

“Watch your thoughts, for they become words. Watch your words, for they become actions. Watch your actions, for they become habits. Watch your habits, for they become character.

Watch your character, for it becomes your destiny.”

Unknown

University of Alberta

**Bayesian Methods for On-Line Gross Error Detection and
Compensation**

by

Ruben Gonzalez

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

Chemical and Materials Engineering

©Ruben Gonzalez

Fall 2010

Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

Examining Committee

Dr. Biao Huang (Supervisor)

Dr. Philip Choi (Chair)

Dr. Vinay Prasad

Dr. Bob Koch (External)

This thesis is dedicated to God, who is the only entity who truly
understands what I do

Abstract

Data reconciliation and gross error detection are traditional methods toward detecting mass balance inconsistency within process instrument data. These methods use a static approach for statistical evaluation. This thesis is concerned with using an alternative statistical approach (Bayesian statistics) to detect mass balance inconsistency in real time.

The proposed dynamic Bayesian solution makes use of a state space process model which incorporates mass balance relationships so that a governing set of mass balance variables can be estimated using a Kalman filter. Due to the incorporation of mass balances, many model parameters are defined by first principles. However, some parameters, namely the observation and state covariance matrices, need to be estimated from process data before the dynamic Bayesian methods could be applied. This thesis makes use of Bayesian machine learning techniques to estimate these parameters, separating process disturbances from instrument measurement noise.

Acknowledgements

First of all, I would like to personally thank God, firstly for actively filling me with joy, strengthening me to persevere, and enabling me to *enjoy* doing this work to a degree that some have considered to be unnatural.

I would like to thank my supervisor Dr. Biao Huang with whom I had the sincere pleasure of working with. He has mastered the art of practical yet tactful honesty, and was instrumental in the developing the essential yet rudimentary skills that I lacked when I first started my work. During weekly meetings Dr. Huang has demonstrated patience and foresight causing these meetings to be effective.

I would also like to thank Dr. Dave Shook for whom I worked as a Lab T.A. and who served as my contact for the ISA for half my tenure as president of the ISA student section. He was always one to provide practical perspectives to my work which I hope never to forget. I would also like to thank Dr. Stevan Dubljevic who served as my ISA contact for the rest of my tenure. I would also like to thank my colleagues who also studied under Dr. Huang; namely Qi Fei, Zhu Yijia, Shao Xinguang, and Elom Domlan who, each in their own way, provided me with encouragement and practical aid. I would also like to thank my non-academic friends for their support, specifically Daniel Scott, Jonathan Ritchie, Zhang Wenzhuo, Flora To, Zhao Lei, and Wei Bingqing.

I would also like to acknowledge the Financial support from Natural Sciences and Engineering Research Council of Canada (NSERC) and Syncrude Canada Ltd.

I am very grateful for my parents who have encouraged me to think for myself and to be consistent with my principals; I thank them for steadfastly believing in my abilities, encouraging me to pursue a career in advanced fields of engineering. Even more specifically, I would like to thank my mother for teaching me to be patient, and my father for teaching me to be persistent.

Contents

1	Introduction	1
1.1	Objective of this thesis	1
1.2	Introduction to the Bayesian Philosophy	1
1.3	Applying Bayesian Philosophy to Data Reconciliation and Gross Error Detection	3
1.4	Outline of the thesis	6
1.5	Contributions of the thesis	8
2	Reconciling continuum and non-continuum data with industrial application	9
2.1	Introduction	9
2.2	Subsystem for Preliminary Analysis	11
2.3	Conditioning Method for WENCO Input	13
2.4	Binomial Distribution and its Effects on Mapping	17
2.5	Principal Component Regression and Performance Metrics	20
2.5.1	Principal Component Regression	20
2.5.2	PCR Performance Metrics	21
2.6	Applying PCR to Verify Smoothing	23
2.7	Applying PCR to Estimate Surge Hopper Scaling	24
2.8	Conclusions	27
3	Estimation of Measurement Noise Variance and Bias Using Bayesian Methods for Process Data Reconciliation	29
3.1	Introduction	29

3.1.1	A note on Notation	31
3.2	Motivation	32
3.2.1	Industrial Example	32
3.2.2	Problem Formulation	32
3.2.3	Mass Balances, Factor Model Representation	34
3.2.4	Conversion to a General Factor Model	35
3.2.5	Bayesian Networks	36
3.3	Bayesian Networks and Variance Estimation	37
3.3.1	Bayesian Example	37
3.3.2	Bayesian Parameter Learning with Hidden Variables	37
3.3.3	Principal Component Analysis	39
3.3.4	Factor Analysis by Modified Principal Components	40
3.4	Estimating the Gross Error Distribution	41
3.5	Simulation of Overall Process	42
3.5.1	Variance Estimation Results: EM by Uninformed Initial Guesses	43
3.5.2	Variance Estimation Results: PCA and FA Results	44
3.5.3	Variance Estimation Results: Bayesian Learning with FA Initial Values	46
3.5.4	Bias Estimation: Bayesian Learning Results	47
3.6	Special Case, Capacity with no Scaling	48
3.6.1	Time Dependency As Information	48
3.6.2	Regression Results Using Industrial Process Data	50
3.7	Industrial Application	50
3.8	Conclusions	52

4 Dynamic Bayesian Approach to Instrument Performance Monitoring and Correction **53**

4.1	Introduction	53
4.1.1	Algorithm Overview	55
4.2	Step(1) State Estimateion	56

4.2.1	Static Observation Model	56
4.2.2	Hidden Dynamic Process Model	56
4.2.3	Augmented Gross Error States	58
4.2.4	Observability	59
4.2.5	Model Estimation	61
4.3	Step (2) Residual Error Distance	61
4.3.1	Residual Error	61
4.3.2	Statistical Distance	63
4.4	Step (3) Systematic Error Updating, Filtering Z-Values	64
4.5	Step (4) Reliability Audit through Hypothesis Testing	64
4.5.1	Recursive Hypothesis Testing	64
4.5.2	C Matrix Based on Fixed P-Values	66
4.5.3	C Matrix Based on Adaptive P-Values	68
4.6	Tuning Steps (1) and (3)	68
4.6.1	Tuning Step (3)	68
4.6.2	Tuning Step (1)	70
4.7	Simulation of Dynamic System	72
4.8	Preliminary Industrial Application	75
4.9	Conclusions	75
5	Conclusions	77
5.1	Concluding Remarks	77
5.2	Recommendations for future work	77
	Bibliography	79

List of Figures

1.1	Static Bayesian Example	2
1.2	Dynamic Bayesian Example	3
1.3	Bayesian Network Example of Kalman Filter	4
1.4	Bayesian Network Example of Factor Model	4
1.5	Gross Error Detection DBN	5
1.6	Proposed Method	5
1.7	Solids Handling and Slurry Preparation	6
1.8	Implementation Prerequisite Scheme	7
2.1	Chapter Methodology	10
2.2	Solids Handling	11
2.3	WENCO and Weightometer Histograms	13
2.4	Optional caption for list of figures	18
2.5	Optional caption for list of figures	19
2.6	Optional caption for list of figures	22
2.7	Optional caption for list of figures	23
2.8	Weightometer-Weightometer Based Estimates of Hopper Scaler	25
2.9	PRS and PD for Weightometer-Weightometer Based Estimates	26
2.10	PRS and PD for Weightometer-Unsmoothed WENCO Based Estimates	27
2.11	PRS and PD for Weightometer-Smoothed WENCO Based Es- timates	27
3.1	Chapter 3 Methodology	31
3.2	Solids Handling and Slurry Preparation	33

3.3	Predictive Reasoning	36
3.4	Diagnostic Reasoning	36
3.5	Bayesian Representaion of Mass Balance Factor Model	38
3.6	Bayesian Representation with Bias	41
3.7	Optional caption for list of figures	44
3.8	Approximate Factor Analysis Model	45
3.9	Optional caption for list of figures	50
3.10	Optional caption for list of figures	51
4.1	Chapter 4 Methodology	54
4.2	Visual Representation of Algorithm	55
4.3	Static Bayesian Network	57
4.4	Dynamic Bayesian Network without Gross Errors	58
4.5	Dynamic Bayesian Network with Gross Errors	60
4.6	Dynamic Bayesian Network without Bias	66
4.7	Visual Representation of Gross Error Hypothesis Test	67
4.8	Gross Error Tracking (instruments 6 and 4)	74
4.9	Gross Error Probability (instruments 6 and 4)	74
4.10	Optional caption for list of figures	76

List of Tables

3.1	Variable Definitions	42
3.2	Correlation Matrix for Direct Nonlinear System	43
3.3	PCA vectors by decreasing PC	44
3.4	Loadings and Noise from FA Estimation vs. Simulated Values	46
3.5	Summary Table of Variance Estimates	46
3.6	Estimated Bias from Bayesian Learning (Zero Bias Initial) . .	47
3.7	Estimated Bias from Bayesian Learning (Modified Bias Initial)	47
3.8	Estimated Variance from Bayesian Learning (FA Initial) . . .	51
3.9	Industrial Estimated Variance from Bayesian Learning	52
4.1	Variable Definitions	72

List of Symbols

a	A vector of indices pertaining to active gross error states
A, C	State transition and observation matrices of a state space model
$A_{(a)}, C_{(a)}$	State transition and observation matrices of a state space model as defined by the set of active bias states a
B_{obs}	Observation of unaccounted gross errors (1 = True, 0 = False)
c_H	Scalar for surge hopper level indicator
d_M	Univariate Mahalanobis distance
D_M	Multivariate Mahalanobis distance
e	A single eigenvector
E	Complete matrix of eigenvectors
E_q	A set of eigenvectors containing the first q principal components
I	Identity matrix
I_Y	An identity matrix having a length equal to the number of process instruments p
$I_{Y(a)}$	An identity matrix with a length equal to the number of active gross error states

$I_{\beta(a)}$	A matrix that contains the columns of identity matrix I_Y that correspond to the values of a ; e.g. if $a = [1, 5, 6]$ $I_{\beta(a)}$ is a $p \times 3$ matrix containing the first, fifth and sixth columns of I_Y .
k_s	Safety factor that takes a value between 0 and 1
K_1, K_2, K_3	Memory constants of states: takes values between 0 and 1, 0 representing no memory, 1 representing perfect memory (random walk)
K_{err}	Error constant used to reduce the weight of corrupted instruments
L	Loading matrix estimated by FA
L_s	Standardized loading matrix estimated by FA
\tilde{L}_s	standardized loading matrix estimated by PCA
L_x	Loading matrix for hidden state
$\mathbf{L}_x(\Delta t)$	Loading matrix for hidden hopper level state (integration time dependent)
M_{eff}	Load Effect Matrix
n	Standardized instrument noise with mean 0 and covariance I
$\sim N(\mu, \Sigma)$	Distributed as a multivariate normal distribution with mean μ and covariance Σ
O	Observability matrix
p	Number of process instruments
P	Posterior covariance matrix from general state space model

$P(e)$	Probability of evidence(normalization constant)
$P(e h)$	Probability of evidence given hypothesis (likelihood)
$P(h)$	Prior probability of hypothesis (prior probability)
$P(h e)$	Probability of hypothesis given evidence (posterior probability)
P_X	Posterior covariance matrix of hidden state estimate for process state Kalman filter
P_Z	Posterior covariance matrix of hidden state estimate for residual error Kalman filter
PD	Percent Definitiveness
PRS	Percent Relative Scatter
PVC	Percent Variance Captured
q_z	Hidden state variance for a single standardized residual error
Q	Hidden state covariance matrix from general state space model
Q_X	Hidden state covarinace matrix for process state Kalman filter
Q_Z	Hidden state covariance matrix for residual error Kalman filter
Q_β	Gross error state covarinace matrix for process state Kalman filter
r	Constant offset reference from Taylor series approximation

r_Z	Observation variance for a single standardized residual error estimation
R	Sample correlation matrix
R_Y	Observation covariance matrix for process state Kalman filter
$R_{\bar{Y}}$	Observation covariance matrix for corrupted instruments
R_Z	Observation covariance matrix for residual error Kalman filter
s	A vector of indices pertaining to static gross error states
S	Sample covariance matrix
S_y	Diagonal elements of sample covariance matrix
v_Y	Observation noise
v_Z	Observation noise of the standardized systematic gross error state
w_X	Process disturbances of hidden process state
w_Z	Deviations in the standardized systematic gross error state
w_β	Process disturbances of hidden gross error state
x	Single process state
X	Complete set of process states
y	Single instrument observation
Y	Complete set of instrument observations
z	Single standardized Gaussian random variable with mean 0 and variance 1 (a single standardized residual error)

\hat{z}_O	Optimal filtered cutoff value for residual error estimates (scalar)
Z	Vector of standardized Gaussian random variables with mean 0 and covariance I (vector of standardized residual errors)
Z_F	Posterior filtered residual errors (gross errors)
\hat{Z}_O	Optimal filtered cutoff value (vector)
Z_u	Standardized systematic error/gross error state
α	Confidence level, or tolerance toward false positive gross errors
β	Gross errors or bias
β_{min}	Desired minimum detectable gross error
δ	Cross-correlated noise (often pertaining to the hidden state or error in dynamic state estimates)
δ_X	Cross-correlated process noise
δ_u	Underlying gross error noise (for gross error filtering)
Δ_r	residual measurement error
Δt	Integration time (sampling time)
λ	A single eigenvalue
Λ	A complete diagonal matrix of eigenvalues
Λ_q	A set of eigenvalues representing the first q principal components
μ	Mean

μ_X	Mean of hidden state X
ρ	Density
Ψ	Specific variance
Ψ_s	Standardized specific variance
σ^2	Variance
σ_1^2	Variance of hopper level readings (integration time independent)
$\sigma_{2(\Delta t)}^2$	Variance of weightometer difference (integration time dependent)
σ_{ZP}	Posterior standard deviation of standardized residual error
$\sigma_{\beta SP}^2$	standardized gross error prior variance
Σ	Covariance matrix
Σ_r	Residual error covariance matrix
Σ_X	State covariance matrix
Σ_Y	Observation covariance matrix
$\Sigma_{\beta 0}$	Prior gross error covariance matrix (used for gross error detection)
θ	Bayesian network parameters
ε	Uncorrelated noise
ε_Y	Uncorrelated measurement noise
ε_β	Uncorrelated noise representing gross error change

List of Abbreviations

<i>WENCO</i>	A database containing mining records, including truck dumps
<i>PCA</i>	Principal Component Analysis
<i>PCR</i>	Principal Component Regression (linear Total Least Squares)
<i>ELR</i>	Expected Load Response (data smoothing model)
<i>FA</i>	Factor Analysis
<i>IID</i>	Independent and Identically Distributed
<i>PVC</i>	Percent Variance Captured
<i>PD</i>	Percent Definitiveness
<i>PRS</i>	Percent Relative Scatter
<i>EM</i>	Expectation-Maximization
<i>BNT</i>	Bayesian Network Toolbox
<i>DBN</i>	Dynamic Bayesian Network
<i>CPT</i>	Conditional Probability Table

Chapter 1

Introduction

1.1 Objective of this thesis

The objective of this thesis is to develop dynamic methods to address the problem of mass balance inconsistency which is common to instrument data within the chemical process industry. Due to the presence of noise within instrument readings, data is often contaminated by random errors, but can also be contaminated by gross errors which are less frequent but often times more serious [22]. Examples of gross errors include process disturbances, instrument bias, process leaks or departure from steady-state [22]. Instrument bias can often be detrimental if it is used as an input for a control system, while process leaks can often affect plant profitability and safety [8]. Thus in many cases, process operation, safety, and bookkeeping can be improved by implementing a method that can detect mass balance inconsistencies as soon as they occur.

Traditional methods of dealing with mass balance consistency are data reconciliation and gross error detection. For a process operating at steady-state, data reconciliation seeks to estimate the set of process values that are most most likely to result in the observed instrument readings. Gross error detection however, uses statistical testing procedures to identify data sets that systematically violate mass balance principals. If a gross error is detected, the corrupted data sets are removed and another attempt is made to estimate the reconciled state.

1.2 Introduction to the Bayesian Philosophy

Data reconciliation and gross error detection are techniques typically performed after the fact; however, there is a demand for implementing real-time techniques to allow for earlier detection of gross errors which is the objective of this thesis. The proposed dynamic approach can be performed using Bayesian techniques.

The key element to Bayesian techniques is the application of Bayes' theorem:

$$P(h|e) = \frac{P(e|h)P(h)}{P(e)} \quad (1.1)$$

where $P(e|h)$ is the probability of the evidence given a hypothesis, and is also referred to as the *likelihood* evidence, $P(h)$ is the prior probability of the

hypothesis, $P(e)$ is the probability of all possible evidence, or is often called a *normalizing constant* to ensure that $P(e|h)$ is a probability density function that integrates to 1, and $P(h|e)$ is the probability of the hypothesis given that it is updated with new evidence, often referred to as the *posterior* probability.

Since the Bayesian philosophy focuses on updating prior hypothetical beliefs using new evidence, Bayesian techniques are an intuitive solution to dynamic estimation problems. Because Bayes' theorem is naturally geared toward diagnosing hidden causes of observed data, it is often helpful to represent the problem as a Bayesian network, which organizes a set of related events in terms of causality. Consider the example shown in Figure 1.1. If one looks at the ground and finds that it is wet and that there are no shadows (possibly due to cloud cover), one can speculate between whether or not it was caused by rain or by a sprinkler. In order to use Equation (1.1) we need to know the prior probability of Rain = T/F and Sprinkler = T/F which defines $P(h)$. Furthermore, we also need to know the conditional probability of any combination of Shadows = T/F, and Wet Ground = T/F given the possible combinations of Rain = T/F and Sprinkler = T/F which defines $P(e|h)$. Often times values of $P(e|h)$ are stored in a *Conditional Probability Table (CPT)*. $P(e)$ is simply the sum of all possible values of $P(e|h)P(h)$ so that probabilities are normalized.

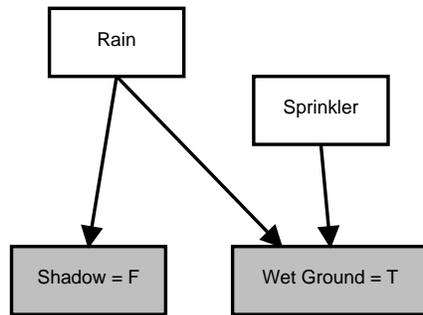


Figure 1.1: Static Bayesian Example

When applied to a dynamic problem, hidden states (both continuous and discrete) at the present time have an effect both on the current observations and the state at the subsequent time interval. When evaluated in this manner, these methods can be classified as Dynamic Bayesian Networks (DBNs). One can convert the static problem in Figure 1.1 into the dynamic problem in Figure 1.2. In this example, we now consider evidence observed 5 minutes ago. If we observed no shadows and wet ground now, the evidence would point toward rain. However, if 5 minutes ago we observed that there were shadows and wet ground, the evidence 5 minutes ago would have pointed toward the sprinkler. This would modify our prior belief toward rain in the current time step, resulting in a stronger posterior belief of the wet ground being caused by the sprinkler.

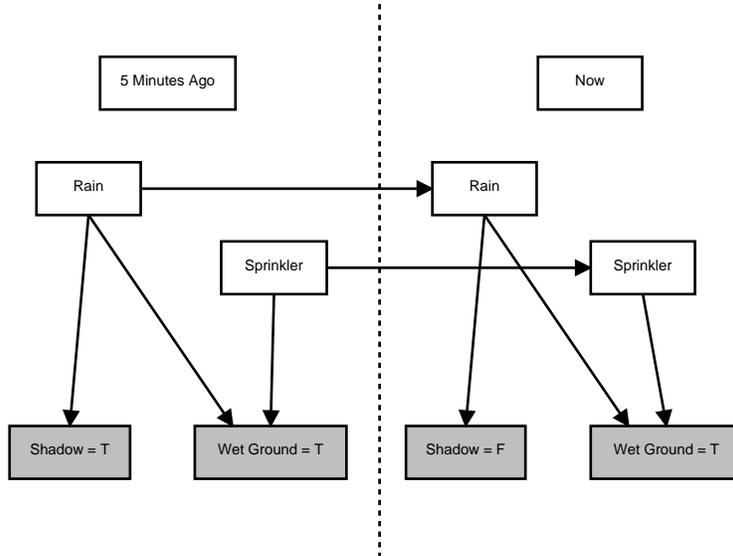


Figure 1.2: Dynamic Bayesian Example

1.3 Applying Bayesian Philosophy to Data Reconciliation and Gross Error Detection

Data reconciliation is a steady-state estimation technique; it attempts to use multiple instruments and combine them to estimate a more reliable steady-state. This works under the assumption that the states are not changing. State estimation however, already has a well established dynamic analogue, that being the Kalman filter which was proposed by R.E. Kalman in 1960 [13]. One in determining this analogue, one must consider the key difference between data reconciliation and Kalman filtering. Data reconciliation obtains the estimate of a steady state that best explains all the collected data [8]. It assumes that the state does not change. However, the optimal Kalman filter obtains the best estimate of the current state given the previous state and its uncertainty [14], it allows assumes that the state can have pre-defined dynamics for change, but is also subject to disturbances caused by various inputs. Thus it weighs the evidence of change against the prior belief of change.

It has been proved that discrete-time Kalman filtering is consistent with Bayes' theorem [17]; in fact, Kalman filtering is a technique that can be used to solve certain DBN structures that consist of Gaussian variables. However, current solution methods to solving DBNs are more general, being able to solve more complex model structures that include discrete-valued variables. Nevertheless, the Kalman filter can easily be converted into a Bayesian network as shown in Figure 1.3 where Gaussian variables are represented by elliptical nodes, and A and C represent the state transition and observation matrices.

Without dynamic relationships, the state space model represented in Figure 1.3 simplifies to a factor model represented in Figure 1.4, where relationships between individual latent variables x and observed variables y are explicitly shown by scalars $c_{[i,j]}$.

Gross error detection makes use of statistical hypothesis testing to deter-

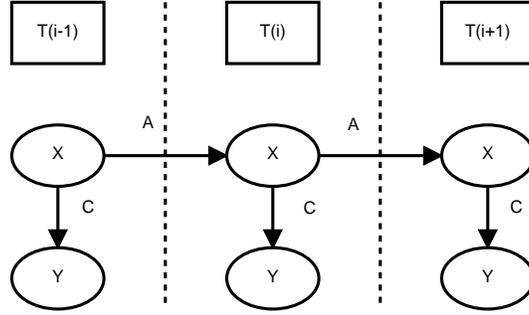


Figure 1.3: Bayesian Network Example of Kalman Filter

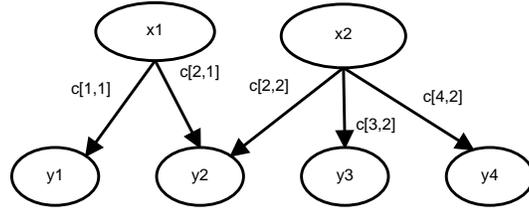


Figure 1.4: Bayesian Network Example of Factor Model

mine whether there is enough evidence to reject the null hypothesis that no gross errors exist. While this type of problem cannot be solved by means of Kalman filtering, gross error detection can nevertheless be formulated as a DBN. Gross error detection deals with a discrete state that can take two possible values: 1, the existence of gross active gross errors, or 0 the absence of active gross errors. Thus, the DBN can be formulated to use test results and their accuracy to recursively monitor the evidence toward the existence of active gross errors as shown in Figure 1.5. In this figure, B represents gross errors and $Bobs$ represents test results while A and C are CPTs pertaining to the probability of switching and false test results respectively.

The end objective of this thesis is to formulate an algorithm that uses a series of Bayesian networks to monitor gross errors such as process leaks and instrument bias. Because of the nature of the problem it addresses, this technique is inherently related to data reconciliation and gross error detection. The overall procedure for the proposed method is discussed in Chapter 4 but can be summarized in Figure 1.6.

Referring to Figure 1.6, Step (1) is performed by Kalman filtering. Due to the fact that beliefs of the instrument reliability change, the state Kalman filter must be appropriately modified according to the reliability audit results. Step (2) uses state estimates to calculate and standardize the residual error. Standardization diagonalizes the covariance matrix so that the statistical distance can be calculated for each individual instrument. Step (3) uses a Kalman filter to smooth out random noise in the residual error estimates, enabling the proposed method to be more sensitive to systematic error. Finally, Step (4) uses filtered statistical distances to calculate P-values which can be used in the DBN. The DBN is used to weigh the probability of the tests being wrong against the probability of an uncorrected gross error existence switching be-

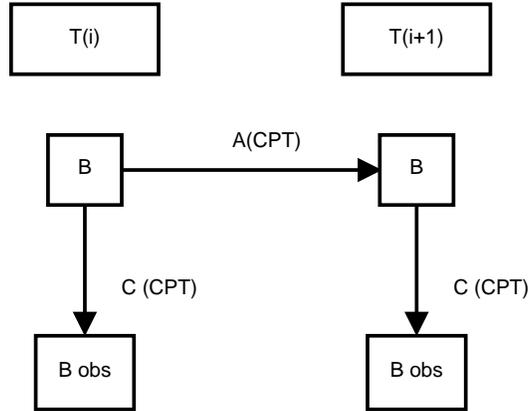


Figure 1.5: Gross Error Detection DBN

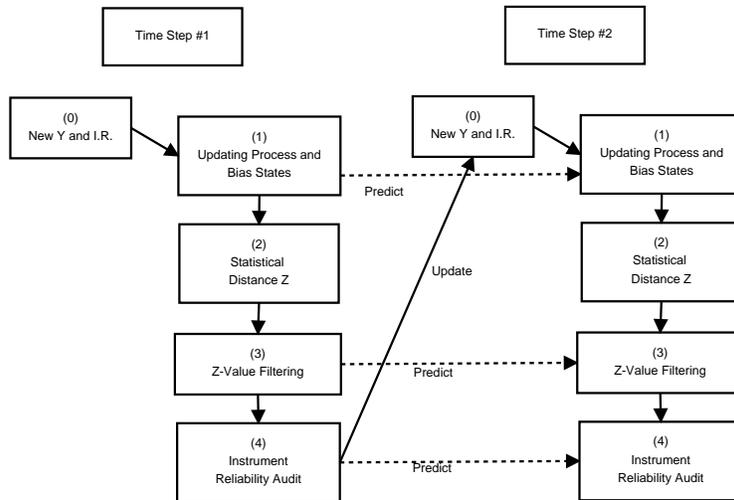


Figure 1.6: Proposed Method

tween T or F .

This thesis applies the proposed method to an industrial solids handling and slurry preparation system shown in Figure 1.7 which contains seven instruments:

1. Truckload values from WENCO Mining Database (WENCO y_1)
2. Crusher Weightometer (WI 1) y_2
3. Surge Level Indicator (LI 1) y_3
4. Mix Box Feed Conveyor Weightometer (WI 2) y_4
5. Slurry Flow Meter (FM S) y_5

6. Slurry Density Meter (DM S y_6)

7. Water Flow Meter (FM W y_7)

The principal concern for the industry are weightometers such as y_2 and y_4 which are often prone to sudden measurement error due to mechanical failure, as well as y_6 which tends to have a bias that gradually drifts. In terms of reconciliation WENCO y_1 data is problematic because it does not exhibit a normal distribution, while the surge pile level indicator y_3 is in units of % Level with no discernable method to convert to mass units. Because of this, both the WENCO data and level indicator data need to be conditioned.

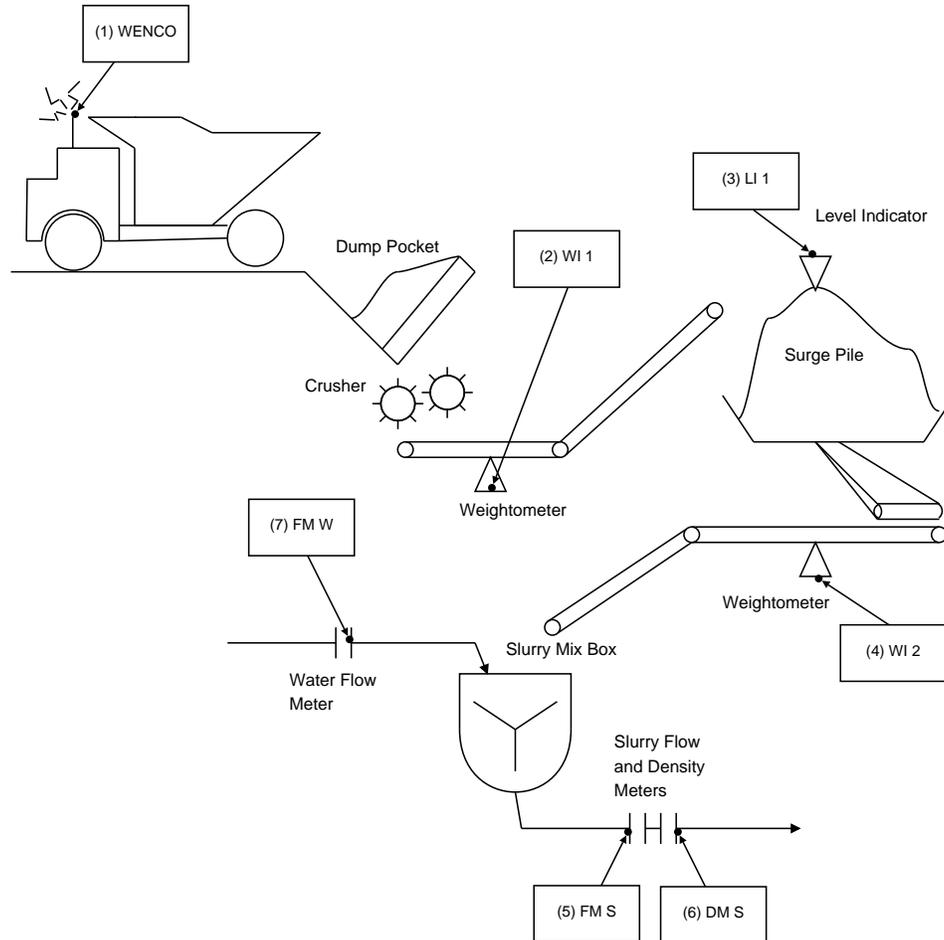


Figure 1.7: Solids Handling and Slurry Preparation

1.4 Outline of the thesis

This chronology of this thesis is based on the process that was required to implement the proposed method. Because of this, the proposed method is not

discussed until Chapter 4, as the model estimation in Chapter 3 and data conditioning in Chapter 2 must be done before the solution can be implemented. The order of prerequisite procedures is shown in Figure 1.8.

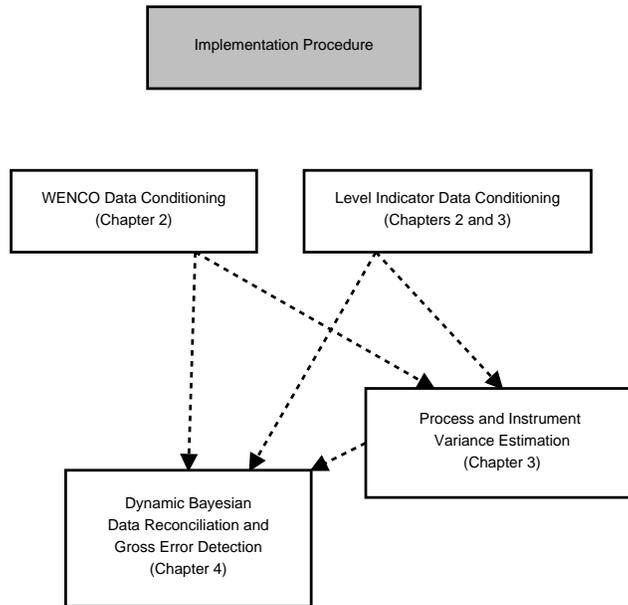


Figure 1.8: Implementation Prerequisite Scheme

Chapter 2 deals with conditioning mass balance information that arrives in discrete loads so that it can be reconciled with continuous data later on in Chapters 3 and 4. An industrial case study is shown where truck load mass is recorded at the time it was dumped into a dump pocket. There is no level indicator on this pocket, and thus it is difficult to perform short-term reconciliation between truck load data and the downstream weightometers readings without knowledge of the hopper contents. This can be resolved by taking into account that truck loads are discrete and follow a binomial distribution, and deriving an appropriate smoothing procedure. The other challenge that is addressed in Chapter 2 is the conditioning of level indicator data by estimating an appropriate scaling factor. The method proposed in Chapter 2 uses Principal Component Analysis (PCA) to estimate the conversion factor between between % capacity and mass.

Chapter 3 focused on the estimation of instrument and process variance from industrial data. Mass balances were used to develop a factor model which was applied to the data; this model was converted to a Bayesian network so that Bayesian learning could be applied. The Bayesian network learning method was used because it allowed the state observation matrix C to be specified, thus constraining the process and instrument measurement noise variance estimates to be consistent with mass balances. Bayesian learning must be performed using a numerical method, a popular method, the Expectation Maximization (EM) algorithm was used for this application. However, as with many numerical estimation techniques, the EM algorithm requires an initial value. Traditional Factor Analysis (FA) was used to obtain an initial value for the problem. Because one cannot constrain FA results to be consistent

with mass balances, estimates were not consistent for all instruments, but FA results provided a very effective initial value for the Bayesian learning routine.

Chapter 3 also revisits the problem of conditioning the level indicator data. Factor model principals were used to formulate a regression model in order to improve surge hopper capacity estimate that was previously calculated in Chapter 2. The regression model proved to exhibit a strong fit toward the industrial data, yielding results similar to the previous estimate but with much more confidence.

Chapter 4 makes use of Bayesian learning techniques to estimate the state covariance matrix in a state space model. The observation matrix was already specified by process knowledge as in Chapter 3, while the state transition matrix can be constructed using knowledge of the control system topology. Thus, the only estimation necessary is the hidden state covariance which was easily estimated using methods in Chapter 3. The rest of Chapter 4 formulates a dynamic Bayesian approach to data reconciliation and gross error detection.

Data reconciliation was performed using a Kalman filter as a dynamic approach to estimate the hidden mass balance variables. Once the mass balance state was estimated, the corresponding measurements were compared to the state estimate in order to calculate residual errors. Residual errors were standardized and filtered using a second Kalman filter based on a random walk model, and filtered estimates were then applied to a dynamic Bayesian network that performed a running hypothesis test for the existence of gross errors. If a gross error was detected, it was added as an augmented dynamic state until a new estimate is stabilized, at which point, the gross error estimate would be removed from the augmented state and remain as a constant correction factor.

1.5 Contributions of the thesis

The contributions of this thesis are summarized as follows:

- Conditioning discrete mass balance data (WENCO) using a smoothing method based on the binomial distribution
- Conditioning unscaled instruments (level indicator) to have mass units using repeated principal components and factor models
- Estimating process noise variance and instrument measurement noise variance of steady-state data using Bayesian network learning
- Using FA results as initial values for process and instrument variance estimation with Bayesian network learning
- Using Bayesian networks as a framework for a dynamic version of data reconciliation and gross error detection
- Estimating gross errors in real time by adding them as augmented states

Chapter 2

Reconciling continuum and non-continuum data with industrial application

Data reconciliation uses a least-squares approach to obtain a maximum-likelihood estimate for hidden states [8]. Furthermore, gross error detection is based on statistical tests, thus it is important to ensure that the residual errors follow a normal distribution [22]. Ensuring normality of residual errors is usually not a concern when reconciling data from continuous processes; however, difficulties often occur when reconciling data and detecting gross errors from dissimilar distributions. The first problem is that if any of the distributions are non-normal, then the weighted least squares techniques are no longer maximum-likelihood estimators [8]. The second problem is that if the distributions are dissimilar, then it is most likely that residual errors are non-normal which renders the popular Gaussian-based statistical tests as invalid.

This chapter focuses on smoothing data that is contaminated by a binomially distributed process so that it can be reconciled with normally distributed data from continuous processes. A case study was done on a system where non-continuum loads from a dump truck were to be reconciled with two downstream continuum weightometers. A smoothing technique was developed from the binomial distribution in order to smooth out the contamination created by the discrete nature of truckload arrivals. Regression analysis based on principal components was used to evaluate the performance of the smoothing algorithm. Furthermore, principal component regression methods were used to condition unscaled level indicator data so that inferences on mass contents could be made.

2.1 Introduction

Data reconciliation is a well established tool for process monitoring and optimization. A common application of data reconciliation is to increase the quality of data for real time optimization of set points and plant model parameters [8]. Most of the focus is on reconciling random error and statistically identifying the presence of gross errors from measurements by malfunctioning instruments so that they do not have an effect on the reconciled measurement [24]. While

the traditional method was to identify gross errors, eliminate the contaminated measurements and restart the algorithm, [8] used an objective function based on the contaminated Gaussian distribution. This was used to simultaneously identify gross errors and perform data reconciliation without iterative steps [8], [16].

Recently, work has been done in the DeBeers mine in integrating data from various weightometers to obtain reconciled measurements [23]. The challenge in this process however, is that due to numerous highly operative mechanical components on weightometers, gross errors are fairly common. The most common causes of gross error are conveyor belt vibration or misalignment, and matter that has spilled onto a weighing component or has collected on the belt itself. Gross errors can be significant, up to 20% and can accumulate rather quickly [23]. In such a scenario it may be advantageous to use soft sensors to estimate gross errors as a hidden state and to track propagation of error from one measurement with respect to others.

A similar problem has been found in weightometers used by various facilities within the Alberta Oil Sands industry. There were significant discrepancies when weightometer data was compared to the WENCO database maintained by mining operations. The WENCO database employs various measurement methods including scales built into trucks, with GPS to record where loads are dumped. Problems arise when smaller time windows are considered, as truck-loads have a very random time delay between time dumped and time registered by the weightometer. Because gross errors are relatively frequent and can develop quickly, shorter time intervals are required in order to capture dynamics. Thus it is important that WENCO data gets appropriately smoothed so that reasonable comparisons can be made with downstream weightometer data. It is also important that proper models be obtained for intermediate holding to ensure mass balances closure.

The work covered in this chapter can be summarized by Figure 2.1, where in Section (3) the smoothing algorithm is developed using process knowledge. Section (4) applied this smoothing method to the WENCO data and observed the conversion to normality. Section (5.1) introduced principal component regression as a method to test the smoothing performance. Section (6) also used principal component regression, but this was used to obtain a conversion factor for conditioning the level indicator data.

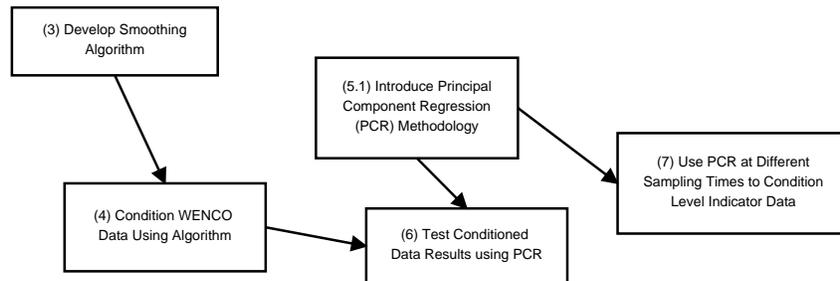


Figure 2.1: Chapter Methodology

2.2 Subsystem for Preliminary Analysis

The subsystem analyzed in this paper is shown in Figure 2.2. Intermediate holding often poses a problem in completing mass balances. As seen in the system schematic, there were two intermediate holding units; the dump pocket which had no instrumentation, and the surge hopper which had a *relative* level indicator.

This system posed two major challenges. The first challenge was to find an appropriate method to balance non-continuum truckload data with continuum data downstream with no instrumentation on the dump pocket. Due to lack of instrumentation, the contents of the dump pocket could not be measured and therefore no intermediate data could be used to normalize the WENCO data with downstream measurements.

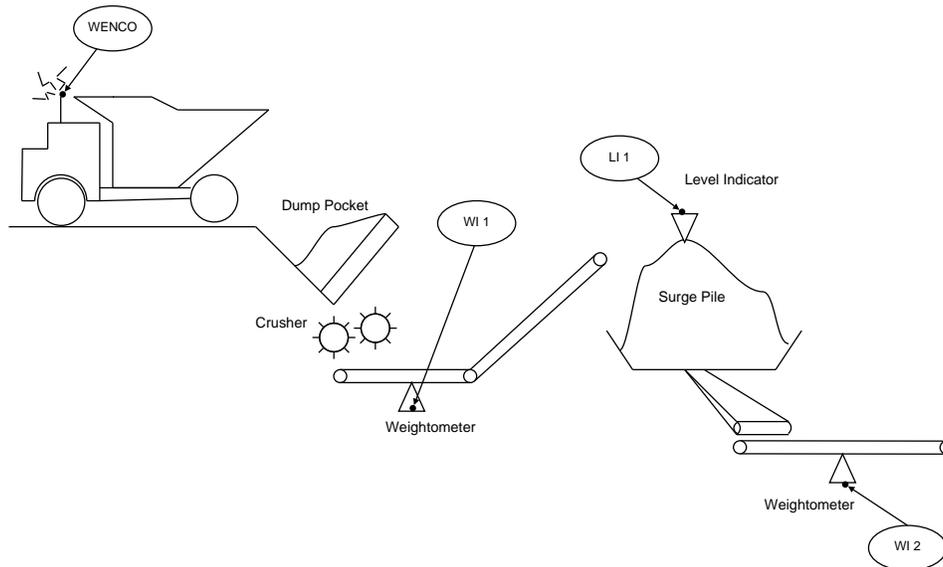


Figure 2.2: Solids Handling

The second challenge was to estimate the maximum capacity of the surge pile or surge hopper. Level measurements were given on a scale of 0% to 100%; however, the maximum capacity was unknown. Previously, least squares regression was used by process engineers to estimate the hopper size; however, results were not useful due to least squares assumptions being invalid (that independent variable is precisely known) and the fact that the correlation was insignificant.

Choosing a reconciliation time window was the first step for this analysis. Traditionally, such reconciliation happens on average over periods of 1-12 hours. This is generally because upstream and downstream equipment can experience significant time delays when reconciliation is employed over the entire plant. However, since all processes within this subsystem are particularly close together, smaller time intervals may still give intelligible results and capture dynamic behavior. For this analysis, reconciliation occurs over one process variable: the train mass flow rate. Within the solids handling system, there are four measurements that take mass flow rate into account.

1. m_{WENCO} = vector of WENCO-Recorded Mass for given truckloads
2. \dot{m}_{WI1} = vector of Mass Flow Rates for Crusher Weightometer
3. P_{LI} = vector of Proportions in terms of Surge Hopper Capacity
4. \dot{m}_{WI2} = vector of Mass Flow Rates for Mix Box Feed Weightometer

Many of these measurements cannot be directly used in a balance as they estimate different system variable types. For direct comparison over a given time window, dynamic measurements must be integrated, WENCO data must be summed, and differences in surge pile level must be calculated and scaled appropriately. Integrated mass values are denoted as $M_{subscript}$ to denote a vector of cumulative mass measurements over each of the i th standardized time windows.

$$\begin{aligned}
M_{WEN(i)} &= \sum_{t=T_i}^{T_{i+1}} m_{WENCO} \\
M_{WI1(i)} &= \int_{T_i}^{T_{i+1}} \dot{m}_{WI1} dt \\
M_{LI} &= c_H (P_{LI(t=T_{i+1})} - P_{LI(t=T_i)}) \\
M_{WI2(i)} &= \int_{T_i}^{T_{i+1}} \dot{m}_{WI2} dt
\end{aligned}$$

Since P_{LI} is a proportional measurement, it must be scaled by c_H which represents the total surge hopper capacity; there was no prior information on this value so c_H must be estimated from the data. When all parameters are defined, vectors of measurements can be directly compared in terms of the mass balances. For example, the first equation is a mass balance comparison between WENCO and the mix box feed conveyor taking into account the intermediate surge hopper level. The second equation is an example of comparing the vector of WENCO values and crusher conveyor. Even without measurement noise these values are not exactly equal because transportation occurs on conveyors and not pipes. Bulk mass accumulation of incompressible matter is not possible in pipes but such unaccountable accumulation is possible for conveyors. Furthermore, there is also a dump pocket that has unmeasured intermediate holding which is also responsible for short term mass imbalance.

$$M_{WEN} \simeq M_{WI2} + M_{LI}$$

or

$$M_{WEN} \simeq M_{WI1}$$

In general, one does not need to include intermediate measurements between two references unless it affects mass balances such as intermediate storage holding. However, ‘distance’ between two reference measurements increases, process noise and time delay begin to play a more significant role.

2.3 Conditioning Method for WENCO Input

While longer time intervals (1-12 hours) are conventional for data reconciliation, short term reconciliation (10-60 minutes) is useful for capturing gross error dynamics for a small system of adjacent instruments. If mass imbalances are normal *as well as* measurement noise, conventional data reconciliation techniques can still be directly applied; however, within this system, non-normal WENCO data poses a problem for short-term reconciliation. Figure 2.3 displays a histogram of the raw WENCO data as well as that for the weightometer over ten minute sampling intervals. Weightometer data is nearly normal. However, the WENCO data is severely contaminated, and can be represented by a combination of Gaussian distributions which is often referred to as a Contaminated Normal distribution [8]. Each peak in the WENCO histogram in Figure 2.3, one can easily observe seven peaks with a possibility of an eighth. Note that in this histogram the bin values are scaled in terms of the process mean, thus the overall mean is centered at a value of 1.

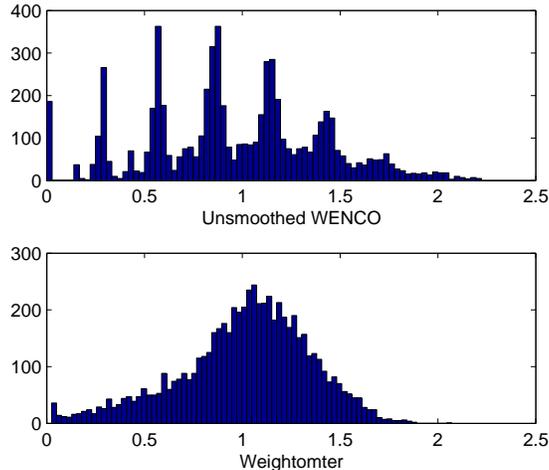


Figure 2.3: WENCO and Weightometer Histograms

For smaller time intervals, one must take into account the discrete nature of dump truck loads, and the inability to measure the contents of the dump pocket, which renders the mass balance incomplete. A simple solution is to apply a weighted average over past readings to mimic the behavior of the dump pocket. Consider the Summed Truck Loads M_{WENti} for a vector of summed truckloads for past time intervals. The *expected load response* (ELR) for the current time interval is given by the relation in (2.1).

$$ELR_{(ti)} = w_1 M_{WEN(ti)} + w_2 M_{WEN(ti-1)} + \dots + w_i M_{WEN(ti-i)} \quad (2.1)$$

Because this system deals with relative weights, the sum of the weights must be equal to 1. Common empirical methods are to use a simple average over previous time windows or exponentially decreasing weights, for instance,

1/2, 1/4, 1/8... This approach works well with dynamic systems, where the next measurement is partially related to the previous one. Another possible method is based on normal distributions with maximum weight at the time delay. Intuitively, this approach requires a delay that is significantly larger than the standard deviation (two or three times larger is feasible), otherwise, future data may be needed to maintain normality. However, the governing process behind this smoothing problem is binomial; thus the smoothing problem should be appropriately derived.

In this problem, varying numbers of trucks arrive over the predefined time intervals to dump material into a dump pocket. One popular method toward modeling arrivals is to use a Poisson's distribution shown in (2.2). This is the classic solution to modeling probabilities of 'rare events' [10]. The Poisson's distribution yields a probability for a given number of 'rare' events k , and the mean number of events for the corresponding time interval λ .

$$f_{Poi\text{ss}}(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (2.2)$$

The emphasis of the Poisson distribution is on the event being 'rare', that is, that the occurrence of events are independent. This has been used to model telephone calls at a call center, and to some extent, airplane arrivals at minor airports [10]. An event is 'rare' when the average number events is much smaller than the practical limit. When using the minor airport example, if the practical airplane frequency limit was 6 airplanes per hour, but only one airplane arrived per day on average, using the Poisson distribution would be appropriate because the time of one airplane arrival would be independent to the time of the previous one.

However, for modeling arrivals for major airports, the Poisson distribution is no longer valid as air traffic scheduling and controllers operate airports to near capacity. This means that the time that the next airplane arrives, depends on the time that the previous plane arrived and that there is a definite possibility of the maximum frequency being reached. If there is a practical maximum frequency of events, the binomial distribution shown in (2.3), becomes more appropriate.

$$f_{Bin}(k; n, p) = \begin{cases} 0 & k < 0 \\ \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} & 0 \leq k \leq n \\ 0 & k > n \end{cases} \quad (2.3)$$

Referring to (2.3), n represents number of trials, p is the probability of getting an event, and k indicates the number of events. The binomial distribution reduces to the Poisson distribution when p is very small and n very large so that $\lambda = np \ll n$. This occurs when the expected number of events np is much less than the maximum n .

In the case of the WENCO truck load example, truck arrivals are not rare events because the expected number of truckloads is not that much smaller than the maximum. This means that the event of one truckload arrival is dependent on the next, thus the binomial distribution better describes this process. When using a binomial approach, n represents the maximum number of truckloads possible (roughly six or seven loads per 10 minute interval according to Figure 2.3), while p represents the expected proportion of truckloads to arrive in terms of the maximum, and k is the query number of truck loads. When formulated in this manner, it is synonymous to dividing the entire time

interval by the maximum number of truckloads n to get n *subintervals*; thus for each subinterval, there is a possibility of one truckload arriving or not, which is represented by the parameter p . This is the classic formulation for a binomial distribution problem. With this in mind, the model takes the form of (2.4), where n_{TL} is the number of truckloads at a given time interval.

$$p(n_{TL} = k) = f_{Bin}(k; n, p) \quad (2.4)$$

Note that the parameter n depends on the length of the time window and breaks down into two components: (1) the maximum number of truckloads that can fit in the dump pocket (constant), and (2) the number of truckloads that the conveyor belt can transport within a time window (time window dependant). The calculation for n is shown in (2.5) while the proportion constant p is shown in (2.6). Since n must be an integer, one should round down as it is a maximum integer number of truck loads.

$$n = \frac{m_H + \dot{m}_C \Delta t}{m_{TL}} \quad (2.5)$$

$$p = \frac{\dot{m}_C \Delta t}{n \cdot m_{TL}} \quad (2.6)$$

where

1. m_H = maximum mass capacity of dump pocket
2. \dot{m}_C = mass flow rate of conveyor
3. Δt = time interval of mass balance evaluation
4. m_{TL} = expected mass per truck load

Note, as the time interval gets very large, the capacity of the pocket becomes less and less significant. If the conveyor belt runs at a steady continuous pace which is equal to the average mass flow rate by the truckload, the residual mass balance error between the truck loads and the conveyor at each interval has an approximate binomial distribution. When recalling the expected truck load mass and the number of truck loads n_{TL} , the mass balance error relationship is shown in (2.7).

$$\epsilon_{MB} \approx m_{TL}(n_{TL} - n \cdot p) \quad (2.7)$$

Residual errors must be normal under typical conditions in order for gross error inference to be applicable. Thus the objective of the weighing function is to provide a statistically sound method to obtain coefficients that best smooth out the binomial nature. As an example, consider the dump pocket system at an interval of 10 minutes, with a dump pocket having an 11.5 truckload capacity. Analysis revealed that the conveyor transported mass at a rate of 4.5 loads per time window, yielding a maximum of 16 loads per time step. A time delay of 6 minutes was estimated. Note that the time delay parameter can be maximized, but dump pocket capacity does not guarantee convergence, so a reasonable number must be assumed if prior information is unavailable.

The philosophy behind weighing is to determine the average weightometer response for an unknown WENCO input. This overall weightometer response

given WENCO input is mainly affected by the time delay and the dump pocket capacity. Since they are independent, one can consider each effect separately and then combine the results. The first case is to assume zero pocket capacity and a fixed time delay. This assumes that once time delay is taken into account, mass balances should always close. For example, if the time delay was 16 minutes on a 10 minute window, 0% of the input would be registered on the first interval, 40% would be registered on the second and 60% would be registered on the third. This simple relation can be summarized by (2.8).

$$p_{ti}(t_d, \Delta t, i) = \begin{cases} 0 & i < (\frac{t_d}{\Delta t} - 1) \\ \frac{t_d}{\Delta t} - i & \frac{t_d}{\Delta t} - 1 \leq i \leq \frac{t_d}{\Delta t} \\ (i + 1) - \frac{t_d}{\Delta t} & \frac{t_d}{\Delta t} \leq i \leq \frac{t_d}{\Delta t} + 1 \\ 0 & i > (\frac{t_d}{\Delta t} + 1) \end{cases} \quad (2.8)$$

The other aspect to be considered was the most probable dump pocket effect. Assuming no time delay, overload from this time interval can affect the loads on the next time interval if a large amount of mass was dumped. For example, if only one truck load was dumped during this time interval (with an average of 4.5), then it would only have a partial effect on this time interval, roughly a quarter in proportion to the mean. However, if five truck loads were dumped, it would have a complete effect on this time interval *and* a partial effect on the next. However, the effects must be weighed with the probability of that particular number of trucks arriving. With probabilities given by the binomial distribution, one can determine the average effect that WENCO input on one interval would have on the successive weightometer readings.

Matrix operations provide a convenient framework to describe the manner in which a given number of truckloads affects multiple time windows. The *load effect matrix* gives the individual effect of a given number of truckloads for a given time interval. This can be later back multiplied by a vector of probabilities for that given number of truckloads. The rows of this matrix represent the time interval assuming no time delay (thus, the first row is the overall effect of a truckload set on the same time interval, the second row would be the effect on the next time interval, so on and so forth), the columns represent the number of truckloads. So, for an $i \times j$ matrix, the elements are constructed as follows.

$$M_{eff}(i, j) = \begin{cases} 0 & j \leq (n \cdot p)(i - 1) \\ \frac{j}{(n \cdot p)} - (i - 1) & (n \cdot p)(i - 1) < j < (n \cdot p)(i) \\ 1 & (n \cdot p)(i) \leq j \end{cases} \quad (2.9)$$

Shown below, is an example of an approximated load effect matrix for the case study system, with a 11.5 truckload dump pocket operating at 4.5 truckloads per interval.

0.2	0.5	0.7	0.9	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0.2	0.4	0.6	0.9	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0.1	0.3	0.6	0.8	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.3	0.5	0.7

From this equation, it can be seen that the effects can range anywhere between zero for “no effect” and 1 for “full effect” denoting the anticipated effect of j truckloads on the i^{th} time interval. This matrix is then multiplied by a vector of probabilities given in (2.11). The vector is generated according to (2.10) by applying the binomial distribution. The resulting vector is the distribution arising from the assumed binomial nature of truckloads. Since the result is a distribution, its elements must add up to 1.

$$P_{bin}(j) = f_{bin}(j; n, p) \quad (2.10)$$

$$P_{eff} = \frac{M_{eff} \times P_{bin}}{\sum M_{eff} \times P_{bin}} \quad (2.11)$$

The final weighing coefficients are a combination of time delay and pocket effect. Element wise, the final weighing parameters which are calculated as follows:

$$w(i) = p_{ti}(i) + p_{ti}(i-1) * P_{eff}(i-1) + \dots + p_{ti}(1) * P_{eff}(1) \quad (2.12)$$

These values are used in (2.1) to calculate the expected mean effect of WENCO input. Recall that optimization of unknown constants must be carefully performed as the objective could be insensitive toward certain parameters. For example, if the objective is minimizing the standard deviation of the error between weighted WENCO reading and weightometer, the maximum pocket capacity cannot be well determined, as the mass balance standard deviation is often insensitive with respect to dump pocket size when close to the optimal region. Nevertheless, dead time parameters often converge, but these estimates are usually more reliable with smaller time windows due to increasing relative effects and greater sensitivity; still, estimations over excessively small time windows tend to be susceptible to noise.

2.4 Binomial Distribution and its Effects on Mapping

In general, trying to obtain statistically useful information from comparing normal and non-normal data is difficult. However, if one smoothes out the binomial tendencies of the truck loads with the appropriate weights, the remaining parameters that cause variance, for example, mass per truck load, can be considered normal. Since converting binomial inputs to normal outputs is a function that the dump pocket physically performs, smoothing out the binomial nature can be considered a statistical simulation. Nevertheless, when time intervals get larger, the binomial distribution starts to gain a normalized shape, and thus for time intervals beyond 30 minutes, smoothing is not generally required. However, weighing profiles adjust accordingly; as the time window increases, the vector of smoothing weights converges to 100% weight on the first time interval. Weighing profiles from 10 to 120 minutes are shown in Figures 2.4(a) to 2.4(d). The smoothing profiles in Figures 2.4(a) and 2.4(b) were applied to WENCO data from a corresponding time interval. Corresponding weightometer data was plotted against the smoothed and unsmoothed values in Figures 2.5(a) to 2.5(d).

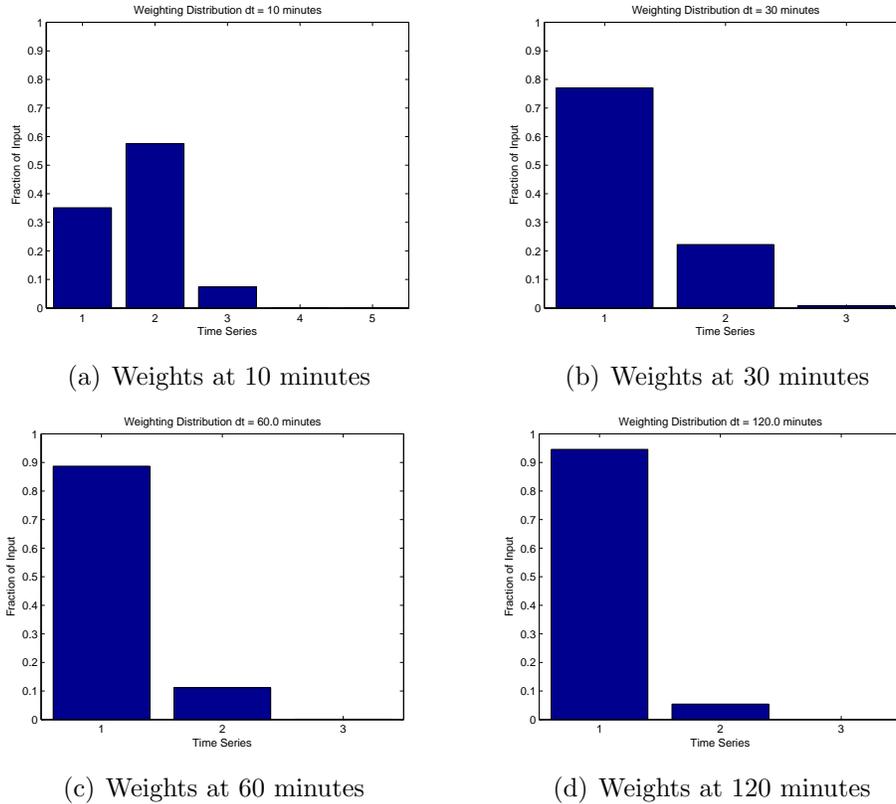


Figure 2.4: Weighting Distribution for Increasing Time Intervals

From Figures 2.5(a) and 2.5(b) dramatic change can be seen in the normality of the data; binomial trends for unsmoothed data are clearly visible at the 10 minute interval. Of course, the data is normally distributed around each binomial concentration as each truck does not dump exactly the same amount of material into the dump pocket, but nevertheless, discrete trends are visible. There is even a slight trend visible in the unsmoothed 30 minute data. However, for time intervals much larger than 30 minutes, the raw WENCO data appears to be much more normal.

Figures 2.5(a) and 2.5(b) display readings relative to the mean, and illustrate how difficult it would be to estimate the mass flow by truckload given the weightometer data, as significantly different truckload values yield similar weightometer results. For example, from Figure 2.5(a), if the weightometer mass was 1.0 means the most likely WENCO mass values would be centered around 0 means, 0.3 means, 0.6 means, 0.9 means and so on. However, for the same weightometer value of 1.0 means, the smoothed WENCO data in Figure 2.5(b) would have a value of roughly 1.0 means. Other predictions from smoothed data also correspond well.

Predicting weightometer data from WENCO data without smoothing is possible, but the distribution over every value of WENCO weight is not necessarily accurate. Time intervals with large WENCO values are often followed by intervals with values that are much smaller. The pocket smooths out these values allowing for the weightometer to run at a constant rate. This often leads to apparent idiosyncracies in the raw data. From the previous exam-

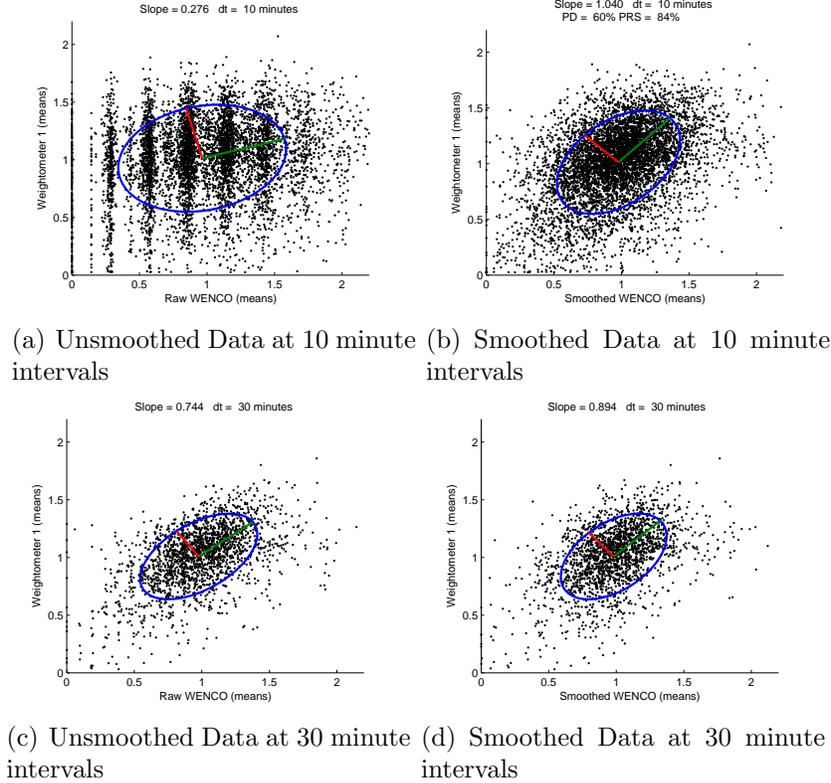


Figure 2.5: Effects of smoothing

ple, given that 0.5 means arrived by truckload, the expected weightometer flow rate would be greater than 0.8 means. By observation, all weightometer values seemed to be centered around 1 mean irrespective to the WENCO value. The smoothing attempts to take irregular summed WENCO values into account and allows the WENCO data to have an estimably unique mapping onto weightometer data. Other smoothing methods could also do this; however, when using weights derived from statistically rigorous methods, it reduces *information loss* from smoothing.

Due to the presence of a dump pocket, current WENCO values often depend on previous ones; because of this, weightometer data is dependent on more than just the single WENCO value at the same time interval. Weighing profiles that take these dependencies into account will *add* information; however, a badly chosen weighing profile could emphasize nonexistent relationships over existent ones, thus *removing* information. For example, a simple average over the past 90 minutes to predict the weightometer response for the next 10 minutes would put far too much emphasis on measurements in the distant past, which are often unrelated to response in question. This would result in oversmoothing of the WENCO data, generating smoothed data that fluctuated much less than the weightometer data; the slope of the elliptical region for oversmoothed WENCO data vs. weightometer data would be steep (much steeper than 1), indicating that variations in oversmoothed WENCO data must be disproportionately scaled in order to predict weightometer data. Since it is expected that WENCO and weightometer data would be consistent, an ideal smoothing profile would generate an ellipse with a slope along

the diagonal, as shown in Figure 2.5(b).

Smoothing profiles that result in coherent relationships suggest that information has been added. Note that numerical values for the slope of elliptical regions can be determined by PCR (principal component regression). While the improvement in results have been showed qualitatively, PCA (principal component analysis) and PCR can be used to quantitatively determine effects of applying the derived weighing profiles.

2.5 Principal Component Regression and Performance Metrics

2.5.1 Principal Component Regression

Principal Component Analysis (PCA) is closely related to Factor Analysis (FA) and is in fact, one possible method of calculating orthogonal factors [12]. Data that can be correlated with many different variables can often be simplified to a smaller dimensional space in the direction of a vector if certain variables seem interdependent. In both PCA and FA, dependencies are represented by a linear combination of variables that are explained by an underlying set of orthogonal factors.

Principal component regression (PCR) has similar objectives to that of partial least squares which is to project both observed and predicted variables onto a new space. In general, it is used for *bilinear factor models* which can be intuitively analyzed using PCA and FA [20]. PCR is a method based on PCA that is used to describe p dimensional data into v dimensions with $p-v$ dimensional noise that is attributed to both variables. This is different from ordinary least squares regression as it tries to reduce p dimensional data into v dimensions with $p-v$ dimensional noise that is attributed to only one variable, which is predicted [12]. Thus, in least squares regression, all the noise is attributed to the predicted variables and it does not take into account the fact that both variables are noisy estimates of an underlying variable.

If there is significant noise within the predictors, the estimated parameters will be biased toward conservative “shallow slope” results as predictor variable inaccuracy reduces predictive power [9]. This effect is often called *regression dilution* [5]. Geometrically, this can be interpreted by noting that minimizing vertical distance between points and a line causes preference toward horizontal lines, even if horizontal position of the points is not necessarily accurate. Orthogonal regression is a family of different methods that are used to minimize the orthogonal distance; [2] describes many of these methods. However, since the system is linear and PCR results have intuitive interpretations that align with PCA and FA, PCR was the chosen as the regression method.

Consider the following model which is attributed to the industrial system. Any underlying variable z is measured by two observations x and y . For all operating conditions, x and y are considered to be normally distributed around the true value z with respective measurement noises ε_{xm} and ε_{ym} . Furthermore, at constant operating conditions, the underlying variable z can also be normal, with a mean around the desired condition z_0 with a variance of σ_z^2 .

$$\begin{aligned}
z &\sim N(z_0, \sigma_z^2) \\
\begin{bmatrix} x_m - x_0 \\ y_m - y_0 \end{bmatrix} &= \begin{bmatrix} a_x \\ a_y \end{bmatrix} z + \begin{bmatrix} \varepsilon_{ym} \\ \varepsilon_{xm} \end{bmatrix}
\end{aligned} \tag{2.13}$$

Because PCR is a regression method that uses the eigenvectors to calculate the slope, it is consistent and unbiased when noise is IID. Such results have been proven, and one such proof is available in the appendix of [11]. It is also recommended that variables be scaled when performing PCA or PCR if the units are not commensurate [12].

2.5.2 PCR Performance Metrics

Regression models often use a metric called the regression factor to indicate fit, a similar metric is often used in PCA. When all eigenvalues are arranged in a vector and scaled so all elements add to one, each eigenvalue represents the fractional variance that is taken into account by its corresponding eigenvector. Thus if zero eigenvalues exist, it means that some variables are linearly dependent on others. In the ideal case, most of the data should be explained by relatively few eigenvectors or principal components.

In order to obtain a performance metric on PCR, the number of components must be chosen; since each component has an eigenvalue that pertains to the degree of which that component explains variance, one merely adds all elements of the scaled eigenvalues that belong to the subspace of interest. In the case of a two-dimensional analysis, one scales a two-eigenvalue vector and obtains the first entry. However, in the case of two-dimensional analysis, the first principal component will never explain less than 50% of the variance, thus users must incorporate this into their discretion when analyzing results. The *percent variance captured* (PVC) is defined in (2.14).

$$PVC = \left(\frac{\lambda_1}{\lambda_2 + \lambda_1} \right) \times 100\% \tag{2.14}$$

where λ_1 is the larger Eigenvalue and λ_2 is the smaller Eigenvalue. Examples of data with 54% and 74% Percent Variance Captured are shown in Figures 2.6(a) and 2.6(b).

While the PVC is a conventional metric for multiple dimensions, for the case of PCR, it was useful to define two unconventional metrics. The first metric is called the PRS (percent relative scatter), which is useful for data sets that have a minimal zero value (as in the case of flow rate data). Because of this, it would be expected that the mean would be significantly far away from zero for the distribution to remain normal. If this is the case, the minor axis length should be short enough to make zero readings sufficiently improbable. This essentially is a comparison between the uncorrelated standard deviation and the mean. For this case, PRS is defined in (2.15) as the 95% confidence interval scalar (given by the adjusted F distribution) times the minor axis length divided by the mean. Thus, for 100% PRS, impossible readings of less than zero are 5% probable according to the multivariate normal distribution. (2.15) is defined in terms of the mean WENCO reading. Generally, PRS values close to or greater than 100% would suggest that the evaluation time window is too small for measurements to be normal.

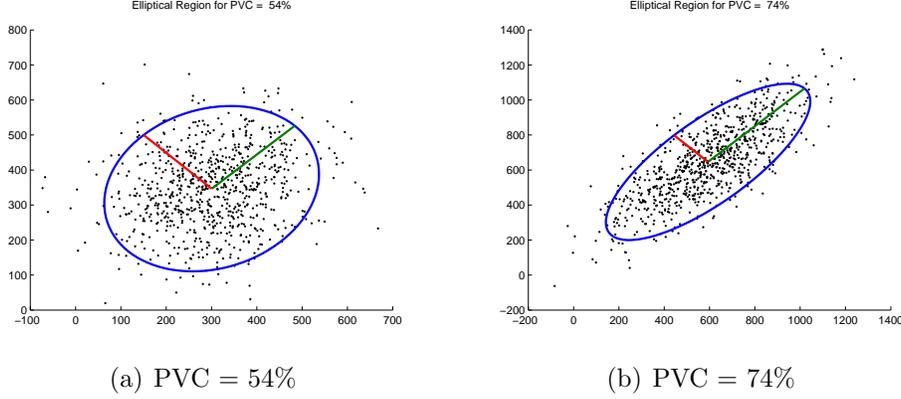


Figure 2.6: Examples of *percent variance capture*

$$PRS = \frac{\sqrt{\lambda_2}}{\mu_{WEN}} \sqrt{\frac{p(n-1)}{n-p} F_{n,p,\alpha}} \times 100\% \quad (2.15)$$

where $F_{n,p,\alpha}$ is the inverse F-Distribution that defines the scalar for a confidence level $1 - \alpha$. Note, for the case of multidimensional analysis, λ_2 is replaced by the summation of eigenvalues of the rejected principal components.

The other useful PCR performance metric that was derived was the PD (percent definitiveness). It is similar to the PVC, except that it is standardized so that the worst case scenario is 0% instead of 50% as in the case of bivariate analysis. This metric is somewhat similar to the coefficient of determination or R^2 value often used in regression. The other trait that makes the PD metric desirable is that its uncertainty can be easily calculated when performing two-dimensional analysis. The PD is defined in (2.16). It is essentially a direct comparison between the relative squared lengths of ellipsoid axes.

$$PD = \left(1 - \frac{\lambda_2}{\lambda_1}\right) \times 100\% \quad (2.16)$$

For multidimensional analysis λ_1 can be replaced by the summed eigenvalues pertaining to the principal components kept while λ_2 pertaining to the discarded principal components. λ_1 spans the total *projected* variance on to all dimensions [12] and is thus similar to the SS_{tot} parameter in the analogous case of the R^2 calculation, while λ_2 spans the *unexplained* or residual variance which is often associated with SS_{err} . Thus the worst case scenario occurs when $\lambda_1 = \lambda_2$ or $SS_{tot} = SS_{err}$. In the same way that SS_{err} is never greater than SS_{tot} , λ_2 is never greater than λ_1 .

As was previously mentioned, the PRS and the PD metrics have uncertainty intervals that can be easily calculated. These intervals make use of the property shown in (2.17), which pertains to eigenvalues of a covariance matrix.

$$\ln \frac{\hat{\lambda}_i}{\lambda_i} \sim N\left(0, \frac{2}{n}\right) \quad (2.17)$$

If the covariance matrix is full rank, then all eigenvalues are independent and standard procedures apply in the addition of independent noise. Con-

confidence intervals for PD and PRS can thus be calculated using confidence intervals on λ .

2.6 Applying PCR to Verify Smoothing

The first application of PCR was to verify improvements of WENCO data by smoothing using the binomial distribution method. Figures 2.7(a) and 2.7(b) display results for PCR on unsmoothed data. Note, unsmoothed data is highly non-normal and thus would not qualify for serious PCR. It is done to illustrate the difficulty in mapping one variable onto another. Figures 2.7(c) and 2.7(d) display results for PCR on smoothed data.

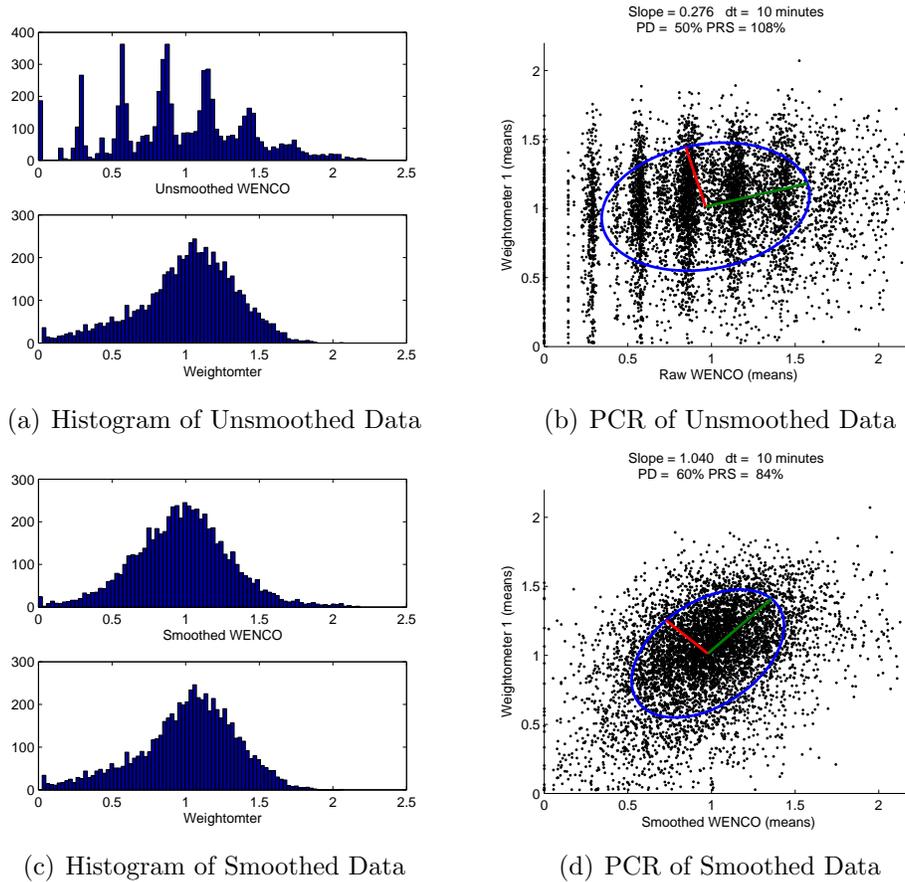


Figure 2.7: Examples of Varying Percent Definitiveness

Results shown in Figures 2.7(a) to 2.7(d) are of significant importance, specifically, the results for smoothed data. It can also be seen from histograms that after smoothing, WENCO data becomes far more normal, as the expected binomial nature has been smoothed out in order to leave the remaining normality behind. Because both measurement methods are measuring the same underlying quantity in the same units, it is known beforehand that the real slope of the relationship is 1. If the calculated slope is the same as the expected slope, the measurement noise is IID, both instruments have equal reliability.

A slope flatter than expected indicates that the the x variable is the less precise observation (regression dilution), while a steeper slope indicates that the y variable is the less precise observation.

After smoothing , WENCO-Weightometer data show that the slope of the ellipsoid is close to 1, indicating that smoothed WENCO measurements now have similar precision to the weightometer measurements. Furthermore, histograms validate this by showing that the measurement variances between data sets are similar. PD is still relatively low from a regression standpoint (60%) but this is simply an indication that measurement noise is not negligible compared to process noise. However, it is important to note that PD increased after smoothing was applied, which indicates that smoothing added information instead of simply smoothing the WENCO data so that it had the appropriate variance.

2.7 Applying PCR to Estimate Surge Hopper Scaling

Another challenge for preliminary analysis was to compare readings from the second weightometer to Wenco data and readings from the first weightometer. Between the two weightometers, there is a surge hopper that has a level indicator. However, the level indicator gives % capacity readings, and the mass for 100% is unknown and must be estimated. The scalar that is required to convert percent readings into mass readings is referred to as the hopper scalar c_H ; rearranging the basic relationship with noise included, the model takes on the form of (2.18) for the weightometer-weightometer comparison, and (2.19) for the WENCO-weightometer comparison.

$$c_H(\Delta M_{LI} + \varepsilon_{LI}) = (M_{Wei2} - M_{WI1}) + (\varepsilon_{WI2} - \varepsilon_{WI1}) \quad (2.18)$$

$$c_H(\Delta M_{LI} + \varepsilon_{LI}) = (M_{WI2} - M_{WEN}) + (\varepsilon_{WI2} - \varepsilon_{WEN}) \quad (2.19)$$

Because the weightometer-weightometer comparisons are more direct, (2.18) is used as a basis for the parametric model in terms of the real mass balance difference.

$$M_{LI} = c_H^{-1}z + \varepsilon_{LI} \quad (2.20)$$

$$M_{WI2} - M_{WI1} = z + \varepsilon_{WI2} - \varepsilon_{WI1} \quad (2.21)$$

While the objective of linear regression would be to choose a hopper scalar value that would result in a minimal *vertical* distance from the line and all the points, the objective for parameter estimation using PCR analysis is quite different. If one scales the raw data, one scales both the hidden causal agent and the noise. Thus there is a tradeoff in terms of matching the underlying correlation *and* keeping noise to a reasonable level; to maximize bidirectional prediction, the slope of the elliptical region should be equal to 1; thus the objective is to choose a hopper scalar that causes this to happen. This is easily done by nonlinear equation solution methods. Figure 2.8 displays hopper scalar estimates generated by this method at different time intervals. In this Figure, the hopper scalar is estimated by comparing differences in weightometer readings upstream and downstream. One should keep in mind that such

results are obtained using a model that assumes % capacity is linearly dependent on the material in the hopper. Estimated values for c_H are a linearized average over the entire operating range.

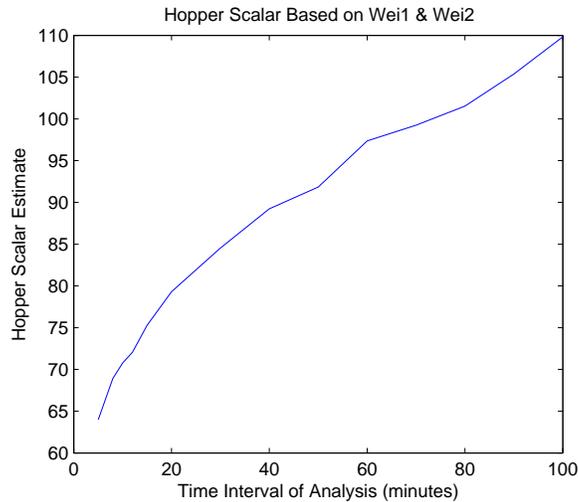


Figure 2.8: Weightometer-Weightometer Based Estimates of Hopper Scaler

When performing PCR to estimate hopper scalars at different time intervals, noise within the hopper level readings does not change in terms of time interval because it is merely a difference between two instantaneous readings. However, weightometer and WENCO readings are summed, so the base value of the random error will increase over larger time windows, but the relative error will decrease. So for excessively small time windows, the hopper level readings will have much more noise than the weightometer readings, but for large time windows, absolute noise from the weightometer readings will be larger because of the summation of errors producing random walk. Since the absolute imprecision of weightometer readings increases with increasing time window, as time windows get larger, c_H will be biased toward larger values.

Since there is no prior knowledge of measurement variance, there is no method or metric to ensure that estimates for the hopper scalar are unbiased at *any given* time window. However, multiple time windows can be considered. PCR is used as an objective function to scale c_H so that hopper data has an elliptical slope of 1. This also causes the hopper level readings and weightometer difference to have a similar absolute noise scale when taking into account *both measurement and process* noise. When the mass change in the hopper is much smaller than the weightometer difference error, definitiveness is likely to be low since hopper level cannot account much for the weightometer error; the converse, wherein weightometer difference is much smaller than the hopper level noise, would also yield relatively low definitiveness. So at small time intervals weightometers are far more sensitive than level indicators, but at large time intervals, level indicators are the far more sensitive measurements. Definitiveness is optimized at the time interval where both instruments are reasonably precise.

Percent definitiveness is an indication of relative dependency between random errors; one of the properties of this system is that dependency is likely

to be maximized when the hopper is scaled correctly. Thus, as long as the relative precision of the two instruments are not excessively dissimilar at the optimal time interval, bias is reasonably small when PD is maximized. Assuming similar relative precision at the optimal time interval is reasonable, as relative precision is expected to be more dissimilar when time windows are far from optimal. Furthermore, this assumption is practical because these two measurement methods are both fairly direct measurements of mass, and thus their inherent variance must be small enough to allow instruments to be useful, but large enough to make them affordable; this makes significant measurement disparity unlikely at the optimum.

Since the covariance matrix is random (as it follows a Wishart Distribution), it is also important to calculate confidence intervals of Percent Definitiveness according to the eigenvalue distribution as shown in (2.17) to ensure that there is adequate resolution for maximization. Figure 2.9 displays both Percent Relative Scatter and Percent Definitiveness trends over increasing time intervals with 90 and 95 percent confidence intervals plotted in dashed lines.

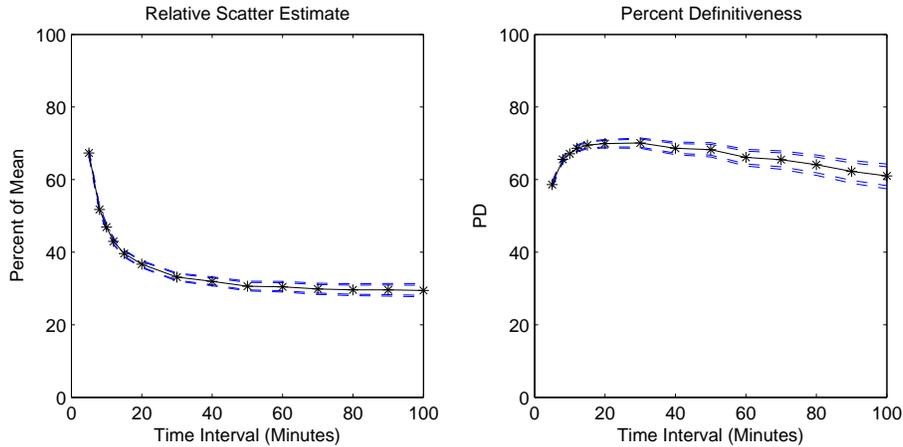


Figure 2.9: PRS and PD for Weightometer-Weightometer Based Estimates

As one can see, Percent Definitiveness is maximized between intervals of 10 minutes to 30 minutes which would correspond to hopper scalar estimates of 70 to 85. While hopper estimates would be biased with biases changing over time, the estimated hopper scalar allows for the best bidirectional prediction at that time interval. Relative scatter is relatively close to its asymptotic minimum at around 30 minutes which means that relative error between weightometer readings will not be significantly reduced by further increasing of time intervals beyond 30 minutes. Hypothetically, one could also use smoothed WENCO readings vs. Weightometer readings to estimate hopper size but it is not advisable. Smoothed WENCO readings have inherently less resolution due to the fact that smoothing causes all measurements to span more time intervals; including weighted averages of data that is not directly dependent will obviously reduce dependency and increase noise.

Figure 2.10 displays PRS and PD parameters for mass balances between unsmoothed WENCO and the second weightometer and Figure 2.11 does the same for smoothed WENCO data. It can be shown that smoothing increases definitiveness between WENCO and weightometer data; however, since more

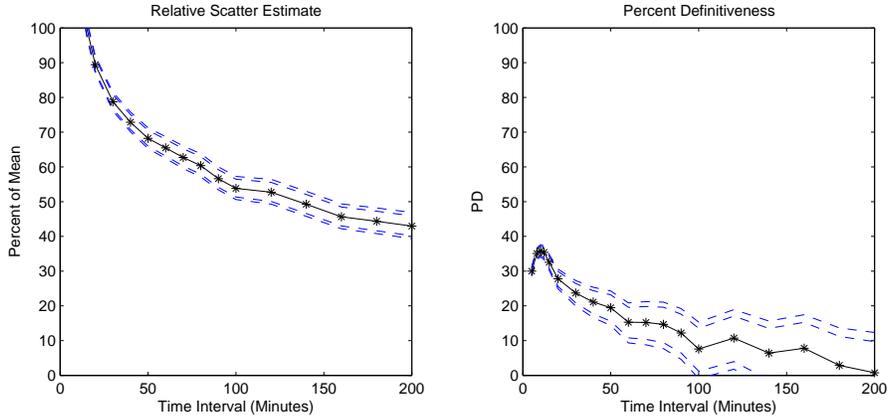


Figure 2.10: PRS and PD for Weightometer-Unsmoothed WENCO Based Estimates

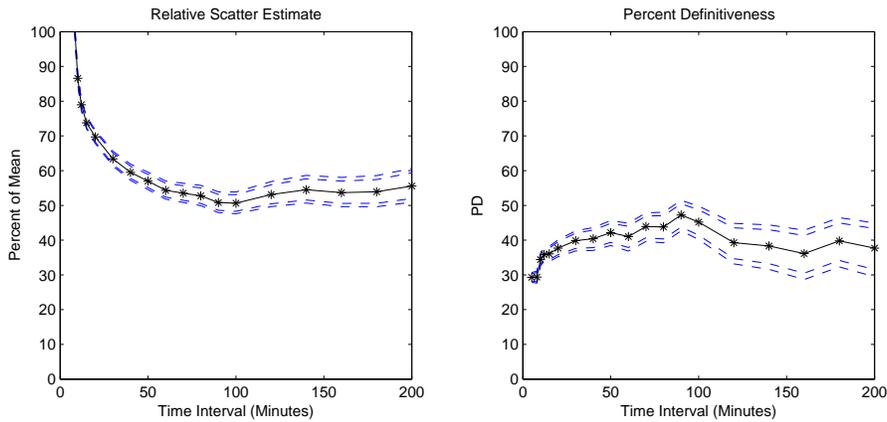


Figure 2.11: PRS and PD for Weightometer-Smoothed WENCO Based Estimates

than one time interval goes into weighing, it is likely that the definitiveness trends will have peaks that are less clear.

2.8 Conclusions

A weighing distribution was derived using the binomial distribution to estimate the expected average effect of a random number of dump trucks arriving with an expected time delay. It was shown through analyzing histograms that this weighing distribution was adequate in normalizing the data. Principal Component Analysis was also used to demonstrate that using this weighing distribution does not result in excessive information loss, because the PCR results of the smoothed data exhibited near ideal conditions, namely, a slope of unity with increased definitiveness.

PCR was also used as an alternative to regression modeling, since the data was actually two measurements measuring the same underlying variable that did not change drastically. It was found that the clearest covariance patterns were found at intervals of 10 to 30 minutes, and it was at these time intervals that the surge hopper parameter was estimated to be on the order of 70 to 85 tonnes per % capacity change. Although PCR-based parameter estimation could not be guaranteed to remove bias on the real value, it is reasonable to assume low bias at time windows with the highest definitiveness. Furthermore, PCR took variance of both measurements into account to prevent excessive noise of one variable to distort the prediction of the other, and thus, such parameters optimize bidirectional prediction when noise is unknown.

1

1A version of this chapter has been accepted for publication. Gonzalez, Huang, Expejo, Almaraj, Lam 2010. Computers and Chemical Engineering 38

Chapter 3

Estimation of Measurement Noise Variance and Bias Using Bayesian Methods for Process Data Reconciliation

This chapter enlarges the scope of the previous system to include more instrumentation. This is required because data reconciliation requires more than two instruments to perform, since if the two measurements are different, there is no objective method to determine which measurement was correct. Using the smoothing techniques introduced in Chapter 1, WENCO data can be added as one of seven instruments when the scope of solids handling system is increased.

A significant challenge to data reconciliation is in obtaining values for measurement noise variance which allows us to attribute weight toward each measurement when estimating the steady-state. Measurements that have more noise variance are given less weight because they are less informative of the state. The problem with directly estimating measurement noise variance from process data is that the process is also subject to disturbance which independently adds to the instrument noise, increasing the variance estimates for the measurements. The focus of this chapter is on applying Bayesian learning to a factor model based on mass balances in order to estimate process disturbance and measurement noise variances.

Furthermore, by introducing the mass balance factor model, an alternate method of estimating the hopper capacity constant c_H is derived, discussed and presented.

3.1 Introduction

Process instrumentation is a crucial element for controlling and monitoring process control systems. However, instrumentation is subject to random measurement error due to imprecision; and more seriously, it is often subject to systematic gross errors caused by instrument bias (or in rare cases, process leaks). Identifying true process operating points and detecting gross errors has traditionally been performed by data reconciliation and gross error detection techniques. However, conventional data reconciliation calculates a weighted

average, where the weights are the inverse of the estimated measurement noise variance. In practice, it is difficult to estimate measurement noise variance from data because of the often inevitable presence of other process disturbances which contaminates these estimates.

Sets of process data contaminated by both process and measurement noise can be described by a factor model based on mass balances. This type of factor model has predefined relationships between the instruments and a set of process variables. The task at hand is to estimate the combination of process and measurement noise variances that best explains the data, a task for which Bayesian learning is an appropriate tool. While the major focus is on estimating process and measurement noise variance—a task that is not explicitly attempted by data reconciliation methods—Bayesian learning can also be used to estimate process means and instrument bias. A better understanding of this work can be obtained by reviewing the progress of data reconciliation.

The first generation of data reconciliation techniques [22] made use of statistical tests such as those implemented by [21], which obtains a chi-squared value from the optimization of the mass balance-based objective function. A later family of techniques proposed by [15] and [6] analyzed residuals of process constraints and applied univariate tests to each measurement. More advanced techniques for data reconciliation were proposed by [22] which used *principal component analysis* (PCA) to draw on principal relations between various measurements. Prior to this, data reconciliation assumed that all hidden process states were invariant and thus independent so that all variance can be attributed to measurements. Using PCA *implicitly* allowed for the tests to take process noise into account, and so considered common trends of the set of measurements over time resulting in a more intelligent testing procedure.

Ozyurt and Pike [8] built on the objective function approach to detect gross errors by introducing a framework based on the contaminated Gaussian distribution; after this, Shladt and Hu [16] continued with the practical use of data reconciliation to enhance soft sensors for key process measurements.

The Bayesian network approach for gross error detection (or bias detection in the absence of process leaks) is closely related to the PCA method used by [22] as it allows for the separation of process noise from measurement noise in order to gain more informative estimates of gross errors. The main difference between these techniques, however, is that the PCA method focuses on learning the covariance matrix from the data and *afterward* imposing mass balance relationships, while Bayesian network techniques focus on learning a model that is *simultaneously* consistent with mass balances and measurement noise covariance. Mass balance factor models contain similar information to the PCA method; however, mass balance factor models are more explicit in their handling process disturbance covariance. These models can be used to apply more advanced methods similar to [16], except in ways that are completely consistent with Bayesian statistics.

This work has made two main contributions. The first contribution is the use of Bayesian networks to separate and estimate measurement and process variance using historical data as well as estimating process means and gross errors. The second contribution is the method of estimating capacity parameters by integrating process data over different time intervals and performing a regression analysis on the covariance matrices; this was a secondary procedure but necessary for practical application.

This chapter is divided into the following sections: A problem from industry is shown in Section 2 with an introduction of the corresponding Bayesian

network. Section 3 contains an overview of the EM algorithm, and the use of PCA and FA to obtain an initial variance guess. Section 4 contains development of gross error estimation from a defined Bayesian network. A simulation of sequential variance and bias estimation is shown in section 5, while section 6 presents a framework for capacity regression. Finally, section 7 contains an industrial application of capacity regression, variance estimation, and bias estimation. Figure 3.1 displays the methodology in terms of key tasks; prerequisite tasks are indicated by arrows; thus it is required that the parent task be completed before undertaking the child task.

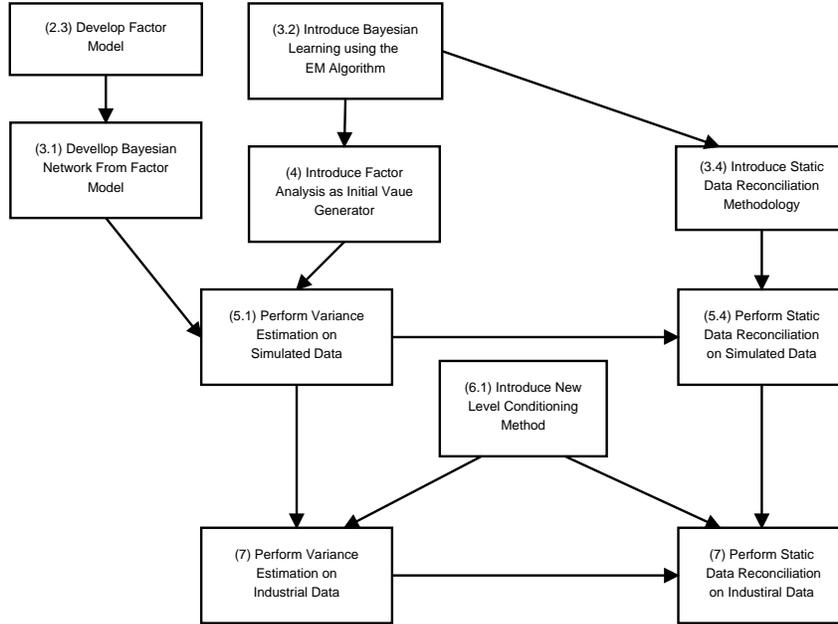


Figure 3.1: Chapter 3 Methodology

3.1.1 A note on Notation

Numerical subscripts indicate the index of an array. For example C_2 would denote the second entry of a vector, while S_{12} would denote the first row second column of a two dimensional array. Subscripts with curved brackets indicate that the variable is defined or specified by the subscript variable within the brackets (this is broader than traditional function notation). For example $\varepsilon_{(\Sigma)}$ would indicate that ε is process noise that corresponds to the observation covariance (measurement noise variances) Σ ; another example would be $\sigma_{2(\Delta t)}^2$ which indicates that measurement noise variance for σ_2^2 is dependent on a particular value of Δt , in the same way $\sigma_{2(1)}^2$ would be the value of σ_2^2 which corresponds to $\Delta t = 1$ (or 1 minute).

3.2 Motivation

3.2.1 Industrial Example

As mentioned in the previous chapter, due to the nature of oil sands operations, weightometers are often unable to achieve their design performance without excessive maintenance. Ultimately, work in this area can enable the on-site monitoring of instrument performance and correction, which could reduce costs of maintenance and aid in dealing with the unavoidable presence of weightometer error. Using the results from the previous chapter, the industrial problem can be expanded so that WENCO and weightometer measurements are compared against more instrumentation. Consider an oil sands slurry preparation system shown in Figure 3.2 which contains seven instruments:

1. Truckload values from WENCO Mining Database (WENCO y_1)
2. Crusher Weightometer (WI 1) y_2
3. Surge Level Indicator (LI 1) y_3
4. Mix Box Feed Conveyor Weightometer (WI 2) y_4
5. Slurry Flow Meter (FM S) y_5
6. Slurry Density Meter (DM S) y_6
7. Water Flow Meter (FM W) y_7

From the flow chart in Figure 3.2, WENCO data is available in a data base which records the mass of each truckload and the time at which it is dumped into the dump hopper. WI 1 is a weightometer on the crusher conveyor belt which feeds into the surge pile. LI1 gives a reading on the relative level of the surge pile (from 0-100%). Since this is a % capacity reading, one of the tasks is to estimate its total capacity. WI 2 is a weightometer for the Mix Box feed conveyor, which transports oil sand from the surge pile to the slurry mix box. In this mix box, slurry is prepared by adding controlled amounts of water to the oil sand. FM W measures the volumetric flow rate of water, and since the density of water is known, conversion to mass flow is trivial. The volumetric effluent slurry flow is measured by FM S, with the density measured by DM S so that a mass flow rate can be calculated.

3.2.2 Problem Formulation

Mass balances serve as a method to keep track of process material over a given time window Δt . Since mass balances are an accumulative comparison of mass flow over time, flow rates should be integrated over time; compositions and properties however, do not have rate-related units and should therefore be averaged over time instead of integrated. Capacity values are inherently cumulative mass values and thus should not be integrated; instead, they should have a difference calculated between the beginning and end of the time interval. The procedure for calculating the mass-balance observation vector Y is shown in Equation (3.1) for the industrial example.

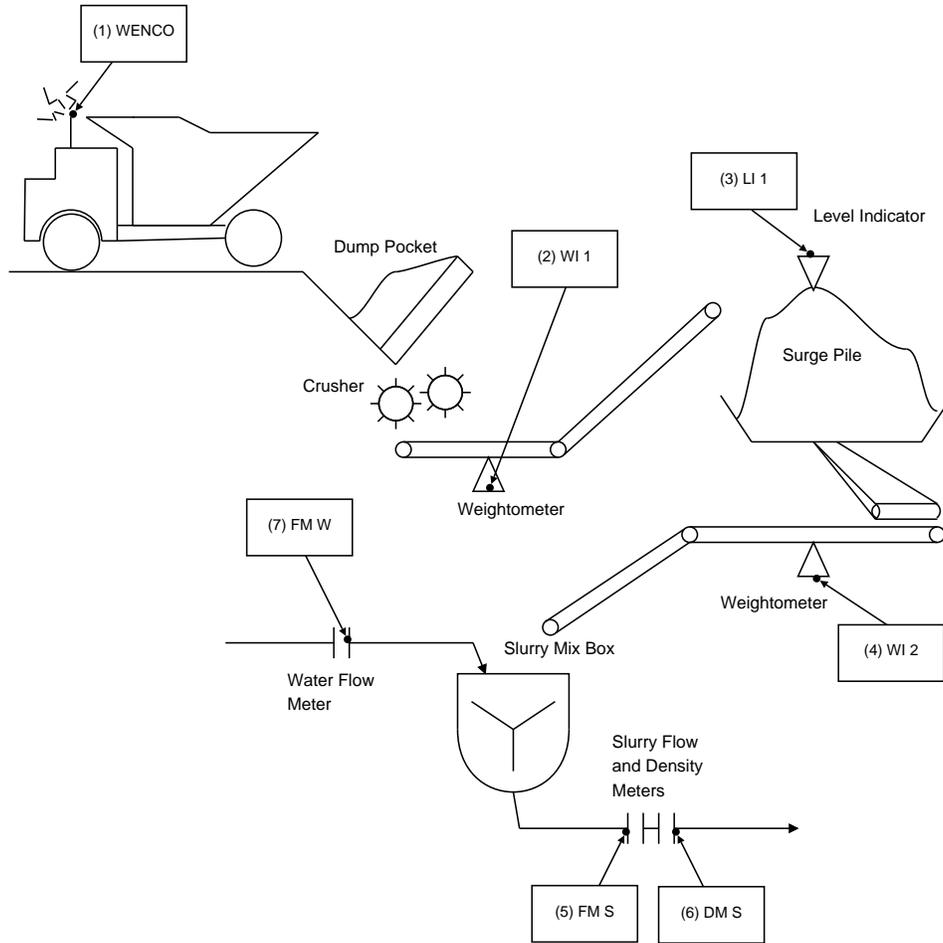


Figure 3.2: Solids Handling and Slurry Preparation

$$\begin{aligned}
 Y_1(i) &= \int_{t_i}^{t_i+\Delta t} \text{WENCO} dt \\
 Y_2(i) &= \int_{t_i}^{t_i+\Delta t} \text{WI1} dt \\
 Y_3(i) &= c_H(\text{LI1}_{(t_i+\Delta t)} - \text{LI1}_{(t_i)}) \\
 Y_4(i) &= \int_{t_i}^{t_i+\Delta t} \text{WI2} dt \\
 Y_5(i) &= \int_{t_i}^{t_i+\Delta t} \text{FMS} dt \\
 Y_6(i) &= \frac{\int_{t_i}^{t_i+\Delta t} \text{DMS} dt}{\Delta t} \\
 Y_7(i) &= \int_{t_i}^{t_i+\Delta t} \text{FMW} dt
 \end{aligned} \tag{3.1}$$

Note, since LI1 gives readings in % capacity, it must be appropriately scaled by c_H to make the conversion to mass units. Since all instrument measurements in Equation (3.1) contain noise, raw measurements can be treated as Gaussian random variables. Due to the integration of noise, measurement noise variance is dependent on the sample time interval Δt ; because of this, mass balance variance estimates are only appropriate given a certain sampling time. The only exception to this rule is intermediate holding which has no integration; such observations would have constant variance regardless of Δt .

3.2.3 Mass Balances, Factor Model Representation

Factor models can be used to represent the mass balance of a process at steady-state. Factor models are composed of two main types of variables, hidden state variables X , and observed variables Y that are affected by the hidden states. Since the factor models of interest represent a mass balance, the hidden states X are a set of minimal process parameters that define the mass balance. The hidden states however, are subject to small amounts of process noise, even at steady state. Because of control systems, noise within the hidden states are cross-correlated, thus X can be represented by Equations (3.2) and (3.3).

$$n \sim N(0, I) \quad (3.2)$$

$$X = \mu + L_x n \quad (3.3)$$

where μ is the process mean vector at a given steady-state, n is a standardized source of variability, and L_x is a loading matrix that produces cross-correlated noise. In this way, cross correlated process noise δ can be summarized as follows:

$$\delta = L_x n \quad (3.4)$$

Since n has covariance I , the covariance of δ can be determined as

$$\Sigma_X = L_x L_x' \quad (3.5)$$

For the industrial example, three hidden states are considered over a given time period: X_1 is the accumulated oil sand throughput (centered around the slurry mix box), X_2 is the accumulated water throughput, and X_3 is the net oil sand accumulation in the surge hopper.

The second part of the factor model deals with the observation variables Y . Equation (3.7) below, is the general nonlinear representation of a factor model with gross errors β which often represents instrument bias (or process leaks in very rare cases).

$$\varepsilon \sim N(0, \Sigma_Y) \quad (3.6)$$

$$Y = f(X) + \beta + \varepsilon \quad (3.7)$$

where $f(X)$ is a nonlinear observation function of the hidden state X , β represents gross errors, and Σ is the measurement noise covariance matrix. Σ_Y is diagonal since it is assumed that measurement noises are mutually uncorrelated. Linear approximations of the observation model take the form shown in the following equation.

$$Y = CX + r + \beta + \varepsilon \quad (3.8)$$

where C is an observation matrix of linearized coefficients and r is a vector of linearized reference points. C and r can be calculated by means of a Taylor series approximation given by the following equations:

$$C_{ij} = \frac{\partial f(\mu)_i}{\partial \mu_j} \quad (3.9)$$

$$r = f(\mu) - C\mu \quad (3.10)$$

where $f(\mu)_i$ is the i^{th} element of $f(\mu)$ (where i ranges from 1 to the number of instruments) and μ_j is the j^{th} element of μ (where j ranges from 1 to the number of hidden states).

For most measurements, Taylor series approximations are not necessary. When measurements are a direct combination of the hidden states having the same units, coefficients of C are simply 1. For example, Y_2 is a direct combination of the *mass flow* through the mix box and the *mass accumulation* within the hopper. Linear combinations can also have simple unit conversions. For example, Y_5 is a volumetric reading that is affected by mass flow rates of oil sand and water which are hidden states; the corresponding elements of C are the specific volume of oil sand and water which serve as unit conversion constants.

For this industrial example, a Taylor series approximation is only required for the density meter which is dependent on oil sand mass flow X_1 and water mass flow X_2 as shown in Equation (3.11).

$$\rho_{slurry} = \frac{\rho_{sand}X_1 + \rho_{water}X_2}{X_1 + X_2} \quad (3.11)$$

Because there is no way to separate this equation into a linear combination of X_1 and X_2 , a Taylor series approximation has been applied.

3.2.4 Conversion to a General Factor Model

The mass balance factor model defined by Equations (3.3) and (3.8) is based on a set of observation variables Y that are defined by a set of cross correlated hidden variables X and an observation matrix C . However, general Factor Analysis methods estimate loadings of hidden variables having a mean of 0 and a covariance of I , which is a property of the standardized variation source n in Equation (3.3).

An general factor model can be obtained by substituting $\mu + L_x n$ for X into Equation (3.8). This results in observations of Y being explained in terms of standardized latent variables n instead of the state variables X

$$Y = C(\mu + L_x n) + r + \beta + \varepsilon \quad (3.12)$$

This results in Y having the following distribution

$$Y \sim N_p(C\mu + r + \beta, \Sigma_Y + C'L'_x L_x C) \quad (3.13)$$

where C and r are specified by process knowledge, so they do not need to be estimated. The principal interest of this work is to estimate measurement and process covariance, Σ_Y and $L'_x L_x$ in a way that is consistent with a mass balance model structure. Estimating these parameters also allow us to obtain a rudimentary estimate of instrument bias β .

3.2.5 Bayesian Networks

Estimating Σ_Y and $L_x L'_x$ is done using a Bayesian network approach which can explicitly estimate statistics of X as well as Y given X by allowing us to fix the values of C . This results in variance estimates that are consistent with the mass balance model structure.

A Bayesian network is a visual representation of causal relationships denoted by nodes and arrows; arrows always point from parent to child which indicates the type of reasoning that must be done. If a parent value is known, it can be used to *predict* the values of child nodes.

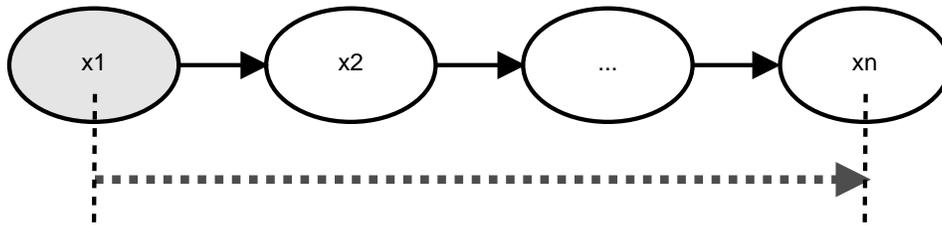


Figure 3.3: Predictive Reasoning

If a child value is known as shown in Figure 3.4 it can be used to *diagnose* the values of the parent nodes.

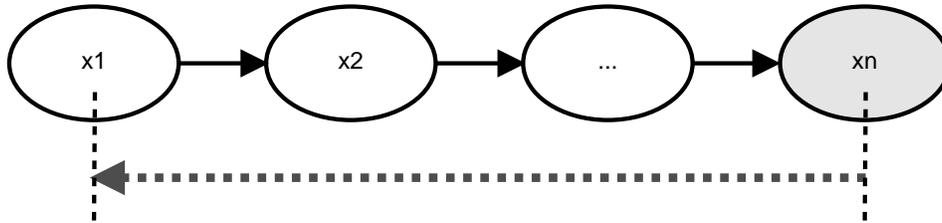


Figure 3.4: Diagnostic Reasoning

A comprehensive description of Bayesian inference for Gaussian distributions can be found in [3].

3.3 Bayesian Networks and Variance Estimation

3.3.1 Bayesian Example

Consider the industrial example. In order to remove bias from the measurements, means are subtracted so that we can focus on variance estimation. This converts Equations (3.3) and (3.8) to the form shown in Equations (3.14) and (3.15).

$$\begin{pmatrix} x_1 - \bar{x}_1 \\ x_2 - \bar{x}_2 \\ x_3 - \bar{x}_3 \end{pmatrix} = \begin{pmatrix} 300 & 0 & 0 \\ 150 & 40 & 0 \\ 0 & 0 & 120 \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \quad (3.14)$$

$$\begin{pmatrix} y_1 - \bar{y}_1 \\ y_2 - \bar{y}_2 \\ y_3 - \bar{y}_3 \\ y_4 - \bar{y}_4 \\ y_5 - \bar{y}_5 \\ y_6 - \bar{y}_6 \\ y_7 - \bar{y}_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1/2.1 & 1.0 & 0 \\ C_o & C_w & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 - \bar{x}_1 \\ x_2 - \bar{x}_2 \\ x_3 - \bar{x}_3 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \end{pmatrix} \quad (3.15)$$

where x_1 represents overall oil sand flow, x_2 represents water flow, and x_3 represents hopper level change while y_i represents an instrument output. C_o and C_w are parameters that are obtained by Taylor series approximations, and are thus dependent on the operating condition.

The model from Equations (3.14) and (3.15) has the corresponding Bayesian network shown in Figure 3.5. Note, solid arrows symbolize relationships that are predetermined and broken ones symbolize relationships that need to be learned. Likewise, solid nodes have known variance, but broken ones require variance estimation. For clarity of relationship, in Figure 3.5, mutually correlated variables are placed in the same node while uncorrelated are placed in separate nodes.

3.3.2 Bayesian Parameter Learning with Hidden Variables

Data from this model contains underlying variables that are hidden (we treat real states as hidden and measurements as observed) thus, the parameter estimation methods must be able to deal with hidden variables. The most widely used parameter estimation methods that can deal with hidden variables are *Gibbs Sampling* (GS) and *Expectation Maximization* (EM) [14]. A popular software used for *Gibbs Sampling* is *WinBUGS*, while the EM algorithm is used in a Bayesian Network toolbox for MATLAB, written by Kevin Murphy [18]. This work however, focuses on the use of the EM algorithm; detailed information can be found in a paper published by Dempster, Laird and Rubin [7]. Being an iterative approach, the EM algorithm obtains a new estimate of parameters $\hat{\theta}_{n+1}$ given a previous set of parameters $\hat{\theta}_n$ by successive substitution and maximization.

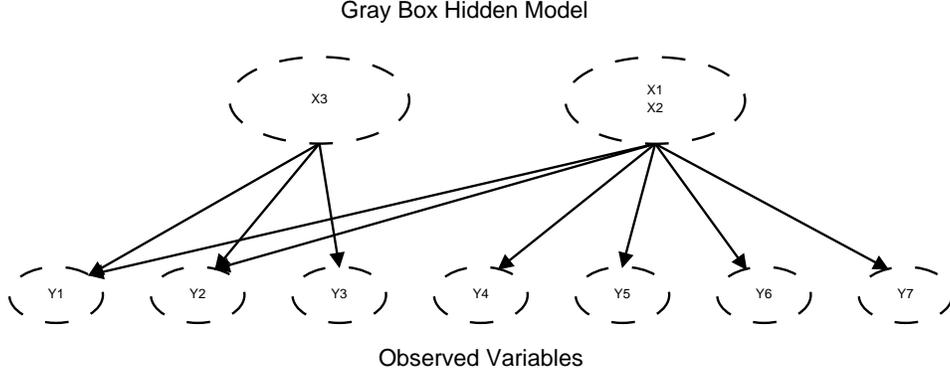


Figure 3.5: Bayesian Representation of Mass Balance Factor Model

$$\hat{\theta}_{n+1} = \arg \max_{\theta} \left[\int P(X|Y, \hat{\theta}_n) \ln [P(Y, X|\theta)] dX \right] \quad (3.16)$$

where X is a missing or hidden variable, Y is a set of observed data, and $\hat{\theta}$ is an estimate of the real parameter set θ . This process can be broken down into two steps:

1. Expectation: When we have a set of initial values for θ_n , we can compute the probability density function (PDF) $P(X|Y, \hat{\theta}_n)$ as a function of X , conditioned by a set of parameters θ_n which are to be held constant.

$$P(X|Y, \hat{\theta}_n) = \frac{P(Y|X, \hat{\theta}_n)P(X|\hat{\theta}_n)}{\int P(Y|X, \hat{\theta}_n)P(X|\hat{\theta}_n)dX} \quad (3.17)$$

where $\hat{\theta}_n$ is a set of node-related parameters such as mean, variance and path coefficients at iteration n , $P(X|\hat{\theta}_n)$ is the multivariate normal PDF given by Equations (3.2) and (3.3), and $P(Y|X, \hat{\theta}_n)$ is another multivariate normal PDF given by Equation (3.13). One should also note that $\int P(Y|X, \hat{\theta}_n)P(X|\hat{\theta}_n)dX$ is simply a normalization constant, since $\hat{\theta}_n$ is constant and X is marginalized out; this results in Equation (3.17) being a relatively simple function of X . At this point it is also useful to note that

$$\ln [P(Y, X|\theta)] = \ln [P(Y|X, \theta)P(X|\theta)] \quad (3.18)$$

which makes explicit use of the model-defined PDFs $P(Y|X, \theta)$ and $P(X|\theta)$; these are similar to those used in Equation (3.17), but allows the parameters θ to vary. Combining Equations (3.17) and (3.18) and marginalizing out X by integration results in the expectation shown below.

$$E_{X|Y, \hat{\theta}_n} \{ \ln P(Y, X|\theta) \} = \int P(X|Y, \hat{\theta}_n) \ln [P(Y, X|\theta)] dX \quad (3.19)$$

This expectation is determined by the previous fixed parameters $\hat{\theta}_n$ and the variable parameters θ which are to be optimized in the maximization step.

2. Maximization: the objective function obtained in Equation (3.19) —a function of θ since X was marginalized out— can be maximized according to Equation (3.16).

Calculations involving Bayesian networks were done using the BNT which is an open-source toolbox for MATLAB written by Kevin Murphy, and can be found at the url in [19]. The files have been made open to the public via the University of British Columbia Computer Science website as well as Google Project Hosting. Details on software use are given by [19] and a paper has been written on the subject [18]. The EM algorithm however, requires initial values to which results can sometimes be sensitive.

3.3.3 Principal Component Analysis

Principal component analysis aims to break down the sample covariance matrix into linearly independent components, and is often used as a first step toward factor analysis to determine the number of latent variables. In general, the goal of PCA is to rotate the axes of p -dimensional covariance matrix so as to coincide with directions of maximum variability, resulting in a diagonal matrix [12]. This allows greater variance to be described by a few major underlying independent variables. Because measurements have very different scales and variances, the standardized PCA method is used. Consequently, the *sample correlation matrix* R is analyzed instead of the *sample covariance matrix* S . R can be obtained by transforming S using S_y which is the main diagonal of S . This transformation is shown in Equation (3.20).

$$R = S_y^{-1/2} S S_y^{-1/2} \quad (3.20)$$

Let $q \leq p$ represent the number of principal components considered when performing PCA. $E_{(q)} = e_1, e_2, \dots, e_q$ would then be a matrix of q eigenvectors, and $\Lambda_{(q)}$, a diagonal matrix of eigenvalues $\lambda_1, \dots, \lambda_q$ of the correlation matrix R . The correlation matrix R , can be redefined according to spectral decomposition in Equation (3.21).

$$R = E_{(p)} \Lambda_{(p)} E'_{(p)} \quad (3.21)$$

$$\tilde{L}_s = E_{(q)} \Lambda_{(q)}^{1/2} \quad (3.22)$$

$$R \approx \tilde{L}'_s \tilde{L}_s \quad (3.23)$$

where \tilde{L}_s is matrix of factor loadings according to PCA, defined according to Equation (3.22). Equation (3.23) should hold true as long as q is large enough to capture sufficient variance. The objective is to find the minimal value of q that would still capture most of the variance expressed in Equation (3.23). PCA is actually an approximate factor analysis solution; the objective of PCA is to determine an adequate number of factor loadings to use in factor analysis.

3.3.4 Factor Analysis by Modified Principal Components

While there are many methods to perform factor analysis (FA), the modified PCA method was used for this chapter. This method of FA is similar to PCA but accounts for noise within the individual observations ($Y_1 \dots Y_p$) themselves, which is denoted as Ψ_s . FA techniques attempt to separate R into its diagonal *specific variance* Ψ_s and its loading-based *communality* $L_s L_s'$ as shown in Equation (3.24).

$$R \approx L_s L_s' + \Psi_s \quad (3.24)$$

L_s and Ψ_s are directly estimated from R using the iterative *Principal Factor Solution* which can be found in [12]. Once Ψ_s and L_s have been estimated, they can be converted to their unstandardized forms according to Equations (3.25) and (3.26).

$$L = S_y^{1/2} L_s \quad (3.25)$$

$$\Psi = S_y^{1/2} \Psi_s S_y^{1/2} \quad (3.26)$$

$$(3.27)$$

FA makes estimates on the assumption that the model can take the form of a zero-mean factor model as shown in Equation (3.28).

$$\tilde{Y} \approx Ln + \varepsilon_{(\Psi)} \quad (3.28)$$

where \tilde{Y} is conditioned observation data (with zero mean), n a source of variation, identical to the n used in Equation (3.3), and $\varepsilon_{(\Psi)}$ is observation noise with an estimated covariance Ψ . Ideally $\varepsilon_{(\Psi)} = \varepsilon$ but estimates of Ψ are not constrained to be consistent with mass balances, thus one must distinguish $\varepsilon_{(\Psi)}$ from ε . Because Equation (3.13) defines the distribution of a zero-mean linear factor model, it is possible to perform a substitution to generate an FA model. When subtracting the sample mean value of Y from Equation (3.12), the factor model reduces to Equation (3.29).

$$Y - \bar{Y} = CL_x n + \varepsilon \quad (3.29)$$

If the data can be arranged in a manner that is consistent with Equation (3.28), FA estimates the loading matrix L as well as the covariance of $\varepsilon_{(\Psi)}$. It becomes clear from Equations (3.28) and (3.29) that FA makes an attempt to estimate a value for $L = CL_x$ as well as $\Sigma_Y = \Psi$. The principal cause of concern for using Ψ as an estimate for Σ_Y is the fact that black box estimation methods such as FA cannot force consistency with the mass balance equation structure onto estimates of L and Ψ . Particular trouble arises in the case where measurements from one instrument appears to be relatively uncorrelated with the others. When this happens, it is difficult to determine whether the noise is coming from the measurement or from a separate hidden state n . Nevertheless, measurements that cause such confusion are often the exception and not the norm, thus as an initial guess for Σ_Y , Ψ is still valuable. The subsequent Bayesian learning will essentially make corrections to Ψ by forcing mass balance consistency to get Σ .

3.4 Estimating the Gross Error Distribution

While the principal focus of this work is using Bayesian networks to estimate process and measurement noise covariance, this method can also be used to estimate gross errors such as instrument bias. Since covariance matrices are now specified, the Bayesian network can be simplified to a form as shown in Figure 3.6

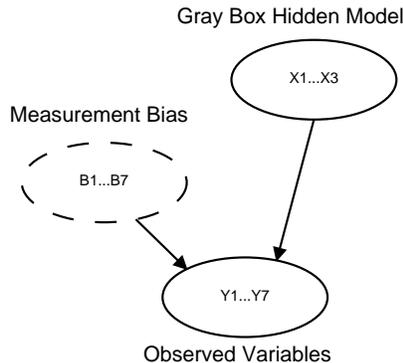


Figure 3.6: Bayesian Representation with Bias

Because it is desirable to estimate instrument bias, sample means should not be removed from observed data; thus, we revert to the direct use of Equations (3.3) and (3.8).

Gross error estimation can be performed through Bayesian learning in the same manner as variance was learned. However, the gross error estimation problem is not consistent with the FA structure because the new network has more parent components than children. Due to this, FA cannot be used to generate initial values.

The first step in this procedure is estimating the process mean μ as defined in Equation 3.3. This can be done by specifying the gross error terms β as equal to zero and learning the process mean. This is consistent with the preliminary step for conventional data reconciliation which assumes that there are no gross errors present; these estimates are weighted mean, weighed according to the noise variance of the corresponding measurement. When learning the process mean from the original data, the best initial value is the design specifications. One can improve the steady-state estimate by simultaneously estimating the steady-state mean with the gross error prior variance. This reduces the contamination of the process mean from gross errors, giving a better steady state estimate for the first step. The improvement comes from the fact that for traditional methods, the only weighting factor that is taken into account is the measurement noise variance. However, when learning the prior gross error variance at the same time, the prior gross error variance that best explains the data is also being taken into account when assessing instrument reliability. Measurements that can be singled out as being inconsistent will have a correspondingly large gross error prior variance, causing that instrument to have less weight on the process mean estimate.

Once a process mean has been estimated, these values can be used to learn the gross error mean and posterior gross error variance. From the gross error mean and variance, hypothesis tests can be performed to determine whether

Table 3.1: Variable Definitions

Symbol	Mean	S.D.	Meaning
x_1	1000	300	Real Oil Sand Flow
x_2	500	155.2	Real Water Flow
x_3	0	120	Real Hopper Level Change
y_1	0	200	WENCO Database Value
y_2	0	90	First Weightometer Readings
y_3	0	130	Hopper Level Readings
y_4	0	80	Second Weightometer Readings
y_5	0	60	Slurry Flow Meter Readings
y_6	0	0.2	Density Meter Readings
y_7	0	50	Water Flow Meter Readings
β_1	0	0	Bias
β_2	-30	0	Bias
β_3	0	0	Bias
β_4	300	0	Bias
β_5	-80	0	Bias
β_6	-0.2	0	Bias
β_7	0	0	Bias

or not a gross error of zero is within the suitable confidence level. If there are any significant gross errors, one might want to re-learn the process mean after replacing the gross error mean value of zero with the new gross error estimate for that instrument.

3.5 Simulation of Overall Process

Before applying these methods to real process data, a simulation of the industrial system was performed by generating data for a fictional operating point. Verification of this method is best done through simulation since the true parameters are available beforehand in a simulation. The real parameters are given in Table 3.1.

Note that gross errors were introduced into this simulation in the form of instrument bias. Gaussian data was generated for the values of n and ε so that

simulated values for X could be obtained using Equation (3.3) and Y could be calculated according to the nonlinear model in Equation (3.7). 900 data points were generated for the system. When performing appropriate substitutions to Equations (3.3) and (3.7), simulated data can be directly calculated according to Equations (3.30) and (3.31)

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1000 \\ 500 \\ 0 \end{pmatrix} + \begin{pmatrix} 300 & 0 & 0 \\ 150 & 40 & 0 \\ 0 & 0 & 120 \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \quad (3.30)$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} x_1 + x_3 \\ x_2 + x_3 \\ x_3 \\ x_1 \\ 2.1^{-1}x_1 + 1.0^{-1}x_2 \\ (x_1 + x_2)^{-1}(2.1x_1 + 1.0x_2) \\ 1.0^{-1}x_2 \end{pmatrix} + \begin{pmatrix} 200n_4 \\ 90n_5 \\ 130n_6 \\ 80n_7 \\ 60n_8 \\ 0.2n_9 \\ 50n_{10} \end{pmatrix} \quad (3.31)$$

where n_i is randomly generated Gaussian noise with mean 0 and variance 1. This simulation resulted in a correlation matrix given in Table 3.2.

Table 3.2: Correlation Matrix for Direct Nonlinear System

y_1	y_2	y_3	y_4	y_5	y_6	y_7
1.00	0.83	0.15	0.78	0.79	-0.03	0.72
0.83	1.00	0.17	0.89	0.88	-0.03	0.82
0.15	0.17	1.00	-0.07	-0.07	-0.02	-0.07
0.78	0.89	-0.07	1.00	0.94	-0.03	0.86
0.79	0.88	-0.07	0.94	1.00	-0.05	0.91
-0.03	-0.03	-0.02	-0.03	-0.05	1.00	-0.07
0.72	0.82	-0.07	0.86	0.91	-0.07	1.00

As mentioned before, all relationships are linear except for the case of y_6 . Performing a Taylor series approximation around the operating points $x_1 = 1000$ and $x_2 = 500$ yields the following parameters: $C_{sand} = 1.836 \times 10^{-4}$ and $C_{water} = -4.492 \times 10^{-4}$.

3.5.1 Variance Estimation Results: EM by Uninformed Initial Guesses

When performing parameter estimation, it was found that certain parameters were sensitive to initial values. Most parameters were well-converged except y_4 and y_7 . Again, if the initial values are not carefully considered, the model may

assume that one instrument is unduely precise in measuring the corresponding underlying variable. A summary of results can be found in Table 3.5.

3.5.2 Variance Estimation Results: PCA and FA Results

PCA was performed with the intent of determining how many hidden variables were needed to explain the covariance (or correlation) matrix. Figure 3.7(a) displays a plot of % variance explained by the n^{th} principal component. Because improvements were marginal after three principal components, three underlying variables are sufficient for this model (which happens to be the number of hidden process states). From these results, it was shown that roughly 92.1% of the variance can be explained using three principal components.

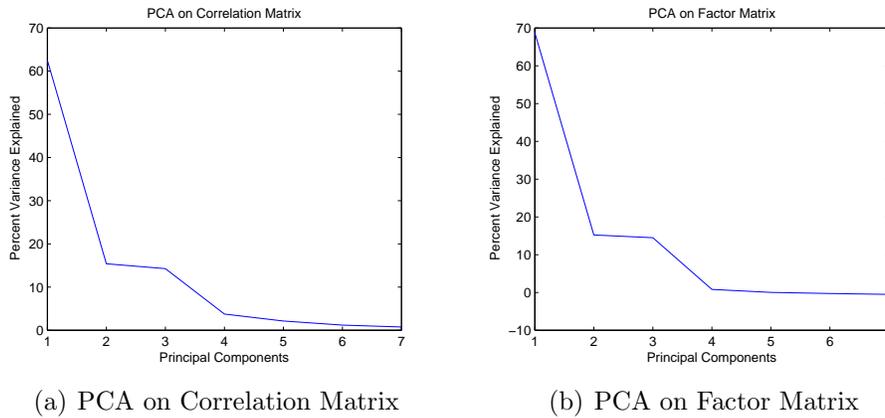


Figure 3.7: FA Validation using PCA

Table 3.3 displays principal components labeled by their scores in the units of % variance explained.

Table 3.3: PCA vectors by decreasing PC

Variable	62.5%	15.4%	14.3%	3.8%	2.1%	1.2%	0.8%
y_1	0.42	-0.16	-0.05	0.82	0.33	0.10	0.05
y_2	0.45	-0.14	-0.04	0.00	-0.42	-0.77	-0.05
y_3	0.01	-0.95	-0.09	-0.24	0.04	0.17	0.01
y_4	0.46	0.10	-0.01	-0.11	-0.49	0.47	0.55
y_5	0.46	0.11	0.01	-0.19	-0.06	0.34	-0.78
y_6	-0.03	0.11	-0.99	-0.04	0.03	0.00	0.00
y_7	0.44	0.12	0.03	-0.47	0.69	-0.15	0.26

When using three factors (as decided from PCA), roughly 98.4% of the variance can be explained by three factors when independent noise was taken into account. One can now use this information to construct a network model. From the overall assumption that three hidden variables cause measurement noise variance, the corresponding Bayesian network is shown in Figure 3.8. Note that the network is simplified by omitting connections between nodes with negligible correlation.

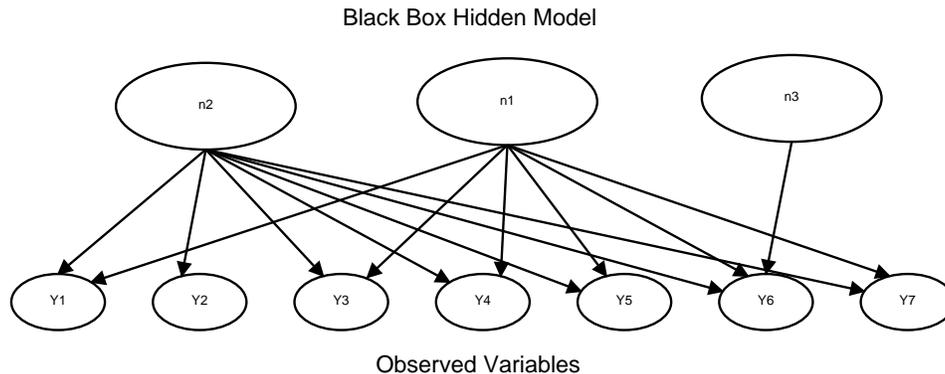


Figure 3.8: Approximate Factor Analysis Model

Factor loadings along with the estimated standard deviation are displayed in Table 3.4. Note that these are scaled so that underlying factors have a mean of zero and standard deviation of 1, according to Equation (3.29). Real values for CL_x are presented in Table 3.4 with the sample standard deviation calculated from simulated data.

Because loadings were known beforehand through mass balances, the principal interest is in estimating measurement noise variance. Overall, noise values produced by FA are fairly close to the simulated values. The only issues lie within the noise variance for y_3 and y_6 , both of which are severely underestimated; this is because the instruments y_3 (surge level) and y_6 (density) are relatively uncorrelated with the rest of the instruments. If an instrument is relatively uncorrelated with the others, FA will attempt to heavily associate this independency toward a separate underlying cause n_i and reduce the effect of n_i on other measured variables. This is easily seen in Table 3.5 where the variance for x_3 is highly overestimated because x_3 is the only hidden variable that has a significant effect on y_3 . Increasing the variance for x_3 , and reducing its effects on other variables explains away much of the independent noise of the y_3 measurements.

With the measurement noise variance calculated, the variance of hidden variables can be estimated either by analyzing the loadings or by subtracting measurement noise variance from the measurement variance. Since measurement noises were assumed to be uncorrelated with process noise, Equation (3.32) can be used to estimate process noise variance.

$$\sigma_{measured}^2 = \sigma_{process}^2 + \sigma_{instrument}^2 \quad (3.32)$$

Results from FA are compared with other results in Table 3.5.

Table 3.4: Loadings and Noise from FA Estimation vs. Simulated Values

Est	L_1	L_2	L_3	Sim	L_1	L_2	L_3
y_1	335.8	-54.9	-20.55	y_1	300	-120	0
y_2	336.3	-53.4	-20.04	y_2	300	-120	0
y_3	4.5	-158.4	-28.2	y_3	0	-120	0
y_4	316.2	33.6	-2.03	y_4	300	0	0
y_5	282.7	30.8	4.16	y_5	292.9	0	40
y_6	-0.01	0.04	-0.185	y_6	0.0012	0	-0.0195
y_7	124	14.1	5.58	y_7	150	0	40

3.5.3 Variance Estimation Results: Bayesian Learning with FA Initial Values

While FA results were suboptimal, they were found to be an effective starting point for Bayesian learning by means of the EM algorithm. Bayesian learning did not bring a dramatic improvement of variance estimation for most instruments when compared to FA initial values. However, it significantly improved variance estimates for difficult-to-estimate instruments such as y_3 and y_6 , leading to a much more reliable estimation overall. A summary of all results is shown in Table 3.5.

Table 3.5: Summary Table of Variance Estimates

Variable	Simulated σ	EM	FA	EM with FA
x_1	319	314	333	314
x_2	161	180	155	174
x_3	117	126	161	121
y_1	222	207	212	203.7
y_2	99.9	105	102	94.3
y_3	144	133	62.8	114.1
y_4	88.8	2.25	94.2	89.8
y_5	66.6	83.7	62.9	56.9
y_6	0.222	0.221	0.063	0.220
y_7	55.5	14.7	59.9	47.69

3.5.4 Bias Estimation: Bayesian Learning Results

After the variance parameters were estimated, Bayesian learning was used to estimate gross errors from bias. A summary of results is given in Table 3.6. Mean and standard deviation values were given and estimation error percentiles were calculated according to the normal distribution. Since the initial guess of bias was zero, the bias estimates were somewhat conservative.

Table 3.6: Estimated Bias from Bayesian Learning (Zero Bias Initial)

Variable	Simulated	EM	σ_β	Percentile Err
β_1	0	-15.2	122	10.5
β_2	-30	-58.5	58.8	37.2
β_3	0	29.9	76.8	30.3
β_4	300	235	50.6	79.8
β_5	-80	-104	43.4	42.7
β_6	-0.2	-0.223	0.199	9.02
β_7	0	4.58	33.3	10.9

Because β_4 was significant, a second learning iteration was used by using this bias estimate instead of zero for the prior bias mean. Estimates shown in Table 3.7 gave improved bias estimation results since the estimated state was less contaminated.

Table 3.7: Estimated Bias from Bayesian Learning (Modified Bias Initial)

Variable	Simulated	EM	σ_β	Percentile Err
β_1	0	-15.7	118	10.6
β_2	-30	-58.7	57.0	38.5
β_3	0	-7.01	74.3	7.52
β_4	300	275	49.0	38.8
β_5	-80	-96.5	41.9	30.5
β_6	-0.2	-0.211	0.199	4.18
β_7	0	-4.38	33.2	10.8

3.6 Special Case, Capacity with no Scaling

3.6.1 Time Dependency As Information

With this factor model framework and philosophy, we can re-visit the previous problem of estimating the hopper constant c_H . Recall that hopper level indicator gave a % capacity reading, but the maximum capacity was not known. As before, the aim is to obtain a hopper scalar c_H that would enable the reliable estimation of the real surge hopper mass accumulation x_3 using surge hopper level readings y_3 .

$$x_3 = c_H y_3 \quad (3.33)$$

Returning to the industrial example in Figure 3.2, consider a subsystem that focuses on WI1, LI1 and WI2 where an intermediate holding vessel is placed between two streams; for this subsystem, the observation vector Y is redefined as follows.

$$y_1 (i) = \text{LI1}_{(t_i+\Delta t)} - \text{LI1}_{(t_i)} \quad (3.34)$$

$$y_2 (i) = \int_{t_i}^{t_i+\Delta t} \text{WEI1-WEI2} dt \quad (3.35)$$

Traditionally, cross-calibration can be done by a comparison of means for flow measurements; however, proper cross-calibration can be difficult for capacity units because they have zero mean. Furthermore, if the process variance is much less than the measurement noise variance (as in a steady-state process), instrument measurement correlation is very low resulting in imprecise estimates of capacity constants. Attempts toward direct regression were made in the previous chapter; however, regression results changed over the different sampling time intervals Δt due to the fact that observation noise levels for weightometers were dependent on Δt . An approximate solution was obtained by using the regression results from the time interval that gave the strongest correlation.

The factor model approach in Equations (3.3) and (3.8) can be applied to this system to obtain the model shown in Equation (3.36)

$$x = L_{x(\Delta t)} n \quad (3.36)$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} x + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_{2(\Delta t)} \end{pmatrix}$$

where subscript (Δt) indicates that the corresponding variable is dependent on Δt , x is a single hidden state—the real surge hopper accumulation—and n is standard Gaussian noise with mean of 0 and variance of 1. y_1 represents the difference in surge hopper readings while y_2 is the summed difference between weightometer readings. C , is the observation matrix and $L_{x(\Delta t)}$ is the loading on the process noise (which is equal to the standard deviation of x) and is dependent on Δt ; $\varepsilon_{2(\Delta t)}$ is the measurement noise from the weightometer difference and is also a function of Δt while ε_1 , the hopper level noise, has constant variance for all values of Δt . The expected covariance structure from this model is represented in Equation (3.37).

$$\text{cov}(y) = \begin{pmatrix} \sigma_1^2 + C_1^2 L_{x(\Delta t)}^2 & C_1 C_2 L_{x(\Delta t)}^2 \\ C_1 C_2 L_{x(\Delta t)}^2 & \sigma_{2(\Delta t)}^2 + C_2^2 L_{x(\Delta t)}^2 \end{pmatrix} \quad (3.37)$$

The covariance at one time interval specifies 3 combinations of parameters, but there are 5 unknown parameters ($L_{x(\Delta t)}$, C_1 , C_2 , σ_1^2 , and $\sigma_{2(\Delta t)}^2$). Thus a solution from a straightforward mass balance factor model is unavailable. Nevertheless, because some parameters vary with Δt while others do not, it is possible to use this information to perform a regression analysis.

Before solving this problem, some parameters can be specified. Continuous flow measurements give direct units; thus C_2 has a value of 1 by definition, while C_1 is equal to c_H^{-1} since C_1 converts x values to y and c_H converts y values to x . When making appropriate substitutions for C_1 and C_2 , Equation (3.37) reduces to (3.38). Let $S_{(\Delta t)}$ be the sample covariance matrix of instrument data, sectioned off according to a given sampling time Δt ; its expected value according to this model is defined as

$$E(S_{(\Delta t)}) = \begin{pmatrix} \sigma_1^2 + c_H^{-2} L_{x(\Delta t)}^2 & c_H^{-1} L_{x(\Delta t)}^2 \\ c_H^{-1} L_{x(\Delta t)}^2 & \sigma_{2(\Delta t)}^2 + L_{x(\Delta t)}^2 \end{pmatrix} \quad (3.38)$$

Sample covariance matrices are directly available for chosen values of Δt ; because of this, it is possible to exploit this model structure and perform regression for covariance values over different values of Δt . Since a 2×2 covariance matrix has three unique entries, there are three possible regression equations.

$$S_{11(\Delta t)} \simeq \sigma_1^2 + \frac{1}{c_H^2} L_{x(\Delta t)}^2 \quad (3.39)$$

$$S_{12(\Delta t)} \simeq \frac{1}{c_H} L_{x(\Delta t)}^2 \quad (3.40)$$

$$S_{22(\Delta t)} \simeq \sigma_{2(\Delta t)}^2 + L_{x(\Delta t)}^2 \quad (3.41)$$

where $S_{ij(\Delta t)}$ is the entry corresponding to the i^{th} row and j^{th} column of the sample covariance matrix $S_{(\Delta t)}$; recall that subscript (Δt) denotes that this variable is a function of Δt . The direct values for $L_{x(\Delta t)}^2$ are not available, but Equation (3.40) indicates that values for $\frac{1}{c_H} L_{x(\Delta t)}^2$ can be directly obtained from the sample covariance matrix. By substitution, relations shown in Equations (3.39) to (3.41) can be reduced to the models shown in Equations (3.42) and (3.43).

$$S_{11(\Delta t)} \simeq \sigma_1^2 + \frac{1}{c_H} S_{12(\Delta t)} \quad (3.42)$$

$$S_{22(\Delta t)} \simeq \sigma_{2(\Delta t)}^2 + c_H S_{12(\Delta t)} \quad (3.43)$$

The model shown in Equation (3.42) can be used directly for regression the since surge level indicator variance σ_1^2 and the surge hopper scalar c_H are independent of Δt ; however, weightometer variance $\sigma_{2(\Delta t)}^2$ is an unknown

function of Δt . If $\sigma_{2(\Delta t)}^2$ and $L_{x(\Delta t)}^2$ are not independent, regression is not possible. Thus, due to direct knowledge of time independence, Equation (3.42) is much more suitable for regression analysis.

3.6.2 Regression Results Using Industrial Process Data

Using industrial data, the first task was to perform regression analysis on Equation (3.42) in order to get an estimate of c_H . Figure 3.9(a) displays the data for linear regression and parameter estimation results for c_H as well as σ_1^2 . Both figures pertain to time intervals between 4 and 30 minutes. Figure 3.9(a) shows a very clear linear trend, which results in a precise value of the c_H parameter which is the inverse of the slope. Since the σ_1^2 parameter is somewhat insignificant at this scale, confidence intervals were wider. For this time interval region, it was also found in Figure 3.9(b) that weightometer variance $\sigma_{2(\Delta t)}^2$ was linear with respect to Δt^2 .

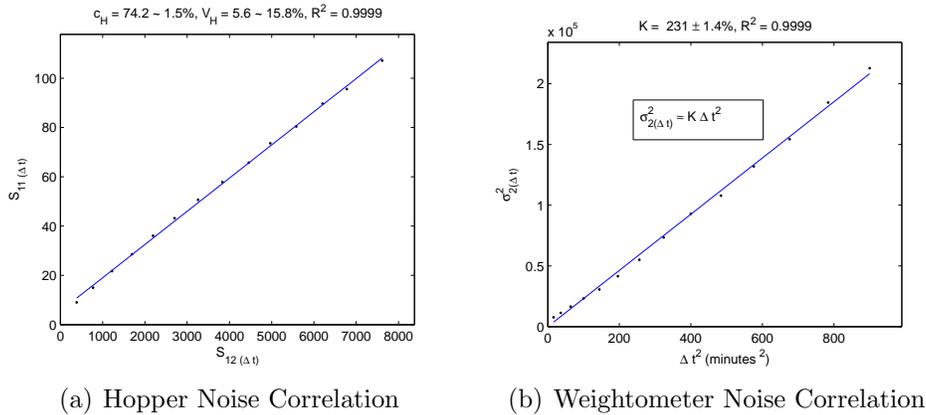


Figure 3.9: Regression Analysis for Noise

In order to get a better estimate of σ_1^2 , data points were generated using smaller time windows, as it is within these time windows that the parameter is better defined. Data was generated for time intervals of 2 to 10 minutes. Confidence intervals for σ_1^2 were much tighter, and marginal plots of σ_1^2 showed little trend activity.

Now that this method has been verified to accurately determine the surge hopper scalar, FA and Bayesian learning can be performed on industrial data.

3.7 Industrial Application

For the industrial case, the variance and bias of instruments were estimated in the same manner as the simulation. Special attention was paid toward obtaining a data set wherein measurements followed a multivariate normal distribution. Summaries for the resulting measurement noise variance are shown in Table 3.7. One observation was that WENCO variance was quite substantial.

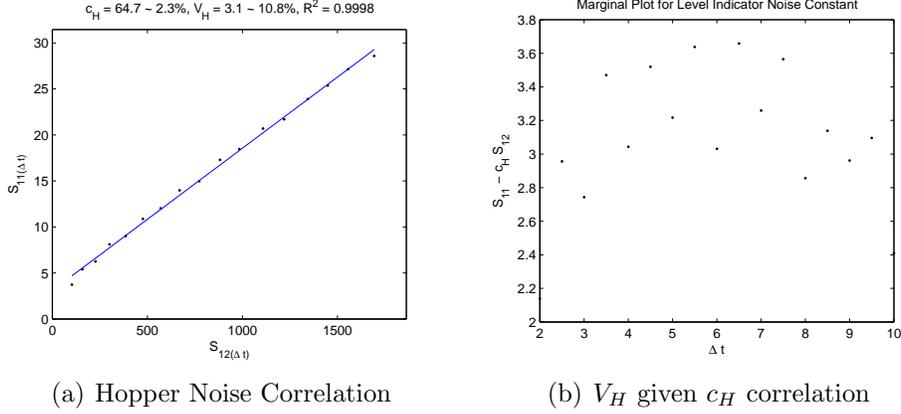


Figure 3.10: Regression focusing on V_H estimation

Table 3.8: Estimated Variance from Bayesian Learning (FA Initial)

Variable	FA	EM with FA
x_1	386	246
x_2	40.8	55.8
x_3	423	329
y_1	364	429
y_2	151	179
y_3	153	207
y_4	51	104
y_5	11.2	12.2
y_6	0.011	0.102
y_7	9.75	18.1

After the variance parameters were estimated, bias means and variances were obtained in the same manner as before using Bayesian learning with results shown in Table 3.7. $\sigma_{\beta Prior}$ represents the RMS error of each instrument, which includes both random and systematic error; the RMS error is used to adjust the instrument measurement weight for the first state estimate. μ_{β} is the estimated mean gross error while $\sigma_{\beta Posterior}$ represents random error of each instrument after the state is estimated. Percentiles in the confidence result. The major concern from the industrial standpoint was the second weightometer Y_4 , which was reported to have biases of up to +30%. One can see from the results that the corresponding bias β_4 has substantial bias with a confidence of 99.997%. This result has been confirmed by engineers within the industry.

Table 3.9: Industrial Estimated Variance from Bayesian Learning

Variable	$\sigma_{\beta Prior}$	μ_{β}	$\sigma_{\beta Posterior}$	Percentile
β_1	250	47.6	237	15.9
β_1	260	80.9	242	26.2
β_1	2.17	-0.01	2.1	0.27
β_1	357	306	80.6	100.0
β_1	22.6	-15.2	14.6	70.3
β_1	0.18	-0.16	0.13	78.8
β_1	19.7	5.22	14.4	28.4

3.8 Conclusions

While Bayesian learning is capable of optimizing parameters in a way that is consistent with both mass balances and measurement covariance, this method comes at a price of requiring an initial value. However, FA is capable of obtaining effective initial values. When FA is used to initialize Bayesian learning, results tend to converge well since Bayesian learning can force mass balance consistency onto estimations. Results for this method were consistent both in simulation and in practice.

When trying to obtain adequate scaling for capacity, the mass balance factor model is adequate in specifying a regression model with respect to the time window. The independency of capacity measurement variance with respect to Δt allowed the use of linear regression analysis to obtain an estimation of the scaling factor. Such an approach can solve the problems of implementing the Bayesian learning approach when intermediate holding measurements are not properly scaled.

1A version of this chapter has been accepted for publication. Gonzalez, Huang, 2010. Industrial Engineering and Chemistry Research

Chapter 4

Dynamic Bayesian Approach to Instrument Performance Monitoring and Correction

This chapter applies results from the previous chapter to employ a Bayesian methodology to detect instrument gross errors and to estimate them in real time; the algorithm consists of four steps. The first step uses a mass balance based Kalman filter to estimate the hidden state, while the second step uses the estimated states and their covariance to calculate standardized residual errors. The third step uses another Kalman filter to estimate systematic trends in the residual errors. The final step calculates P-values from traditional hypothesis test procedures, and uses a dynamic Bayesian network to refine statistical evidence from hypothesis tests. This method is tested on a simulation and then applied to industrial data. Simulations demonstrate that this method is able to detect rapidly propagating gross errors, but as one would expect, there is a tradeoff between precision and the speed of detection. In the presence of multiple gross errors, as in the case of industrial data, this method tends to prioritize correcting large gross errors first.

4.1 Introduction

The previous chapter focused on the use of Bayesian learning techniques to identify noise and gross error parameters of a static process model based on mass balances. The method introduced in this chapter makes use of Bayesian learning to estimate a dynamic model which is used to detect gross errors in real time. This method is inherently related to data reconciliation and gross error detection; as mentioned before, traditional data reconciliation techniques attempt to find the hidden steady-state that best explains the data, while gross error detection performs statistical tests on measurements to determine if there is any systematic deviation from the reconciled steady state. Upon the detection of significant gross errors, suspicious measurements are discarded and data reconciliation is then re-performed with corrupted instruments removed.

The proposed method is a dynamic analogue of the traditional methods. A dynamic model based on mass balances is learned, and a Kalman filter is used to replace data reconciliation in order to obtain a reconciled dynamic

state estimate. Gross error detection is performed by filtering residual errors and calculating P-values according to the normal distribution; these P-values are used to enter evidence into a Dynamic Bayesian Network (DBN) which is a way of monitoring test results over time, checking for systematic unusual behavior. When gross errors are detected, the Kalman filter is modified in order to heavily penalize the corrupted instruments. The proposed method however, performs an additional task of correcting a corrupted instrument by estimating a corresponding gross error term so that a corrected instrument can be re-introduced into the state estimate.

The focus of this chapter is the practical application of a dynamic data reconciliation and gross error detection algorithm, similar to the objectives given in [16]. The difference is that this algorithm is inherently tailored toward a dynamic process and is consistent with Bayesian statistics. The key advantage to the proposed dynamic method is that it can detect the sudden development of gross errors in real time. Consequently, while multiple gross errors can be a problem in the static case, the dynamic method deals with errors as soon as they occur; this enables the dynamic method to deal with multiple gross errors in small manageable (or observable) groups. Figure 4 displays the methodology shown in this chapter.

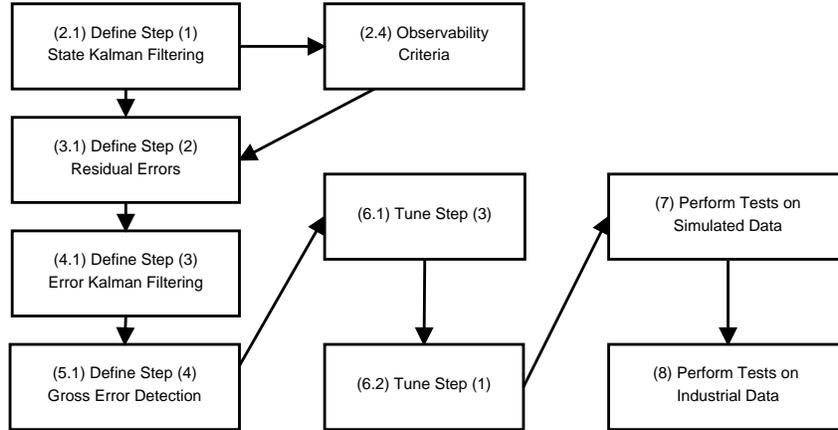


Figure 4.1: Chapter 4 Methodology

The four steps are briefly introduced in the Algorithm Overview section. In Section 2.1 Kalman filtering is introduced as a state estimation tool for Step (1), note that the instrument noise and process disturbance covariances are tuned using methods introduced in Chapter 3. Because this method adds gross errors as augmented states, the observability criteria must be introduced in Section 2.4. Section 3.1 addresses the calculation and standardization of residual errors done in Step (2) while Section 4.1 introduces the filtering procedure in Step (2) that filters these standardized residual errors. Finally, in Section 5.1, the Dynamic Bayesian Network is introduced as a tool for a dynamic version of hypothesis testing in Step (4). Section 6.1 uses principles addressed in Step (4) to develop an objective-based tuning method in Step (3). Section 6.2 uses results from the tuning of Step(3) to derive a tuning method for gross error state variance in Step (1). Finally, Sections 7 and 8 discuss results from simulation tests and industrial applications respectively.

4.1.1 Algorithm Overview

A schematic of the algorithm is shown in Figure 4.2. For each time step, the algorithm performs the following procedure: (1) Update the process and gross error states as new observations of instrument reliability become available; note that instruments that are determined as unreliable have a weighing penalty applied, and dynamic gross error states are only applied to unreliable instruments. (2) Calculate the statistical distance of each measurement. It is similar to data reconciliation techniques proposed by [22]. (3) Update the continuous gross error state, which serves as a filter for statistical distance results. (4) Perform an instrument reliability audit, using a DBN to track test results. This step consists of using filtered Z -values to test both the existence of a bias change and the nonexistence of a bias change; the successful test is the one which has a stronger P -value. The P -values are also used to enter the evidence into the DBN, which is then used to audit the instruments for reliability. Results from the reliability audit are used to assign reliability states for the next time step.

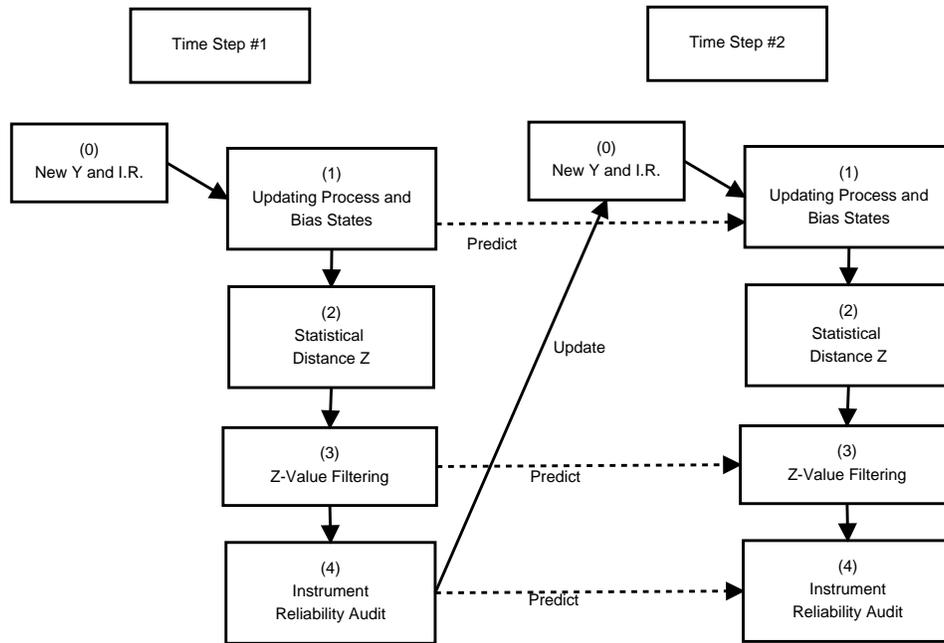


Figure 4.2: Visual Representation of Algorithm

One might try to simultaneously estimate the hidden process state as well the complete gross error states, eliminating steps (2) (3) and (4); however, as shown later, estimating a complete set of gross errors in addition to the hidden state causes the system to become unobservable.

It would also be possible to construct a large *Switching Kalman filter* which includes reliability states that can modify the Kalman filter. This would also eliminate steps (2) (3) and (4), but such an approach would be severely limited in two ways: (1) Computational power is exponential in terms of the number of instruments used; this is a severe limitation as gross error correction requires the use of a significant number of instruments in order to be reliable. (2) Unreliable instruments cannot be significantly penalized since actual instrument

noise variance must increase substantially in order for large penalization to be favored. Because of these two drawbacks, the proposed method was developed in order to reduce the computational burden, and to allow the instrument correction algorithm to err on the side of caution by heavily penalizing unreliable instruments.

4.2 Step(1) State Estimateion

Process model formulation deals with specifying a model and estimating model parameters so that step (1) in Figure 4.2 can be performed. While the previous chapter used a factor model for static conditions; dynamic conditions are now used which means that a state-space model is more appropriate. The first step uses a Kalman filter as a dynamic data reconciliation method. Some attempts have been previously made to use Kalman filtering for data reconciliation; however, its implementation has been limited due to difficulties in tuning the process model [1]. Because of this, an algorithm entitled Dynamic Data Reconciliation (DDR) has been developed by [1]. However, the Bayesian learning method discussed in Chapter 3 can be applied to dynamic models in order to overcome such tuning difficulties. This allows for the feasible application of the Kalman filter.

4.2.1 Static Observation Model

In the most general sense, the vector of observed measurements Y are a non-linear function $g(X)$ of the hidden states X with added gross errors β and uncorrelated instrument noise v .

$$Y = g(X) + \beta + v \quad (4.1)$$

Because gross error detection is based on mass balances, X is a minimal set of mass process variables that complete the mass balance. This methodology deals with linear systems, thus the nonlinear function $g(X)$ must be linearized by a first order Taylor Series expansion. This way, the function g is approximated by a linear observation matrix C and a first-order linearization reference r .

$$g(X) \approx CX + r \quad (4.2)$$

The visual representation of this static model can be represented by a Bayesian network with gross errors as shown in Figure 4.3. As in the previous chapter, Bayesian networks can be used to model causal relationships between random variables; nodes represent sets of variables, and arrows indicate the causal relationship.

Note that the node sizes as well as C and I are given in terms of the industrial example.

4.2.2 Hidden Dynamic Process Model

Because this method deals with a dynamic problem, the dynamic propagation of the states $f(X)$ must also be considered in Equation (4.3)

$$X_{(t)} = f(X_{(t-1)}, \mu_X) + w_{X(t)} \quad (4.3)$$

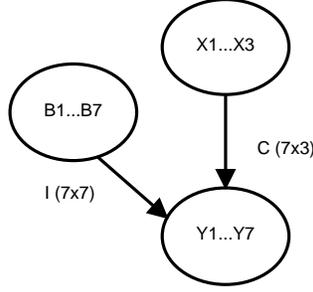


Figure 4.3: Static Bayesian Network

where $w_{X(t)}$ is cross-correlated process noise at time t , and μ_X is a process operating condition or steady-state. Combining Equations (4.1) and (4.3) results in a state-space model.

In practice, $f(X)$ and $g(X)$ are handled differently. On one hand, because $g(X)$ deals with instrument relationships toward hidden process states, the function $g(X)$ is relatively well understood. Most of the time, the relationship between the instrument observation and the hidden state is linear; however, nonlinear instruments can be adequately approximated by a Taylor Series approximation. On the other hand, $f(X)$ is a dynamic function of hidden states; the true process model is often times not well understood. Furthermore, because it deals with relationships between time steps, $f(X)$ is highly dependent on the time window Δt . For any known $f(X)$, the linear model is obtained by means of a Taylor Series approximation about the process steady-state.

$$X(t) \approx A(X_{(t-1)} - \mu_X) + \mu_X + w_{X(t)} \quad (4.4)$$

It is also common practice to remove the steady-states from the Y observations in order to simplify the model for X .

$$X(t) \approx AX_{(t-1)} + w_{X(t)} \quad (4.5)$$

For most cases, the purpose for the dynamic hidden state is to track the process operating point, which could be changed by the plant operator or slightly affected by process noise. A mixed model comprised of *random walk states* and *tracking states* is ideal for this application. Independent variables are assigned a random walk model; these variables are often flow rates of limiting reagents which are subject to plant scheduling. However, some variables are controlled in order to accommodate changes in the random walk variable.

For example, if the oil sand flow rate x_1 was scheduled, but a constant slurry density was required, then the water flow rate x_2 would track the oil sand flow rate x_1 which is assume to exhibit random walk. Thus the steady-state of x_2 is the value that maintains the desired ratio of \bar{x}_2/\bar{x}_1 . x_3 is the surge hoper accumulation and essentially independent of x_1 and x_2 ; however, it is constrained by the maximum hopper capacity and thus follows a fading-memory assumption rather than random-walk. These conditions result in the following hidden state model.

$$\begin{pmatrix} x_{1(t)} \\ x_{2(t)} \\ x_{3(t)} \end{pmatrix} = \begin{pmatrix} K_1 & 0 & 0 \\ (1 - K_2)\frac{\bar{x}_2}{\bar{x}_1} & K_2 & 0 \\ 0 & 0 & K_3 \end{pmatrix} \begin{pmatrix} x_{1(t-1)} \\ x_{2(t-1)} \\ x_{3(t-1)} \end{pmatrix} + w_X(t) \quad (4.6)$$

where K_1 , K_2 and K_3 are constants between 0 and 1 which pertain to the degree of memory of each variable (1 denotes a random walk state, and 0 denotes a static state). Because x_1 is the random-walk variable, K_1 has a value of 1 by definition. Of particular interest is state x_2 , where \bar{x}_2/\bar{x}_1 denotes the desired ratio between x_2 and x_1 to be maintained by the control system. In this way, for a given value of x_1 , the controller will react in a way to preserve the ratio between x_1 and x_2 . K_2 is related to how well the controller adjusts x_2 to maintain its desired proportion to x_1 ($K_2 = 0$ for perfect tracking, $K \approx 1$ for laggy or imprecise control).

While can also use Bayesian learning to estimate A , but this often results in a process being centered around a mean value. Specifying a model similar to Equation (4.6) allows the mean to move; using this type of model allows the dynamic state estimation to be more consistent with traditional data reconciliation, which estimates states with no prior knowledge of the process mean. When combined with the observation model in Equation (4.1), a state space model is created which has the dynamic Bayesian network (DBN) representation shown in Figure 4.4. Note that DBNs are recursive representations of static Bayesian networks; hidden states at time T_k are considered parents of the hidden states at time T_{k+1} .

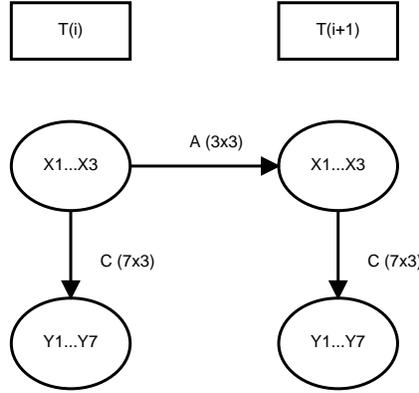


Figure 4.4: Dynamic Bayesian Network without Gross Errors

4.2.3 Augmented Gross Error States

In order to obtain dynamic estimates of gross errors (or instrument bias), they must be included in the state space model as augmented states. The gross error terms themselves are assumed to follow a random walk model shown in Equation (4.7)

$$\beta_{(t)} = I\beta_{(t-1)} + w_{\beta(t)} \quad (4.7)$$

When gross errors are detected, the corresponding gross error states are included in the process model given by Equation (4.6) as augmented states, resulting in Equation (4.8) below.

$$\begin{aligned} \begin{pmatrix} X(t) \\ \beta(t) \end{pmatrix} &= \begin{pmatrix} A & \mathbf{0} \\ \mathbf{0} & I_{\beta(a)} \end{pmatrix} \begin{pmatrix} X(t-1) \\ \beta_{(a,t-1)} \end{pmatrix} + \begin{pmatrix} w_{X(t)} \\ w_{\beta(t)} \end{pmatrix} \\ Y(t) &= \begin{pmatrix} C & I_{Y(a)} \end{pmatrix} \begin{pmatrix} X(t) \\ \beta_{(a,t)} \end{pmatrix} + \beta_{(s)} + v_{Y(t)} \end{aligned} \quad (4.8)$$

where a is a set of indices pertaining *active* gross error states, s is a set of *static* gross error states, $I_{\beta(a)}$ is an identity matrix with the same length of a , and $I_{Y(a)}$ is the a^{th} columns of an identity matrix of the same length as Y . In this way, if instruments 3 and 6 were found to have active gross errors, $a = (3, 6)$, $s = (1, 2, 4, 5, 7)$, $I_{\beta(a)}$ would be a 2×2 identity matrix, and $I_{Y(a)}$ would be the 3rd and 6th columns of I_Y (I_Y is an identity matrix having the same length as Y). For notational convenience, we will use $A_{(a)}$ and $C_{(a)}$ to represent augmented matrices that are specified by the indicated active gross errors a .

$$\begin{aligned} \begin{pmatrix} X(t) \\ \beta(t) \end{pmatrix} &= A_{(a)} \begin{pmatrix} X(t-1) \\ \beta_{(a,t-1)} \end{pmatrix} + \begin{pmatrix} w_{X(t)} \\ w_{\beta(t)} \end{pmatrix} \\ Y(t) &= C_{(a)} \begin{pmatrix} X(t) \\ \beta_{(a,t)} \end{pmatrix} + \beta_{(s)} + v_{Y(t)} \end{aligned} \quad (4.9)$$

This can also be converted to a DBN as shown in Figure 4.5, which displays the case for all bias states being active. Note that in this figure, (a) pertains to the active gross errors while (s) pertains to the static gross errors, or gross errors that are not active. The node $B(s)$ is a static node and is thus noise free as it acts as a static parameter.

4.2.4 Observability

Augmented gross error states are added when residual error tests indicate that unknown gross errors are present, but these states are removed when tests indicate that the gross errors have been accounted for. It is important that these augmented gross error states only be included when necessary; including all gross error models into the augmented state will cause the model to become unobservable. This can be shown by applying an observability criterion; [4] contains a number of different observability criteria on page 171 of his book, the most convenient form for this application is shown in Equation (4.10). A linear system is observable if

$$O = \begin{pmatrix} A - \lambda I \\ C \end{pmatrix} \quad (4.10)$$

has full column rank for every eigenvalue λ , where O is a simplified observability matrix. When applying a complete set of hidden gross error states, the simplified observability matrix becomes

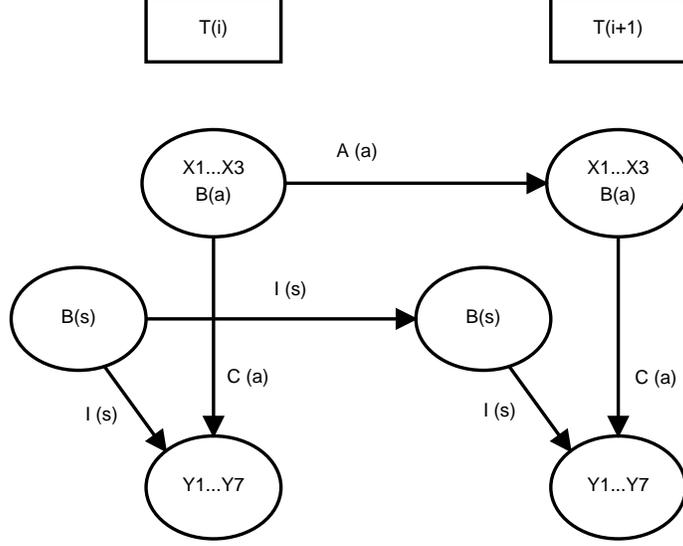


Figure 4.5: Dynamic Bayesian Network with Gross Errors

$$O = \begin{pmatrix} A - \lambda I_X & \mathbf{0} \\ \mathbf{0} & I_Y - \lambda I_Y \\ C & I_Y \end{pmatrix} \quad (4.11)$$

where I_X is an identity matrix with the same length as X , and I_Y is an identity matrix with the same length as Y . From this matrix, one can see that I_Y forms a basis of C , and that $I_Y - \lambda I_Y$ forms a basis for 0 at all values of λ , the only basis in the left columns that is not formed by the right is $A - \lambda I_X$. Thus O will have full column rank only if $A - \lambda I_X$ has full column rank for every eigenvalue λ ; this is by definition impossible because the values of λ for A are the ones that cause $A - \lambda I_X$ to have a zero determinant and hence, an incomplete rank.

This result of non-observability when introducing a full set of gross error states is the intuitive consequence of trying track $p + n$ independent states with only p instruments. Any set of gross error states that allow $I_{Y^{(a)}}$ to form a basis for C is thus a *sufficient condition* to cause the linear system to be unobservable.

While analyzing the observability matrix in Equation (4.11) shows that complete set of gross error states is a sufficient condition to cause unobservability, it is not a necessary condition. If there is insufficient instrument redundancy, there could be a single gross error state that could cause the system to be unobservable. This causes gross errors for that particular instrument to be unobservable even if all other instruments were functioning properly. Thus, it is important to ensure that every gross error is observable. This can be done by setting the bias index a to every possible *single instrument* index, and then calculating the simplified observability matrices according to Equation (4.12)

$$O = \begin{pmatrix} A^{(a)} - \lambda I \\ C^{(a)} \end{pmatrix} \quad (4.12)$$

If instrument redundancy is greater than minimal, the model will be observable for many different multi-instrument combinations of a ; this method is able to simultaneously detect any combination of instrument gross errors a for which the model is observable.

When augmented gross error states are activated as hidden states, they will converge to a value that minimizes the residual error. It is also helpful to increase the corresponding instrument noise variance elements of R_Y to $R_{\tilde{Y}}$ when a gross error has been detected. $R_{\tilde{Y}}$ is the instrument noise covariance of unreliable instruments, which can be obtained by scaling the original value of R_Y by an gross error penalization constant k_{err} .

$$R_{\tilde{Y}} = k_{err} R_Y \quad (4.13)$$

Using $R_{\tilde{Y}}$ instead of R_Y will reduce the effects of contaminated instruments on the estimation of the hidden process state X .

4.2.5 Model Estimation

In the previous chapter, R_Y was only estimated for the static case but for the dynamic case, estimates of R_Y are still valid because there is no change in the observation variance (instrument noise variance) of Y given X . Thus, the only additional parameter to be estimated is Q_X . After parameters have been estimated, performing Step (1) is a simple application of the Kalman filter.

4.3 Step (2) Residual Error Distance

4.3.1 Residual Error

For detecting gross errors, conventional methods rely mostly on calculating the statistical distance between the instrument readings and the expected readings given the reconciled state estimate. A similar approach is used in Step (2) for the dynamic case, however, a Kalman filter is used to determine the hidden reference state instead of data reconciliation. For each state estimate, the expected process measurements and their uncertainty can be obtained by posterior state distributions. The difference between the measurements and the estimated reference value is the residual error Δ_r , which is defined according to Equation (4.15).

$$\Delta_{r(t)} = Y_{(t)} - C_{(a)} \hat{X}_{(a,t)} - \beta_{(s)} \quad (4.14)$$

For simplicity of derivation and notation, we treat observations Y as if biases were already removed, and hidden states X as if active biases are part of the state.

$$\Delta_{r(t)} = Y_{(t)} - C_{(a)} \hat{X}_{(t)} \quad (4.15)$$

When calculating the residual error, one cannot use the real values of $X_{(t)}$ directly, but one can use the posterior Kalman filter estimate $\hat{X}_{(t)}$ which contains estimation noise. However, noise in $\hat{X}_{(t)}$ and $Y_{(t)}$ are dependent, and must be broken down into independent components.

$$\hat{X}_{(t-1)} = X_{(t-1)} + \delta_{(t-1)} \quad (4.16)$$

$$\hat{X}_{(t)} = X_{(t)} + \delta_{(t)} \quad (4.17)$$

$$X_{(t)} = A_{(a)}X_{(t-1)} + w_{X(t)} \quad (4.18)$$

$$Y_{(t)} = C_{(a)}X_{(t)} + v_{Y(t)} \quad (4.19)$$

From these four equations, we can see that the previous state estimate $\hat{X}_{(t-1)}$ is contaminated by the previous state estimate error $\delta_{(t-1)}$, the current state estimate $\hat{X}_{(t)}$ is contaminated by the current state estimate error $\delta_{(t)}$. While $\delta_{(t)}$ and $v_{Y(t)}$ are dependent, $\delta_{(t-1)}$, $w_{X(t)}$ and $v_{Y(t)}$ are all independent; thus, the task is to break the covariance of $\hat{\Delta}_{r(t)}$ into terms that are contaminated by independent noise.

From the Kalman filter, the current state estimate is calculated from the current instrument observations and the previous state estimate as follows:

$$\hat{X}_{(t)} = A\hat{X}_{(t-1)} + K(Y_{(t)} - CA\hat{X}_{(t-1)})$$

Substituting in Equations (4.16) and (4.19) yields,

$$\hat{X}_{(t)} = A(X_{(t-1)} + \delta_{(t-1)}) + K[(CX_{(t)} + w_{X(t)}) - CA(X_{(t-1)} + \delta_{(t-1)})]$$

Further substitution of Equation (4.17) for $\hat{X}_{(t)}$ (and canceling out $CA\hat{X}_{(t-1)}$), and the substitution of Equation (4.18) for $X_{(t)}$ on the right yields

$$AX_{(t-1)} + w_{X(t)} + \delta_{(t)} = AX_{(t-1)} + Kv_{Y(t)} + KCw_{X(t)} + (I - KC)A\delta_{(t-1)}$$

which can be solved for $\delta_{(t)}$, breaking it down into its independent components.

$$\delta_{(t)} = Kv_{Y(t)} + (I - KC)(A\delta_{(t-1)} - w_{X(t)}) \quad (4.20)$$

With $\delta_{(t)}$ being broken down into independent noise components, we can now return to Equation (4.15). By canceling out $CX_{(t)}$ from the right hand side of Equation (4.15), the expression for $\hat{\Delta}_r$ is reduced to

$$\hat{\Delta}_{r(t)} = v_{Y(t)} - C\delta_{(t)} \quad (4.21)$$

Since $v_{Y(t)}$ and $\delta_{(t)}$ are dependent, the expression for the residual error must be broken down into independent components by substituting Equation (4.20) for $\delta_{(t)}$ and collecting terms:

$$\hat{\Delta}_{r(t)} = (I - CK)v_{Y(t)} + C(I - KC)(w_{X(t)} - A\delta_{(t-1)}) \quad (4.22)$$

All noise terms are independent, so the covariance is a linear combination of these terms. The covariance of $v_{Y(t)}$ and $w_{X(t)}$ were already given as R_Y and Q_X respectively, while the covariance of $\delta_{(t-1)}$ is actually $P_{X(t-1)}$ which is the posterior state estimate covariance given by the Kalman filter for the previous step. Also, the effects of active gross error states are explicitly taken into account.

$$\begin{aligned} \Sigma_{r(t)} = & (I - C_{(a)}K_{(a)})(R_Y)(I - C_{(a)}K_{(a)})^T + \\ & C_{(a)}(I - K_{(a)}C_{(a)})(A_{(a)}P_{X(\bar{a},t-1)}A_{(a)}^T + Q_{X(\bar{a})})(I - K_{(a)}C_{(a)})^T C_{(a)}^T \end{aligned} \quad (4.23)$$

where $P_{X(\bar{a},t-1)}$ is the posterior state covariance given by the Kalman filter with penalization scaling removed from the gross error states, and $K_{(a)}$ is the optimal Kalman gain. When determining the value of Σ_r , one should recall that the posterior covariance $P_{X(\bar{a},t-1)}$ is updated at each time step and the Kalman gain $K_{(a)}$ changes according to the active gross error states.

A proper understanding Equation (4.23) requires a book-keeping exercise. When observations are penalized, the corresponding value for R_Y is replaced with a fictionally large value $R_{\tilde{Y}}$. This is done to prevent unreliable observations from corrupting the state estimate. However, these corrupted observations must be used to estimate the corresponding gross errors; in order for the penalized observations to have an effect on gross error state estimates, the gross error states must also be penalized themselves. This results in $Q_{X(a)}$ and $P_{X(a,t-1)}$ also having fictionally large variances.

Since residual errors are used to detect systematic error, the residuals should be standardized in terms of the actual random errors and state uncertainty. Thus for Equation (4.23) $Q_{X(\bar{a})}$ should have entries of 0 for all columns and rows that pertain to $\beta_{(a)}$, while the covariance of X should remain unchanged. $P_{X(\bar{a},t-1)}$ however reflects state uncertainty. For $\beta_{(a)}$, entries in $P_{X(a,t-1)}$ are fictionally large, so they must be divided by k_{err} so that only the actual uncertainties in $\beta_{(a)}$ are taken into account. Once again, entries of $P_{X(a,t-1)}$ that pertain to X should remain unchanged when calculating $P_{X(\bar{a},t-1)}$.

4.3.2 Statistical Distance

The residual error can be standardized by $\Sigma_{r(t)}$ to determine the statistical distance for each individual observation. A popular distance to use is the Mahalanobis distance. For the univariate case, the Mahalanobis is simply the difference between the observation and the expected value μ divided by the standard deviation σ .

$$d_M = z = \sigma^{-1}(y - \mu) \quad (4.24)$$

Because there is only one component, the z value is equal to the Mahalanobis distance. For the multivariate case, the Mahalanobis distance is defined as

$$D_M = \sqrt{\Delta_r^T \Sigma_r^{-1} \Delta_r} \quad (4.25)$$

This distance however, can be broken down component-wise to calculate the statistical distance of each element of $\Delta_{r(t)}$

$$Z_{(t)} = \Sigma_{r(a,t)}^{-1/2} \hat{\Delta}_{r(t)} \quad (4.26)$$

where Z is a vector of statistical distances, having a covariance matrix of I , and where $\Sigma_{r(a,t)}^{-1/2}$ is calculated by spectral decomposition since $\Sigma_{r(a,t)}$ is positive definite.

4.4 Step (3) Systematic Error Updating, Filtering Z-Values

When residual errors are due to random noise, Z vector has a covariance matrix of I due to standardization. However, gross errors will not be detectable unless their statistical distance is significant compared to random error. Nevertheless, we can detect small systematic offsets in the standardized residual Z by applying another Kalman filter based on a random walk model. Such filtering allows us to detect smaller gross errors; however, there is a tradeoff between the precision and the speed at which this error can be detected. The state space model is shown in Equation (4.27).

$$\begin{aligned} Z_{u(t)} &= Z_{u(t-1)} + w_{z(t)} \\ Z_{(t)} &= IZ_{u(t)} + v_{z(t)} \end{aligned} \quad (4.27)$$

where $w_{z(t)}$ represents disturbances in the standardized systematic gross error $Z_{u(t)}$, and $v_{z(t)}$ is the random noise in the values of $Z_{(t)}$ caused by instrument observations. These noise terms are distributed as follows:

$$w_{z(t)} \sim N(0, Q_Z) \quad (4.28)$$

$$v_{z(t)} \sim N(0, R_Z) \quad (4.29)$$

$$R_Z = I \quad (4.30)$$

where Q_Z is a diagonal covariance matrix with entries less than 1 and represents noise attributed to gross error changes. In reality Q_Z is nearly zero except during times where gross errors change significantly, but specifying small average increases precision by filtering out noise in Z . When applying the a Kalman filter based on Equation (4.27), Q_Z is a tuning parameter; smaller values allow for greater precision in detecting gross errors, but could potentially slow down detection. Obtaining optimal values of Q_Z based on specifications is discussed later.

In Equation (4.29), the observation noise is defined by the covariance matrix $R_Z = I$. This is due to the fact that when there is no systematic variation, standardization causes $cov(Z_{(t)}) = I$. The Kalman filter based on Equation (4.27) returns the state estimates $\hat{Z}_{(t)}$ and the diagonal posterior covariance matrix $P_{Z(t)}$.

4.5 Step (4) Reliability Audit through Hypothesis Testing

4.5.1 Recursive Hypothesis Testing

The Kalman filter in step (3) filters values of Z in order to obtain a corresponding filtered posterior estimate \hat{Z} and a posterior covariance estimate P_Z . \hat{Z} is standardized by P_Z according to

$$\hat{Z}_{S(t)} = P_{Z(t)}^{-1/2} \hat{Z}_{(t)} \quad (4.31)$$

Since the states are independent, P_Z is diagonal, and the univariate procedure can be equivalently applied. Let $p_{z(t)}$ be the square-root of each diagonal entry of $P_{Z(t)}$, then

$$\hat{z}_{S(t)} = \frac{\hat{z}_{(t)}}{p_{z(t)}} \quad (4.32)$$

Normal percentile values (P-values) can be calculated from standardized residual errors by using the error function.

$$P_{value} = 1 - erf\left(\frac{|\hat{z}_{(t)}|}{p_{z(t)}\sqrt{2}}\right) \quad (4.33)$$

A simple method to perform instrument reliability auditing is to choose a minimum P value of α that would trigger the flagging of a corrupted instrument. This confidence level α (the probability of a false-positive result) would correspond to a particular absolute Z-value that would serve as a cutoff threshold value.

A more rigorous approach would be to employ a discrete-valued DBN which acts as a Kalman filter for discrete states, allowing the management of probabilistic evidence from P-values over time and reducing the frequency of false alarms. Traditional statistical hypothesis testing would require that we treat one state as being the null hypothesis; for this case, the null hypothesis would state that there are no unidentified gross errors. However, the DBN does not make the distinction between null and alternate hypothesis, and instead, allows us to consider previous probabilistic evidence as priors. Unlike the traditional method which only considers evidence toward the alternate hypothesis (that being “there is an unknown gross error”), the Bayesian method allows us to simultaneously consider evidence toward both the existence and the nonexistence of unknown gross errors.

The discrete valued DBN uses a direct application of Bayes’ Theorem

$$P(h|e) = \frac{P(e|h)P(h)}{P(e)} \quad (4.34)$$

where $P(h|e)$ is the probability of the hypothesis given the evidence, or the *posterior probability*, $P(h)$ is the prior probability of the hypothesis or the *prior probability*, $P(e|h)$ is the probability of the evidence given the hypothesis, or the *likelihood*, and $P(e)$ is the total probability of the evidence, which serves as a *normalization* factor. The corresponding Bayesian Network is given for the problem in Figure 4.6.

Because every step has a simple 3-node chain inference, updating $P(B = T)$ with evidence is a simple application of Bayes’s rule. An example of this application can be found in [14]. Once observing positive or negative test results for observed gross error B_{obs} we can use the Bayesian network to update our belief of condition $B = T$. Updating our believe of an unaccounted gross error ($B = T$) requires knowledge of the conditional probability for these related events which is often shown in a *conditional probability table* (CPT). This is related to the state transition matrix for a state-space model; the CPT is thus labeled as A and is calculated as follows:

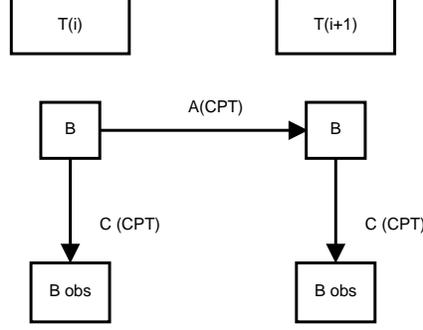


Figure 4.6: Dynamic Bayesian Network without Bias

$$A = \begin{pmatrix} P(B_{(t+1)} = T|B_{(t)} = T) & P(B_{(t+1)} = T|B_{(t)} = F) \\ P(B_{(t+1)} = F|B_{(t)} = T) & P(B_{(t+1)} = F|B_{(t)} = F) \end{pmatrix} \quad (4.35)$$

Values for A are often treated as tuning parameters which dictate how much evidence is required in order to change our belief in the discrete gross error state. A second CPT, labeled C can be interpreted as the reliability of the test in performing a statistical inference.

$$C = \begin{pmatrix} P(B_{obs} = T|B = T) & P(B_{obs} = T|B = F) \\ P(B_{obs} = F|B = T) & P(B_{obs} = F|B = F) \end{pmatrix} \quad (4.36)$$

If one assumes that the prior gross error distribution is normal, C matrix entries have a visual representation shown in Figure 4.7. The probability of a false positive is the area under ‘filtered Z posterior’ curve that lies outside a given Z value; by contrast, the probability of a false negative is the area under the ‘standardized gross error prior’ curve that lies inside the same Z value. The optimal threshold value is the \hat{Z} value at which false positive and false negative probabilities are equal.

4.5.2 C Matrix Based on Fixed P-Values

When using fixed P-Values, one applies a hypothesis test, and the observed result (T or F) is then used for the discrete Bayesian network. The probability of false positives $P(B_{obs} = T|B = F)$ is set by the confidence level α at which we reject the null hypothesis that unaccounted gross error is zero. In fact,

$$\alpha \equiv P(B_{obs} = T|B = F) \quad (4.37)$$

nevertheless, determining the probability of false negatives requires the standardized prior distribution of the possible values of gross error. The prior distribution of the gross error β_0 is given by

$$\beta_0 \sim N(0, \Sigma_{\beta_0}) \quad (4.38)$$

where Σ_{β_0} is a diagonal covariance matrix for the prior gross error distribution. The prior gross error distribution can be standardized in the same way that Y

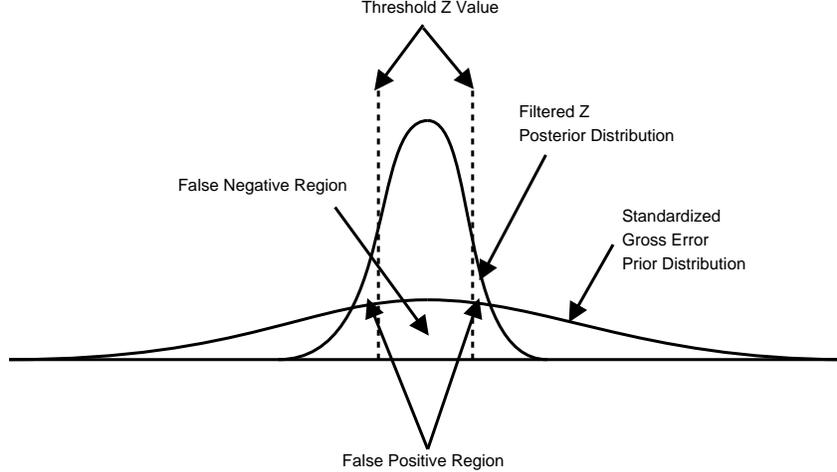


Figure 4.7: Visual Representation of Gross Error Hypothesis Test

values were standardized. Firstly, one must take the diagonal matrix Σ_{β_0} and convert it to a vector of standard deviations $\vec{\sigma}_{\beta_0}$. Applying Equation (4.39) standardizes these values into the Z-value space as was done to the residual errors in Step (2).

$$\vec{\sigma}_{\beta_0 S} = \Sigma_{r(\infty)}^{-1/2} \vec{\sigma}_{\beta_0} \quad (4.39)$$

where $\Sigma_{r(\infty)}$ is the residual error covariance matrix at steady-state with no active bias. $\Sigma_{r(\infty)}$ is used instead of $\Sigma_{r(t)}$ so that the statistical distance of $\vec{\sigma}_{\beta_0}$ from 0 is conserved for static gross error conditions, preventing state uncertainty from relaxing confidence requirements for gross error detection. Using the standardized values $\sigma_{\beta_0 S}$, entries of the C matrix can be calculated according to Equations (4.40) to (4.43). The optimal cutoff values \hat{Z}_O would be the threshold value, which causes $P(B_{obs} = T|B = F) = P(B_{obs} = F|B = T)$. Equivalently, the threshold value would cause the C matrix to be symmetric.

$$P(B_{obs} = T|B = F) = 1 - erf\left(\frac{|\hat{z}_O|}{p_{z(\infty)}\sqrt{2}}\right) \quad (4.40)$$

$$P(B_{obs} = T|B = T) = 1 - P(B_{obs} = T|B = F) \quad (4.41)$$

$$P(B_{obs} = F|B = T) = erf\left(\frac{|\hat{z}_O|}{\sigma_{\beta_0 S}\sqrt{2}}\right) \quad (4.42)$$

$$P(B_{obs} = F|B = F) = 1 - P(B_{obs} = F|B = T) \quad (4.43)$$

where $p_{z(\infty)}$ is the the steady-state value of the posterior standard deviation used in Equations (4.32) and (4.32), obtained using the diagonal matrix $P_{Z(\infty)}$ given by Step (3). Recall that the posterior covariance $P_{Z(t)}$ changes during the Kalman filter startup period; however, it eventually reaches a steady-state value. Thus, it is best to calculate the optimal value \hat{Z}_O using the steady-state

values of $P_{Z(\infty)}$. One should also keep in mind that since there is a hypothesis test for each instrument, a discrete DBN must be made for each individual instrument, each with its own C matrix.

4.5.3 C Matrix Based on Adaptive P-Values

While using fixed P-Values can slightly reduce computational demand, using adaptive P-Values is more rigorous and can significantly increase the information that is introduced toward the discrete DBN. The difference in methodology is that for every step, a new C matrix is calculated according to Equations (4.44) to (4.47). Thus, instead of using optimal values of \hat{Z}_O and steady state values of $p_{z(\infty)}$ for all cases, the posterior filtered values \hat{Z} and $p_{z(t)}$ are directly applied for every step.

$$P(B_{obs} = T|B = F) = 1 - erf\left(\frac{|\hat{z}(t)|}{p_{z(t)}\sqrt{2}}\right) \quad (4.44)$$

$$P(B_{obs} = T|B = T) = 1 - P(B_{obs} = T|B = F) \quad (4.45)$$

$$P(B_{obs} = F|B = T) = erf\left(\frac{|\hat{z}(t)|}{\sigma_{\beta_0 S}\sqrt{2}}\right) \quad (4.46)$$

$$P(B_{obs} = F|B = F) = 1 - P(B_{obs} = F|B = T) \quad (4.47)$$

In order to determine whether to observe $P_{obs} = T$ or $P_{obs} = F$, one must calculate the probabilities in Equations (4.44) and (4.46). If the probability of a false positive (Equation (4.44)) is smaller than that of a false negative (Equation (4.46)), observing a ‘positive’ is more informative; otherwise, it is best to observe a ‘negative’. The advantage to this approach is that the strength of a positive or negative result is adjusted according to the filtered Z values.

4.6 Tuning Steps (1) and (3)

4.6.1 Tuning Step (3)

While the use of a Kalman filter for the standardized residual errors Z has been discussed in Step (3), determining a suitable value for the model variance Q_Z is not a trivial procedure. There are three parameters that can affect the resulting Z-value filter: the gross error threshold value (determined by the minimum detectable gross error β_{min}), the fractional allowance of false positives α , and the prior standard deviation of gross errors $\vec{\sigma}_{B0}$. However, because this method is constrained to have a predetermined threshold value as the point where $P(B_{obs} = T|B = F) = P(B_{obs} = F|B = T)$, specifying any two of these values would define a third. This can be shown mathematically. The value for α is implicitly specified by the equality constraint in Equation

(4.48) resulting in explicit specifications in Equations (4.49) and (4.50).

$$P(B_{obs} = F|B = T) = P(B_{obs} = T|B = F) \quad (4.48)$$

$$\alpha = P(B_{obs} = T|B = F) \quad (4.49)$$

$$\alpha = P(B_{obs} = F|B = T) \quad (4.50)$$

Let the vector of threshold Z values be defined as

$$Z_{Sp} = \Sigma_r^{-1/2} \beta_{min} \quad (4.51)$$

The probability terms in Equation (4.48) have already been defined by Equations (4.44) and (4.46); by making the appropriate substitutions,

$$1 - erf\left(\frac{|z_{Sp}|}{p_{z(\infty)}\sqrt{2}}\right) = erf\left(\frac{|z_{Sp}|}{\sigma_{\beta_0 S}\sqrt{2}}\right) \quad (4.52)$$

where z_{Sp} is the scalar value of the vector Z_{Sp} for any given element. The specifiers $\sigma_{\beta_0 S}$ and z_{Sp} are related to α by using Equations (4.40) and (4.42) to make appropriate substitutions for $P(B_{obs} = T|B = F)$ and $P(B_{obs} = F|B = T)$ into Equations (4.49) and (4.50).

$$\alpha = 1 - erf\left(\frac{|z_{Sp}|}{p_{z(\infty)}\sqrt{2}}\right) \quad (4.53)$$

$$\alpha = erf\left(\frac{|z_{Sp}|}{\sigma_{\beta_0 S}\sqrt{2}}\right) \quad (4.54)$$

These equations can be rearranged to solve for z_{Sp}

$$|z_{Sp}| = \sqrt{2}p_{z(\infty)}erf^{-1}(1 - \alpha) \quad (4.55)$$

$$|z_{Sp}| = \sqrt{2}\sigma_{\beta_0 S}erf^{-1}(\alpha) \quad (4.56)$$

The desired posterior standard deviations $p_{z(\infty)}$ define the desired posterior covariance $P_{Z(\infty)}$ for the end of Step (3). Values of $p_{z(\infty)}$ are obtained by making appropriate substitutions and can be defined in the three following ways

$$p_{z(\infty)} = \frac{|z_{Sp}|}{\sqrt{2}erf^{-1}\left(1 - erf\left(\frac{|z_{Sp}|}{\sigma_{\beta_0 S}\sqrt{2}}\right)\right)} \quad (4.57)$$

$$p_{z(\infty)} = \frac{z_{Sp}}{\sqrt{2}erf^{-1}(1 - \alpha)} \quad (4.58)$$

$$p_{z(\infty)} = \sigma_{\beta_0 S} \frac{erf^{-1}(\alpha)}{erf^{-1}(1 - \alpha)} \quad (4.59)$$

Equation (4.57) was obtained by solving Equation (4.50) for $p_{z(\infty)}$, Equation (4.58) was obtained by solving Equation (4.55) for $p_{z(\infty)}$, and Equation (4.59)

was obtained by setting Equations (4.55) and (4.56) equal to each other and solving for $p_{z(\infty)}$. Equations (4.57), (4.58) and (4.59) specify the *posterior* Z variance $P_{Z(\infty)}$, however, the Kalman filter requires that we specify the *model* covariance Q_Z . Because the posterior covariance changes as the Kalman filter runs through iterations, one cannot choose a single value for Q_Z that would yield the design criteria at all times. However, the posterior Z variance does converge to a steady-state value.

In general, the posterior covariance of the Kalman filter P_t is related to the prior covariance P_{t-1} according to Equation (4.60).

$$P_{(t)} = (I - (AP_{(t-1)}A^T + Q)C^T(C(AP_{(t-1)}A^T + Q)C^T + R)^{-1}C) \times (AP_{(t-1)}A^T + Q) \quad (4.60)$$

where A is the state transition matrix, C is the observation matrix, Q is the process noise covariance and R is the observation noise covariance. In the Z -value Kalman filter, A and C are both equal to 1. Furthermore, the observation noise variance R_Z is replaced with $R_Z = I$. At steady-state, $P_{(t)} = P_{(t-1)} = P_{Z(\infty)}$; by making substitutions for the standardized residual model and solving the equation component-wise. Let $p_{z(\infty)}$, q_z , and r_z represent the square-root of any given diagonal component of $P_{Z(\infty)}$, Q_Z , and R_Z respectively.

$$p_{z(\infty)}^2 = \left(1 - \frac{p_{z(\infty)}^2 + q_z^2}{p_{z(\infty)}^2 + q_z^2 + 1}\right) (p_{z(\infty)}^2 + q_z^2) \quad (4.61)$$

Solving for q_z^2 yields

$$q_z^2 = \frac{p_{z(\infty)}^4}{1 - p_{z(\infty)}^2} \quad (4.62)$$

$$(4.63)$$

Note that each element of $P_{Z(\infty)}$ must be less than 1; this is the maximum value of uncertainty $P_{Z(\infty)}$ that can be obtained from a random walk model with $R_Z = I$. This is not a major issue; it simply means that there is a limit to the posterior estimate uncertainty that the model in Equation (4.27) can have. If the specifications of α , z_{Sp} and $\sigma_{\beta_0 S}$ result in values of $P_{Z(\infty)}$ that is greater than the limit, then these specifications are too lenient to result in a model that is consistent with Equation (4.27). Specification of Q_Z can also be done as a matrix operation.

$$Q_Z = (P_{Z(\infty)} - I)^{-1} P_{Z(\infty)}^2 \quad (4.64)$$

4.6.2 Tuning Step (1)

Since Q_Z determines the speed at which gross errors can be detected, it can also be used to specify the speed at which gross errors are corrected in Step (1). Recall that we have not yet specified a value for Q_β which is the covariance for w_β in Equation (4.7). In order to determine Q_β from Q_Z two facts about

residual errors must be recalled: when biases are static, (1) residual errors $\Delta_{r(t)}$ have covariance Σ_r and (2) standardized residual errors $Z_{(t)}$ have covariance $R_Z = I$. Active gross errors must have a proportional effect on both the residual error and the standardized residual error.

$$\Sigma_{r(\infty)}^{-1} Q_\beta = (I)^{-1} Q_Z \quad (4.65)$$

Because Q_β is a diagonal matrix, a matrix $\Psi_{r(\infty)}$ containing the diagonal elements of $\Sigma_{r(\infty)}$ must be used. Furthermore, because instruments that are associated with active gross errors are penalized by k_{err} , the Q_β used in the Kalman filter must be multiplied by the same constant. This way, corrupted instruments will have reduced effect on process state estimates, but will not have a reduced effect on gross error state estimates.

$$Q_\beta = k_{err} \Psi_{r(\infty)} Q_Z \quad (4.66)$$

However, appropriate values of Q_β are often quite small due to the fact that gross errors are not common. In reality, while average values of Q_β are small, local values can become quite large during the event of a significant and sudden change in gross error. By using information in the filtered standardized residual errors \hat{Z} , the parameter Q_β can be varied so that it can quickly track large and sudden gross errors.

Before a gross error can be identified, the values of \hat{Z} must be allowed to diverge. Because the estimates \hat{Z} track disturbances from $w_z(t)$, for squared values of \hat{Z} that are significantly larger than the diagonal elements of $P_{Z[i,i]}$, the local value of Q_β can be approximated as

$$\hat{Q}_{\beta[i,i]}(t) = k_p k_{err} \Psi_{r[i,i](\infty)} (\hat{Z}_{[i]}^2(t) - P_{Z[i,i]}(t)) \quad (4.67)$$

where k_p is a proportionality constant. The relationship is approximately proportional due to the fact that it takes a certain amount of time to detect a major gross error. In the time it takes to detect a gross error, the values of \hat{Z} are allowed to diverge, and will do so at a rate that is proportional to the local value of $Q_\beta^{1/2}$ (which is proportional to the gross error magnitude). In practice, due to the nature of P-values and the DBN in Step (4), when gross errors are larger than $3Q_\beta^{1/2}$, increasing the size of the gross errors will have a marginal effect on the time required to detect the gross error. Nevertheless, the adjustments for the local values of Q_β tend to be conservative due to the fact that the values of \hat{Z} have slightly less time to propagate before gross error compensation takes place. Adjusting the value of Q_β is only recommendable when the difference in Equation (4.67) is positive, resulting in the following heuristic:

$$\begin{aligned} & \text{If } \hat{Z}_{[i]}^2(t) > P_{Z[i,i]}(t) \\ \hat{Q}_{Z[i,i]} &= Q_{Z[i,i]} + k_p (\hat{Z}_{[i]}^2(t) - P_{Z[i,i]}(t)) \end{aligned} \quad (4.68)$$

$$\begin{aligned} & \text{Otherwise} \\ \hat{Q}_{Z[i,i]} &= Q_{Z[i,i]} \end{aligned} \quad (4.69)$$

Table 4.1: Variable Definitions

Symbol	Mean	S.D.	Meaning
x_1	1000	300	Real Oil Sand Flow
x_2	500	155.2	Real Water Flow
x_3	0	120	Real Hopper Level Change
y_1	0	200	WENCO Database Value
y_2	0	90	First Weightometer Readings
y_3	0	130	Hopper Level Readings
y_4	0	80	Second Weightometer Readings
y_5	0	60	Slurry Flow Meter Readings
y_6	0	0.2	Density Meter Readings
y_7	0	50	Water Flow Meter Readings

With this adjusted value for Q_Z , the local estimate \hat{Q}_β can be specified

$$\hat{Q}_\beta = k_{err} \Psi_{r(\infty)} \hat{Q}_Z \quad (4.70)$$

In practice, k_p is a tuning parameter that can be obtained using a given value for the switching probability in Step (4). Once properly tuned, the time required to correct a gross error is much less dependent on the gross error magnitude. Furthermore, once gross error estimation process begins, updates in gross error estimates will reduce the values of $\hat{Z}_{[i]}^2$, reducing the local estimates for \hat{Q}_β . This allows for more precise estimation of β later on, once the major corrections have taken place.

Using Equation (4.66) to define \hat{Q}_β is sufficient for gross errors that tend to be small or constant in magnitude. However, for systems with gross errors that have unpredictable magnitudes exceeding $Q_\beta^{1/2}$, using Equation (4.70) is more recommendable as it allows for appropriate adjustment in the local values of \hat{Q}_β .

4.7 Simulation of Dynamic System

A hypothetical operating point was considered for this system with means and variances as shown in Table 4.7.

A simulation of the system was performed by sequential generation of Gaussian random variables n and applying the parameters in Table 4.7 to Equations

(4.5) and (4.1), resulting in Equations (4.71) to (4.73); this generated values of ideal process measurements.

$$w_X = \begin{pmatrix} 300 & 0 & 0 \\ 150 & 40 & 0 \\ 0 & 0 & 120 \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \quad (4.71)$$

$$\begin{pmatrix} x_{1(t+1)} \\ x_{2(t+1)} \\ x_{3(t+1)} \end{pmatrix} = (0.7)I \begin{pmatrix} x_{1(t)} \\ x_{2(t)} \\ x_{3(t)} \end{pmatrix} + (0.3) \begin{pmatrix} 1000 \\ 500 \\ 0 \end{pmatrix} + w_X \quad (4.72)$$

$$\begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \\ y_5(t) \\ y_6(t) \\ y_7(t) \end{pmatrix} = \begin{pmatrix} x_{1(t)} + x_{3(t)} \\ x_{2(t)} + x_{3(t)} \\ x_{3(t)} \\ x_{1(t)} \\ \frac{x_{1(t)}}{2.1} + \frac{x_{2(t)}}{1.0} \\ \frac{2.1x_{1(t)} + 1.0x_{2(t)}}{x_{1(t)} + x_{2(t)}} \\ 1.0^{-1}x_2 \end{pmatrix} + \begin{pmatrix} 200n_4 \\ 90n_5 \\ 130n_6 \\ 80n_7 \\ 60n_8 \\ 0.2n_9 \\ 50n_{10} \end{pmatrix} \quad (4.73)$$

After this, a constant bias with a magnitude of -2 standard deviations was added to instrument Y_4 (the second weightometer) at 20 time steps, and a bias with a magnitude of +2 standard deviations was added to instrument Y_6 (the density meter) at 120 time steps. The gross error detection filter was tuned so that $\alpha = 0.05$ at a magnitude of 0.18 standard deviations. Note that specifying the desired detection level and the false positive allowance α assumes no prior knowledge of possible gross error values. This approach simply assigns a prior distribution that has a significance level α for a given magnitude of gross error.

The discrete-valued Bayesian network was tuned according to Equation (4.35) with $P(B_{(t+1)} = T|B_{(t)} = F) = 10^{-10}$ and $P(B_{(t+1)} = F|B_{(t)} = T) = 10^{-10}$, which corresponds to a switching probability of 10^{-10} . Small values were used for the switching probability in order to reduce the number of false positives. Since no prior assumptions were made about the possible magnitudes of the gross errors, adjustable values of \hat{Q}_β were used, and k_p was tuned based on simulations with gross errors of 4 standard deviations. This resulted in $k_p = 0.05$. The simulation results are shown in Figures 4.8 and 4.9.

In general, the algorithm performed quite well despite using a simulation that was designed to test its limits. There was some trouble when random errors in Y_2 and Y_7 tripped the correction alarm; this was most likely due to imperfect estimates of errors in Y_4 and Y_6 causing mild corruption in the state estimates. However, problems in Y_2 and Y_7 occurred at a time when most gross errors were accounted for resulting in relatively little corruption in the reference state estimate; thus, they re-converged to a reasonable value. This is why fast detection and correction are important to ensure as little as corruption as possible; as long as state estimates are not significantly corrupted, flagged instruments will re-converge to their proper values. Problems can still occur however, if biases are introduced to the system in an unobservable manner; this simply means that there are too many active bias states to ensure an uncorrupted state estimate.

False alarm rates can also be reduced by using low switching probabilities as was used in this simulation (a switching probability of 10^{-10}). Low switching

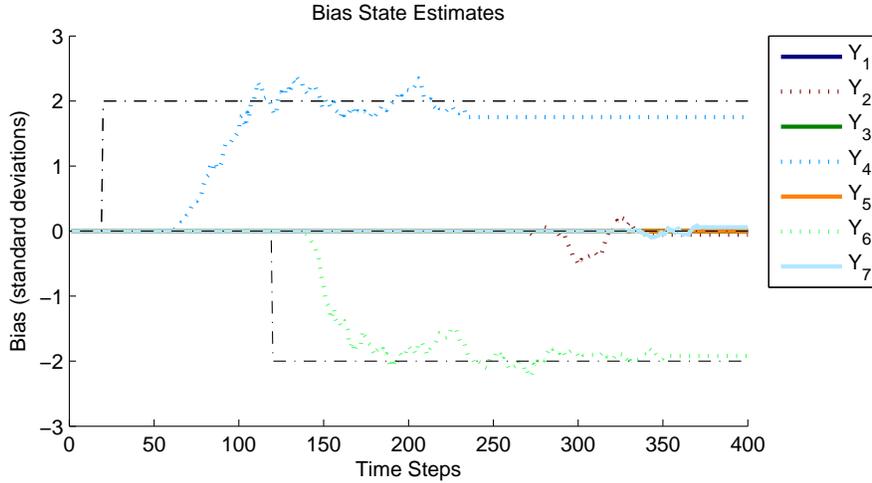


Figure 4.8: Gross Error Tracking (instruments 6 and 4)

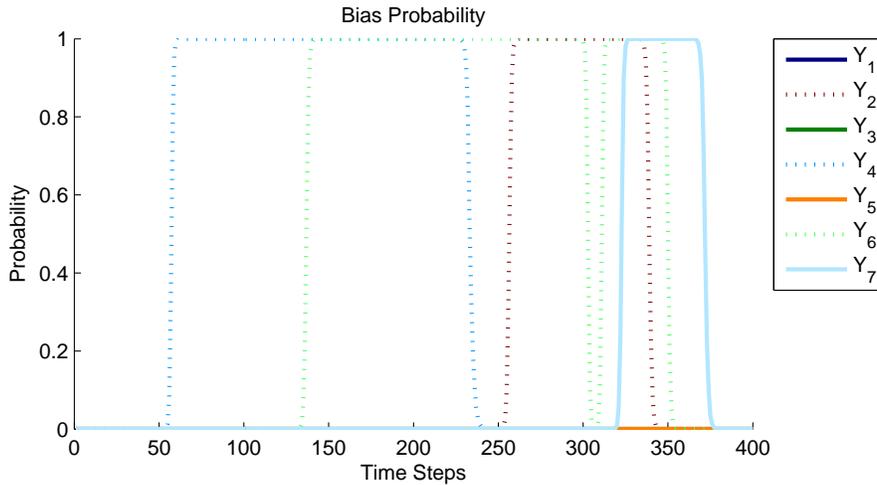


Figure 4.9: Gross Error Probability (instruments 6 and 4)

probabilities also result in a delayed response, as is apparent in Figure 4.8; this is because it takes a considerable amount of time to obtain enough evidence of bias to overcome the prior beliefs against switching. Delayed switching is a consequence of caution against false alarms. However, one must take caution not to set switching probabilities too low since delayed detection also means that the state estimate is corrupted for a longer period of time. Allowing for longer corruption periods can sometimes result in detecting the wrong gross error if there is not a high degree of instrument redundancy. From Figure 4.8, one can observe that at this error magnitude, detection occurs within approximately 30 time steps with no incorrectly identified instruments shortly after the major corruption occurred. This indicates that these tuning parameters allowed the algorithm to be resilient to corruption.

4.8 Preliminary Industrial Application

This technique was also applied to an industrial process. However, because this was a scheduled industrial process where dynamic model was not known beforehand, the hidden state model was defined according to Equation (4.6) instead of Equation(4.71). The corresponding constants were set to the following values: $K_1 = 1$, $K_2 = 0.1$, and $K_3 = 0.3$. The first case was applied to Train 2, a system wherein the second weightometer Y_4 was of primary concern; bias was for Y_4 reported to be on the area of 30%. Unfortunately, no data was found wherein the bias significantly shifted, thus there are many simultaneous biases in place at the start of this data. This method assumes that there is no bias at startup; however, in the case of simultaneous biases at startup, it tends to prioritize the correction of obvious bias before correcting less detectable bias, mitigating contamination from multiple simultaneous biases. Samples were taken at 10 minute intervals.

Figure 4.10(a) displays results for Train 2, where it appears that Y_4 clearly shows bias even at the 10 minute interval level. There also appears to be minor problems with the density meter, which is not uncommon (the density meter was more of concern for Train 3).

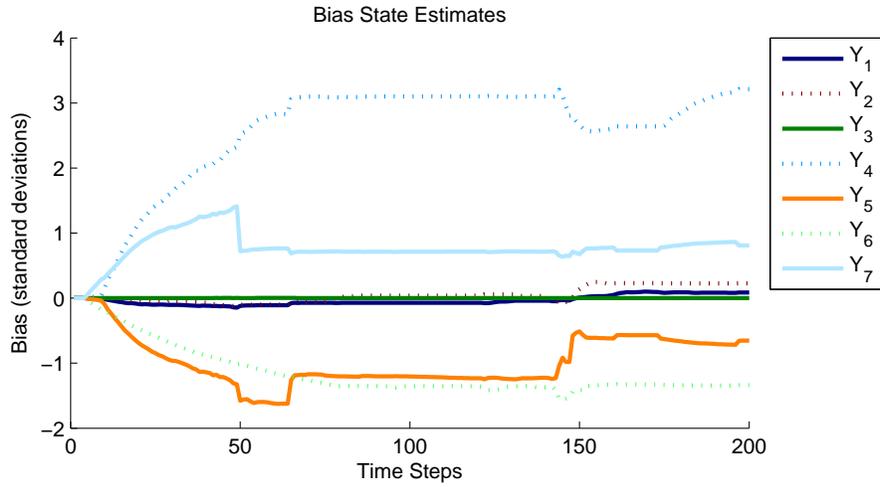
This method was also applied to Train 3 where it was reported that the second weightometer Y_4 was also having trouble, but that this bias tended to propagate as a slow drift. More trouble however, was noticed in the data where the density meter was well calibrated for when the system was off and the density was pure water, but not so well calibrated for measuring slurry density. In order to illustrate weightometer drift, data was taken at much longer time intervals (4 hour sampling time) and results are shown in Figure 4.10(b).

One would notice from the results in Figure 4.10(b) that the bias estimate for Y_6 is very sensitive toward on/off operating conditions. Corresponding state estimates were not included so as not to disclose proprietary process information; nevertheless, it was found that the bias estimate for Y_6 tended to sharply decrease when the system was turned off, but would quickly return to the previous estimate when the process resumed normal operation. This indicates that the instrument is well calibrated for when the system is turned off, but that it is poorly calibrated for when the system is running. These trends may also suggest that the bias for Y_6 is a function in terms of the state as opposed to an additional constant. Identifying a consistent bias estimate for this case would require prior knowledge of the bias function of the state and the use of nonlinear state estimation techniques.

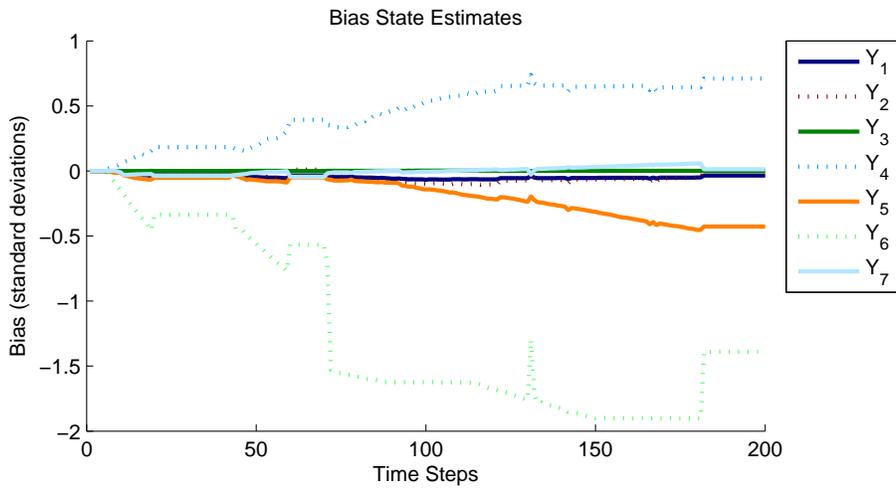
4.9 Conclusions

A methodology for the real-time detection and and quantification of instrument gross error has been developed using dynamic Bayesian methods. Furthermore, a sound methodology has been derived that tunes this procedure according to two of three possible specifications: (1) desired detection margin, (2) α which corresponds to desired fraction of instrument noise and gross error rejection, and (3) the prior gross error distribution.

From simulations it has been shown that the proposed algorithm is able to detect rapidly-propagating instrument error, and to selectively quantify it without contaminating error estimate of other instruments. In the industrial



(a) Gross Error Tracking Train 2



(b) Gross Error Tracking Train 3

Figure 4.10: (Industrial Application)

case where simultaneous errors are present, it has been shown to give higher priority to instruments that exhibit major gross errors.

Chapter 5

Conclusions

5.1 Concluding Remarks

The main contributions of this thesis can be summarized as follows:

- Deriving a method of smoothing out process data contaminated by a discrete process using the Binomial distribution for the purpose of data reconciliation
- Combining Bayesian learning (via EM algorithm) with black box FA initial values in order to estimate variance parameters for a gray box factor model based on mass balances
- Using Bayesian Learning with known instrument variance to perform data reconciliation and gross error detection
- Using a Dynamic Bayesian network based on residual error P-values for gross error detection
- Using a mass balance Kalman filter tuned by Bayesian learning for data reconciliation
- Conditionally adding gross error states to the mass balance Kalman filter as augmented states
- Justifying conditional gross error estimation for mass balance Kalman filter by investigating observability criteria
- Tuning the dynamic method based on common engineering specifications

5.2 Recommendations for future work

Research initiatives on the topic of the current research and the following related field are worthy of future investigations:

- There has been no theoretical development for tuning the switching parameters for the discrete DBN. It has been noticed that for any given switching parameter and size of gross error, there is a value of Q_Z that

minimizes detection time. This is because decreasing the values of Q_Z increases the resolution of the residual errors, allowing for the evidence requirements to switch the DBN to be met faster; however, small values of Q_Z means it takes more time for the filtered residual errors to reach a point where the evidence is significant. Being able to determine the optimal value of Q_Z for a given error magnitude and switching parameter may give insight to simultaneously tuning the Z-value filter and the discrete DBN to ensure harmonized results.

- This work has only considered the identification of gross errors; however, no attempt has been made to distinguish between instrument bias and process leaks. The use of fault diagnosis techniques to distinguish between the two types of gross errors may be worth investigating.
- Another more difficult uninvestigated problem would be to use Bayesian techniques to distinguish state discrepancies that are caused by process leaks, and discrepancies that are caused by bias instruments serving as inputs to control systems. This would be more difficult than distinguishing between leaks and single instrument biases, because both events may affect multiple instruments.

Bibliography

- [1] S. Bai, J. Thibault, and D. McLean.
- [2] A. Bjorck. *Numerical Methods for Least Squares Problems*. Siam, first edition, 1996.
- [3] William M. Bolstad. *Introduction to Bayesian Statistics*. John Wiley and Sons, first edition, 2004.
- [4] C.T. Chen. *Linear System Theory and Design*. Oxford University Press, third edition, 1999.
- [5] R. Clarke and et al. Underestimation of risk associations due to regression dilution in long-term follow-up of prospective studies. *American Journal of Epidemiology*, 150(4):341–353, 1999.
- [6] C. M. Crowe, Y. A. Garcia Campos, and A. Hrymak. Reconciliation of process flow rates by matrix projection 1. the linear case. *AIChE Journal*, 29(6):881–888, 1983.
- [7] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. 1967.
- [8] Ralph W. Pike Derya B. Ozyurt. Theory and practice of simultaneous data reconciliation and gross error detection for chemical processes. *Computers and Chemical Engineering*, 28:381–402, 2004.
- [9] Z. Griliches and J.A. Hausman. Errors in variables in panel data. *Journal of Econometrics*, 31:93–118, 1984.
- [10] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, third edition, 2001.
- [11] Biao Huang. Process identification based on last principal component analysis. *Journal of Process Control*, 11:19–33, 2001.
- [12] Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, sixth edition, 2008.
- [13] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [14] K. B. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence*. Chapman & Hall/CRC, first edition, 2004.

- [15] R. S. H. Mah and A. C. Tamhane. Detection of gross errors in process data. *AIChE Journal*, 28(5):828–830, 1982.
- [16] Bei Hu Markus Schladt. Soft sensors based on nonlinear steady-state data reconciliation in the process industry. *Chemical Engineering and Processing*, 46:1107–1115, 2007.
- [17] R. J. Meinhold and N. D. Singpurwalla. Understanding the kalman filter. *The American Statistician*, 37(2):123–127, 1983.
- [18] Kevin P. Murphy. The bayes net toolbox for matlab. In *The 33rd Symposium of the Interface of Computing Science and Statistics*, Orange County California, October 2001.
- [19] Kevin P. Murphy. Bayes net toolbox for matlab, 2007. <http://code.google.com/p/bnt/>.
- [20] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.
- [21] P.M. Reilly and R.E. Parbani. Application of statistical theory of adjustments to material balances. *13*, 1963.
- [22] H. Tong and C. M. Crowe. Detection of gross errors in data reconciliation by principal component analysis. *AIChE Journal*, 41(7):1712–1722, 1995.
- [23] N. Selby C. Visser T. Letwaba X.W. Pan, G. Metzner and K. Coleman. Development of weightometer soft sensor. *The Journal of The South African Institute of Mining and Metallurgy*, 38:291–296, 2003.
- [24] Ouguan Xu and Jian Chu Yu Miao, Hongye Su. Support vector regression approach for simultaneous data reconciliation and gross error or outlier detection. *Ind. Eng. Chem. Res.*, XXX(xxx):000, 2009.