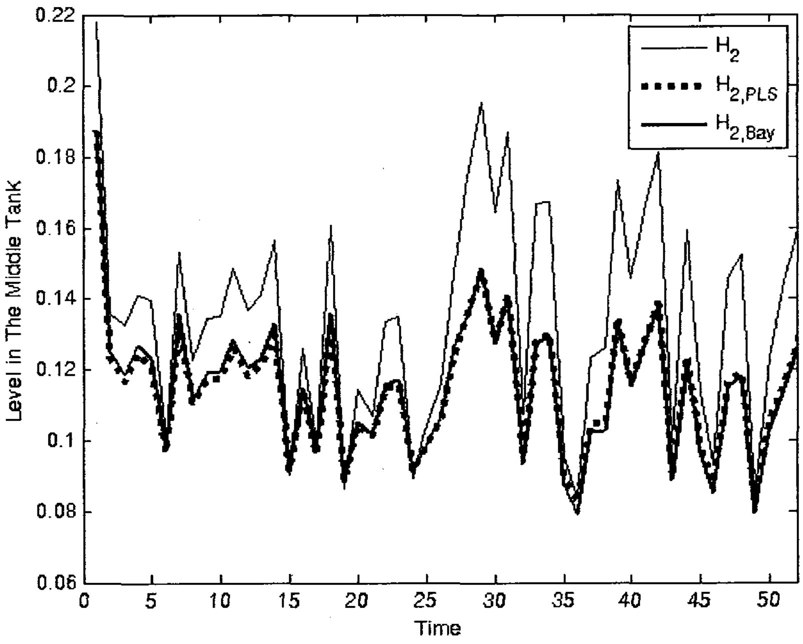

Figure 4.14. Trend comparison of prediction results obtained from the PLS and Bayesian models using a completely observed testing data set

The fitted models are then applied to performing prediction on 52 different observed cases. The prediction results for completely observed testing data are plotted in Figure 4.14. As mentioned earlier, PLS and Bayesian have the same prediction performance under linear noise and complete data conditions. In the next step, we remove part of the measurements from the testing data set as if they had not been observed. Two partially observed testing data sets are constructed. Without loss of generality, the missing values for the upper tank level can be taken as being the first half elements of the correlated data vector. At the same time, the last half elements of the data vector associated with lower tank level are also deleted. In the second data set, the last half of the upper tank level measurements and the first half of the lower tank level measurements are taken out. For each data set, the predicted values obtained from both models are shown in Figure 4.15 and Figure 4.16. When the upper tank level is not measured, it is obvious that the prediction performance of the PLS model deteriorates. On the other hand, missing the lower tank level does not seriously affect the prediction performance of PLS. Studying the PLS model, we see that very little weight is given to the lower tank level in comparison with the upper tank level.

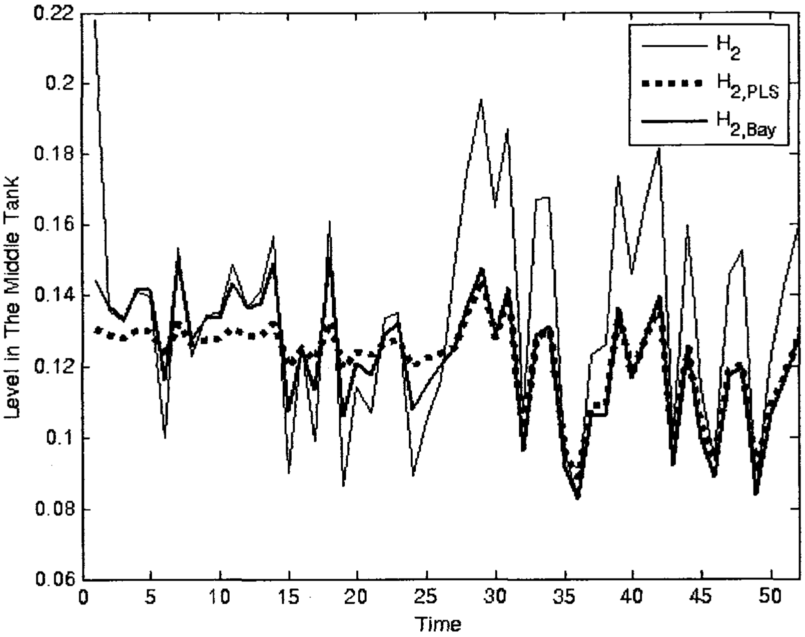

Figure 4.15. Trend comparison of prediction results obtained from the PLS and Bayesian models using the first partially observed testing data set

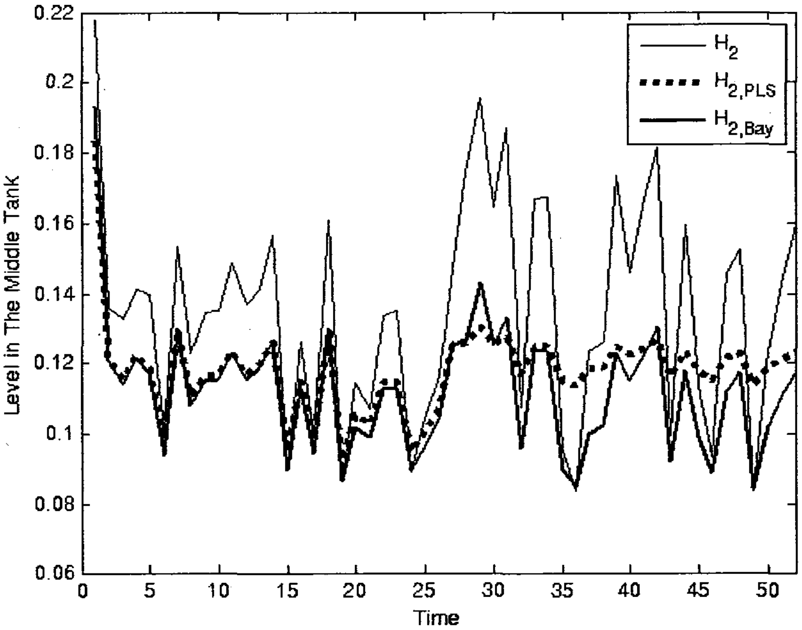

Figure 4.16. Trend comparison of prediction results obtained from the PLS and Bayesian models using the second partially observed testing data set

As a result, the fact that PLS fails to handle the missing values is much clearer when the upper level is not observed. Consequently, PLS does not capture the significant changes of the level in the middle tank from incomplete data set. In contrast, Figure 4.15 and Figure 4.16 reveal that the absence of either upper or lower tank level measurements does not influence the performance of the Bayesian model in predicting the level of the middle tank.

4.10 Industrial evaluation: the froth treatment process

We now turn to the application and evaluation of Bayesian inference in the context of soft sensor implementation for the froth treatment process discussed in Chapter 2. To improve the performance of the developed soft sensor, we need to handle the problem of missing and unreliable values. To do so, a probabilistic framework is applied to the investigation of the distribution of input variables. Applying a Bayesian way of thinking in the implementation, outliers and missing values in a variable will be replaced by the most probable value of that variable. The performance of the soft sensor models is reviewed using a testing set of records collected from April 2004 to December 2006.

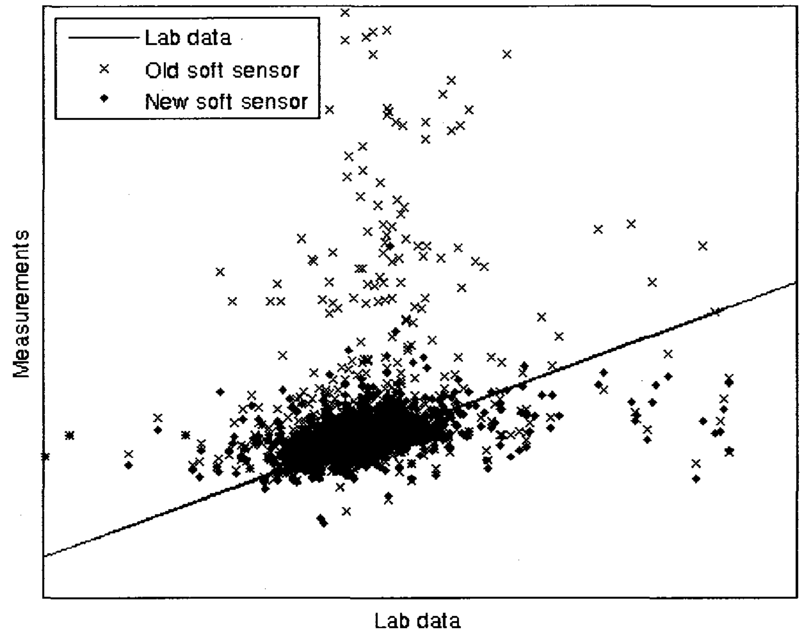

Figure 4.17. Scatter plot comparison of Bayesian and non-Bayesian N:B soft sensors

Scatter plot comparisons of each N:B soft sensors vis-à-vis the lab data are presented in Figure 4.17. It is obvious that unreliable measurements result in over-estimation of NB values in non-Bayesian soft sensor model; however, we are able to handle unreliable or missing input variables in the new soft senor. Figure 4.18 depicts a comparison of soft sensor and refractometer outputs. This figure demonstrates that the soft sensor provides reasonably successful prediction as a result of capturing changes in both measured and quality variables. To analyze the prediction error for each approach, their mean absolute errors, standard deviations, and mean squared errors are also presented in Table 4.9.

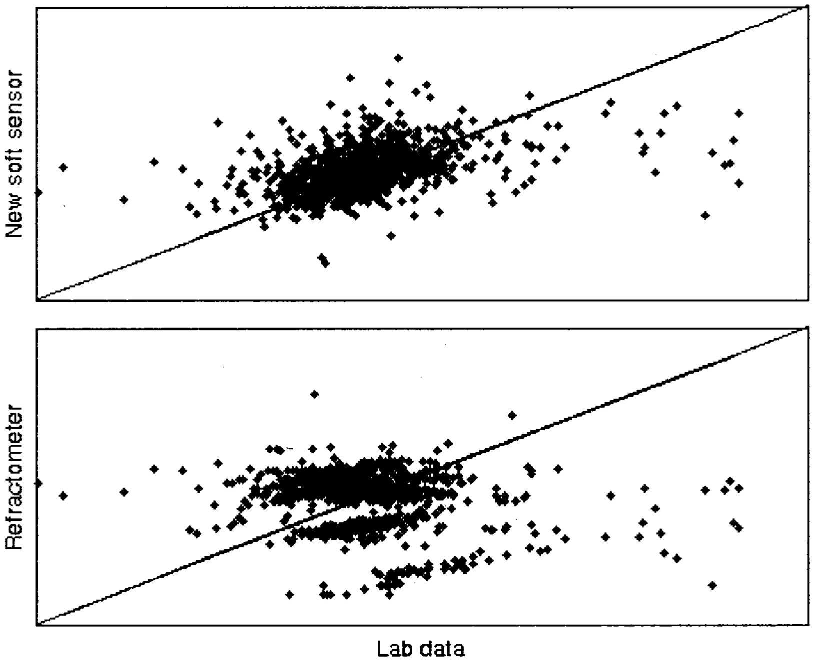

Figure 4.18. Scatter plot comparison of Refractometer Bayesian N:B soft sensor

Table 4.9. Mathematical comparison of N:B Measurements

	Refractometer	**New Soft Sensor**
Mean Absolute Error	0.0731	0.0392
Standard Deviation	0.0689	0.0531
Mean Squared Error	0.0098	0.0038

4.11 Conclusion

In this work, the practical issues of soft sensor development have been studied. To derive a soft sensor model from industrial data, it is important to deal with missing measurements and outliers. Several approaches for handling incomplete data sets have been reviewed: casewise deletion, mean substitution, LOFC, regression imputation, NIPALS algorithm, and EM algorithm. Robust regression and box plot criteria were also presented as the most commonly used techniques for detecting and dealing with outliers.

Bayesian inference theory and algorithms have been discussed in detail. Since the Bayesian approach enables us to cope with missing values, it is tempting to formulate the soft sensor problem in a Bayesian framework. A Bayesian learning/inference algorithm (algorithm 4.1) has also presented for dealing with outliers. Simulated data sets, designed specifically to accentuate the presence of outliers, missing values, and lognormal errors, have been used to assess the performance of each different approach in soft sensor development and implementation. Experimental and industrial case studies have also been analyzed in order to show the effectiveness of Bayesian soft sensor on real data sets.

4.12 Bibliography

1. Bishop C. M. and M. E. Tipping, *Bayesian Regression and Classification,* in Advances in Learning Theory: Methods, Models and Applications, J.A.K. Suykens *et al.* (Editors), IOS Press, NATO Science Series III: Computer and Systems Sciences, Vol. 190
2. Brandel, J., *Empirical Bayes Methods for Missing Data Analysis*, Technical Report 2004:11, Department of Mathematics, Uppsala University, June 2004
3. Chen C., *Robust Regression and Outlier Detection with the ROBUSTREG Procedure*, Proceedings of the 27th Annual Users group International Conference; SAS Institute, Cary, NC, 2002
4. D'Agostini, G, *Bayesian Inference in Processing Experimental Data: Principles and Basic Applications*, Reports on Progress in Physics, Vol. 66 ,pp.1383-1420, 2003
5. Dempster A. P., N. M. Laird, and D. B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society Series B, Vol. 39, pp. 1-38, 1977
6. Friedman, N., *The Bayesian Structural EM Algorithm*, Proceedings of the Fourteenth Conference on IUA, Madison, Wisconsin, pp. 129-138, 1998
7. Ghahramani, Z., *Learning Bayesian Networks*, Lecture Notes in Artificial Intelligence, Department of Computer Science, University of Toronto, October 1997
8. Heckerman, D., *A Tutorial on Learning Bayesian Networks*, Technical Report MSR-TR-95-06, Microsoft Research, November 1996
9. Helsper, E. L., C. Van der Gaag, *Building Bayesian networks through Ontologies*, Proceedings of the 15th European Conference on Artificial Intelligence, Lyon, France, pp. 680-684, July 2002
10. Huber, P. J., *Robust Estimation of a Location Parameter*, Annals of Mathematical Statistics, Vol. 101, pp. 35-73, 1964
11. Johnson, R. A., D. W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice Hall, Upper Saddle River, NJ, 1998
12. Korb, K. B., A. E., Nicholson, *Bayesian artificial intelligence*, Chapman & Hall/CRC, 2004

13. Leray P., O. Francois, *Bayesian Network Structural Learning and Incomplete Data*, Proceedings of AKRR'05 International and Interdisciplinary Conference, Helsinki, Finland, June 2005

14. Lewicki P., T. Hill, *Statistics Methods and Applications*, StatSoft, Inc., November 2005

15. Lin B., B. Recke, J. K.H. Knudsen, and S. B. Jørgensen, *A Systematic Approach for Soft Sensor Development*, Computers & Chemical Engineering, Vol. 31(5-6), pp. 419-425, May 2007

16. Murphy K. P., *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. Thesis, Department of Computer Science, University of California, Berkely, Fall 2002

17. Nelson P. R. C., P. A. Taylor, and J. F. MacGregor, *Missing Data Methods in PCA and PLS: Score Calculations with Incomplete Observations*, Chemometrics and Intelligent laboratory Systems, Vol. 35, No. 1, pp. 45-65, November 1996

18. Park, S., C. Han, *A Nonlinear Soft Sensor Based on Multivariate Smoothing Procedure for Quality Estimation in Distillation Columns*, Computers and Chemical Engineering, Vol. 24, pp. 871-877, 2000

19. Rubin, D., *Inference and Missing Data*, Biometrika, Vol. 63, pp. 581-592, 1976

20. Shrive F. M., H. Stuart, H. Quan, and W. A. Ghali, *Dealing with Missing Data in a Multi-Question Depression Scale: A Comparison of Imputation Methods*, BMC Medical Research Methodology, December 2006

21. Tipping, M. E., *Bayesian Inference: An Introduction to Principles and Prediction in Machine Learning*, in Advanced Lectures on Machine Learning, O. Bousquet, U. von Luxburg, and G. Rätsch (Editors.), pp. 41-62, 2004

22. Wang, D., R. Srinivasan, J. Liu, P. N. S Guru, and K. M. Leong, *Data-driven Soft Sensor Approach For Quality Prediction in a Refinery Process*, 4th International IEEE Conference on Industrial Informatics, Singapore, August 2006

23. Wold, S., L. Eriksson, J. Trygg, N. Kettaneh, *The PLS method -- partial least squares projections to latent structures and its applications in industrial RDP*, Institute of Chemistry, Umeå University, June 2004

Chapter 5

Dynamic Bayesian Models and Multimodal Processes

This chapter introduces dynamic Bayesian networks (DBNs) as an alternative for modeling dynamic processes that evolve with time. We begin with a short review of the theory behind dynamic Bayesian representation, learning, and inference. The review demonstrates the power of DBNs to model dynamic processes and shows that DBNs are useful for a wide range of applications. This chapter will then focus on switching Kalman filter (SKF) models that represent switching dynamics in multimodal processes as an approximation of non-linear processes. The application of SKF models to estimating the model parameters and determining the operating mode is then demonstrated on the three-tank system.

5.1 Introduction

Thus far, we have considered only static systems; however, temporal data arises in many areas of science and engineering. As a result, many real-world processes need to be naturally modeled as dynamic systems in order to express their behavior over time. In soft sensor applications, this reflects both the static and dynamic characteristics of the process and provides more flexibility for the developed soft sensor.

State-space models are among the formulations extensively used to represent, and hence model, dynamic systems [3]. In a state-space model, we assume that a sequence of real-valued observations has been generated from a sequence of hidden-state variables. Furthermore, the models take the first-order Markovian assumption that the present state is conditionally independent of the entire past, given the immediately-preceding state. Hidden Markov models (HMMs) [9] and Kalman filter models (KFMs) [12] are the two most common types of state-space models. In HMMs, a sequence of either continuous or discrete observations is modeled by assuming that each observation depends on a discrete hidden variable. HMMs have been applied in a wide variety of fields including speech recognition, modeling protein and DNA sequences in Bioinformatics, and fault diagnosis [9,2,13]. The KFMs are analogous to HMMs, the key difference being that the hidden-state variables are continuous.

Dynamic Bayesian networks (DBNs) [7], a more general type of state-space models, are an extended form of static Bayesian models. A DBN is composed of identical sub-models duplicated over each time-slice. Each time-slice is linked to others through a set of inter-slice connections. Many time series models, including HMMs and KFMs, are just special cases of DBNs. KFMs assume that the hidden state must be unimodal, an assumption which is inappropriate for many soft sensor problems. It is well-known that soft sensors are valid for only the particular region in which the data are collected. If a process operation condition changes significantly, the system has multiple modes or regimes of behavior and the soft sensor should switch between different models. A switching linear dynamic (SLD) model is an example of a hybrid DBN that can capture richer structure than KFMs for addressing the multimodal processes [6].

The novelty of this chapter is soft sensor development by representing a dynamic process as an SLD. This innovation proves useful in the development of soft sensors for multimodal systems. In particular, we are interested in applying SLDs to modeling piecewise linear behavior in order to approximate non-linear models. Once a multimodal system is represented as an SLD, learning algorithms can be applied to the estimation of model parameters. Applying inference algorithms, we are able to determine the operating regime and predict hidden variables.

This chapter first provides a brief overview of dynamic Bayesian models. Subsequently, it discusses the problem of learning the parameters of DBNs using the EM algorithm and reviews the main kinds of inference we might want to perform using a DBN. We next turn our attention to the applications of dynamic Bayesian modeling, focusing on SKF models that represent switching dynamics. To illustrate this approach, an SKF model is developed to represent the piecewise linear behavior of the top tank of the three-tank system discussed in Chapter 3. Finally, dynamic modeling of this tank is applied to determining its operating mode and estimating noise-free liquid level.

5.2 Representation

A dynamic Bayesian model is one that represents sequences of variables as they evolve over time. Depending on the types of variables, a DBN can be continuous, discrete, or hybrid, i.e. can contain continuous variables as well as discrete ones. Suppose that a set of variables $\boldsymbol{Z} = \{Z_1, ..., Z_n\}$ are of interest at each time step. These variables typically represent the input ($\boldsymbol{U}$), hidden variables ($\boldsymbol{X}$), and outputs ($\boldsymbol{Y}$) of a state-space model. Assigning a time index to each variable, a DBN for modeling dynamics can be constructed:

- The current time step is represented by $\boldsymbol{Z}^t = \{Z_1^t, ..., Z_n^t\}$.
- The previous time step is represented by $\boldsymbol{Z}^{t-1} = \{Z_1^{t-1}, ..., Z_n^{t-1}\}$.
- The future time step is represented by $\boldsymbol{Z}^{t+1} = \{Z_1^{t+1}, ..., Z_n^{t+1}\}$.

In a DBN, relationships between variables within a time-slice are represented by intra-slice connections, $Z_i^T \rightarrow Z_j^T$, while the relationships between variables at successive time steps are represented by inter-slice connections, $Z_i^T \rightarrow Z_i^{T+1}$ and $Z_i^T \rightarrow Z_j^{T+1}$. The relationships within and between time-slices are quantified by the conditional probability distribution associated with each variable. The CPDs within and between slices are repeated for each $t > 0$, so that DBNs are time-invariant. Although it is not a requirement, it is also assumed that the states of a DBN satisfy the first-order Markov condition, which is defined as follows:

$$P\left(\boldsymbol{Z}^t \mid \boldsymbol{Z}^0, ..., \boldsymbol{Z}^{t-1}\right) = P\left(\boldsymbol{Z}^t \mid \boldsymbol{Z}^{t-1}\right) \qquad \textbf{(5.1)}$$

The joint probability distribution of the DBN is then defined as:

$$P\left(\boldsymbol{Z}^t \mid \boldsymbol{Z}^{t-1}\right) = \prod_{i=1}^{n} P\left(Z_i^t \mid Par\left(Z_i^t\right)\right) \qquad \textbf{(5.2)}$$

where $Par\left(Z_i^t\right)$ are the parents of node Z_i. As mentioned earlier, the parents of a node can be either in the current time-slice or in the previous time-slice.

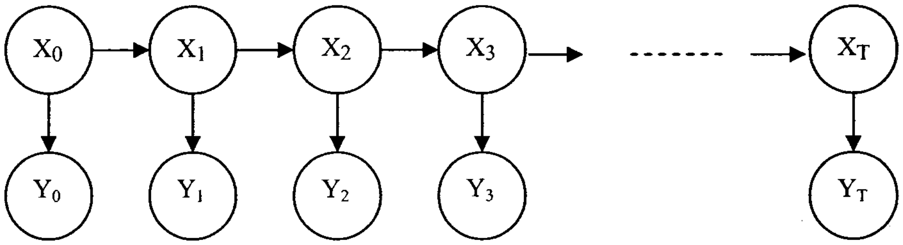

Figure 5.1. A dynamic Bayesian network representing a state-space model

5.3 Learning in dynamic Bayesian models

The purpose of learning in a DBN is to estimate the parameters of $P\left(\boldsymbol{Z}^0\right)$ and $P\left(\boldsymbol{Z}^t \mid \boldsymbol{Z}^{t-1}\right)$ given a number of sequences of observations. The techniques for parameter estimation in a dynamic Bayesian model are straightforward extensions of the learning algorithms applied in Bayesian models such as EM and gradient descent algorithms which have been discussed in Chapter 3. To illustrate the use of EM algorithm in learning DBNs from a single run of output observations, we formulate and solve the problem of ML parameter estimation for a linear system driven by white Gaussian noise.

Assume that a sequence of observations, $\{Y_1,...,Y_T\}$, is generated by the finite dimensional linear-Gaussian state-space model (also known as Kalman filter model) as shown in Figure 5.1 and formulated by:

$$\begin{cases} X_t = AX_{t-1} + w_t \\ Y_t = BX_t + v_t \end{cases} \qquad \textbf{(5.3)}$$

where

- X is a K dimensional hidden state variable,
- A is the state transition matrix,
- B is the observation matrix, and
- v_t and w_t are uncorrelated, zero-mean random noise vectors.

The probability of the hidden states and observations for this model can be written as:

$$P\{X_t, Y_t\} = P(X_0)P(Y_0 \mid X_0)\prod_{t=1}^{T} P(X_t \mid X_{t-1})P(Y_t \mid X_t) \qquad \textbf{(5.4)}$$

Thus

$$\log\left(P\{X_t,Y_t\}\right)=\log\left(P(X_0)\right)+\sum_{t=0}^{T}\log\left(P(Y_t\mid X_t)\right)+\sum_{t=1}^{T}\log\left(P(X_t\mid X_{t-1})\right) \qquad \textbf{(5.5)}$$

The joint log likelihood of Equation 5.5 is a sum of quadratics:

$$\begin{aligned}L(\boldsymbol{X},\boldsymbol{Y},\theta)=&-\frac{1}{2}\log|\Sigma|-\frac{1}{2}(X_0-\mu)'\,\Sigma^{-1}(X_0-\mu)\\&-\frac{T}{2}\log|Q|-\frac{1}{2}\sum_{t=1}^{T}(X_t-AX_{t-1})'\,Q^{-1}(X_t-AX_{t-1})\\&-\frac{T+1}{2}\log|R|-\frac{1}{2}\sum_{t=0}^{T}(Y_t-BX_t)'\,R^{-1}(Y_t-BX_t)+constant\end{aligned} \qquad \textbf{(5.6)}$$

where

- R is the covariance of the observation noise v_t
- Q is the covariance of the states noise w_t

For the complete data, $\{X_0,X_1,\ldots,X_T,Y_0,Y_1,\ldots,Y_T\}$, the ML parameters can be solved by maximizing $\log L$ with respect to the free parameters A, B, Q, and R. Therefore, the ML estimates are[1]:

$$\hat{A}=\Gamma_4\Gamma_3^{-1} \qquad \textbf{(5.7)}$$

$$\hat{B}=\Gamma_6\Gamma_1^{-1} \qquad \textbf{(5.8)}$$

$$\hat{Q}=\Gamma_2-\Gamma_4\Gamma_3^{-1}\Gamma_4'=\Gamma_2-\hat{A}\Gamma_4' \qquad \textbf{(5.9)}$$

$$\hat{R}=\Gamma_5-\Gamma_6\Gamma_1^{-1}\Gamma_6'=\Gamma_5-\hat{B}\Gamma_6' \qquad \textbf{(5.10)}$$

where the sufficient statistics are,

- $\Gamma_1=\dfrac{1}{T+1}\displaystyle\sum_{t=0}^{T}X_tX_t'$
- $\Gamma_2=\dfrac{1}{T}\displaystyle\sum_{t=1}^{T}X_tX_t'$

[1] Please see [1] for derivations. Here, the final results are stated without proof.

- $\Gamma_3 = \frac{1}{T}\sum_{t=1}^{T} X_{t-1}X_{t-1}{}'$
- $\Gamma_4 = \frac{1}{T}\sum_{t=1}^{T} X_t X_{t-1}{}'$
- $\Gamma_5 = \frac{1}{T+1}\sum_{t=0}^{T} Y_t Y_t'$
- $\Gamma_6 = \frac{1}{T+1}\sum_{t=0}^{T} Y_t X_t'$

Since the states are hidden variables, we consider applying the EM algorithm to estimate the expected values:

0. Start from initial guesses for the desired parameters.

1. **Expectation Step:** In this case, the conditional distributions of the states given the observations are Gaussian, because the process disturbances are assumed to be Gaussian:

$$P(X_t \mid \boldsymbol{Y}) \sim N(\hat{X}_{t|T} \mid \Sigma_{t|T}) \qquad \textbf{(5.11)}$$

It follows that the sufficient statistics for maximizing $Q\left(\theta^{k+1} \mid \theta^k\right)$ can be computed from:

$$E_{\theta^k}\left\{Y_t X_t' \mid \boldsymbol{Y}\right\} = Y_t E_{\theta^k}\left\{X_t' \mid \boldsymbol{Y}\right\} = Y_t \hat{X}'_{t|T}$$

$$E_{\theta^k}\left\{Y_t Y_t' \mid \boldsymbol{Y}\right\} = Y_t Y_t'$$

$$E_{\theta^k}\left\{X_t X_t' \mid \boldsymbol{Y}\right\} = \Sigma_{t|T} + \hat{X}_{t|T}\hat{X}'_{t|T}$$

$$E_{\theta^k}\left\{X_t X_{t-1}{}' \mid \boldsymbol{Y}\right\} = E_{\theta^k}\left\{\left(X_t - \hat{X}_{t|T}\right)\left(X_{t-1} - \hat{X}_{t-1|T}\right) \mid \boldsymbol{Y}\right\}\Sigma_{t|T} + \hat{X}_{t|T}\hat{X}'_{t-1|T}$$

Note that the subscript θ^k represents the parameter vector that is used in calculating the expectations. These terms have been calculated using the fixed-interval Kalman smoothing algorithm in [1].

2. **Maximization Step:** compute the new ML estimates for the system parameters from Equation 5.7 to Equation 5.10; these estimates maximize the following quantity:

$$Q\left(\theta^{k+1} \mid \theta^{k}\right)=E_{\theta^{k}}\left\{L\left(X, Y, \theta^{k}\right) \mid Y\right\} \tag{5.12}$$

Other efficient methods for learning the parameters and structure of dynamic Bayesian networks can be found in [7].

5.4 Inference in dynamic Bayesian networks

The general inference problem in DBNs is to compute $P\left(Z_i^{t_k} \mid Y^{t_i},...,Y^{t_j}\right)$, where $Z_i^{t_k}$ represents the i$^{\text{th}}$ output or hidden variable at time t_k and $Y^{t_i},...,Y^{t_j}$ denote all the observations between times t_i and t_j. The main inference tasks that we might want to perform in DBNs can be usually categorized as one of four possible types of query:

1. *Filtering*: The most common inference problem is to estimate the belief state, which is defined as $P\left(X^t \mid Y^0,...,Y^t\right)$.
2. *Smoothing*: Given a sequences of observations, $\left\{Y^0,...,Y^t\right\}$, we can also estimate the states of the hidden variables at previous time-slices, i.e., compute $P\left(X^{t-h} \mid Y^0,...,Y^t\right)$. As mentioned in the previous section, smoothing is important for parameter learning.
3. *Prediction*: Given all the observations up to the current time, sometimes we want to predict future outputs or hidden variables, i.e., compute $P\left(X^{t+h} \mid Y^0,...,Y^t\right)$ or $P\left(Y^{t+h} \mid Y^0,...,Y^t\right)$.
4. *Viterbi decoding*: Another interesting inference problem is to compute the most likely sequence of hidden variables based on data from past observations, i.e., compute $\underset{X^0,...,X^t}{\arg\max} P\left(X^0,...,X^t \mid Y^0,...,Y^t\right)$.

For a detailed discussion of inferences in DBNs readers are referred to [5,7,14].

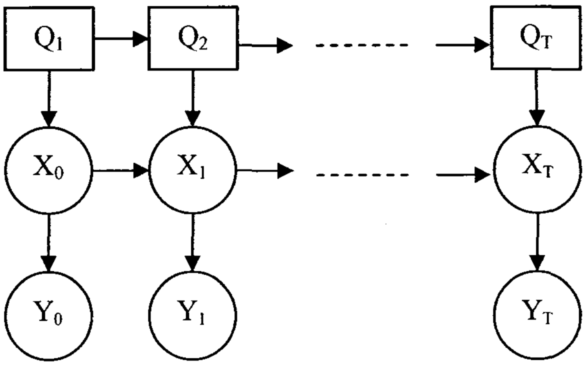

Figure 5.2. A switching Kalman filter model

5.5 Switching Kalman filters

The remainder of this chapter is devoted to a discussion of applying DBNs to representing multimodal dynamic processes. We shall consider hybrid models, i.e. models that contain both discrete and continuous variables. Consider the model in Figure 5.2, which shows a generic switching Kalman filter (SKF). The discrete variables in this network are represented by rectangles and the continuous variables are represented by circles. The CPDs for this model are as follows:

$$P(Q_t = j \mid Q_{t-1} = i) = A(i,j) \quad \textbf{(5.13)}$$

$$P(X_t = x_t \mid X_{t-1} = x_{t-1}, Q_t = i) = N(\mu_i^X + H_i x_{t-1}, R_i) \quad \textbf{(5.14)}$$

$$P(Y_t = y_t \mid X_t = x_t) = N(\mu^Y + W x_t, \sigma) \quad \textbf{(5.15)}$$

One of the most important applications of switching dynamics is for modeling multimodal processes in general and piecewise linear systems in particular. To illustrate the concept of hybrid dynamic Bayesian modeling, an SKF model will be developed to represent the piecewise linear behavior of the top tank in the three-tank setup; this setup was introduced in Chapter 3.

5.5.1 Modeling the piecewise linear behavior of the top tank in the three-tank setup

Consider the three-tank system discussed in Chapter 3. Under normal operating conditions, the liquid level in each tank is assumed to be constant if the inlet flow to the top tank is constant or subject to very little variation. The measured variables are contaminated, however, by measurement noises, so that:

$$H_i = H_i^{\circ} + e \tag{5.16}$$

$$q = q^{\circ} + v \tag{5.17}$$

In this study, we focus on the top tank, which is represented by the following non-linear model[1]:

$$F\left(H^{\circ}, q\right) = \frac{dH^{\circ}}{dt} = \frac{1}{A}q - \frac{1}{A}C\sqrt{H^{\circ}} \tag{5.18}$$

where,

- q is the measured inflow to the first tank,
- H° is the noise-free liquid level in the first tank,
- A is the cross sectional area of the first tank, and
- C is the resistance of the output orifice of the first tank.

If the system operates around an equilibrium point and if the perturbations involved are small, then it is possible to approximate this non-linear system by a linear system. Hence, the linearized equation is obtained by the Taylor expansion of Equation 5.18 around an equilibrium state. If the normal operating condition corresponds to $\bar{q}$[2] and $\bar{H}$, then Equation 5.18 is linearized about these points as follows:

$$F\left(H^{\circ}, q\right) = F\left(\bar{H}^{\circ}, \bar{q}\right) + \frac{\partial F}{\partial H^{\circ}}|_{H^{\circ}=\bar{H}^{\circ}, q=\bar{q}} \times \left(H^{\circ} - \bar{H}^{\circ}\right) + \frac{\partial F}{\partial q}|_{H^{\circ}=\bar{H}^{\circ}, q=\bar{q}} \times \left(q - \bar{q}\right)$$

$$= -\frac{C}{2A\sqrt{\bar{H}^{\circ}}}\left(H^{\circ} - \bar{H}^{\circ}\right) + \frac{1}{A} \times \left(q - \bar{q}\right) \tag{5.19}$$

Thus, the noise-free level at time $t+1$, which is H_{t+1}°, can be calculated as:

$$H_{t+1}^{\circ} = \left(1 - \frac{C^2 \Delta t}{2A\bar{q}}\right) H_t^{\circ} + \frac{\Delta t}{2A}\bar{q} + \varepsilon_t \tag{5.20}$$

where Δt is the sampling time and ε_t is the noise term.

[1] For simplicity, henceforth, we will drop the index 1 that denotes the first tank.

[2] $\bar{q} = q^{\circ}$

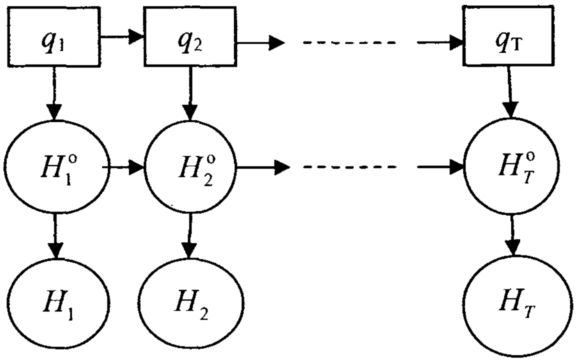

Figure 5.3. The top tank of the three-tank system modeled as an SKF model

We can model all of this using an SKF model as shown in Figure 5.3. In this model, each steady state value of the inflow corresponds to a nominal operating mode. The sampling time is 20s and two nominal steady state values considered here are $\bar{q}^{(1)} = 50\,cm^3/s$ and $\bar{q}^{(2)} = 80\,cm^3/s$, respectively. The prior probabilities and the transition probabilities used in simulation are given in Table 5.1 and Table 5.2, respectively.

First, we develop an SKF model based on simulated data. The rationale is that both algorithm performance and model inaccuracies cause errors in real data. Using simulated data, however, only the second type of error presents and we can better evaluate the performance of our approach. A set of 21 cases containing 19 time-slices is generated from the three-tank simulated model. The magnitude of inflow varies between two steady state values resulting in a standard deviation of 5.55 for measured liquid level. Therefore, it is assumed that the measurement noise is distributed as $\varepsilon \sim N\left(0,(0.56)^2\right)$.

Table 5.1. Prior probabilities of the operating modes

	$\bar{q}^{(1)}$	$\bar{q}^{(2)}$
Prior Probability	0.4	0.6

Table 5.2. Transition probabilities of the operating modes

	$\bar{q}^{(1)}$	$\bar{q}^{(2)}$
$\bar{q}^{(1)}$	0.97	0.03
$\bar{q}^{(2)}$	0.01	0.99

Table 5.3. Parameters of the SKF model obtained from simulated data

	Parameters of interest	
	$\bar{a}^{(1)}$	$\bar{a}^{(2)}$
True	0.2339	0.5212
SKF	0.2504	0.5172

First, we assume that the prior probabilities and the transition matrix between the two equilibrium points are known a priori. The model representing the piecewise linear behavior of the tank is identified as follows:

- First operation level: $H^{\circ}_{t+1} = 0.2504\times H^{\circ}_{t} + 0.0571$
- Second operation level: $H^{\circ}_{t+1} = 0.5172\times H^{\circ}_{t} + 0.0914$

The estimated parameters together with their true values are presented in Table 5.3. Comparing these with the true parameters, it is obvious that the SKF model effectively estimates model parameters. The most interesting feature of representing the liquid tank as an SKF model, however, is that it enables us to determine its operating mode and estimate the noise-free liquid level on-line, i.e. as a dynamic soft sensor. In this respect, both filtering and smoothing task can be performed. Since our interest is in the dynamic estimation problem, we apply moving-horizon strategy and estimate the filtered estimates. Figure 5.4 shows the estimated operating mode. This figure proves that the SKF model does an excellent job to capturing the operating mode. In addition, the changes in operating mode are detected at very latest after two time steps. To investigate the effect of noise level on the performance of the SKF model, standard deviation of the measurement noise associated with liquid level increases from 0.56 to 1.78 and 2.87. The confusion matrices, which show actual vis-à-vis predicted group membership, are reported in Table 5.4.

Given these data, the error rates are then easily calculated (see Table 5.5). Error rate represents the percentage of misclassification followed by the incorrect estimation of operating mode. As shown in Table 5.5, our SKF model has a near perfect performance in determining the level of operation. Although the error rate becomes larger as the noise variance increases, the detection performance is still satisfactory.

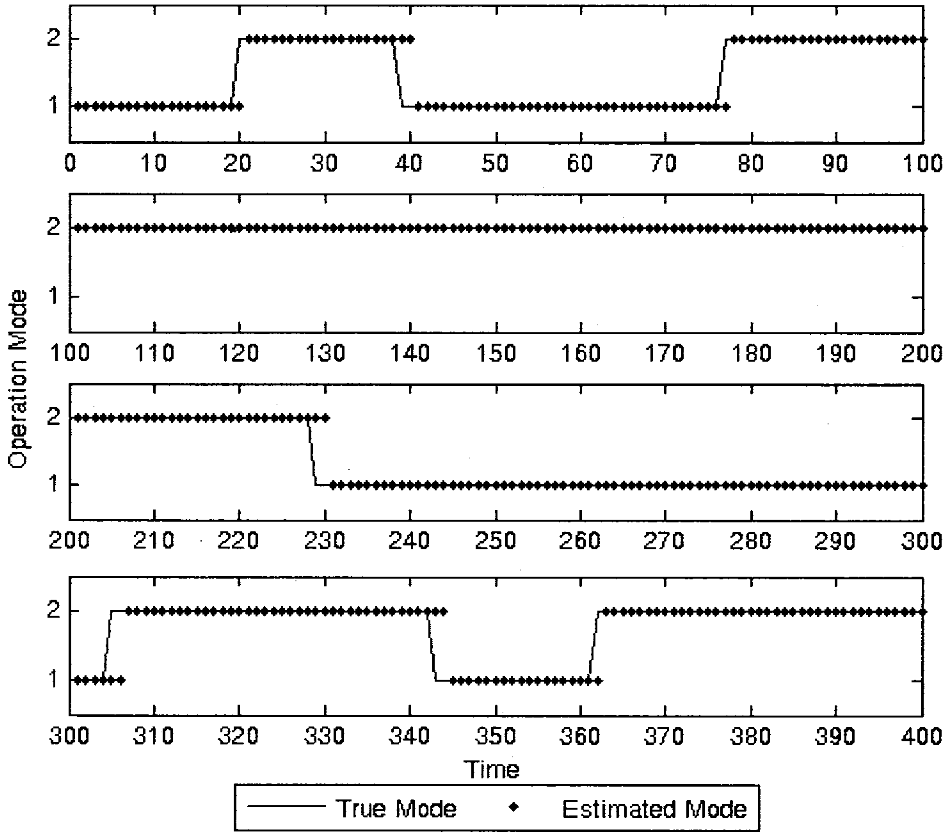

Figure 5.4. Estimated level of operation with a measurement noise distributed as $\varepsilon \sim N\left(0,(0.56)^2\right)$

Table 5.4. Confusion matrices of operating mode estimation for different levels of measurement noise

Standard Deviation of Measurement Noise	*Actual Membership*	*Predicted Membership*	
		1st Mode	**2nd Mode**
0.56	**1st Mode**	146	6
	2nd Mode	5	243
1.78	**1st Mode**	145	7
	2nd Mode	9	239
2.87	**1st Mode**	143	9
	2nd Mode	10	238

Table 5.5. Error rate in estimating operating modes for different levels of noise

Standard Deviation of Measurement Noise	*Error Rate (%)*
0.56	2.75
1.78	4
2.87	4.75

Next, we are interested in evaluating the performance of the SKF model in estimating the noise-free liquid level. The measured values are compared with the estimated ones for $\varepsilon \sim N\left(0,(1.78)^2\right)$ and $\varepsilon \sim N\left(0,(2.87)^2\right)$, respectively, in Figure 5.5 and Figure 5.6. Taking H^{o} as a reference, we can calculate the accuracy of $\hat{H}^{\text{o}}$ and H. The results of our mathematical comparison are summarized in Table 5.6. Both trend and mathematical comparison reveal that the estimates are closer to the true noise-free liquid levels than are the measurements; therefore, the proposed representation of the liquid tank is able to recover noise-free variables from the measurements. It is noteworthy that the Bayesian EIV framework proposed in Chapter 3 needs more than one variable in order to be able to capture measurement noises.

Hitherto, we assumed that the prior probabilities and the transition matrix are known as a priori. It is also interesting to investigate how an SKF model handles the unknown probabilities. An SKF model is built in the absence of prior knowledge. The estimated prior probabilities and the transition probabilities, respectively, for different levels of measurement noise are reported in Table 5.7 and Table 5.8.

Table 5.6. Mathematical comparison of measured and recovered noise-free variables

Standard Deviation of Measurement Noise	*Absolute Error Average*	
	X_1	$\hat{X}_1^{\text{o}}$
0.56	0.0046	0.0027
1.78	0.0145	0.0078
2.87	0.0234	0.0126

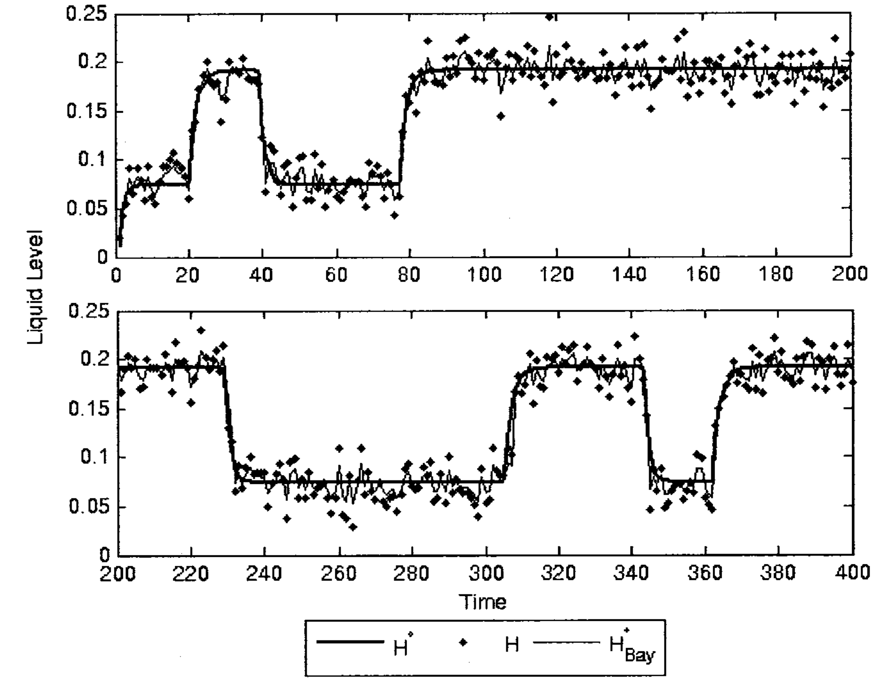

Figure 5.5. Trend comparison of measured liquid level and estimated noise-free liquid level with $\varepsilon \sim N\left(0,(1.78)^2\right)$

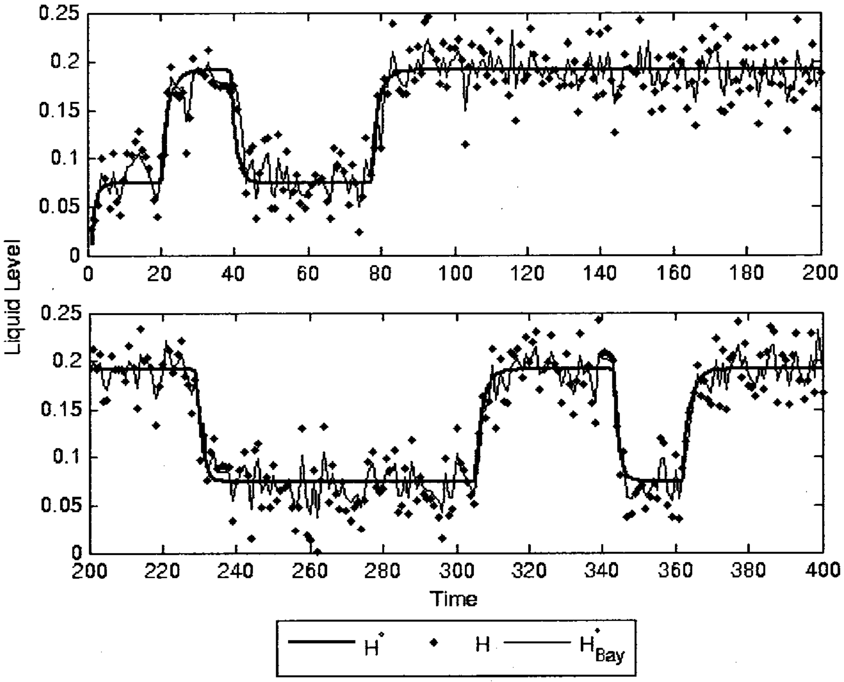

Figure 5.6. Trend comparison of measured liquid level and estimated noise-free liquid level with $\varepsilon \sim N\left(0,(2.87)^2\right)$

Table 5.7. Prior probabilities of the operating modes for different levels of noise

Standard Deviation of Measurement Noise	*Prior Probability*	
	$\bar{q}^{(1)}$	$\bar{q}^{(2)}$
0.56	0.4053	0.5947
1.78	0.4116	0.5884
2.87	0.4019	0.5981

Table 5.8. Transition probabilities of the operating modes for different levels of noise

Standard Deviation of Measurement Noise	*Transition Probability*		
		$\bar{q}^{(1)}$	$\bar{q}^{(2)}$
0.56	$\bar{q}^{(1)}$	0.9719	0.0158
	$\bar{q}^{(2)}$	0.0281	0.9842
1.78	$\bar{q}^{(1)}$	0.9741	0.0259
	$\bar{q}^{(2)}$	0.0143	0.9857
2.87	$\bar{q}^{(1)}$	0.9556	0.0444
	$\bar{q}^{(2)}$	0.0322	0.9678

Comparing the estimates with the true values (see Table 5.1 and Table 5.2), we conclude that prior information about prior and conditional probability distributions of the mode transition between equilibrium points are not required to develop a SKF model. It is obvious that the SKF representation is capable of estimating these probabilities in the modeling phase.

We now turn our attention to experimental data obtained from the three-tank setup. As before, inflow to the top tank varies between the two steady state values of 50 and 80. First, the parameters of the SKF model are estimated from collected data. These estimates are presented in Table 5.9. It is clear that the estimated values are close to the values provided by vendor; yet, the estimates obtained from simulation data are more accurate.

Table 5.9. Parameters of the SKF model obtained from experimental data

	Parameters of interest	
	$\bar{a}^{(1)}$	$\bar{a}^{(2)}$
Vendor	0.2339	0.5212
SKF	0.2706	0.4653

Table 5.10. Confusion matrix of operating mode estimation obtained from experimental data

Actual Membership	*Predicted Membership*	
	1st Mode	**2nd Mode**
1st Mode	145	7
2nd Mode	8	240

In the next step, we estimate the mode of operation from the SKF model. The results are summarized in the confusion matrix reported in Table 5.10. According to Table 5.10, the error rate is 3.75%, which once again shows the SKF model's good ability to detect operating mode. As verified by simulated data, we are also able to estimate the noise-free variables from a sequence of measurements. The estimated noise-free and the measured liquid levels are plotted in Figure 5.7. In the case of experimental data, there is no reference available to us by which to determine the accuracy of estimated noise-free variables. From our experience, however, we know that our setup produces valid results only under high pump speed conditions. Nevertheless, the first steady state value of inflow, $\bar{q}^{(1)}$, corresponds to a low pump speed. It is therefore expected that the measurements collected under the first operating mode will be highly affected by noise. Hence, the values for the noise-free liquid level should be greater than the observed values. As presented in Figure 5.7, this fact has been detected by our SKF model.

5.6 Conclusion

The focus of this chapter has been to illustrate the potential of the dynamic Bayesian networks (DBNs) to represent dynamic systems and multimodal processes. The fundamentals of dynamic Bayesian modeling have been reviewed and the learning and inference in DBNs have been discussed. We concentrated on switching Kalman filter (SKF) model as an example of a hybrid DBN. We proposed an SKF representation for multimodal dynamic processes and illustrate this representation by modeling the piecewise linear behavior of the top tank of the three-tank system. The SKF model has been developed from both simulated and experimental data. It has been verified that such a representation enables us to determine operating modes and estimate noise-free variables solely from measured variables. Our conclusion is that DBNs not only have the same strengths as Bayesian models, they also offer more advantages in some circumstances due to temporal aspects.

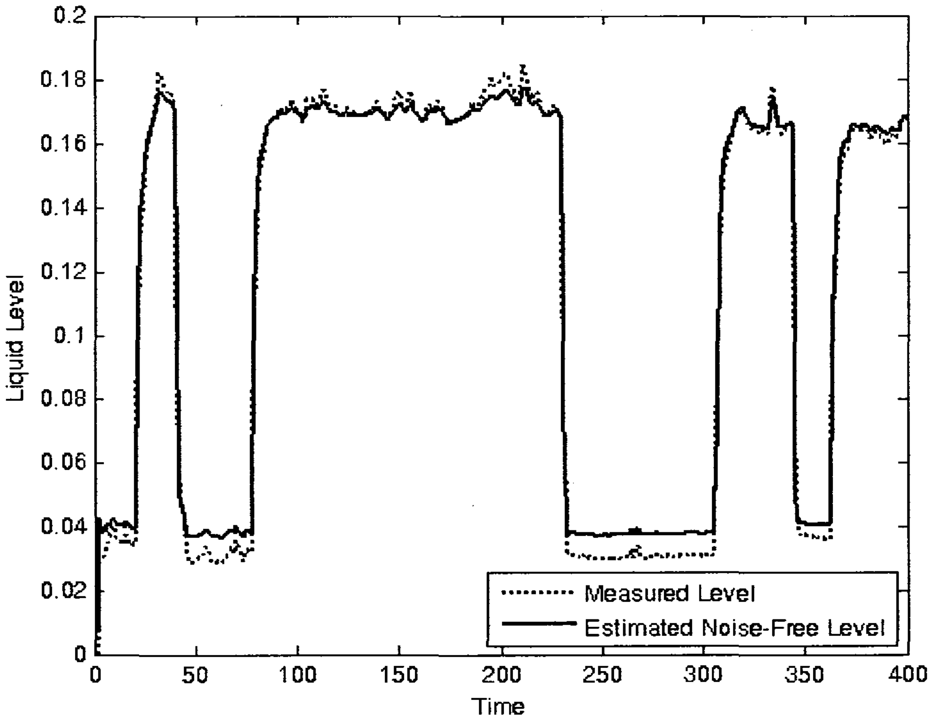

Figure 5.7. Trend comparison of measured liquid levels and estimated noise-free liquid levels for experimental data

5.7 Bibliography

1. Digalakis V., J.R. Rohlicek, and M. Ostendorf, *ML Estimation of a Stochastic Linear System with the EM Algorithm and Its Application to Speech Recognition*, IEEE Transactions on Speech and Audio Processing, Vol. 1, No.4, pp. 431-442, October 1993
2. Durbin R., S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1999
3. Franklin G. F., J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems, Third Edition*, Englewood Cliffs, NJ: Prentice-Hall, 1998
4. Huang B., Q. G. Wang, *Overview of Emerging Bayesian Approach to Nonlinear System Identification*, 2006
5. Korb K. B. and A. E. Nicholson, *Bayesian artificial intelligence*, Chapman & Hall/CRC, 2004
6. Lerner U. N., *Hybrid Bayesian Networks for Reasoning about Complex Systems*, Ph.D. Thesis, Department of Computer Science, Stanford University, October 2002
7. Murphy K. P., *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. Thesis, Department of Computer Science, University of California, Berkely, Fall 2002
8. Pernestål A., *A Bayesian Approach to Fault Isolation with Application to Diesel Engine Diagnosis*, Licentiate Thesis, KTH School of Engineering, Stockholm, Sweden, February 2007
9. Rabiner L. R., *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, Vol. 77, No. 2, pp. 257–286, February 1989
10. Rao C. V., J. B. Rawlings, *Constrained Process Monitoring: Moving-Horizon Approach*, AIChE Journal, Vol. 48, No.1, January 2002
11. Shumway R. H., D. S. Stoffer, *An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm*, Journal of Time Series Analysis, Vol. 3, No. 4, pp. 253-264, 1982
12. Sorenson H., *Kalman Filtering: Theory and Application*, IEEE Press, 1985
13. Tokatli F., A. Cinar, *Fault Detection and Diagnosis in a Food Pasteurization Process with Hidden Markov Models,* Canadian Journal of Chemical Engineering, Vol. 82, No. 6, pp. 1252-1262, 2004

14. Zweig G. G., *Speech Recognition with Dynamic Bayesian Networks*, Ph.D. Thesis, Department of Computer Science, University of California, Berkely, Spring 1998

Chapter 6

Conclusions and Future Work

In the previous chapters, the applications of Bayesian approach in soft sensor development and implementation have been illustrated. This chapter discusses the main conclusions and gives some suggestions for future research.

6.1 Summary

In this thesis, we have tried to achieve two main goals: develop a better understanding of soft computing and Bayesian method, and show that Bayesian models can be useful for soft sensor applications.

First, background material about soft sensors and Bayesian networks was presented in Chapter 1. It was shown that a Bayesian probabilistic framework can represent real-world systems, i.e. variables are described by prior distributions and dependencies are formulated by conditional distributions. We next developed a soft sensor that aimed to provide online estimation of a quality variable in the froth treatment process in Chapter 2. To do so, we combined our basic understanding of the process and non-linear regression technique to develop a gray box model. The goal was to give an introduction into soft sensor development and illustrate theoretical aspects and steps in soft sensing through the industrial case study.

Development and implementation of soft sensors entail many challenges due to the quality of collected data. Some of these problems, including measurement noises, missing measurements, and outliers, have been discussed in Chapters 3 and 4. We have demonstrated that formulating the soft sensor problem in a Bayesian framework enables us to solve these problems. The price to be paid is increased computational time due to iterative nature of Bayesian learning and inference algorithms. Chapter 3 addresses the problem of parameter estimation from noise-contaminated input and output data. Classical least squares regression and TLS have been reviewed as traditional parameter estimation approaches. Based on the available literature, classical least squares regression is not robust to data contaminated with measurement noise. Besides, the satisfactory performance of TLS is only obtained under a limiting assumption that measurement errors are independent random variables with zero mean and equal variances; otherwise, the covariance of measurement noise needs to be known. We have provided an introduction to Bayesian learning and have proposed the EM-based Bayesian algorithm (algorithm 3.2) that is robust to noisy data. The presented Bayesian-EIV framework overcomes the typical limitations of conventional parameter estimation methods. In addition, the proposed algorithm has put forward some ideas for improvement of the EM algorithm, based on iteratively freezing some parameters. The efficiency and effectiveness of the Bayesian-EIV framework have been demonstrated on numerical simulations and a pilot-scale experiment.

Chapter 4 began with an introduction to Bayesian inference theory and algorithms. Since that chapter deals with incomplete and inconsistent datasets, some of the existing treatments of missing values such as deletion and imputation methods have been reviewed. It was shown how OLS and PLS regression techniques handle missing measurements with the use of these techniques. In addition, advantages and disadvantages of each of these approaches have been discussed. A brief introduction to existing outliers detection methods such as robust regression and box plot construction criteria are provided as well. Next, Bayesian learning and Bayesian inference are combined to outline a new way of deriving and employing soft sensor models. On the other hand, we have proposed a Bayesian learning/inference algorithm (algorithm 4.1) which improves treatment of outliers. If outliers represent the random measurement errors, our algorithm is applied to compensate for the noise. Finally, the feasibility and performance of Bayesian approach was proven by demonstrating how well it can handle practical issues associated with various simulation, experimental and industrial cases. We believe that these evaluations have verified that the Bayesian approach has some real advantages over the others. Note that a Bayesian soft senor is able to address the discussed issues altogether.

Up to this point in the thesis, we have concentrated on static models. In many soft sensor applications, however, we need to model dynamic behavior of the system. We therefore turned our attention to temporal models. In Chapter 5, dynamic Bayesian networks were introduced as an alternative for modeling dynamic processes. We provided an overview of DBNs representation and briefly discussed parameter learning and probabilistic inference in these models. Next, we concentrated on SKF models that represent switching dynamics in multimodal processes as an approximation of non-linear processes. To illustrate this approach, a SKF model has been developed to represent the piecewise linear behavior of the top tank in the three-tank setup discussed in Chapter 3. It has been verified that such a representation enables us to determine operating modes and estimate noise-free variables solely from measured variables. This case study may prove of use in the development of soft sensors for multimodal systems.

6.2 Limitations of Bayesian modeling

In spite of the remarkable potential of Bayesian models in soft sensor development, there are some limitations to their application as listed below.

1. All conclusions drawn from the posterior distribution depends on the quality and extent of the prior probability used in Bayesian inference processing. As a result, a Bayesian network is only as useful as this prior knowledge is reliable. Nevertheless, the more data that are collected, the less influence the prior distribution has on the posterior distribution.
2. Bayesian theory and software are well-established for handling discrete variables. However, they are not as mature in dealing with continuous variables.
3. Once a Bayesian model is developed, it might be unable to respond to some previously unforeseen cases. This limitation should be carefully considered in soft sensor applications including fault diagnosis.

6.3 Future works

It is our hope that this thesis demonstrates the effectiveness of Bayesian modeling in soft sensor applications and provides useful algorithms and frameworks to address some of the practical issues of soft sensor development. However, there are several open issues to be investigated to further extend the applicability of the Bayesian models in soft sensors. Here, we summarize some of the potential future works as follows.

Some of the practical issues in soft sensor development have been identified and a Bayesian approach has been applied to deal with them. Clearly much more needs to be done along these lines. For instance, it has been shown that the proposed EM-based Bayesian algorithm is robust to noisy measurements under a normal error assumption. However, it would be more practical to also consider the case in which measurement noises have non-Gaussian distributions. In addition, the performance of the presented algorithms and frameworks can be further evaluated through their applications on more complex models.

DBNs are presented as an alternative for modeling dynamic processes. It would be interesting to compare performance of DBNs with other representations of dynamic systems such as unscented Kalman filters (UKF) and particle filters.

Further, the use of hybrid dynamic Bayesian models in fault diagnosis is an attracting area of application. We believe that the case study presented in Chapter 5 will serve as a motivation for other researches to pursue the use of Bayesian modeling in this field.

Appendeix A

Nomenclature

A'	Cross section of the IPS I vessel
A''	Cross section of the IPS II vessel
C_i	Resistance of the output orifice of i^{th} tank
$\mathcal{D}$	Set of independent and identically distributed observations
$\boldsymbol{D}_m$	Unobserved variables in a data set
$\boldsymbol{D}_o$	Observed variables in a data set
$E(X)$	Expectation of X
F_1	Undiluted feed flow rate
F_2	Under-flow rate from IPS
F_3	Naphtha flow rate
F_4	Diluted feed flow rate
G	Total feed flow rate to IPS I
H_i	Fluid level in the i^{th} tank
I_1	Diluted feed flow rate to the first IPS I
J_1	Diluted feed flow rate to the first IPS II
$\mathcal{M}$	Model structure
$N(0,\varepsilon_i)$	Normal Gaussian distribution
$P(X)$	Probability distribution of X
$P(X \mid Y)$	Conditional distribution of X given Y

$P(X \mid Y = y)$	Conditional distribution of X given $Y = y$
$P(X = x)$	Probability that X takes the value x
$Par(X)$	Set of parents of X
Q	Covariance of the state noise
$Q_{0.25}$	First quartiles
$Q_{0.75}$	Third quartiles
R	Covariance of the observation noise
T	Total feed flow rate to IPS II
V_i	Fluid volume in the i^{th} tank
W	Regression matrix
Wb	Weight ratio of dry bitumen
Wn	Weight ratio of Naphtha
Ww	Weight ratio of water
X°	Noise-free X
e_i	Noise associated with the measured variable X_i
p_{ji}	Path coefficients for X_i a X_j
q	Inflow to the top tank
r_{Yi}	Correlation coefficient between Y and X_i
v_d	Contribution from direct effects
v_i	Contribution from indirect effects
v_t	Observation noise
v_u	Contribution from unknown sources
w_t	States noise
Δt	sampling time
Θ	Model parameters
α	Normalizing constant
ε_i	Variance of X_i

γ_c	Completeness index
γ_d	Significance index
μ_i	Mean of X_i
ρ	Density

Appendeix B

Algorithm Description of Total Least Squares (TLS)

This appendix provides a brief overview of the total least squares algorithm used in Chapter 3. For detailed discussion of this method the reader is referred to [1].

The total least square (TLS) method is the extension of the classical regression technique that has been applied to compensate for variable noises. Under specific conditions, the TLS solution computes optimal parameter estimates in classical EIV models. These models are represented by exact linear relations of form (B.1):

$$Z_n^T l = r(n) \tag{B.1}$$

where,

- Z_n is a vector that corresponds to m measurements sampled at time n
- l plays the role of an unknown parameter vector that characterizes the special model
- $r(n)$ denotes a linear combination of the measurement noises and the other disturbances

In order to identify the true parameters of the models, the sum of squared of the equation error need to be minimized:

$$J = \frac{1}{2N}\sum_{n=1}^{N} r^2(n) \qquad \text{subject to} \quad l^T l = I \tag{B.2}$$

Thus, we have

$$\frac{1}{N}\sum_{n=1}^{N} Z_n Z_n^T l = \lambda l \tag{B.3}$$

The TLS estimates, say l_0, are given by the eigenvector of $\frac{1}{N}\sum_{n=1}^{N} Z_n Z_n^T l$ which corresponds to the smallest eigenvalue, say λ_0 :

$$\frac{1}{N}\sum_{n=1}^{N} Z_n Z_n^T l_0 = \lambda_0 l_0 \Rightarrow \lambda_0 = \frac{1}{N} l_0^T \sum_{n=1}^{N} Z_n Z_n^T l_0 \qquad \textbf{(B.4)}$$

The above stated algorithm corresponds to EIV regression model with the restrictive condition that the measurement errors are independent random variables with zero mean and equal variances.

Bibliography

1. Huffel S. V. and J. Vandewalle, *The Total Least Squares Problem : Computational Aspects and Analysis*, Society of Industrial and Applied Mathematics, Philadelphia, 1991

Appendeix C

Comments on Bayesian Software

This appendix comments on particular features of some of the major Bayesian software. The software package names, their authors, and their web locations are given in Table C.1. Note that we have personal experience, through this research project, with the following software: BNT, WinBUGS, and Netica. A resource guide to the other Bayesian packages can be found in [1].

C.1 Netica

Netica is a user-friendly Bayesian tool that is commercially available since 1995. It can be used to build and learn Bayesian networks, as well as perform different types of inference tasks. It is also capable of representing Dynamic Bayesian networks. Netica can learn probabilistic relations from data through the application of Spiegelhalter & Lauritzen parameterization, EM, or gradient descent algorithms; missing values are allowed. In addition, the relationships between variables may be entered as individual probabilities or in the form of equations. However, only parameter learning is supported by this package. Netica discretizes the continuous variables by partitioning their domain into some finite number of subsets. Since the representation of the resulting discretized model is exponential in the number of variables, this approach becomes problematic in large or complex networks. For example, the number of parameters represent a Gaussian distribution over N variables is $O(N^2)$ .If we discretize these variables into m ranges, then this approach requires $O(m^N)$ elements to be learnt and stored.

Netica's exact general probabilistic inference is based on the message passing in a junction tree of cliques, which is the fastest available algorithm. Once a Network is created, we can answer queries or find optimal decisions. Given a case of new observations, both posterior probability of queries and most probable explanation (MPE) can be found. Netica has facilities to enter and update only individual cases, and it does not handle sets of cases.

In conclusion, Netica is suitable for application in the following areas: diagnosis, prediction, decision analysis, sensor fusion, expert system building, probabilistic modeling, and certain kinds of statistical analysis [3].

C.2 Bayesian Net Toolbox (BNT)

Bayes Net Toolbox (BNT) is the other Bayesian modeling and inference packages. Taking advantages of MATLAB features, BNT has become a widely used and powerful Bayesian software since 2002. BNT suffers from the lack of GUI, which is currently made up by MATLAB visualization tools; a preliminary attempt to make a GUI has been done by Philippe LeRay [2]. It can build and learn static and dynamic Bayesian networks, as well as answer queries or find optimal solutions using its powerful inference engine. BNT does not allow the entry of probabilistic relations by equation. BNT supports both parameter learning and structure learning by several learning algorithms such as EM and MCMC algorithms. BNT treats continuous variables as continuous without trying to discretize them. It allows only linear relations between the continuous variables and does not allow discrete nodes to have continuous parents. In addition, non-Gaussian probability distributions of continuous variables are not supported. Inference task in static and dynamic Bayesian networks are performed by many different exact and approximate inference algorithms.

Finally, BNT is applicable for implementation of the following probabilistic models: linear regression, logistic regression, mixtures of Gaussians, DBNs (such as HMM, Kalman filters, switching Kalman filters and ARMAX models), factor analysis, probabilistic PCA, and many others [2].

Table C.1. The Bayesian software packages experienced through the course of this thesis

Name	*Webpage*	*Author*
BNT	http://bnt.sourceforge.net/	Murphy
Netica	http://www.norsys.com/	Norsys
WinBUGS	http://www.mrc-bsu.cam.ac.uk/bugs/	MRC/Imperial College

C.3 WinBUGS

WinBUGS is the most advanced version of BUGS (Bayesian Inference Using Gibbs Sampling) that provides Bayesian analysis of statistical models using MCMC. Since MCMC is inherently less robust to the prior information than analytic statistical methods, prior knowledge plays an important role in the accuracy of a Bayesian model identified by WinBUGS. A wide range of non-Gaussian probability distributions for discrete and continuous variables is provided. WinBUGS allows the entry of probabilistic relations by equation, and supports nonlinear relations between the continuous variables. If model parameters are strongly related, however, the model convergence may be very slow. Therefore, this program is inefficient for time series structures such as HMM. A new observation can be predicted by specifying it as missing in the data set and assigning it a uniform prior.

Bibliography

1. Korb K. B. and A. E. Nicholson, *Bayesian artificial intelligence*, Chapman & Hall/CRC, 2004
2. Murphy K. P., *How to Use Bayes Net Toolbox,* Tutorial of BNT Toolbox, Department of Computer Science, University of California, Berkley, June 2004
3. Norsys Software Corp., *Netica*: *Application for Belief Networks and Influence Diagrams*, User's Guide, 2007