uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Shuanghua Bai
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Chemical Engineering)
GRADE / DEGREE

Department of Chemical Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Data Reconciliation for Dynamic Processes

TITRE DE LA THÈSE / TITLE OF THESIS

Jules Thibault
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

David McLean
CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Marc Dubé

Atef Fahim

Jean Paris

David Taylor

Gary W. Slater
LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

Data Reconciliation for Dynamic Processes

by

Shuanghua Bai

A thesis submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in the Department of Chemical Engineering
UNIVERSITY OF OTTAWA

February 2006

Your file Votre référence
ISBN: 978-0-494-15007-8
Our file Notre référence
ISBN: 978-0-494-15007-8

STATEMENT OF CONTRIBUTIONS OF COLLABORATORS

I hereby declare that I am the sole author of this thesis. My supervisors, Professors David D. McLean and Jules Thibault in the Department of Chemical Engineering at the University of Ottawa, have provided excellent collaboration throughout this research program. Their helpful comments, suggestions and editorial corrections have significantly improved the quality of this thesis and publications.

Signature: Date: Feb. 4, 2006

ABSTRACT

In a modern chemical plant, the implementation of a distributed control system leads to a large number of measurements that are available online for process monitoring, control, optimization and management decision making. Unfortunately, these measurements often contain errors that degrade the information quality obtained from the raw data. This thesis is dedicated to the development of **d**ynamic **d**ata **r**econciliation (DDR) algorithms for the optimal estimation of variables in dynamic processes. More importantly, the DDR algorithms were implemented within the structures of feedback control loops, and the performance of the DDR algorithms as well as the controllers was quantitatively assessed via a series of process simulations. The DDR algorithms, acting as digital filters, were compared to commonly used filters, such as the **e**xponentially **w**eighted **m**oving **a**verage (EWMA), **m**oving **a**verage (MA), Kalman and extended Kalman filters. Methodologies to use the DDR algorithms to deal with autocorrelated noise were also investigated.

The DDR algorithms integrate information from both measurements and process dynamic models such that, at each sampling time, the estimates obtained by the DDR algorithms provide more precise representations of the current state of the process. Three DDR algorithms were developed, namely, **n**onlinear **p**rogramming (NLP) based DDR, predictor-corrector based DDR, and **a**uto**a**ssociative **n**eural **n**etwork (AANN) based DDR.

Evaluations of these DDR algorithms were conducted via simulations of three chemical processes, namely a cylindrical storage tank, a spherical storage tank and a binary distillation column. Results demonstrated that the DDR algorithms are efficient and effective tools for the estimation of dynamic processes. They perform significantly better than the EWMA and MA filters. Furthermore, compared to the Kalman filter, the DDR algorithm is easier to understand and to implement. Studies also showed that the structure of process models has considerable impact on the performance of the DDR. The use of the DDR algorithms embedded in feedback control loops significantly enhanced the

controller performance. For example, the cost function of the control system in the distillation column was reduced by 28~39% when linear, adaptive linear and nonlinear DDR algorithms were used. The cost function of the controller in the cylindrical storage tank was reduced by 46% using DDR, while it was reduced by 33% when using a EWMA filter.

RÉSUMÉ

Dans une usine chimique moderne, l'implantation d'un système réparti de commande automatique fournit un grand nombre de mesures qui sont accessibles en ligne et servent à la surveillance des procédés, à la commande automatique, à l'optimisation et à la prise de décisions. Toutefois, ces mesures contiennent souvent des erreurs qui dégradent la qualité de l'information obtenue à partir des données brutes. Cette thèse est consacrée au développement d'algorithmes dynamiques de réconciliation des données (DDR) pour l'estimation optimale des variables des procédés dynamiques. De plus, les algorithmes DDR développés sont incorporés à l'intérieur des boucles de régulation automatique, et la performance des algorithmes DDR comme celle des régulateurs sont évaluées par l'intermédiaire d'une série de simulations. Les algorithmes DDR, utilisés comme filtres numériques, sont comparés aux filtres généralement utilisés, tels les filtres à moyenne mobile pondérée exponentiellement (EWMA), à moyenne mobile (MA), Kalman et les filtres de Kalman étendus. Les méthodologies pour permettre l'utilisation des algorithmes DDR pour traiter le cas des bruits autocorrélés sont également étudiées.

Les algorithmes DDR intègrent l'information des mesures et des modèles dynamiques des procédés de sorte que, à chaque période d'échantillonnage, les estimations obtenues par les algorithmes DDR fournissent une représentation plus exacte de l'état actuel du procédé. Trois algorithmes DDR sont développés, à savoir, un algorithme DDR basé sur la programmation non-linéaire, un algorithme DDR sous la forme d'un prédicteur-correcteur, et un algorithme DDR sous la forme d'un réseau neuronal autoassociatif.

La performance des algorithmes DDR est évaluée par l'intermédiaire de simulations de trois procédés chimiques, soit un réservoir de stockage cylindrique, un réservoir de stockage sphérique et une colonne binaire de distillation. Les résultats démontrent que les algorithmes DDR sont des outils efficaces pour l'estimation des procédés dynamiques. Ils sont de beaucoup supérieurs aux filtres EWMA et MA. De plus, comparés au filtre de Kalman, ils sont plus faciles à comprendre et à implanter. La

structure des modèles dynamiques des procédés a un impact considérable sur la performance des algorithmes DDR. L'utilisation des algorithmes DDR à l'intérieur des boucles de commande peut augmenter de manière significative les performances du système de régulation. Par exemple, la valeur du critère de performance du système de commande de la colonne de distillation est réduite de 28~39% lorsque les algorithmes DDR linéaire, non-linéaire, et linéaire adaptatif sont utilisés. Le critère de performance du système de commande pour le réservoir cylindrique de stockage a été réduit de 46% en utilisant l'algorithme DDR et de 33% avec le filtre EWMA.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep gratitude to my supervisors Dr. David D. Mclean and Dr. Jules Thibault, who offered me this opportunity to carry out the PhD program. I sincerely appreciate the financial support they have provided which made this work possible. I thank them for their precious time to discuss the research problems during our regular meetings and whenever I went to their offices to see them. They not only have shared their knowledge on chemical engineering science, but also educated me how to write scientific papers, how to use the English language properly, and more importantly, how to become a scientific researcher. I realize that they have done tremendous work on my behalf. Their valuable instructions, helpful suggestions and beneficial guidance are great contributions to this thesis. I am deeply indebted to them. Their humanism, enthusiasm and professionalism have deeply influenced me and undoubtedly will benefit me forever.

Besides my supervisors, I would like to thank the rest of the professors and staff in the department who have given me assistance and support during the past five years. I also would like to take this moment to thank Dr. Jesse Shen, the president of Crechem Technologies Inc. in Ottawa, who has encouraged me to pursue this degree. Without his encouragement, this thesis would have been impossible.

Last, but not least, I thank my family. I owe tremendous debts to my wife, Yuning Cui. She has carried on most of our family work so that I could concentrate on this research program. I am also greatly indebted to my oldest child, Rulin Bai, and my newborn baby, Christopher Bai, to whom I should devote more time for their care. Finally, special thanks go to my parents who have given me endless love and support all the time.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER II

CHAPTER III

CHAPTER IV

CHAPTER V

CHAPTER VII

ANNEX B

LIST OF FIGURES

CHAPTER II

CHAPTER III

CHAPTER IV

CHAPTER V

CHAPTER VI

CHAPTER VII

CHAPTER VIII

ANNEX A

ANNEX B

CHAPTER I

Introduction

1 PREAMBLE

One of main thrusts of modern plant operation is to improve the quality of online information in distributed control systems (DCS). Accurate and reliable information about the current state of a process is paramount for plant monitoring and control. Unfortunately, sampled values of plant variables are usually corrupted by measurement noise. The presence of measurement noise not only prevents plant operators from identifying the true values of process variables, but also deteriorates controller performance when the raw measurements are directly transmitted to controllers to calculate control moves.

To cope with these problems, digital filters, such as exponentially weighted moving average (EWMA) or moving average (MA) filters have been widely applied. The EWMA and MA filters use measurement temporal redundancy where past and current measurements are averaged to give estimates for the current values of the process variables. These classical filters provide satisfactory performance for steady-state or slow dynamic processes. However, under transient process conditions, they inevitably introduce larger time delays, thereby reducing their effectiveness for noise attenuation.

Optimal estimation of dynamic processes is an important topic that has been studied for a long time by academic researchers and practicing engineers. Advanced model-based filters have been developed. The model-based filters employ both measurement temporal and spatial redundancy, where information from both past measurements and process models is used for state estimation. A well known model-based filter is the Kalman filter developed in 1960. It employs stochastic linear state-space process and measurement models. The most attractive advantage of the Kalman filter lies in its optimal estimation, in the sense of minimum mean squared prediction errors, and it has acquired a reputation as a panacea for process state estimation and prediction (Kamen and Su, 1999). However, the optimality of the Kalman filter requires two restrictive prerequisites: linear state-space models and independent white Gaussian noise for both process models and measurements. Although applications of the Kalman filter in tracking moving objects

such as aircraft and missiles have been common, their applications in chemical engineering are relatively infrequent. In these situations, difficulties are associated with identification of reliable stochastic state-space models as well as the complex mathematical manipulations in implementing the Kalman filters (Roffel and Chin, 1987; Wilson, et al., 1998; Brosilow and Joseph, 2002).

In recent years, there has been a nascent interest in using *data reconciliation* techniques for process state estimation. The principle of data reconciliation lies in its integration of information from both measurements and process models to provide more reliable estimates of process variables. More importantly, the reconciled data are consistent with known relationships between process variables, such as mass and heat balances. The concept of data reconciliation was initially developed to calculate mass balances for steady-state processes (Kuehn and Davidson, 1961; Mah et al., 1976; Hodouin and Everell, 1980; Crowe et al., 1983; Crowe, 1986). The interest in applying data reconciliation in chemical processing plants started in the late 1980s when plant managers realized that data reconciliation could turn raw process data into more consistent, reliable information, and that such information is critical for effective plant operation and management. Nowadays, data reconciliation for steady-state processes is a mature technology and has been widely applied in chemical, petrochemical and mineral industries. However, chemical processes are prone to frequent external disturbances and process state setpoint changes. Process conditions are continually undergoing variation. Moreover, some chemical processes are intrinsically dynamic (e.g., a batch reactor). The development of data reconciliation techniques for dynamic processes is critical and therefore is the theme of this thesis.

2 BACKGROUND AND LITERATURE REVIEW

Plant measurements are usually contaminated by measurement errors consisting of random noise, bias, and/or outliers. Random noise is due to irreproducible factors that randomly affect the measurements. Measurement bias represents a discrepancy between the expected value of the measurements and the true value of the process variable, and is

attributed to improper instrument installation and/or miscalibration. Spikes occurring in measurements due to nonrandom events such as a sudden failure of a measurement device characterize outliers. Measurement bias and outliers are often referred to as gross errors.

2.1 Steady-state data reconciliation

Steady-state data reconciliation is a technique used to improve the accuracy and precision of measurements by reducing the impact of random measurement errors as well as gross errors in steady-state process data. The concept of data reconciliation was initiated by Kuehn and Davidson (1961). Since then, it was studied extensively (Mah et al., 1976; Stanley and Mah, 1977; Hodouin and Everell, 1980; Romagmoli and Stephanopoulos, 1981; Crowe et al., 1983; Jordache et al., 1985; Tamhane and Mah, 1985; Serth and Heenan, 1986; Crowe, 1986, 1989; Narasimhan and Mah, 1987; Rosenberg et al., 1987; Tjoa and Biegler, 1991; Rollins and Davis, 1992; Tong and Crowe, 1995; Sanchez and Romagnoli, 1996). The basic principle of data reconciliation lies in its integration of information from both measurements and process models to provide more accurate estimates of process variables. For a steady-state process, the reconciled or estimated values of the process variables are obtained by minimizing a constrained, weighted sum of squared measurement errors, i.e.,

$$\text{Minimize } J(\hat{\mathbf{y}},\hat{\mathbf{z}}) = (\mathbf{y}-\hat{\mathbf{y}})^{\mathrm{T}}\,\mathbf{V}^{-1}(\mathbf{y}-\hat{\mathbf{y}}) \tag{1}$$

$$\text{subject to } \mathbf{f}(\hat{\mathbf{y}},\hat{\mathbf{z}}) = \mathbf{0}$$

$$\mathbf{g}(\hat{\mathbf{y}},\hat{\mathbf{z}}) \geq \mathbf{0}$$

where $\mathbf{y}$ is a M×1 vector of measurements for M process variables. Values in $\mathbf{y}$ are often averaged values of raw measurements over a period of time (e.g., one-hour period). $\hat{\mathbf{y}}$ is a M×1 vector of reconciled values for the M process variables, $\hat{\mathbf{z}}$ is a N×1 vector of estimates for N unmeasured process variables or model parameters, $\mathbf{V}$ is a M×M covariance matrix of the M measurements, $\mathbf{f}$ is a functional vector of model equality

constraints, and $\mathbf{g}$ is a functional vector of model inequality constraints including simple upper and lower bounds for $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$.

The simplest example in steady-state data reconciliation is to reconcile flow rates around a plant where all flows are measured. Because the flow rates must satisfy the mass balances around each unit, the reconciled flow rates are obtained in order to

$$\text{Minimize } J(\hat{\mathbf{y}}) = (\mathbf{y} - \hat{\mathbf{y}})^{\mathrm{T}} \mathbf{V}^{-1} (\mathbf{y} - \hat{\mathbf{y}}) \qquad (2)$$
$$\text{subject to } \mathbf{A}\hat{\mathbf{y}} = \mathbf{0}$$

where $\mathbf{A}$ is the incidence matrix of the plant. Each row of the incidence matrix represents a particular unit (a node) in the plant and each column represents a given stream, respectively. Each element in $\mathbf{A}$ is either 1, -1 or 0 depending on whether the corresponding flow stream is an input stream, an output stream or a stream that is not associated with this node, respectively. The optimization problem of Equation (2) can be solved using the method of Lagrange multipliers. It can be shown that the final solution is

$$\hat{\mathbf{y}} = \mathbf{y} - \mathbf{V}\mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{V}\mathbf{A}^{\mathrm{T}})^{-1}\mathbf{A}\mathbf{y} \qquad (3)$$

If unmeasured flows appear in the model constraints of Equation (2), it is possible to estimate the unmeasured flows by data reconciliation using the projection matrix method proposed by Crowe et al. (1983). The constraint equations of the mass balances are first partitioned into terms of measured and unmeasured variables; then the unmeasured variables are eliminated by multiplying by a projection matrix, and then the measured flows are reconciled. Finally, the unmeasured flows are estimated by least-squares (Hodouin et al., 1998; Narasimhan and Jordache, 2000; Romagnoli and Sanchez, 2000).

It is also possible to use other process models (e.g., thermodynamic equations, reaction kinetic equations, etc.) as constraints to reconcile the associated raw plant measurements. Because these process models, used as the constraints, are often nonlinear in terms of

reconciled or unmeasured variables, the reconciled values of process variables in Equation (1) cannot be solved analytically, but numerically by techniques of nonlinear programming (NLP). Detailed descriptions of nonlinear programming in optimization can be found in many monographs (e.g., Edgar et al., 2001).

Data reconciliation employs measurement redundancies to improve the accuracy and consistency of process data. Redundancy arises from the fact that at least some information about the process is known and this information relates measurements to each other. Some important concepts in data reconciliation necessarily need clarification. A *redundant variable* is a measured variable that can be estimated by other measured variables via process models, in addition to its own measurement. A *nonredundant variable* is a measured variable that cannot be estimated other than by its own measurement. An *observable variable* is an unmeasured variable that can be estimated from measured variables through process models. A *nonobservable variable* is a variable for which no information is available for its estimation. A comprehensive study for classifying these process variables in steady-state data reconciliation can be found in Crowe (1989).

Nowadays, with the implementations of DCS in the plants, steady-state data reconciliation has been interfaced with plant databases and runs online. The data reconciliation module can perform mainly two roles:

1. *Mass and heat balance reconciliation for the plant.* The balanced plant data can be used for plant performance monitoring, management reporting and other technical applications.
2. *Plant parameter estimation.* Accurate estimates of plant parameters, such as heat transfer coefficients, fouling factors of heat exchangers and tray efficiencies of distillation towers, play important roles for tracking equipment performance. One approach to the parameter estimation is to solve the estimation problem simultaneously with the data reconciliation. The model parameters are treated as unmeasured variables and adjusted via the data reconciliation algorithm to match the

plant measurements and process model constraints (MacDonald and Howat, 1988). The reconciled model parameters are expected to be more accurate and can be used with greater confidence.

2.2 Dynamic data reconciliation

Beginning in the early nineties, the benefits of applying data reconciliation to steady-state processes have triggered some researchers to develop data reconciliation techniques for dynamic processes. For example, Darouach and Zasadzinski (1991), Liebman et al. (1992), Ramamurthi et al. (1993), Albuquerque and Biegler (1996), Hodouin and Makni (1996), Bagajewicz and Jiang (1997) and Binder et al. (2002) extended the concept of steady-state data reconciliation to dynamic processes. Mathematically, the estimates of the measured process variables, as well as unmeasured variables or model parameters, at each sampling time t, are obtained by solving the constrained least-squares optimization problem expressed as

$$\text{Minimize } J(\hat{\mathbf{y}}_i, \hat{\mathbf{z}}_i) = \sum_{i=0}^{t} \left[(\mathbf{y}_i - \hat{\mathbf{y}}_i)^T \mathbf{V}^{-1} (\mathbf{y}_i - \hat{\mathbf{y}}_i) \right] \tag{4}$$

$$\text{subject to } \mathbf{f}\left[\frac{d\hat{\mathbf{y}}}{dt}, \hat{\mathbf{y}}, \hat{\mathbf{z}} \right] = \mathbf{0}$$

$$\mathbf{g}(\hat{\mathbf{y}}, \hat{\mathbf{z}}) = \mathbf{0}$$

$$\mathbf{h}(\hat{\mathbf{y}}, \hat{\mathbf{z}}) \geq \mathbf{0}$$

where $\mathbf{y}_i$ is a M×1 vector of measurements for M process variables at sampling time t = i, $\hat{\mathbf{y}}_i$ is a M×1 vector of estimates (reconciled values) for the M process variables at time t = i, $\hat{\mathbf{z}}_i$ is a vector of estimates for unmeasured process variables or model parameters, **f** represents a functional vector of differential equations, **g** and **h** represent functional vectors of algebraic equality and inequality constraints.

Comparing Equation (4) to Equation (1), it is observed that steady-state data reconciliation uses measurement spatial redundancy, meaning that the measurements in different locations of a plant can be related to each other by means of algebraic equations

such as the mass and heat balances. On the other hand, *dynamic data reconciliation (DDR)* uses both spatial and temporal redundancies in the measurements. The temporal redundancy of a measurement implies that the current measurement can be related to its past measurements via differential equations.

For the DDR problem formulated by Equation (4) with model constraints consisting of linear differential equations, Darouach & Zasadzinski (1991) developed a recursive scheme, while Bagajewicz & Jiang (1997) proposed an integral approach. With model constraints consisting of nonlinear differential and algebraic equations, Liebman et al. (1992) proposed the use of orthogonal collocation on finite elements, Albuquerque and Biegler (1996) applied an implicit Runge-Kutta method, and Binder et al. (2002) proposed a non-uniform discretization method to discretize the differential equations within a moving window over a time horizon. In each case the DDR problem was converted to minimizing the objective function constrained by algebraic equations so that the optimization problem was solved by nonlinear programming.

Although these approaches to solving the DDR problem formulated by Equation (4) have been well established in the literature, the DDR algorithm suffers from several shortcomings, notably:

1. A critical assumption in Equation (4) is that the models represent the true dynamics of the process, and the reconciled data exactly satisfy the models. However, no mathematical model is a perfect representation of a real plant, as they invariably contain some degree of model mismatch (model error). The reconciled values should therefore be obtained considering both measurement and model errors.
2. First principle models, such as reaction rate equations, were used as constraints in reconciling the raw measurements. Unfortunately, such phenomenological models are rarely available in practice for chemical plants. Consequently, the difficulties to obtain such phenomenological models would be one factor that has impeded DDR for a wide range of applications.

3. The DDR algorithm requires complex and sophisticated mathematical manipulations. As a result, online computations to iteratively solve the nonlinear optimization problem for dynamic data reconciliation can be significant.

In an early attempt to overcome these difficulties, Makni et al. (1995) and Hodouin and Makni (1996) modified the steady-state data reconciliation algorithm to perform real-time data reconciliation considering process dynamics. More importantly, the effect of model errors on the reconciled values was considered. In this case, the reconciled values of measured process variables and the estimates of unmeasured variables at each sampling time, t, were obtained by simultaneously minimizing the weighted sum of squared measurement and model errors expressed as

$$\text{Minimize } J(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) = \sum_{i=t-K}^{t} \left[(\mathbf{y}_i - \hat{\mathbf{y}}_t)^T \mathbf{V}^{-1} (\mathbf{y}_i - \hat{\mathbf{y}}_t) \right] + \mathbf{g}^T(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) \Psi^{-1} \mathbf{g}(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) + \mathbf{f}^T(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) \Omega^{-1} \mathbf{f}(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) \tag{5}$$

where $\mathbf{g}(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t)$ is a functional vector of process steady-state models such as mass conservation equations, $\mathbf{f}(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t)$ is a functional vector of dynamic process models in discretized formats and Ψ and Ω are the associated covariance matrices of the model errors (i.e., $\Psi = \text{Cov}(\zeta_t)$ with $\zeta_t = \mathbf{g}(\mathbf{y}_t, \mathbf{z}_t)$, and $\Omega = \text{Cov}(\mathbf{e}_t)$ with $\mathbf{e}_t = \mathbf{f}(\mathbf{y}_t, \mathbf{z}_t)$). The current and recent past measurements $\mathbf{y}_{t-K}$, $\mathbf{y}_{t-K+1}$, ..., $\mathbf{y}_t$ within a moving window [t-K, t] were used in reconciling the current measurements as well as estimating the current unmeasured variables. Minimization of the objective function attempts to smooth the measured values, while forcing the reconciled data to obey the steady-state models. However, it also allows the reconciled data to vary to a certain extent with process dynamics. The trade-off between these properties for the reconciled values is balanced by adjusting the window width, K, and the elements in Ψ and Ω, all of which are treated as tuning parameters in this data reconciliation algorithm. The optimization of Equation (5) can be solved analytically or numerically by nonlinear programming, depending on the complexity of the problem to be tackled. The DDR algorithm formulated by Equation (5)

is very useful for processes having slow dynamics. Although, this DDR algorithm can also be applied to processes having fast dynamics by setting a small number for the moving window width K and very large numbers for the elements in Ψ, the incorporations of the moving window and the steady-state models $\mathbf{g}(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t)$ in the objective function make the algorithm complex.

3 SCOPE OF RESEARCH

In this work, it is assumed that the plant measurement error contains random noise only. Other issues for measurement bias and outliers detection for dynamic processes are not considered. They have been addressed, however, in many papers and monographs (e.g., McBrayer and Edgar, 1995; Chen and Romagnoli, 1998; Gertler, 1998; Chiang et al., 2001; Abu-el-zeet et al., 2002; Luo et al., 2003). The main objectives of this thesis were:

1. *To develop efficient DDR algorithms.* The developed DDR algorithms were based on statistical fundamentals and principles, and are expected to have theoretical and practical significance.
2. *To investigate the impact of model structure on the performance of DDR.* DDR uses process models as redundant information to compensate for measurement error and, as a result, process models play an important role in the performance of DDR. Often, first principle models such as reaction kinetics, thermodynamics for a chemical process are difficult or impractical to obtain. Consequently, black-box models have to be identified and used. The black-box models have a variety of structures (e.g., linear or nonlinear). The following model structures were studied: linear model, linear model having adaptive gain, and nonlinear model (neural networks).
3. *To evaluate controller performance with embedded DDR strategies.* Investigations of DDR to this point have been exclusively focused on open loop operation. This work, however, studied the use of DDR within the structure of feedback control loops to mitigate the impact of measurement noise on controller performance. The performance of DDR as well as the controllers was quantitatively evaluated. In

addition, the performance of DDR was compared to that of traditionally used EWMA and MA filters for their ability to improve controller performance.

4. *To compare the proposed DDR algorithm with the Kalman filter.* Both the Kalman filter and DDR utilize process models for the estimation of dynamic processes. Relationships between the two strategies were discussed. Assessments of the DDR and the Kalman filter to deal with measurements corrupted by both white noise and autocorrelated noise were conducted.
5. To develop a novel AANN-based DDR algorithm. An approach to use **a**uto**a**ssociative **n**eural **n**etworks (AANN) to perform DDR was explored in this work. Architectures of AANNs for dynamic processes were proposed. Network training methodologies for effective dynamic data reconciliation were studied. Closed-loop performance of the AANN-based DDR was quantitatively evaluated.

These studies were carried out via process simulation, using developed FORTRAN programs. Three simulated processes, namely a cylindrical storage tank, a spherical storage tank, and a binary distillation column were employed. The cylindrical storage tank and the spherical storage tank are simple processes used for ease of understanding and simplicity of analysis, whereas the distillation column is a complex process used to evaluate more typical chemical engineering processes. Respective descriptions for the three processes are presented in each chapter whenever they are cited as illustrative examples in this thesis.

4 THESIS STRUCTURE

This thesis is presented in paper format. Chapter 2, the first paper, investigates the impact of measurement noise on controller performance. Also, two data reconciliation filters, namely, a single-set data reconciliation (SSDR) filter and a moving-window data reconciliation (MWDR) filter are developed in this paper. The SSDR and MWDR filters are categorized as quasi-steady-state data reconciliation algorithms because stationary process models are used in the data reconciliation. The controller performance with the embedded filters is quantitatively assessed. A simulated distillation column is employed

in this study. Simulation results show that the use of data reconciliation filters can effectively attenuate noise propagation inside feedback control loops, therefore allowing implementation of more aggressive controllers, and thereby yielding better performance of the control system. Furthermore, it shows that the degree of knowledge available in the process measurements is an important factor in the performance of the data reconciliation filters. It demonstrates that the data reconciliation filters can deal with biased and autocorrelated measurements.

Chapter 3, the second paper, develops dynamic data reconciliation (DDR) algorithm where process dynamic models replace the quasi-steady-state models in Chapter 2 in order to overcome their limitations. Based on a least-squares criterion where the weighted sum of squared deviations between reconciled and measured values and deviations between reconciled and model predicted values are minimized, the developed DDR algorithm has a predictor-corrector form. The DDR algorithm, as a filter, is evaluated when it is embedded in feedback and feedforward/feedback control schemes. In addition, the performance of the DDR filter is compared to that of commonly used EWMA and MA filters for improved process performance. The simulated cylindrical storage tank and the distillation column processes are employed as numerical examples to demonstrate DDR strategies using phenomenological and empirical process models.

The work reported in the first and the second papers is an extension of the candidate's Master's thesis on the same research topics. These papers were significantly revised and improved during the candidate's PhD program.

Chapter 4, the third paper, rigorously derives the DDR algorithm based on statistical properties of measurements and models using Bayesian arguments. The DDR algorithm has a convenient predictor-corrector format like the Kalman filter. The relationships between the DDR and the Kalman filter are studied, showing that the Kalman filter is a subset of the DDR. Statistical analysis for reconciled values shows the DDR is an unbiased estimator and it results in more precise estimates than using either raw measurements or model predictions individually. Modifications to the DDR algorithm are

introduced to allow its applications to situations involving autocorrelated measurement noise. For cases involving autocorrelated measurement noise, an effective shortcut procedure is also developed to simplify implementations. The effectiveness of the DDR algorithm embedded in the feedback control system for the distillation column is demonstrated, and compared to that of Kalman filter via the simulation.

Process dynamic models provide an integral part of the DDR algorithm. When phenomenological models are unavailable or impractical to develop, black-box models can be identified and used in the DDR. Chapter 5, the fourth paper, investigates the impact of the structures of black-box models on the performance of the DDR. It shows, despite their different forms, black-box models can be tailored and used in DDR. Implementations of linear DDR, adaptive-linear DDR and nonlinear DDR strategies are demonstrated for the simulated distillation column. It is shown that the structures of the models have a considerable impact on the performance of the DDR as well as the performance of controllers. A linear DDR can successfully attenuate the measurement noise. However, further improvement can be achieved using adaptive-linear and nonlinear DDR which employ more comprehensive models that can more efficiently capture underlying dynamics of the process.

Implementations of the DDR algorithms developed in Chapters 2-5 require calculating model predictions (e.g., one-step-ahead predictions) at each sampling time, or solving a dynamic optimization problem for cases where model predictions cannot be explicitly calculated. For complex processes, the real-time optimization may need longer computation time. To remedy this problem, a method to train an autoassociative neural network (AANN) to perform the DDR is developed in Chapter 6, the fifth paper. In this study, the architecture of an AANN for a dynamic process is developed. Then the AANN is trained by minimizing the data reconciliation criterion. A methodology to train the AANN to perform DDR is proposed. It is shown that, after offline training, the AANN can be implemented online to perform dynamic data reconciliation to efficiently attenuate the impact of measurement noise on controllers.

The AANN-based DDR algorithm developed in Chapter 6 is evaluated for the control of a linear process, the cylindrical storage tank, and a nonlinear process, the spherical storage tank. The storage tanks have multiple inputs and a single output. Phenomenological models (mass balances) for the two processes are used to train the AANNs to perform DDR. Unfortunately, such phenomenological models are often unavailable for complex chemical processes. Consequently, Chapter 7, the sixth paper, explores the methodologies of AANN-based DDR using process empirical models. Moreover, strategies to apply AANN-based DDR to multiple input and multiple output processes are demonstrated. The multivariable control of the distillation column process is employed as an illustrative example in this study. Two AANN architectures for dynamic data reconciliation, namely a feedforward AANN and a recurrent AANN, are proposed. The performance of the recurrent AANN is shown to be more effective than that of the feedforward AANN. The AANN-based DDR is also shown to be robust to changes of noise level in plant measurements. Also, the performance of the AANN-based DDR is compared to that of the Kalman filter in the control of the distillation column.

After the presentations of the preceding studies, Chapter 8 summarizes this research work, draws some overall conclusions and emphasizes some important issues. Some recommendations and directions for future studies are also discussed in this chapter.

The DDR algorithms developed in this work are based on the assumption that the measurement noise and model prediction error are normally distributed. If this assumption is valid, it can be shown that the reconciled values are also normally distributed and the reconciliation error is less than both model prediction and the measurement errors. In order to validate this assumption, histograms of measurement errors, model prediction errors and reconciliation errors for the storage tank and distillation column processes are presented in Annex A. It is shown that both the model prediction errors and reconciliation errors very closely resemble normal distribution, and the reconciliation errors have smaller standard deviations than the model prediction errors and the measurement errors. It can therefore be concluded that the assumption of normal distribution made in this work is valid.

Annex B presents the initial development of a simple dynamic model for an autogenous/semi-autogenous (AG/SAG) grinding process. Compared to conventional AG/SAG models reported in the literature, the proposed model has a smaller number of parameters that can be more easily calibrated using plant data. Based on the proposed models, steady-state conditions and dynamic behaviors of the grinding process are investigated. This work provides a starting point for future studies on the applications of DDR algorithms in an industrial grinding process. In fact, this work started with the collaboration of COREM, a Quebec mineral processing consortium, with the objective to calibrate and use the model for on-line applications. However, priorities of the company shifted and the project is not pursued at the present time. It is believed, however, that this model could, following a proper calibration, be used efficiently for dynamic data reconciliation, process control and optimization in the mineral industry.

Finally, all simulation software developed and used in these studies is packaged into a CD. Details about the contents of the CD are provided in Annex C.

REFERENCES

Abu-el-zeet, Z.; V.M. Becerra and P.D. Roberts (2002). Combined bias and outliers identification on dynamic data reconciliation. *Computers Chem. Engng*, **26**, 921-935.

Albuquerque, J. and L. Biegler (1996). Data reconciliation and gross error detection for dynamic systems. *AIChE J.*, **42**, 2841-2856.

Bagajewicz, M. and Q. Jiang (1997). Integral approach to plant linear dynamic reconciliation. *AIChE J.*, **43**, 2546-2558.

Binder, T.; L. Blank; W. Dahmen and W. Marquardt (2002). On the regularization of dynamic data reconciliation problems. *J. Process Control*, **12**, 557-567.

Brosilow, C. and Joseph, B. (2002). *Techniques of model-based control*. Prentice Hall, New Jersey.

Chen, J. and J.A. Romagnoli (1998). A strategy for simultaneous dynamic data reconciliation and outlier detection. *Computers Chem. Engng*, **22**, 559-562.

Chiang, L.H.; E.L. Russell and R.D. Braatz (2001). *Fault detection and diagnosis in industrial systems*, Springer, London.

Crowe, C.M.; Y.A.G Campos and A. Hrymak (1983). Reconciliation of process flow rates by matrix projection. Part I: Linear case. *AIChE J.*, **29**, 881-888.

Crowe, C.M. (1986). Reconciliation of process flow rates by matrix projection. Part II: Nonlinear case. *AIChE J.*, **32**, 616-623.

Crowe, C.M. (1989). Observability and redundancy of process data for steady-state data reconciliation. *Chem. Eng. Sci.*, **44**, 2909-2917.

Darouach, M. and M. Zasadzinski (1991). Data reconciliation in generalized linear dynamic systems. *AIChE J.*, **37**, 193-201.

Edgar, T.F.; D.M. Himmelbau and L.S. Lasdon (2001). *Optimization of chemical processes.* 2nd Ed., McGraw Hill, Boston.

Gertler, J.J. (1998). *Fault detection and diagnosis in engineering system*. Marcel Dekker, New York.

Hodouin, D. and M.D. Everell (1980). A hierarchical procedure for adjustment and material balancing of mineral processes data. *Int. J. Miner. Process.*, 7, 91-116.

Hodouin, D. and S. Makni (1996). Real-time reconciliation of mineral processing plant data using bilinear material balance equations coupled to empirical dynamic models. *Int. J. Miner. Process*, **48**, 245-264.

Hodouin, D.; A. Mirabedini; S. Makni and C. Bazin (1998). Reconciliation of mineral processing data containing correlated measurement errors. *Int. J. Miner. Process*, **54**, 201-215.

Jordache, C.; R.S.H. Mah and A.C. Tamhane (1985). Performance studies of the measurement test for detecting of gross error in process data. *AIChE J.*, **31**, 1187-1201.

Kamen, E.W. and Su, J.K. (1999). *Introduction to Optimal Estimation*. Springer, London.

Kramer, M.A. (1992). Autoassociative neural networks. *Computers Chem. Engng*, **16**, 313-328.

Kuehn, D.R. and H. Davidson (1961). Computer control, II Mathematics of control. *Chem. Eng. Prog.*, **57**, 44-47.

Liebman, M.J.; T.F. Edgar and L.S. Lasdon (1992). Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers Chem. Engng*, **16**, 963-986.

Luo, R.; M. Misra; T. Soderstrom and D.M. Himmelblau (2003). Sensor fault detection via multiscale analysis of prediction model residuals. *Ind. & Eng. Chem. Res.*, **42**, 3372-3380.

MacDonald, R.J. and C.S. Howat (1988). Data reconciliation and parameter estimation in plant performance analysis. *AIChE J.*, **34**, 1-8.

Mah, R.S.H.; G. Stanley and D. Downing (1976). Reconciliation and rectification of process flow and inventory data. *Ind. Eng. Chem. Proc. Des. Dev.*, **15**, 175-183.

Makni, S. and D. Hodouin and C. Bazin (1995). A recursive node imbalance method incorporating a model of flowrate dynamics for on-line material balance of complex flowsheets. *Miner. Eng.*, **8**, 753-766.

McBrayer, K.F. and T.F. Edgar (1995). Bias detection and estimation in dynamic data reconciliation. *J. Process Control*, **4**, 285-289.

Narasimhan, S. and Jordache, C. (2000). *Data reconciliation & gross error detection, an intelligent use of process data.* Gulf Publishing Company, Houston.

Narasimhan, S. and R.S.H. Mah (1987). Generalized likelihood ratio method for gross error identification. *AIChE J.*, **33**, 1514-1521.

Ramamurthi, Y.; P.B. Sistu and B.W. Bequette (1993). Control relevant dynamic data reconciliation and parameter estimation. *Computers Chem. Engng*, **17**, 41-59.

Roffel, B. and P. Chin (1987). *Computer control in the process industries*. Lewis Publishers, Chelsea, Michigan.

Rollins, D.K. and J.F. Davis (1992). Unbiased estimation technique for identification of gross errors. *AIChE J.*, **38**, 563-571.

Romagnoli, J.A. and G. Stephanopoulos (1981). Rectification of process measurement data in the presence of gross errors. *Chem. Eng. Sci.*, **36**, 1849-1863.

Romagnoli, J.A. and M.C. Sanchez (2000). *Data processing and reconciliation for chemical process operations.* Academic Press, San Diego.

Rosenberg, J.; R.S.H. Mah and C. Jordache (1987). Evaluation of schemes for detecting and identification of gross errors in process data. *Ind. & Eng. Chem. Proc. Des. Dev.*, **26**, 555-564.

Sanchez, M. and J. Romagnoli (1996). Use of orthogonal transformations in data classification-Reconciliation. *Computers Chem. Engng*, **20**, 483-493.

Serth, R.W. and W.A. Heenan (1986). Gross error detection and data reconciliation in steam-metering systems. *AIChE J.*, **30**, 743-747.

Stanley, G.M. and R.S.H. Mah (1977). Estimation of flows and temperatures in process network. *AIChE J.*, **23**, 642-650.

Tamhane, A.C. and R.S.H. Mah (1985). Data reconciliation and gross error detection in chemical process network. *Technometrics*, **27**, 409-422.

Tjoa, I.B. and L.T. Biegler (1991). Simultaneous strategies for data reconciliation and gross error detection of nonlinear systems. *Computers Chem. Engng*, **15**, 679-690.

Tong, H. and C.M. Crowe (1995). Detection of gross errors in data reconciliation by principle component analysis. *AIChE J.*, **41**, 1712-1722.

Wilson, D.I., M. Agarwal and D.W.T. Rippin (1998). Experience Implementing the Extended Kalman Filter on an Industrial Batch Reactor. *Computers Chem. Engng*, **22**, 1653-1672.

CHAPTER II

Closed-loop Data Reconciliation for the Control of a Binary Distillation Column

Shuanghua Bai, Jules Thibault* and David D. McLean

Department of Chemical Engineering

University of Ottawa

Ottawa, Ontario, Canada K1N 6N5

Published in **Chemical Engineering Communications**, **Volume 192, 1444-1467, 2005.**

* Corresponding author. Tel: 613-562-5800 ext. 6094

Email: thibault@genie.uottawa.ca

ABSTRACT

Data reconciliation is a procedure that makes use of process models along with process measurements to give more precise and consistent estimates for process variables. Data reconciliation has been traditionally used to provide a more representative set of data to calculate steady-state inventories and process yields. For dynamic systems, the use of data reconciliation is relatively nascent. This article examines the potential use of data reconciliation in closed-loop control as a filter to attenuate the noise in measurements of the controlled variables so that the controllers can access more accurate sets of data. Data reconciliation filters were implemented in simulations of a PID control system for a binary distillation column. Results showed that data reconciliation could efficiently reduce the propagation of measurement noise in control loops, so that the overall performance of the controller is enhanced.

Keywords: Measurement noise, Data reconciliation, Distillation, Process control.

1 INTRODUCTION

Reliable knowledge of the current state of a process is paramount to its assessment, control, and optimization. However, chemical plant measurements are inevitably corrupted by process noise. The impact of such noise on process measurements can be represented by

$$\mathbf{y} = \mathrm{E}[\mathbf{y}] + \boldsymbol{\varepsilon} \tag{1}$$

where $\mathbf{y}$ is a $M \times 1$ vector of the M observed values of the measured variables, $\mathrm{E}[\mathbf{y}]$ is a $M \times 1$ vector of expected values of these process variables and ε is a $M \times 1$ vector of the measurement noise, which is usually assumed to be normally distributed with mean zero and known variance.

The measurement noise is usually of high frequency, and results in high-frequency oscillations of manipulated variables that deteriorate the performance of the control system. In order to approach optimal control, it is therefore necessary to attenuate the measurement noise before calculating the control actions. Analog and digital filters are widely used to cope with these problems. Narasimhan and Jordache (2000) have discussed classical digital filters, including linear/nonlinear exponential filters, moving average filters, and polynomial filters. These classical digital filters provide satisfactory performance under steady-state conditions if they are well tuned. However, under transient conditions, their efficiency to reduce noise is decreased and time delays are introduced. Today, the use of high-speed computers in process control makes it possible to implement more complex algorithms to filter the measurement noise. One such approach is data reconciliation (DR), a technique based on using process models as well as process measurements to obtain a more accurate estimate of system variables, particularly controlled variables in the context of this study.

Data reconciliation has been widely applied in a variety of processing industries. However, most applications have been limited to process monitoring, gross error

detection, and process inventory calculations. For example, Bussani et al. (1995), Pierucci et al. (1996), Chiari et al. (1997), and Li et al. (2001) respectively applied on-line steady-state data reconciliation and optimization to a hydrogen plant, olefin plant, refinery, and crude oil distillation unit. McBrayer et al. (1998) applied open-loop nonlinear dynamic data reconciliation to a feed-blend tank to estimate unmeasured flow rates. Soderstrom et al. (2000) successfully implemented a real-time dynamic data reconciliation strategy to improve inventory calculations of a diluent for a chemical plant. Ramamurthi et al. (1993) showed that dynamic data reconciliation in nonlinear systems could improve the closed-loop performance of a nonlinear predictive controller. Recently, Abu-el-zeet et al. (2002) investigated the use of dynamic data reconciliation and systematic bias detection in model predictive control (MPC).

Despite these applications of DR, no reports of the use of DR with widely used proportional-integral-derivative (PID) control schemes were found. The use of data reconciliation within the control loop is shown schematically in Figure 1. The data reconciliation algorithm is treated as a digital filter to reconcile the raw measurements; then the reconciled data are used by the controllers to calculate adjustments for the manipulated variables.

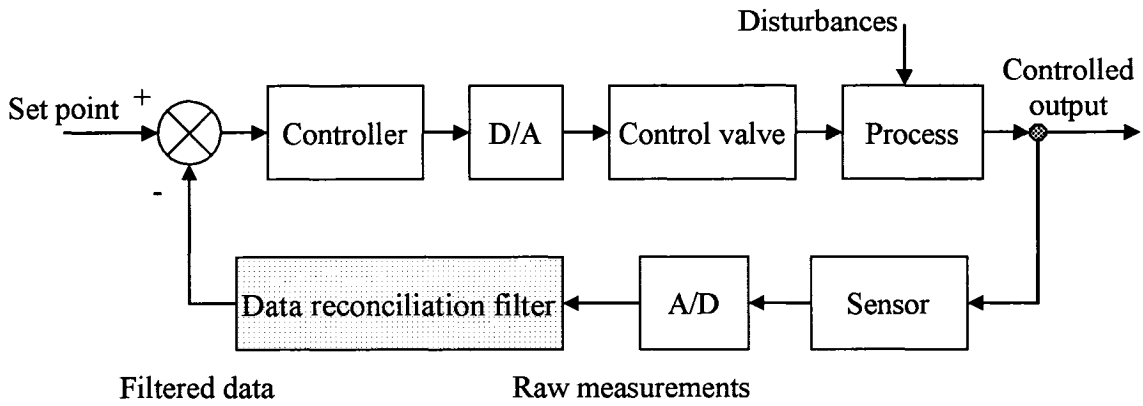


Figure 1 Application of data reconciliation algorithms in process control.

The objective of this work was to evaluate the use of quasi-steady-state data reconciliation to improve the performance of conventional PID controllers when measurement noise is significant. The presence of measurement noise distorts the

dynamics of the process, leading to the detuning of controllers that, in turn, results in more sluggish control action. In this work, data reconciliation algorithms, embedded within control loops, were used as filters to reduce the impact of measurement noise and its propagation through the control loop. It was expected that noise reduction would allow higher controller gains to be used such that the performance of the controllers would be enhanced. The performance of a quasi-steady-state data reconciliation filter for the control of a binary distillation column was evaluated and compared to the use of classical filters.

2 FORMULATION OF DATA RECONCILIATION FILTERS

Data reconciliation techniques are used to obtain precise estimates of the true (expected) values of measured or unmeasured process variables that are consistent with underlying process models. Data reconciliation can be formulated as a constrained weighted least-squares optimization problem such that reconciled values of process variables are those that

$$\text{Minimize } J(\hat{\mathbf{y}},\hat{\mathbf{z}}) = (\mathbf{y}-\hat{\mathbf{y}})^T \mathbf{V}^{-1}(\mathbf{y}-\hat{\mathbf{y}}) \tag{2}$$

$$\text{subject to } \mathbf{f}(\hat{\mathbf{y}},\hat{\mathbf{z}}) = \hat{\delta}$$

$$\mathbf{y}_l \le \hat{\mathbf{y}} \le \mathbf{y}_u$$

$$\mathbf{z}_l \le \hat{\mathbf{z}} \le \mathbf{z}_u$$

where $\hat{\mathbf{y}}$ is a $M\times 1$ vector of estimates for the true values of measured process variables, $\hat{\mathbf{z}}$ is a $N\times 1$ vector of estimates for the true values of unmeasured variables, $\mathbf{z}$ is a $N\times 1$ vector of true values of the unmeasured variables, $\mathbf{V}$ is a $M\times M$ covariance matrix for the measured variables. Vectors $\mathbf{y}_l$, $\mathbf{y}_u$, $\mathbf{z}_l$, and $\mathbf{z}_u$ are vectors of lower and upper bounds for the measured and unmeasured variables, $\mathbf{f}$ is a $C\times 1$ model constraint vector whose elements are process model equations (e.g., mass and energy balances) that may be algebraic or differential and $\hat{\delta}$ is a $C\times 1$ vector of model residuals that are the estimates of model random error, $\mathbf{f}(\mathbf{y},\mathbf{z}) = \delta$.

Conventional data reconciliation algorithm formulated by Equation (2) considers the process models to be exact such that all elements of the model residual vector $\hat{\delta}$ are equal to zero. This implies that reconciled values are adjusted to perfectly satisfy these models. However, perfect models rarely exist and the reconciliation of measured values will more likely be a compromise between inaccuracies in both measurements and process models. In such cases, at each sampling time t, the reconciled values of the process variables, $\hat{\mathbf{y}}_t$ and $\hat{\mathbf{z}}_t$, should be obtained by minimizing the weighted sum of squared measurement and model errors defined as

$$\text{Minimize } J(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) = (\mathbf{y}_t - \hat{\mathbf{y}}_t)^T \mathbf{V}^{-1} (\mathbf{y}_t - \hat{\mathbf{y}}_t) + \mathbf{f}^T(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) \boldsymbol{\Omega}^{-1} \mathbf{f}(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) \qquad (3)$$

$$\text{subject to } \mathbf{y}_{t,l} \le \hat{\mathbf{y}}_t \le \mathbf{y}_{t,u}$$

$$\mathbf{z}_{t,l} \le \hat{\mathbf{z}}_t \le \mathbf{z}_{t,u}$$

where $\boldsymbol{\Omega} = \{v_{i,j}\}$ $i = 1, 2, \ldots, C$; $j = 1, 2, \ldots, C$, is a $C \times C$ covariance matrix of model error.

For linear models, the variances and covariances in the $\boldsymbol{\Omega}$ matrix can be obtained analytically. For nonlinear models, they can be calculated using linear approximations or they can be calculated more rigorously by Monte Carlo simulation. If correlations between the model errors are neglected, $\boldsymbol{\Omega}$ becomes diagonal.

Equation (3) will be called the Single Set Data Reconciliation (SSDR) filter, since it only uses information available at the current sampling instant. It should be noted that: i) the process models, $\mathbf{f}(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) = \hat{\delta}$, employed in the SSDR filter, can be fundamental or empirical; ii) since process models contain unmeasured variables, the number of models must be greater than the number of unmeasured variables in order that the models can provide the redundancy required for the data reconciliation in the SSDR case.

In order to further reduce the noise level, a moving average filter can be integrated within the structure of a SSDR filter to make use of past information about process measurements. If the differences of the current estimates from the most recent L

measurements are minimized simultaneously with the minimization of the model errors, a Moving Window Data Reconciliation (MWDR) filter can be formulated as

$$\text{Minimize } J(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) = \sum_{k=0}^{L-1} \left[(\mathbf{y}_{t-k} - \hat{\mathbf{y}}_t)^T \mathbf{V}^{-1} (\mathbf{y}_{t-k} - \hat{\mathbf{y}}_t) + \mathbf{f}^T (\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) \Omega^{-1} \mathbf{f} (\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) \right] \quad (4)$$

$$\text{subject to } \mathbf{y}_{t,l} \le \hat{\mathbf{y}}_t \le \mathbf{y}_{t,u}$$

$$\mathbf{z}_{t,l} \le \hat{\mathbf{z}}_t \le \mathbf{z}_{t,u}$$

where L is the window width in which L sets of recent measurements are used. The MWDR filter considers the process as being stationary within the window. The window width, L, can be considered as a tuning parameter for the filter. If L is large, the filtered data should be smoother at the expense of longer time delays under dynamic changes. When the variance of the model error is significantly larger than the variance of the measurement error, the MWDR filter approaches the simple moving average (MA) filter.

3 SIMULATIONS OF DISTILLATION DYNAMICS

3.1 Distillation system

The SSDR and MWDR filters were implemented in the control of a binary (benzene/toluene) distillation column by computer simulation. The schematic diagram of the column is shown in Figure 2, and the geometric parameters of the column are summarized in Table I.

Table I Distillation column geometric parameters.

Parameter	Value	Parameter	Value
Column diameter	1.016 m	Weir height	0.051 m
Tray type	Sieve	Downcomer width	0.132 m
Total number of trays	21	Hole diameter	4.8 mm
Tray spacing	0.457 m	Hole pitch	14 mm
Tray bubbling area	0.68 m^2	Fractional hole area	0.1
Downcomer area	0.065 m^2	Reflux drum diameter	1.016 m
Weir length	0.681 m	Column base diameter	1.016 m

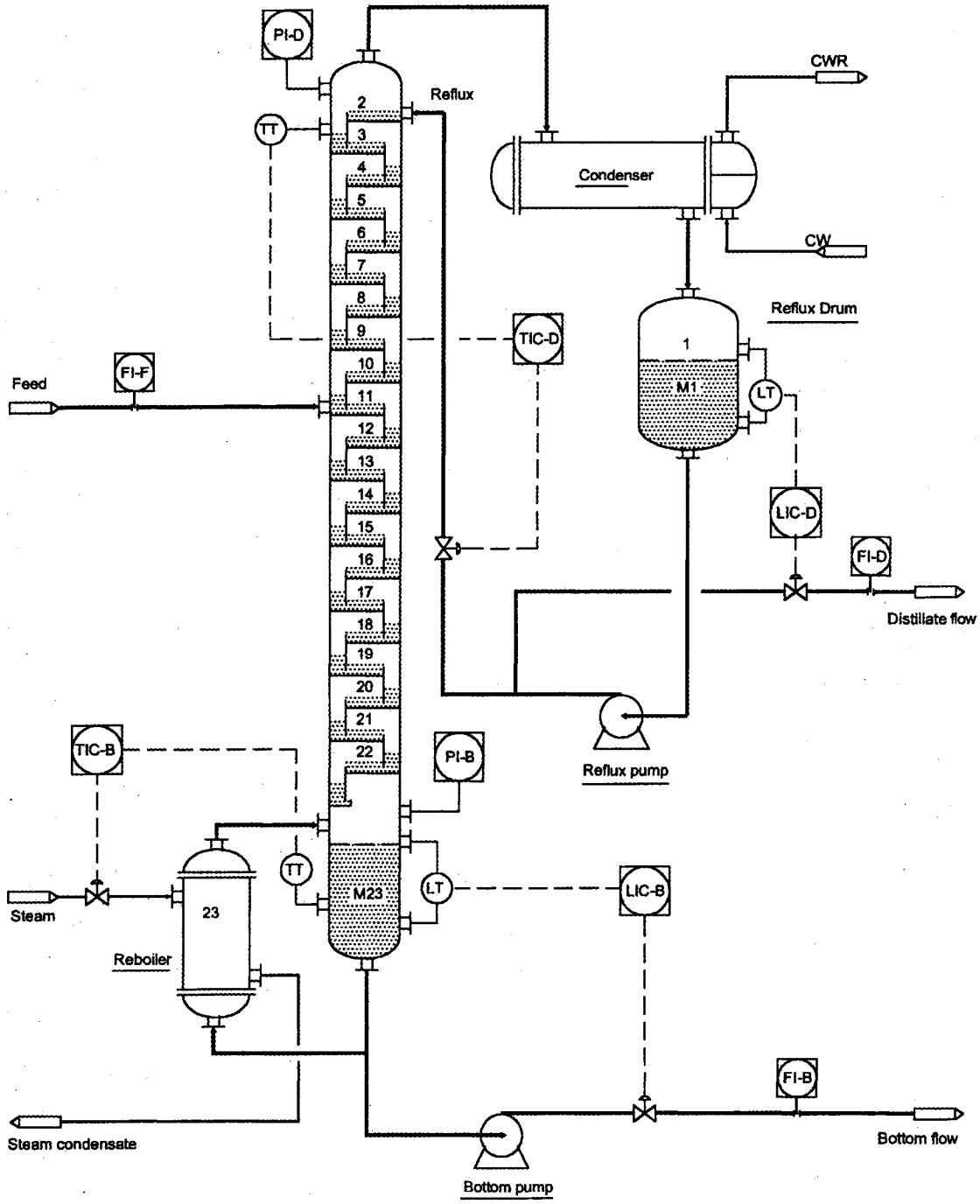


Figure 2 Schematic diagram of benzene-toluene distillation column.

The control configuration of the distillation column consists of four control loops. Two inferential feedback PI controllers, TIC-D and TIC-B, are used to control the top and bottom compositions of benzene by manipulating the reflux flow rate and the reboiler

heat duty, respectively. Two other feedback PI controllers, LIC-D and LIC-B, are used to control the reflux drum and column base liquid levels by manipulating the distillate flow and the bottom product flow rates, respectively. Three flow meters, FI-F, FI-D, and FI-B, monitor the feed, distillate, and bottom flow rates, respectively. The pressure sensors, PI-D and PI-B, measure the column top and bottom pressures. The nominal steady-state values for the measurements and their typical noise levels are listed in Table II.

Table II Nominal steady-state values and noise levels of measured variables.

Variable	Unit	Steady-state value	Standard deviation
Feed flow, F	kmol/h	100.0	0.50
Distillate flow, D	kmol/h	29.90	0.50
Bottom flow, B	kmol/h	70.10	0.50
Top temperature, T_D	°C	84.20	0.25
Bottom temperature, T_B	°C	117.4	0.25
Reflux drum level, H_D	m	0.500	0.02
Column base level, H_B	m	0.700	0.02
Top pressure, P_D	kPa	111.2	0.50
Bottom pressure, P_B	kPa	125.6	0.50

Mathematical models for the distillation column consist of a set of differential-algebraic equations (DAEs) that describe the transient behavior of the column (Gani et al., 1986). These models include relationships describing mass and heat balances, vapor-liquid phase equilibria, tray hydraulics, physical properties, and controller algorithms. The dynamic simulator of the column was based on the following assumptions:

- The liquid holdups on trays, in the downcomers, reflux drum, and column base were considered to be well-mixed, continuous, stirred tanks.
- The vapor hold up was neglected.
- The tray efficiency was assumed to be 100%.
- The dynamics of the condenser, reboiler, measuring devices, and control valves were negligible compared with the dynamics of the distillation column.

The mass and heat balances for each stage (trays, condenser and reflux drum, reboiler and column base) were ordinary differential equations (ODEs), while the other phenomenological equations (e.g., phase equilibria and tray hydraulics) were algebraic equations. The PI control algorithms were in difference form since the controllers were operated at discrete time. This set of mathematical models of the distillation column formed an ensemble of coupled DAEs that needed to be solved as a function of time. An explicit Euler's method was used to integrate the ODEs (Pantelides and Barton, 1993). The simulator was coded in FORTRAN.

3.2 Closed-loop responses without measurement noise

For each pair of controlled and manipulated variables, the open-loop response of the controlled variable to a step change in the manipulated variable, was first simulated to obtain the process reaction curve. The open-loop responses were approximated by either first-order-plus-dead-time, or pure-integrator-plus-dead-time models. The discrete sampling time, Δt, was set to 30 seconds for all measurements. Ziegler-Nichols tuning rules (Ogunnaike and Ray, 1994) were used to determine initial estimates of the discrete PI controller parameters, K_C and τ_I (see Table III). The performance of the controllers was tested for disturbance regulation with the initial Z-N tuning parameters and without measurement noise using a 20% step increase in the feed flow rate. Results of this simulation are presented in Figure 3. These results illustrate that the controllers have good performance in the absence of measurement noise.

Table III Controller parameters.

Controller	LIC-D		TIC-D		LIC-B		TIC-B	
	K_C kmol.h^{-1}.m^{-1}	τ_I min	K_C kmol.h^{-1}.°C^{-1}	τ_I min	K_C kmol.h^{-1}.m^{-1}	τ_I min	K_C MJ.h^{-1}.°C^{-1}	τ_I min
Initial Z-N values	-960.0	46.5	-87.0	48.3	-960.0	46.5	850.7	71.6
Detuned values	-24.7	5.0	-1.8	5.0	-45.5	5.0	40.8	5.0

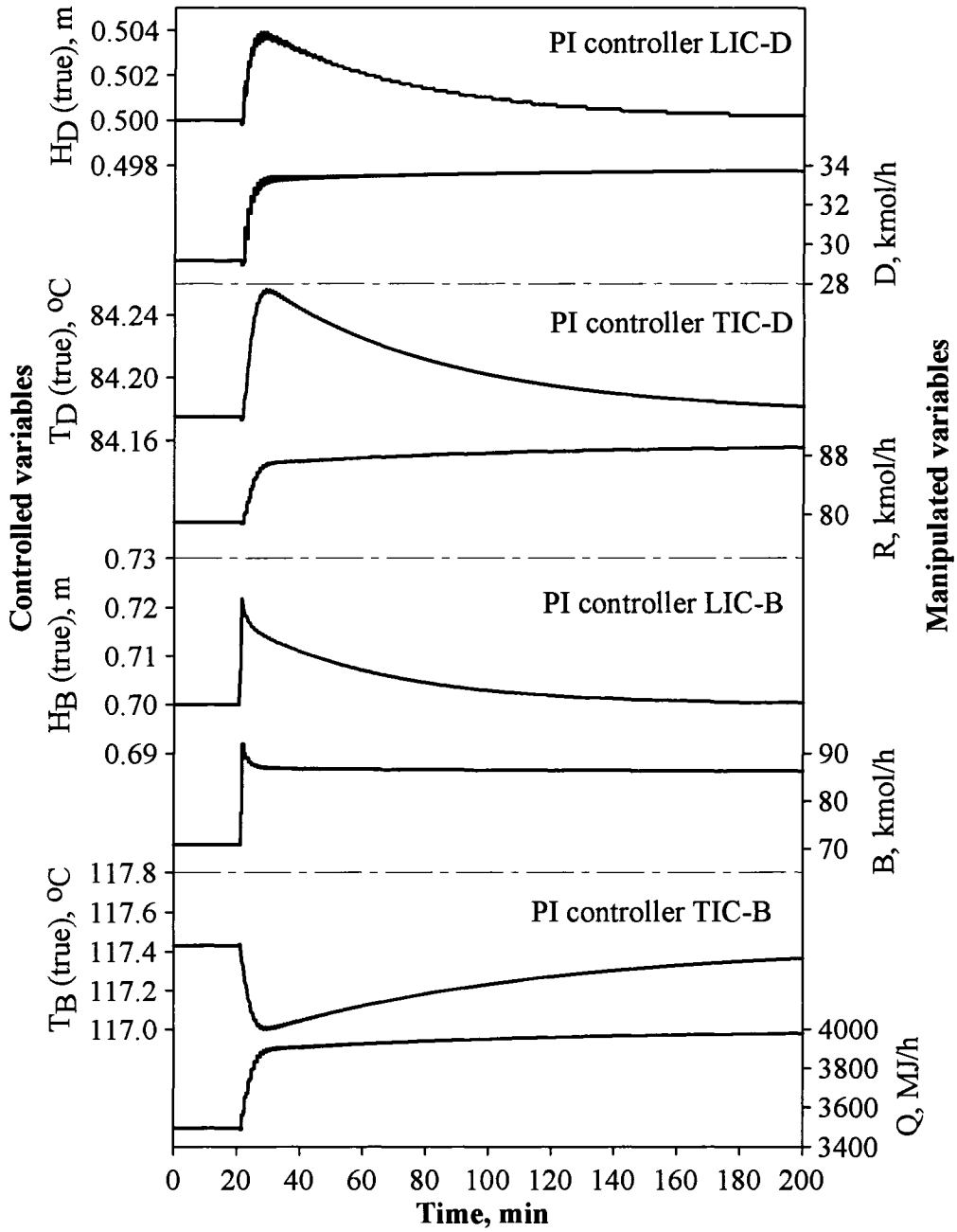


Figure 3 Closed-loop responses with initial Z-N tuning parameters without measurement noise for a 20% step increase in feed flow rate at t = 20 min.

3.3 Closed-loop responses with measurement noise

In real plants all measurements are inevitably corrupted by noise. Using the standard deviations given in Table II, Gaussian white noise was added to the true values of the process variables to provide more realistic measured values. Subsequently, these

noisy measured values were used by the controllers to generate closed-loop responses. Results for the same feed flow disturbance are presented in Figure 4. The presence of the measurement noise significantly reduced the performance of all control loops. The dynamic responses of the system were characterized by large high-frequency, often saturated oscillations of manipulated variables that completely masked the expected responses following the feed flow rate disturbance.

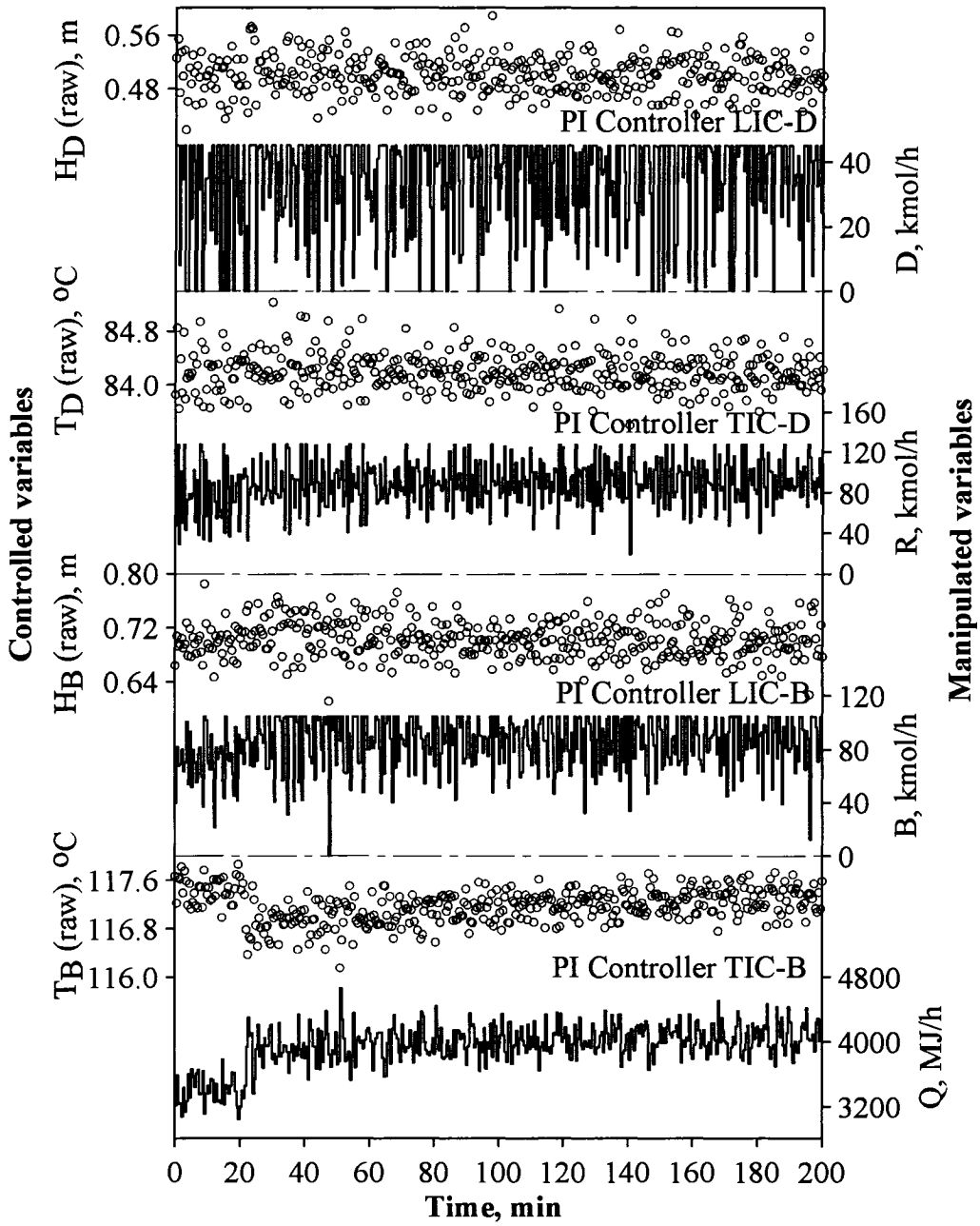


Figure 4 Closed-loop responses with initial Z-N tuning parameters with measurement noise levels listed in Table II, for a 20% step increase in feed flow rate at t = 20 min.

To regain an acceptable system performance, the four controllers were retuned using noisy measurements having standard deviations given in Table II. For this multiple-input multiple-output (MIMO) discrete PI control system, the tuning objective function was formulated as

$$\text{Minimize } J(\mathbf{K_C}, \tau_I) = \sum_{i=1}^{K} \left\{ W_i \sum_{t=1}^{t_s} (y^*_{t,i} - y_{t,i})^2 \Delta t + w_i \sum_{t=1}^{t_s} (u_{t,i} - u_{t-1,i})^2 \Delta t \right\} \qquad (5)$$

In Equation (5), $J(\mathbf{K_C}, \tau_I)$ is defined as the cost function of the control system, $\mathbf{K_C}$ is the vector of controller gains, τ_I is the vector of controller integral times, K is the total number of control loops. $y_{t,i}$ is the true value of controlled variable of controller i at sampling time t, $y^*_{t,i}$ is the setpoint for controller i, $u_{t,i}$ is the value of the manipulated variable for controller i at sampling time t. W_i is the weighting factor for the integral of the squared error (ISE) of the controlled variable of controller i, w_i is the weighting factor for the integral of the squared differences (ISDU) of the values of manipulated variable between sampling times t and t-1 for controller i. t_s represents a time sufficiently long to approach steady state.

It should be noted that, in order to bring the scales of the ISE and ISDU components in the objective function (5) to similar levels, the ISE and ISDU for each controller were scaled via division by their initial values (ISE_0 and $ISDU_0$) obtained using the initial Z-N tuned controllers, before starting minimization of the objective function. The values of the weights, W_i and w_i, for each controller were set at 0.15 and 0.1, respectively. The optimization was performed in two steps: a rough grid search was carried out to identify the optimal region, followed by a rigorous optimization using a quasi-Newton method with lower and upper bounds.

Values of new optimal controller parameters are presented in Table III, along with the original Z-N values. The integral times, obtained when the measurement noise was considered, hit their lower bounds, which were set at 5 minutes. Meanwhile, the controller gains were reduced significantly. The controller gains were, on average, thirty

times smaller whereas the integral times were, on average, ten times smaller. Closed-loop responses with the new controller parameters are presented in Figure 5. The oscillations of the manipulated variables were significantly reduced compared to those in Figure 4. The dynamics of the controlled variables could be observed despite the noisy measurements, but at the expense of more sluggish control that resulted in larger deviations from setpoint during the transient period.

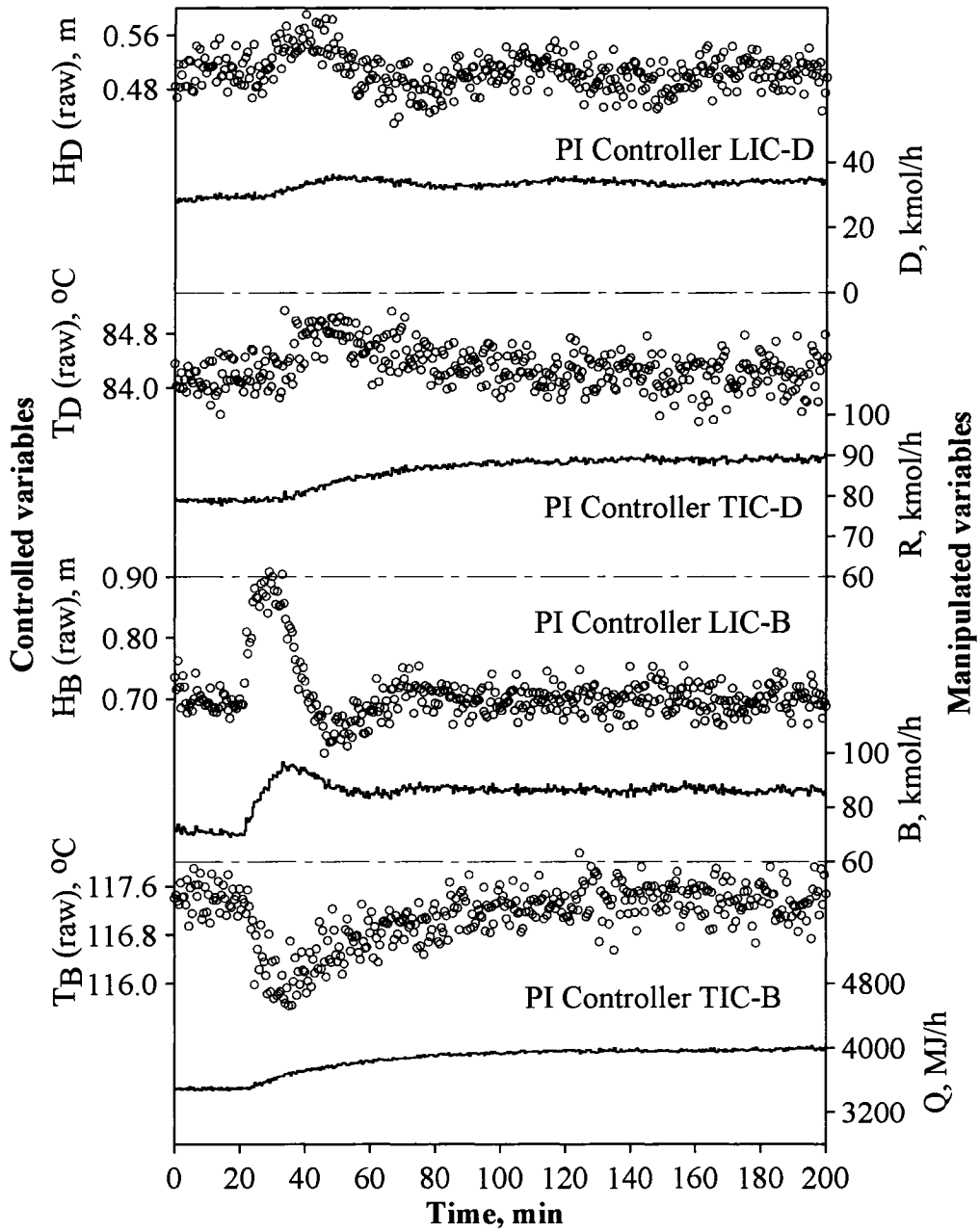


Figure 5 Closed-loop responses with detuned controllers with measurement noise level listed in Table II, for a 20% step increase in feed flow rate at t = 20 min.

The values of ISE, ISDU and cost function for each controller evaluated when the controllers, tuned with Z-N, were used without and with measurement noise are presented as Cases A and B of Table IV, whereas the values obtained with the detuned controllers, considering measurement noise, are presented as Case C of Table IV. As expected, these results showed that the presence of measurement noise greatly enlarged the ISEs and particularly the ISDUs, resulting in very large values of the cost function. When the detuned controllers were used, the ISDUs were reduced by about three orders of magnitude, the ISEs for the two top control loops were reduced, and ISEs for the base control loops were increased. The overall cost function with the detuned controllers decreased from 248 to 0.391.

The effects of the variations of the manipulated variables on the true values of process variables were investigated. Figure 6 shows the true values of the four controlled variables at the nominal process steady state when the four control loops were closed with noisy measurements but without external disturbances. The resulting variances of the true values of controlled variables were significantly magnified due to the large changes in the manipulated variables with the initial Z-N tuning parameters. On the other hand, using the detuned controllers, the variances of the true values of the controlled variables were relatively small due to smaller changes in the manipulated variables. The calculated variances of the controlled variables with the initial Z-N tuning parameters were, on average, 20 times greater than those with the detuned controllers. It is obviously important to attenuate unnecessary variations of the manipulated variables caused by process noise in order to attain good controller performance.

Table IV Performance of controllers without and with data reconciliation filters ($J_i = W_i \times ISE_i + w_i \times ISDU_i$)

Case	DR Filter	LIC-D (W=0.15, w=0.1)			TIC-D (W=0.15, w=0.1)			LIC-B (W=0.15, w=0.1)			TIC-B (W=0.15, w=0.1)			$\sum_{k=1}^{4} J_k$
		ISE_1	$ISDU_1$	J_1	ISE_2	$ISDU_2$	J_2	ISE_3	$ISDU_3$	J_3	ISE_4	$ISDU_4$	J_4	
A	No	0.002	0.004	0.001	0.015	0.011	0.003	0.005	0.086	0.009	0.285	0.057	0.048	0.061
B	No	0.623	583	58.4	0.889	1385	138.6	0.178	399	39.9	0.366	111	11.1	248
C	No	0.212	0.238	0.056	0.349	0.198	0.072	0.354	0.482	0.101	0.888	0.287	0.162	0.391
D	SSDR	0.243	0.038	0.040	0.355	0.201	0.073	0.376	0.200	0.076	0.891	0.277	0.161	0.350
E	MWDR	0.293	0.004	0.044	0.373	0.015	0.057	0.469	0.027	0.073	0.910	0.021	0.139	0.313
F	MWDR	0.160	0.028	0.027	0.172	0.093	0.035	0.102	0.136	0.029	0.330	0.123	0.062	0.153

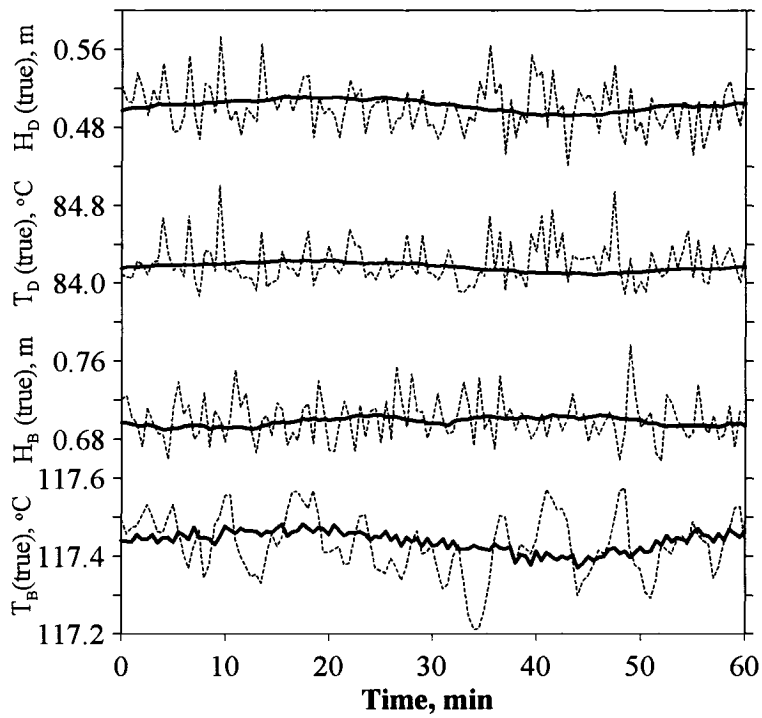


Figure 6 True values of controlled variables at process nominal steady-state conditions. - - - - - : with initial Z-N tuning controllers; ——: with detuned controllers.

4 IMPLEMENTATION OF DATA RECONCILIATION FILTERS

To attenuate the propagation of the measurement noise, the SSDR and the MWDR filters formulated by Equations (3) and (4) were implemented within the feedback control loops in the distillation column. For this distillation column, the measurement vector is denoted by $\mathbf{y}_t = [F_t, D_t, B_t, T_{D,t}, T_{B,t}, H_{D,t}, H_{B,t}, P_{D,t}, P_{B,t}]^T$, and the vector of model residuals is given by

$$\mathbf{f}[\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t] = \begin{bmatrix} \hat{F}_t - \hat{D}_t - \hat{B}_t - \dfrac{(\hat{H}_{D,t} - \hat{H}_{D,t-1})A_D\rho_D}{\Delta t} - \dfrac{(\hat{H}_{B,t} - \hat{H}_{B,t-1})A_B\rho_B}{\Delta t} \\ \hat{F}_t x_F - \hat{D}_t \hat{x}_{D,t} - \hat{B}_t \hat{x}_{B,t} - \dfrac{(\hat{H}_{D,t}\hat{x}_{D,t} - \hat{H}_{D,t-1}\hat{x}_{D,t-1})A_D\rho_D}{\Delta t} \\ \qquad\qquad \cdots\cdots - \dfrac{(\hat{H}_{B,t}\hat{x}_{B,t} - \hat{H}_{B,t-1}\hat{x}_{B,t-1})A_B\rho_B}{\Delta t} \\ \dfrac{\hat{P}_{D,t}\hat{x}_{D,t}}{10^{(A_1 - \frac{B_1}{\hat{T}_{D,t}+C_1})}} + \dfrac{\hat{P}_{D,t}(1-\hat{x}_{D,t})}{10^{(A_2 - \frac{B_2}{\hat{T}_{D,t}+C_2})}} - 1 \\ \dfrac{10^{(A_1 - \frac{B_1}{\hat{T}_{B,t}+C_1})}\hat{x}_{B,t}}{\hat{P}_{B,t}} + \dfrac{10^{(A_2 - \frac{B_2}{\hat{T}_{B,t}+C_2})}(1-\hat{x}_{B,t})}{\hat{P}_{B,t}} - 1 \end{bmatrix} \tag{6}$$

The first two rows of the model residuals are the equations of conservation of total mass and of one of the components, benzene, respectively. The assumption of process stationarity was partly relaxed by adding accumulation terms for both the reflux drum and the column base. On the other hand, the mass and component accumulations on each tray were neglected. A_D and A_B are the cross-sectional areas of the reflux drum and the column base. ρ_D and ρ_B are the top and bottom liquid densities, which are assumed to be constant, and x_F is the feed composition. The third row is the dew point equation of the vapor leaving tray 2 employed to estimate the unmeasured variable $x_{D,t}$ using the measurements $P_{D,t}$ and $T_{D,t}$. The fourth row is the bubble point equation for the bottom liquid, used to estimate the unmeasured variable $x_{B,t}$ using measurements $P_{B,t}$ and $T_{B,t}$. Parameters A_j, B_j, and C_j are Antoine constants.

The random error of the four models was assumed to be uncorrelated and the variances of the individual model errors were calculated by Monte Carlo simulation. The minimization of the objective function for the data reconciliation filters was implemented using a quasi-Newton method with lower and upper bounds. The filtered data vector for the raw measurements obtained by optimization is denoted as $\hat{\mathbf{y}}_t = [\hat{F}_t, \hat{D}_t, \hat{B}_t, \hat{T}_{D,t}, \hat{T}_{B,t}, \hat{H}_{D,t}, \hat{H}_{B,t}, \hat{P}_{D,t}, \hat{P}_{B,t}]^T$, where the reconciled values of the

controlled variables, $\hat{H}_{D,t}$, $\hat{T}_{D,t}$, $\hat{H}_{B,t}$ and $\hat{T}_{B,t}$, were used by the four controllers to calculate the manipulated variables. The auxiliary vector of the estimates of unmeasured variables is $\hat{\mathbf{z}}_t = [\hat{x}_{D,t}, \hat{x}_{B,t}]^T$.

The performances of the SSDR filter and the MWDR filter, having window width four, were evaluated for a 20% step increase of the feed flow rate at time $t = 30$ min. Results are presented in Figures 7 and 8, respectively.

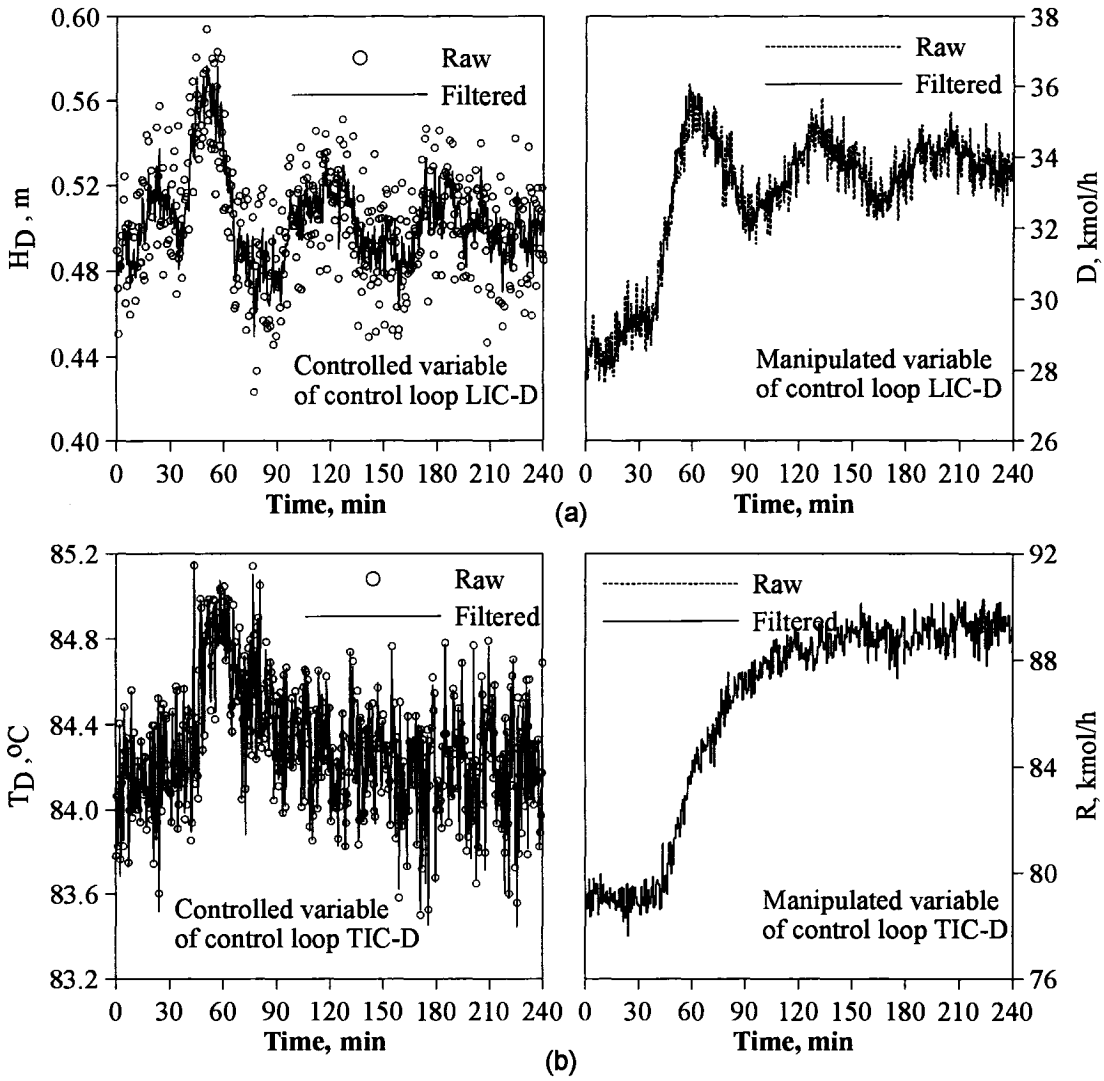


Figure 7 Responses for a 20% step increase in the feed flow rate at t = 30 min with a SSDR filter. (a) reflux drum level control loop; (b) top temperature control loop.

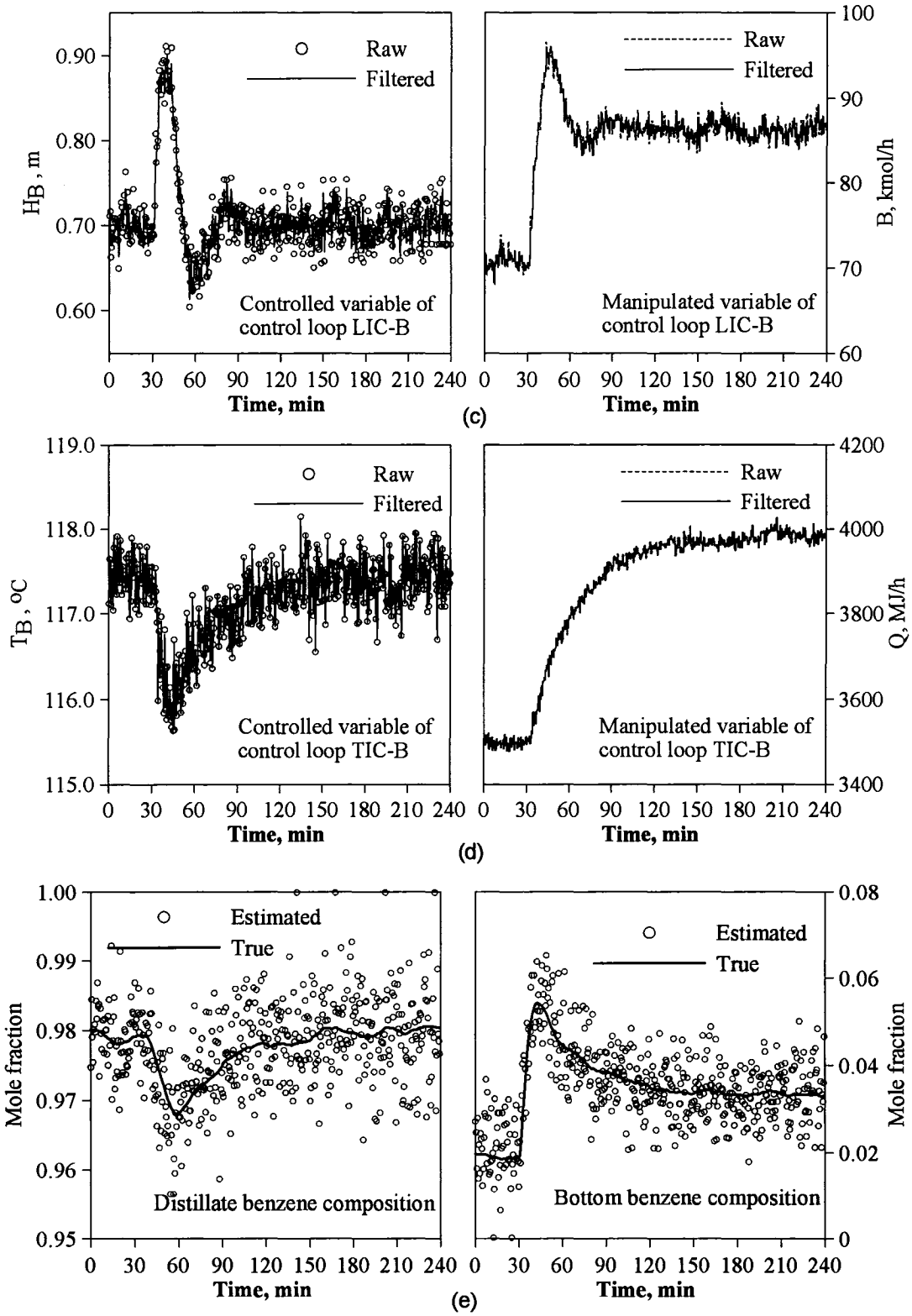


Figure 7 (*continued*) Responses for a 20% step increase in the feed flow rate at t = 30 min with a SSDR filter. (c) column base level control loop; (d) bottom temperature control loop; (e) estimation of the distillate and bottom benzene compositions.

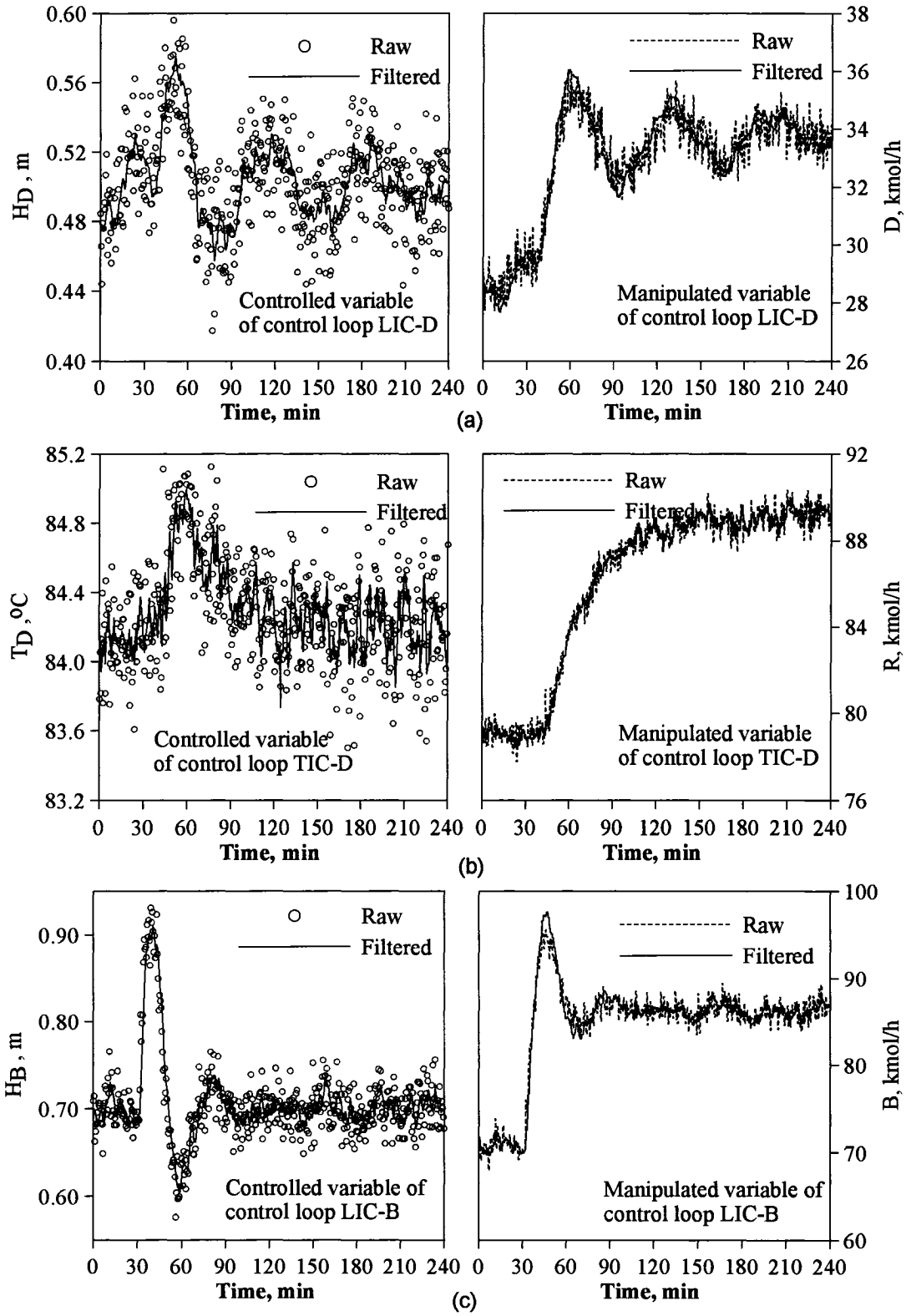


Figure 8 Responses for a 20% step increase in the feed flow rate at t = 30 min with a MWDR filter. (a) reflux drum level control loop; (b) top temperature control loop; (c) column base level control loop.

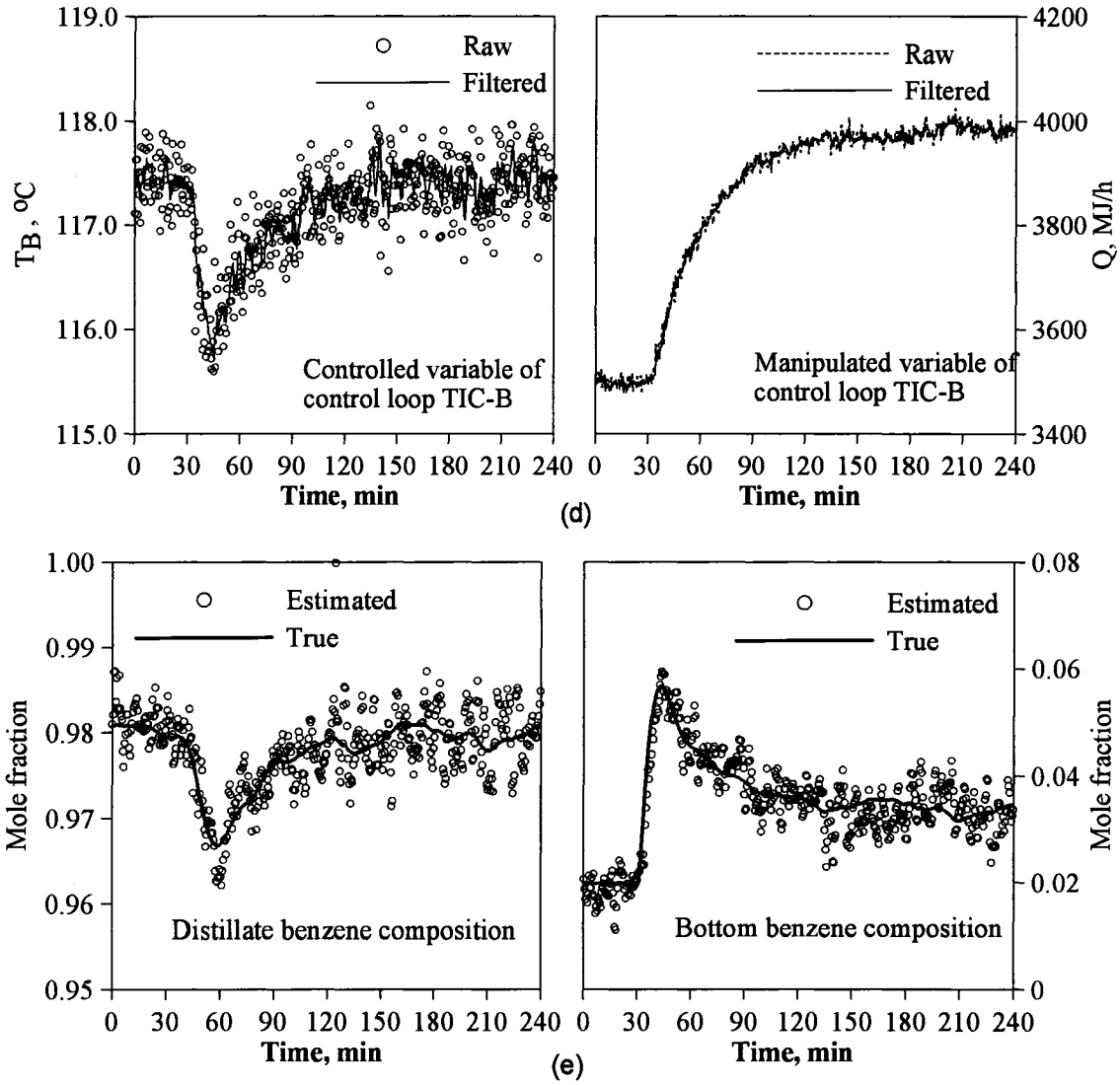


Figure 8 (*continued*) Responses for a 20% step increase in the feed flow rate at t = 30 min with a MWDR filter. (d) bottom temperature control loop; (e) estimation of the distillate and bottom benzene compositions.

It should be noted that, in Figures 7(e) and 8(e) some values of the estimated composition reached their upper or lower bound and were consequently set to 1.0 and 0 respectively. The SSDR filter was effective in reducing the noise propagation for the two liquid level control loops, whereas its filtering effects on the two temperature control loops were hardly noticeable. These results indicated that the mass balances played the dominant role in the data reconciliation filter with a secondary role for the thermodynamic models. The benzene compositions estimated by the SSDR filter are unbiased; however they are still

relatively noisy. On the other hand, the MWDR filter was effective for all control loops, and the noise reductions through the control loops were significant. The estimates of the benzene compositions by the MWDR filter were much less noisy than the estimates obtained by the SSDR filter. The overall performance of MWDR filter was better than that of the SSDR filter. This is because the MWDR filter used several past measurements which provided more information for the filter. This can also be interpreted in terms of the improvement in precision of the measured values as result of averaging the past several measurements.

Values of ISE and ISDU calculated for each control loop, along with the evaluations of the cost function for each controller with the DR filters are presented as Cases D and E of Table IV, respectively. The values of ISDU using the SSDR filter was reduced by approximately one order of magnitude for LIC-D, and 50% for LIC-B; for TIC-D and TIC-B, there were no significant differences in the ISDU. The values of ISDU using the MWDR filter for all loops were decreased by at least one order of magnitude. On the other hand, the values of ISE were increased for all loops using either the SSDR or MWDR filters. The cost function of the controller decreased due to the significant reduction in ISDUfor all cases except for the temperature loops using the SSDR filter for which the values of the cost function were essentially unchanged. The overall cost function of the control system was reduced from 0.39 to 0.35 when the SSDR filter was used, and to 0.31 with the MWDR filter.

Due to the presence of measurement noise, it was necessary to considerably detune all control loops to obtain acceptable performance. Since data reconciliation filters were able to reduce the effect of noise, more aggressive controllers could therefore be used. To evaluate the performance of the control loops using a MWDR filter with larger controller gains, the process was subjected to a 20% step increase of the feed flow rate. Results are presented in Figure 9 where the values of ISE and ISDU of each control loop are plotted as a function of the controller gain ratio K_C/K_{C0}. K_{C0} corresponds to the optimal detuned value of the gain listed in Table III when process noise was considered. The gains of all controllers were incremented simultaneously. Figure 9 illustrates that the ISE decreased

significantly when the controller gains were increased by a factor of 3. However, for higher values of the gain ratios ($3.0<K_C/K_{C0}<7.0$), the change of the ISE values for the three controllers, TIC-D, LIC-B and TIC-B, remained relatively small, but the ISE of the controller LIC-D showed a larger increment. Eventually, for large enough values of the ratio K_C/K_{C0}, the ISEs for all the other control loops would be expected to increase significantly. In this situation, the ISE for LIC-D increased very rapidly with the increase of the gain ratio because of its greater sensitivity to the manipulations.

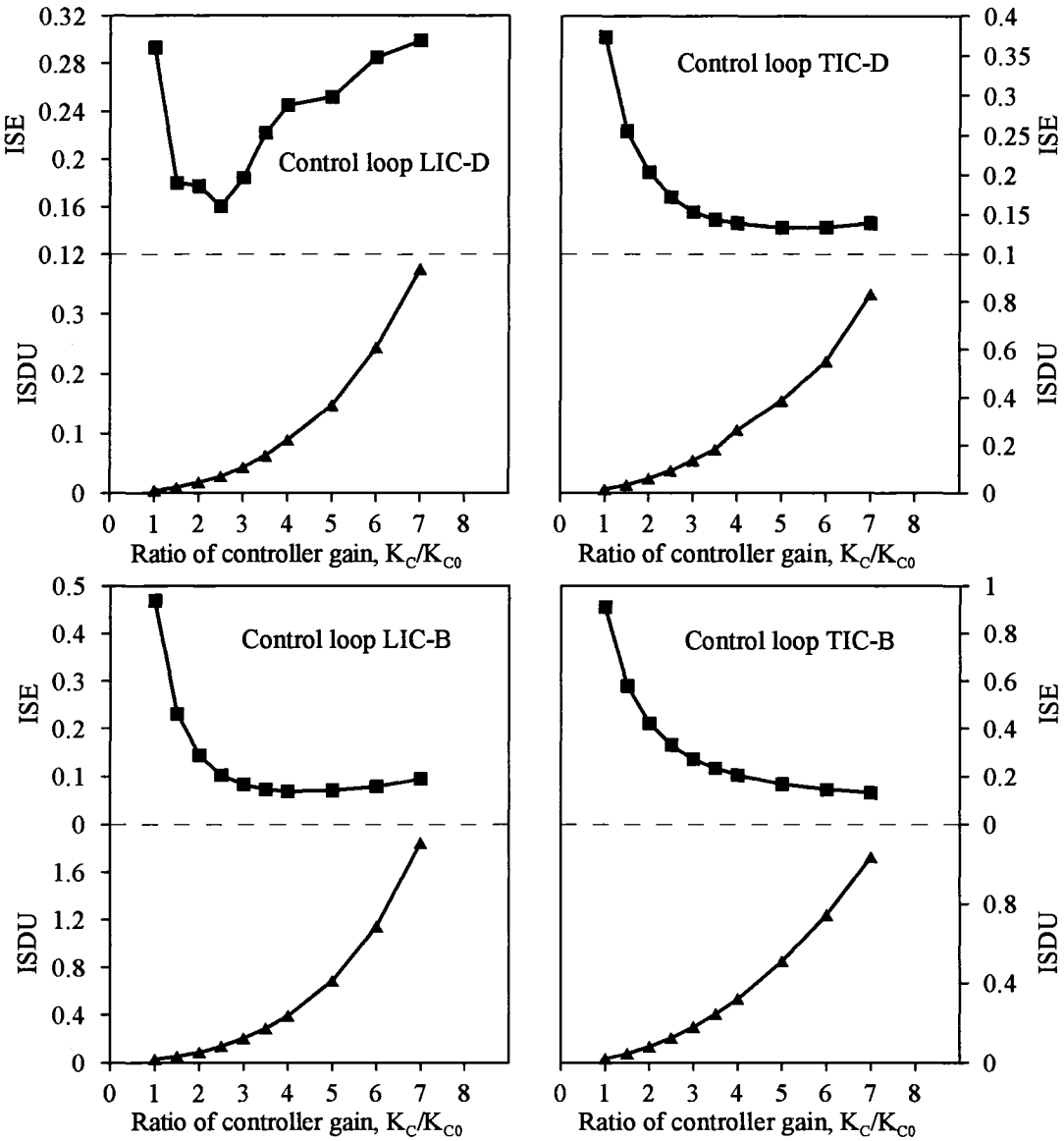


Figure 9 Performance of control loops with MWDR filter with higher controller gains (K_{C0}: detuned controller gains listed in Table III).

The cost function for each control loop with the increase of K_C/K_{C0} was also calculated, and the results are presented in Figure 10. For the control loop TIC-B, a minimum value of the cost function was obtained at $K_C/K_{C0} = 3.0$. For other control loops, the minimum values of their cost function were obtained when $K_C/K_{C0} = 2.5$. Using the new optimal controller gains, the values of ISE, ISDU, and cost function for each control loop were evaluated and the results are listed in Case F of Table IV. The overall cost function of the control system was reduced from 0.313 to 0.153 when the new optimal controller gains were used with the embedded MWDR filter.

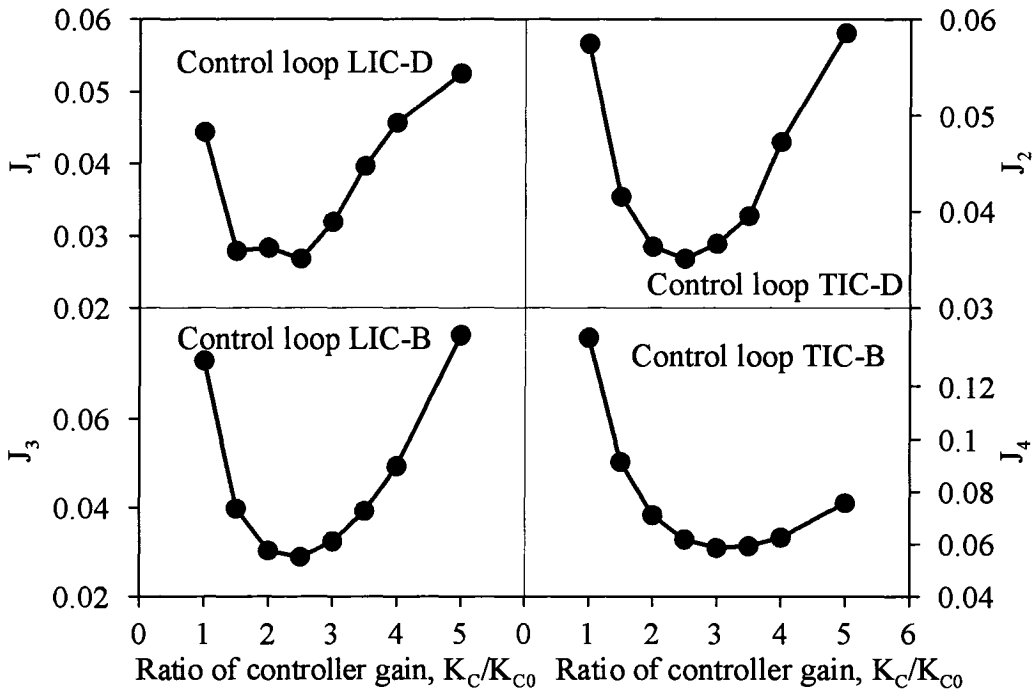


Figure 10 Cost function of control loop with MWDR filter with higher controller gains (K_{C0}: detuned controller gains listed in Table III).

Data reconciliation filters have been shown to significantly attenuate the impact of measurement noise, thereby enhancing the performance of controllers. However, as indicated in Figure 7 and Table IV, the SSDR filter did not perform as well as the MWDR filter. In order to demonstrate how the level of knowledge affects the performance of a SSDR filter for the distillation column, the problem was slightly modified. First, two additional variables were measured; the top and bottom benzene compositions were assumed to be measured without time delay and with standard

deviation of 0.0005 mole fraction. Second, it was assumed that the two pressures were measured more precisely with one-tenth of the standard deviation listed in Table II. Third, the SSDR filter only used pressure, temperature and composition measurements ($\mathbf{y}_t = [x_{D,t}, x_{B,t}, T_{D,t}, T_{B,t}, P_{D,t}, P_{B,t}]^T$) and two model residuals consisting of the two thermodynamic models of Equation (6). The performance of this reduced SSDR filter was tested for a 20% step increase of feed flow at time t = 30 min. The resulting raw measurements and filtered values of the compositions, temperatures and pressures are presented in Figure 11. These results show that significant filtering occurred for the two temperatures, whereas, the filtering of the compositions and pressures was hardly observable (i.e., the raw and filtered values are indistinguishable in the figure). This was because the temperatures had relatively larger errors compared to those of the pressures and compositions. The high precision of measurements of the compositions and pressures resulted in a much heavier weight on correcting the temperatures.

The performance of the reduced SSDR filter in attenuating the measurement noise of the two temperatures was compared to that of an exponentially weighted moving average (EWMA) filter,

$$\hat{T}_t = \theta \hat{T}_{t-1} + (1-\theta) T_t \tag{7}$$

and a moving average (MA) filter

$$\hat{T}_t = \frac{1}{L} \sum_{k=0}^{L-1} T_{t-k} \tag{8}$$

for the same 20% step increase of feed flow. The mean squared error (MSE) defined as

$$MSE = \frac{1}{n-1} \sum_{t=1}^{n} \left(T_t - T_t^{true} \right)^2 \tag{9}$$

for the raw measurements and with T_t replaced by $\hat{T}_t$ for the filtered data are presented in Figure 12. These results indicated that the performance for the SSDR filter for reconciling the temperature was clearly superior to that of EWMA and MA filters.

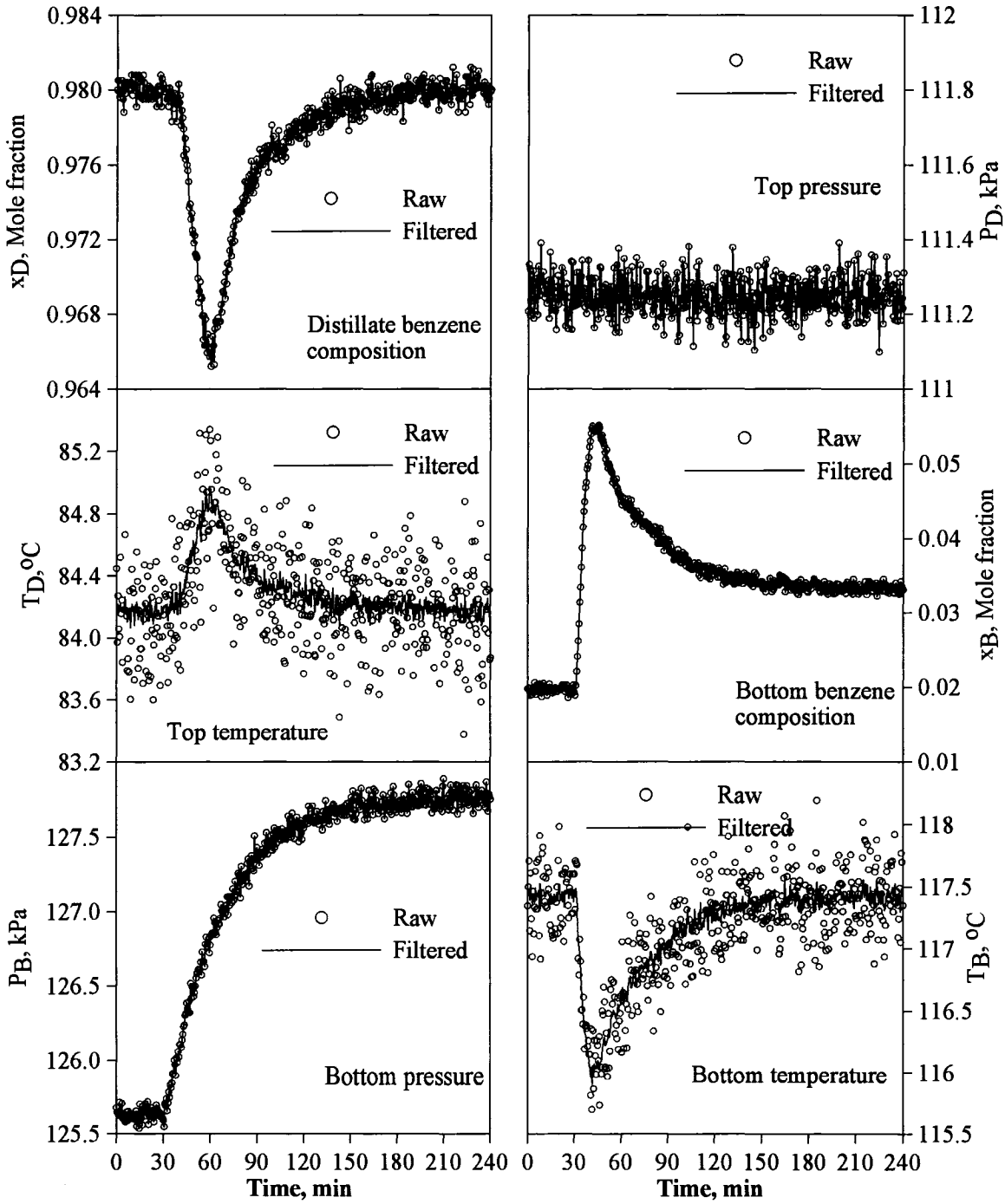


Figure 11 Performance of the reduced SSDR filter for a 20% step increase in feed flow rate at t = 30 min.

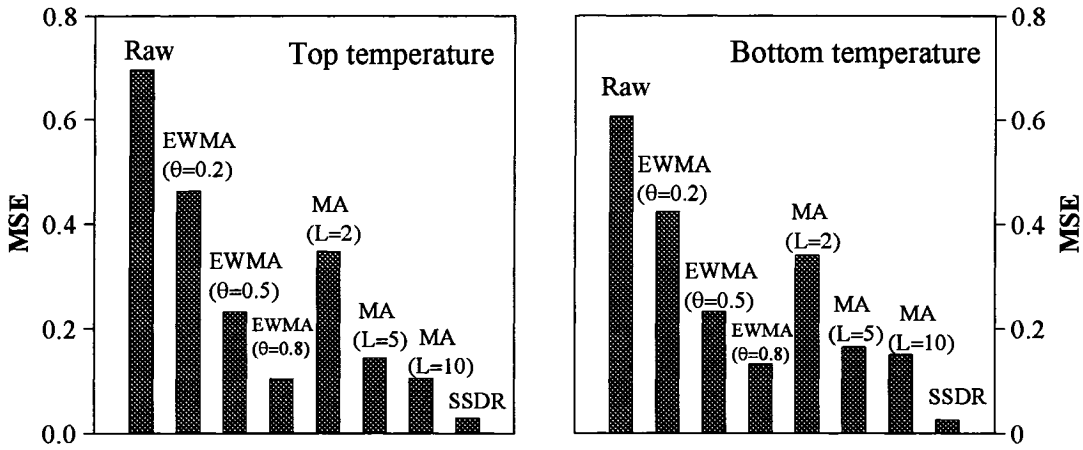


Figure 12 Comparison of performance of the reduced SSDR filter with EWMA and MA filters during process transient condition.

The closed-loop performance of the reduced SSDR filter to deal with biased measurements and autocorrelated measurements was also studied. Figure 13 shows the results when the top temperature measurements are biased by 1.0 °C, and the measurements of the bottom temperature are highly autocorrelated and given by

$$T_{B,t} = 0.9T_{B,t-1} + a_t \qquad (10)$$

where a_t is white noise. The SSDR filter effectively compensated for the measurement bias as the filtered values for the top temperature were much closer to the true values than to the raw measurements. In addition, despite the strong autocorrelations of the measurements of the bottom temperature, the SSDR filter displayed good performance in tracking the true values of the controlled variable.

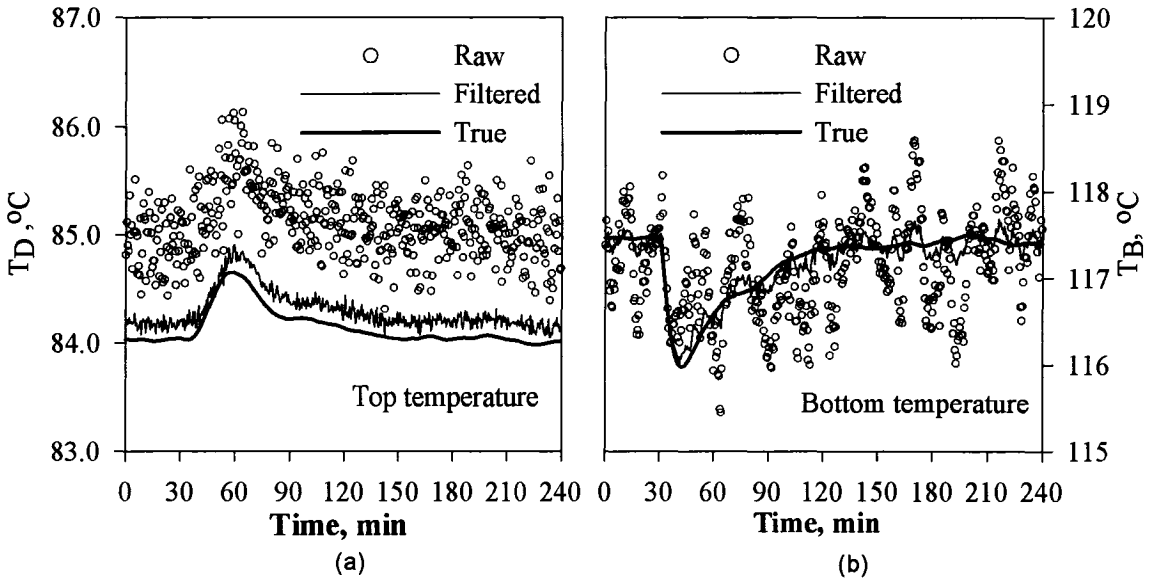


Figure 13 Performance of the reduced SSDR filter in dealing with (a) biased and (b) autocorrelated measurements.

5 CONCLUSION

The magnitude of measurement noise must be considered when designing controllers using models of the process. Otherwise, the controllers are too aggressive, resulting in excessive variations in the controlled and manipulated variables. On the other hand, the detuning of controllers due to the presence of noise leads to more sluggish responses and lower controller performance. In this work, two data reconciliation filters were developed: the SSDR filter, which uses only the current set of measurements and process models, and the MWDR filter, which uses a series of past measurements and process models to further decrease the effect of noise. The DR filters allowed the implementation of more aggressive controllers, thereby yielding better overall performance of all control loops. The DR filters also provided estimates for unmeasured process variables. The degree of knowledge available in the process measurements and models is an important factor for good performance of a DR filter. With sufficient knowledge, a SSDR filter was shown to perform better than classical EWMA and MA filters. The SSDR filter was also shown to be effective in dealing with biased and autocorrelated measurements.

NOMENCLATURE

A:	Cross-sectional area of the column, m^2
A_j, B_j, C_j:	Antoine constants
B:	Distillation bottom flow rate, kmol/h
C:	Number of process models
D:	Distillate flow rate, kmol/h
E[**y**]:	M×1 vector of expected values of process variables
f:	C×1 model constraint vector
F:	Feed flow rate, kmol/h
H:	Liquid level, m
$\mathbf{K}_C$:	Vector of controller gains
K:	Total number of control loops
L:	Window width
M:	Number of measured variables
N:	Number of unmeasured variables
P:	Pressure, kPa
Q:	Reboiler heat duty, MJ/h
R:	Reflux flow rate, kmol/h
Δt:	Discrete sampling time interval, s
T:	Temperature, °C
u:	Manipulated variable
V:	M×M covariance matrix of the measurements
W:	Weighting factor for ISE
w:	Weighting factor for ISDU
x:	Composition, mole fraction
y:	M×1 vector of raw measurements
z:	N×1 vector of estimates for unmeasured variables

Greek letters

ε:	M×1vector of random variables
σ^2:	Variance
δ:	C×1 vector of model random error

Ω: C×C covariance matrix of model error

τ_I: Vector of integral times for each controller

ρ: Liquid density, kg/m^3

θ: Parameter in EWMA filter

Superscripts and Subscripts

B: Column bottom

D: Distillate

F: Feed

l: Lower bound

t: Time instant

u: Upper bound

$\wedge$: Estimate

*: Setpoint

Acronyms

DAEs: Differential-algebraic equations

EWMA: Exponentially weighted moving average

ISE: Integral of squared errors for controlled variable

ISDU: Integral of squared differences of manipulated variable

MIMO: Multiple-input multiple-output

MA: Moving average

MWDR: Moving window data reconciliation

MPC: Model predictive control

MSE: Mean squared errors

ODEs: Ordinary differential equations

SSDR: Single set data reconciliation

REFERENCES

Abu-el-zeet, Z.H., Roberts P.D. and Becerra V.M. (2002). "Enhancing model predictive control by data reconciliation", *AIChE J.*, **48**, 324-333.

Bussani, G., Chiari M., Grottoli M.G., Pierucci S., Faravelli T., Ricci G. and Gioventu, G. (1995). "Application of data reconciliation and optimization procedure to hydrogen plant", *Computers Chem. Engng*, **19**, Suppl. S299-S304.

Chiari, M., Bussani G., Grottoli M.G. and Pierucci S. (1997). "On-line data reconciliation and optimization: Refinery applications", *Computers Chem. Engng*, **21**, Suppl. S1185-S1190.

Gani, R., Ruiz C.A. and Cameron I.T. (1986). "A generalized model for distillation columns – I: model description and applications", *Computers Chem. Engng*, **10**, 181-198.

Li, B. Chen B., Jiang J. and Cong S. (2001). "Steady-state online data reconciliation in a crude oil distillation unit", *Hydrocarbon Processing*, **80**(3), 61-64.

McBrayer, K.F., Soderstrom T.A., Edgar T.F. and Young R.E. (1998). "The application of nonlinear dynamic data reconciliation to plant data", *Computers Chem. Engng*, **22**, 1907-1911.

Narasimhan, S. and Jordache C. (2000). "*Data reconciliation & gross error detection, an intelligent use of process data*", Gulf Publishing Company, Houston.

Ogunnaike, B.A. and Ray W.H. (1994). "*Process dynamics, modeling, and control*", Oxford University Press, New York.

Pantelides, C.C., and Barton P.I. (1992). "Equation-oriented dynamic simulation: current status and future perspectives", European Symposium on Computer Aided Process Engineering-2, October 1992, *Computers Chem. Engng*, **17**, Suppl. 1, 263-285.

Pierucci, S., Brandani P., Ranzi E. and Sogaro A. (1996). "An industrial application of an on-line data reconciliation and optimization problem", *Computers Chem. Engng*, **20**, Suppl., S1539-S1544.

Ramamurthi, Y., Sistu P.B. and Bequette B.W. (1993). "Control - relevant dynamic data reconciliation and parameter estimation", *Computers Chem. Engng*, **17**, 41-59.

Soderstrom, T.A., Edgar T.F., Russo L.P. and Young R.E. (2000). "Industrial application of a large-scale dynamic data reconciliation strategy", *Ind. Eng. Chem. Res.* **39**, 1683-1693.

CHAPTER III

Enhancing Controller Performance via Dynamic Data Reconciliation

Shuanghua Bai, David D. McLean and Jules Thibault*

Department of Chemical Engineering

University of Ottawa
Ottawa, Ontario, Canada K1N 6N5

Published in **Canadian Journal of Chemical Engineering, Volume 83, 515-526, 2005.**

* To whom correspondence should be addressed. Tel: 613-562-5800 ext. 6094
Email: thibault@genie.uottawa.ca

Preface

Chapter 3 develops a dynamic data reconciliation (DDR) algorithm where process dynamic models replace the quasi-steady-state models that were used in Chapter 2 in order to improve the performance of data reconciliation for processes having significant dynamics.

ABSTRACT

Measured values of process variables are subject to measurement noise. The presence of measurement noise can result in detuned controllers in order to prevent excessive adjustments of manipulated variables. Digital filters, such as exponentially weighted moving average (EWMA) and moving average (MA) filters, are commonly used to attenuate measurement noise before controllers. In this article, we present another approach, a dynamic data reconciliation (DDR) filter. This filter employs discrete dynamic models that can be phenomenological or empirical, as constraints in reconciling noisy measurements. Simulation results for a storage tank and a distillation column under PI control demonstrate that the DDR filter can significantly reduce propagation of measurement noise inside control loops. It has better performance than the EWMA and MA filters, so that the overall performance of the control system is enhanced.

Keywords: measurement noise, filter, data reconciliation, controller performance

1 INTRODUCTION

Measured values of process variables are subject to measurement noise. When noisy measurements are directly used to calculate control moves, the manipulated variables can display excessive adjustments and controllers have to be detuned for the process to have an acceptable behavior. However, with detuning, processes show more sluggish responses to disturbance and setpoint changes. To remedy this problem, filters, such as exponentially weighted moving average (EWMA) and moving average (MA) filters are usually used inside feedback loops to attenuate the measurement noise before transmission to controllers. In this article, we propose another approach, a dynamic data reconciliation (DDR) filter.

Data reconciliation (DR) is a technique to compensate for measurement errors by using prior knowledge of the process in the form of mathematical models so that more reliable and accurate estimates of the process states are obtained. Data reconciliation techniques for steady-state processes have been well documented by Romagnoli and Sanchez (2000). Only in recent years, has data reconciliation for dynamic processes been investigated (e.g., Liebman et al., 1992; Ramamurthi et al., 1993; Albuquerque and Biegler, 1996; Binder et al., 2002; Abu-el-zeet et al., 2002). In such studies, reconciled data were obtained by minimizing a weighted least-squares objective function, subject to dynamic process models consisting of differential-algebraic equations (DAEs). Most of this work has been devoted to optimal estimation of process states under open-loop conditions. Few papers have assessed controller performance using DDR strategies instead of raw measurements. Ramamurthi et al. (1993) proposed a DDR algorithm that led to better closed-loop performance for a nonlinear predictive controller. Abu-el-zeet et al. (2002) claimed that DDR, in conjunction with systematic bias detection, enhanced a model predictive control scheme. However, the degree of improvement for the controller performance was not specified.

In our work, an algorithm for a DDR filter was embedded in feedback loops to reconcile noisy raw measurements before calculating the control actions and the controller

performance was quatitatively assessed. The formulation of the DDR filter is presented in the next section. Then, implementations of the DDR filter, as an integral part of conventional PI control loops, are demonstrated in simulated storage tank and distillation column processes. Comparisons of the performance of the DDR with EWMA and MA filters are carried out.

2 FORMULATION OF DYNAMIC DATA RECONCILIATION ALGORITHM

The DDR filter formulated in this paper is valid for filtering random measurement errors. The output variables of the process are assumed to be measured without systematic bias and, at sampling time t, to follow the additive noise model,

$$\mathbf{y}_t = \mathbf{x}_t + \boldsymbol{\varepsilon}_t \tag{1}$$

where $\mathbf{y}_t$ is a M×1 vector of measured values of the M output variables, $\mathbf{x}_t$ is a M×1 vector of true values of the M output variables, and $\boldsymbol{\varepsilon}_t$ is a M×1 vector of random variables that are usually assumed to be normally distributed with mean values of zero and known variances and covariances (i.e., $\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \mathbf{V})$). Since any covariance between the measured variables can arise only if some relationship exists between the measured values (e.g., as a result of a dependency through calibrations based on peak area fraction for chromatographic analysis), it can often be assumed that covariances are not significantly different from zero so that $\mathbf{V}$ is diagonal. In addition, it is assumed in this work that there is no serial correlation between the measurement errors. Furthermore, it is assumed the covariance of the measurement error is constant over all operating conditions. If we take expectations of Equation (1), then $E(\mathbf{y}_t) = \mathbf{x}_t$, which means the "best" estimates for the process output variables at time t are just the realizations (i.e., measurement values) of $\mathbf{x}_t$. If other information about the dynamics of the process is available, such as a model of the process, such information could be used in conjunction with the measurements to obtain better estimates of process variables. Because a process model is never a perfect representation of the real plant, model predictions inevitably

contain some degree of error. We can write the predicted values of the M output variables in the additive noise form

$$\hat{\mathbf{y}}_t = \mathbf{x}_t + \delta_t \tag{2}$$

where $\hat{\mathbf{y}}_t$ is a M×1 vector of model predictions at time t, δ_t is a M×1 vector of model prediction error. Often, the model predicted values of the output variables can be explicitly calculated by

$$\hat{\mathbf{y}}_t = \mathbf{f}(\hat{\mathbf{x}}_{t-1}, \hat{\mathbf{x}}_{t-2}, \ \cdots, \mathbf{u}_{t-d}, \mathbf{u}_{t-d-1}, \ \cdots) \tag{3}$$

where $\hat{\mathbf{x}}_{t-i}$ is a M×1 vector of past reconciled outputs at time t-i (to be defined by Equation (6)), $\mathbf{u}_{t-d}$ is a N×1 vector of values of inputs including manipulated and disturbance variables at time t-d, and d represents the process time delay associated with each input. **f** represents a M×1 vector of functional forms for which any suitable model structures (linear or nonlinear) could be used.

The model prediction error, δ_t, is a complex function of various factors such as improper model form (e.g., linear approximations of nonlinear processes), uncertainties in values of model parameters, inaccurate information in $\hat{\mathbf{x}}_{t-i}$ or measurement errors in $\mathbf{u}_{t-d-i}$. Because stochastic properties of δ_t are difficult to evaluate, δ_t is often simply assumed to be Gaussian white noise (i.e., $\mathbf{e}_t \sim N(\mathbf{0}, \mathbf{R})$). This implies that **R** is constant over all operating conditions. Moreover, serial correlation of the model prediction error is negligible. Another assumption is that δ_t is independent of ε_t. Based upon above assumptions, the optimal estimates of the true values of process output variables at time t, can be obtained by simultaneously minimizing the weighted sum of squared measurement and model errors, that is

$$\text{Minimize } J(\hat{\mathbf{x}}_t) = (\mathbf{y}_t - \hat{\mathbf{x}}_t)^T \mathbf{V}^{-1} (\mathbf{y}_t - \hat{\mathbf{x}}_t) + (\hat{\mathbf{y}}_t - \hat{\mathbf{x}}_t)^T \mathbf{R}^{-1} (\hat{\mathbf{y}}_t - \hat{\mathbf{x}}_t) \tag{4}$$

where $\mathbf{V}$ is the covariance matrix of the measurement error ε_t. $\mathbf{R}$ is the covariance matrix of the model prediction error δ_t. At the current time t, since the values of the model predictions (e.g., one step ahead prediction), $\hat{\mathbf{y}}_t$, are known by Equation (3), the values of $\hat{\mathbf{x}}_t$ can be solved analytically. Taking partial derivatives of the objective function with respect to $\hat{\mathbf{x}}_t$, and setting it to zero yields

$$\frac{\partial J}{\partial \hat{\mathbf{x}}_t} = -2\mathbf{V}^{-1}(\mathbf{y}_t - \hat{\mathbf{x}}_t) - 2\mathbf{R}^{-1}(\hat{\mathbf{y}}_t - \hat{\mathbf{x}}_t) = \mathbf{0}$$

Solving for $\hat{\mathbf{x}}_t$ gives

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}(\mathbf{V}^{-1}\mathbf{y}_t + \mathbf{R}^{-1}\hat{\mathbf{y}}_t)$$
$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\mathbf{y}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{R}^{-1}\hat{\mathbf{y}}_t$$

Adding and subtracting $\mathbf{V}^{-1}$ to $\mathbf{R}^{-1}$ in the second term gives

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\mathbf{y}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\left[(\mathbf{V}^{-1} + \mathbf{R}^{-1}) - \mathbf{V}^{-1}\right]\hat{\mathbf{y}}_t$$

Expending and rearranging yields

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\mathbf{y}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}(\mathbf{V}^{-1} + \mathbf{R}^{-1})\hat{\mathbf{y}}_t - (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\hat{\mathbf{y}}_t$$
$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}(\mathbf{y}_t - \hat{\mathbf{y}}_t) \quad (5)$$

Rewriting Equation (5) in a simple form gives

$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}}_t + \mathbf{K}(\mathbf{y}_t - \hat{\mathbf{y}}_t) \quad (6)$$

where $\mathbf{K} = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}$. Equation (6) is the final expression for the DDR filter.

It is worth noting that the reconciled values of the output variables, $\hat{\mathbf{x}}_t$, are given in a predictor-corrector form. The first term, $\hat{\mathbf{y}}_t$, in the right-hand of Equation (6) represents the model predictions, and the second term $\mathbf{K}(\mathbf{y}_t - \hat{\mathbf{y}}_t)$ represents corrections to $\hat{\mathbf{y}}_t$ based on the current measurements $\mathbf{y}_t$. The matrix, $\mathbf{K}$, is regarded as the gain of the DDR filter.

For a univariate output process, Equation (6) of the DDR filter can be reduced to

$$\hat{x}_t = \hat{y}_t + \frac{\upsilon^2}{\upsilon^2 + \sigma^2}(y_t - \hat{y}_t) = \hat{y}_t + \frac{\upsilon^2/\sigma^2}{\upsilon^2/\sigma^2 + 1}(y_t - \hat{y}_t) \tag{7}$$

where υ^2 is the variance of the model predictions, and σ^2 is the variance of the measurements. This equation indicates that the filtered data follow an asymptotic curve as a function of υ^2/σ^2, and are bounded by measured and model predicted values. If the level of confidence in the model is low compared to that in the measurement, the ratio υ^2/σ^2 is large and the gain of the filter approaches 1. In this case, the filtered data are close to raw measurements. On the other hand, if a perfect model is available, the ratio υ^2/σ^2 is small, the gain of the filter approaches 0, and the filtered data are close to model predictions. As a result, reliable process models are paramount for good performance of the DDR filter in reducing the measurement error.

It is interesting to note that the form of the DDR filter is similar to that of a Kalman filter. The use of a Kalman filter demands a specific structure for process state/measurement models, namely,

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} + \mathbf{w}_{t-1}, \tag{8}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \boldsymbol{\varepsilon}_t \tag{9}$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are deterministic matrices with appropriate dimensions respectively. If all the state variables of the process are directly measured, $\mathbf{C}$ is an

identity matrix. $\mathbf{w}_t$ denotes process model noise which is assumed to be Gaussian white noise and independent of ε_t, $\mathbf{w}_t \sim N(\mathbf{0}, \mathbf{S})$. An important assumption in the Kalman filter is that all uncertainties in the models of the process can be efficiently translated to the process white noise $\mathbf{w}_t$. For the problem defined by Equations (8) and (9), the optimal estimates of process state variables by the Kalman filter are given by

$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}}_t + \boldsymbol{\kappa}_t \left(\mathbf{y}_t - \mathbf{C}\hat{\mathbf{y}}_t \right), \tag{10}$$

where

$$\hat{\mathbf{y}}_t = \mathbf{A}\hat{\mathbf{x}}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} \tag{11}$$

and the Kalman gain, κ_t, is recursively calculated by

$$\mathbf{P}_t^- = \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^T + \mathbf{S} \tag{12}$$

$$\boldsymbol{\kappa}_t = \mathbf{P}_t^- \mathbf{C}^T \left(\mathbf{C}\mathbf{P}_t^- \mathbf{C}^T + \mathbf{V} \right)^{-1} \tag{13}$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \boldsymbol{\kappa}_t \mathbf{C}\mathbf{P}_t^- \tag{14}$$

For nonlinear models, nonlinear terms in the models have to be linearized at each sampling time in order to implement the Kalman filter. The Kalman filters were developed in 1960, but their applications in chemical engineering have been limited (Roffel & Chin, 1987). This is not only because the apparent mathematical complexity of the Kalman filter is a deterrent for its wider use, but also because it is difficult to determine the covariance matrix of the process noise $\mathbf{S}$. As a consequence, the elements in $\mathbf{S}$ are commonly viewed as tuning parameters rather than measurable constants in implementing the Kalman filters (Wilson et al., 1998).

It has been observed that, if matrix $\mathbf{C}$ in the Kalman filter is an identity matrix, the form of the DDR filter (Equation (6)) is identical to that of the Kalman filter (Equation (10)), but their gain matrices are calculated differently. Furthermore, if $\mathbf{R}$ for the DDR filter and $\mathbf{S}$ for the Kalman filter are appropriately tuned, the two filters will be equivalent to each other. Compared to the Kalman filter, the concept of DDR filter is relatively concise and straightforward so that it is easier to understand and implement. Also, the

DDR filter can use a variety of structures for process models, unlike the Kalman filter which requires a state-space formulation. However, similar to the Kalman filters, it is also difficult to determine the covariance matrix of the model predictions (matrix **R**) in the DDR filter. Again, for simplicity, a diagonal gain matrix **K** is used, and its elements treated as tuning parameters in the DDR filter (Makni et al., 1995).

3 SIMULATION EXAMPLES

In order to illustrate the implementation of the DDR filters for closed-loop control, two simulation examples are now considered. One example is a simple process, a storage tank with one control loop. The other example is a more complex process, a distillation column with four control loops. The distillation column is first studied with feedback control, followed by the study of feedforward/feedback control with DDR filters.

3.1 Storage Tank

The schematic diagram of the cylindrical storage tank process is shown in Figure 1. The diameter and the height of the tank are 1.0 m and 1.2 m, respectively. A PI controller is used to regulate the liquid level of the tank by manipulating the outlet flow. The feed flow to the tank is measured but not controlled. The sampling interval is 1 minute and zero-order hold is used. At steady state, the feed flow rate and the liquid level are 30.0 L/min and 60.0 cm, respectively. Measured values were assumed to be normally distributed with standard deviations of 0.6 L/mim and 2.4 cm, respectively. The dynamics of the measuring device and control valve were neglected.

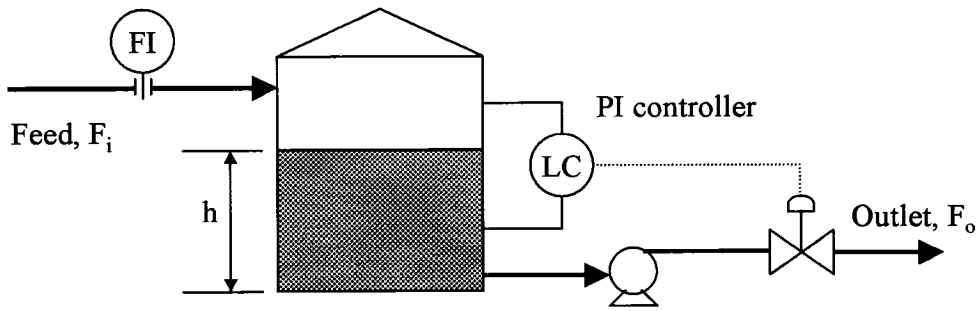


Figure 1 Schematic diagram of a storage tank process.

With measurement noise, the controller gain K_C and integral time τ_I were obtained by subjecting the storage tank to a series of step changes in the feed flow rate having magnitude of 20%, -40% and 20% of steady-state value respectively, followed by a series of controller setpoint changes with magnitude of -33%, 66% and -33% of steady-state value respectively, and then minimizing the objective function,

$$\Phi(K_C, \tau_I) = \alpha \sum_{t=0}^{t_s} (h_t^* - h_t)^2 \Delta t + \beta \sum_{t=0}^{t_s} (F_{o,t} - F_{o,t-1})^2 \Delta t \qquad (15)$$

where Φ denotes the cost function for controller performance, h_t^* is the setpoint, h_t is the true value of the tank level at sampling time t, $F_{o,t}$ is the value of the manipulated variable, the outlet flow, at time t, Δt is the sampling time interval. α is a weighting factor for the integral of squared errors of the controlled variable (ISE), and β is a weighting factor for the integral of squared differences of the manipulated variable between sampling times t and t-1 (ISDU). t_s represents the total number of process sampling periods over which the integration is performed. The minimization of the ISE term in the objective function attempts to maintain the controlled variable as close as possible to its setpoint, and the minimization of the ISDU term attempts to prevent excessive adjustments of the manipulated variable.

Minimization of $\Phi(K_C, \tau_I)$ was carried out using a quasi-Newton method to find optimal values of the controller parameters. The minimization was carried out 100 times, each time with different white noise sequences. The average values and their standard errors evaluated for the optimal controller parameters and the cost function for the controller performance are presented in the first row of Case I of Table 1. On average, the optimal controller gain was $\bar{K}_C = -1.48 \pm 0.007\, m^2.h^{-1}$, the integral time was $\bar{\tau}_I = 12.89 \pm 0.266\, min$, and the controller resulted in cost function of controller performance $\bar{\Phi}_{No\ filter} = 14.05 \pm 0.07$.

Table 1 Comparison of optimal controller performance without and with embedded filters for the storage tank (Values listed are sample means plus and minus one standard errors based on 100 random noise sequences).

Case	Filter	Optimal filter parameter	Optimal controller parameters		Objective function of controller performance		
			K_C, $m^2.h^{-1}$	τ_I, min	ISE (α=0.3)	ISDU (β=0.7)	$\Phi(K_C, \tau_I)$
I	No	-	-1.48±0.007	12.89±0.266	32.86±0.165	5.99±0.041	14.05±0.074
$\sigma_{h_t^m} = 2.4$cm	DDR	0.05±0.003	-2.49±0.003	7.69±0.019	15.76±0.023	4.02±0.008	7.54±0.011
$\sigma^2_{F_{i,t}^m} = 0.6$L/min	EWMA	0.66±0.004	-2.76±0.016	15.95±0.517	19.04±0.046	5.34±0.013	9.45±0.019
	MA	6	-2.98±0.001	20.33±0.095	18.18±0.044	5.18±0.008	9.08±0.017
II	No	-	-1.82±0.005	5.82±0.017	20.59±0.045	4.66±0.012	9.44±0.016
$\sigma_{h_t^m} = 1.2$cm	DDR	0.08±0.005	-2.49±0.003	7.66±0.015	15.73±0.022	4.02±0.007	7.51±0.009
$\sigma^2_{F_{i,t}^m} = 0.6$L/min	EWMA	0.45±0.001	-2.52±0.003	8.58±0.030	17.45±0.024	4.75±0.008	8.56±0.011
	MA	5	-3.01±0.002	18.36±0.065	17.06±0.024	4.97±0.005	8.60±0.009
III	No	-	-0.65±0.009	4.32±0.143	59.46±0.611	7.86±0.170	23.34±0.134
$\sigma_{h_t^m} = 4.8$cm	DDR	0.02±0.001	-2.62±0.003	9.01±0.050	15.94±0.026	4.21±0.008	7.42±0.011
$\sigma^2_{F_{i,t}^m} = 0.6$L/min	EWMA	0.80±0.001	-2.83±0.006	24.71±0.234	24.05±0.093	6.37±0.018	11.67±0.036
	MA	6	-2.56±0.005	17.89±0.155	23.10±0.096	6.07±0.020	11.18±0.038

Because noisy measurements were used for the controller, the controller performance was inevitably deteriorated. To attenuate the propagation of measurement noise throughout the control loop, a DDR filter for the storage tank was developed and embedded inside the feedback loop. The DDR filter (using Equation (6)) for the storage tank had the form

$$h_t^f = \hat{h}_t + k(h_t^m - \hat{h}_t) \tag{16}$$

where h_t^f is the filtered tank level used to calculate the control moves, k is the filter gain, h_t^m is the measured tank level, and $\hat{h}_t$ is the model predicted tank level given by the mass balance around the tank

$$\hat{h}_t = h_{t-1}^f + \frac{\Delta t}{A}(F_{i,t-1} - F_{o,t-1}) \tag{17}$$

In the Equation (17), A is the cross-sectional area of the tank, and $F_{i,t-1}$ is the raw measurements of feed flow rate at time t-1. Using the embedded DDR filter, the optimal controller parameters along with the filter gain were obtained by minimizing the objective function of Equation (15) when the tank was subjected to the same external disturbances and controller setpoint changes. Results of average values and their standard errors for the controller and filter parameters obtained for the same 100 sequences of Gaussian random noise, are presented in the second row of Case I of Table 1. With the embedded DDR filter, the optimal controller gain was increased by a factor of 1.7, while the controller integral time was decreased by a factor of 1.7, indicating that a more aggressive controller was obtained. The cost function of the controller performance decreased from $\bar{\Phi}_{\text{No filter}} = 14.05 \pm 0.07$ to $\bar{\Phi}_{\text{DDR}} = 7.54 \pm 0.01$, a 46% reduction. The average optimal DDR filter gain was $\bar{k} = 0.05 \pm 0.003$, such a small value meaning that significant confidence had been put on model predictions in the DDR filter.

Results of raw measurements, reconciled (filtered) and true values of controlled and manipulated variables for the storage tank using the optimal controller and the DDR filter

are presented in Figure 2. It shows that the noise associated with the filtered data of the liquid level was significantly less than the original measurements. The filtered tank level was very close to its true value and the high-frequency swings in the control moves were eliminated by the DDR filter.

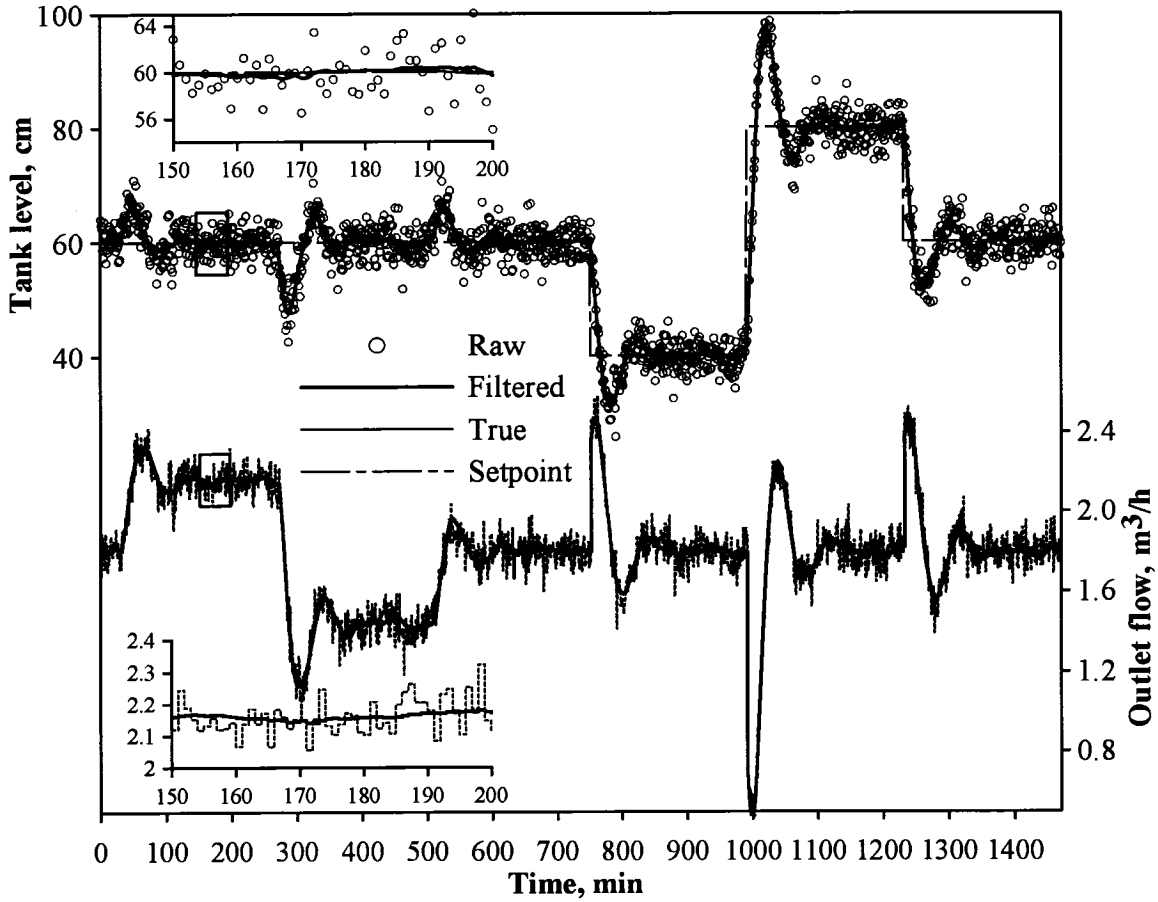


Figure 2 Performance of the DDR filter to reduce noise propagation through the control loop in the storage tank. The dashed line for manipulated variable (outlet flow rate) represents the control moves without DDR filter.

The enhanced controller performance using the DDR filter was compared to that using common EWMA and MA filters. The EWMA filter, for the filtered level, is given by

$$h_t^f = \theta h_{t-1}^f + (1-\theta) h_t^m \tag{18}$$

where θ, the tuning parameter of the filter, is the fraction of the estimated tank level at the previous sampling instant retained for the current estimated value. The MA filter, for the filtered tank level, is given by

$$h_t^f = \frac{1}{L}\sum_{i=0}^{L-1} h_{t-i}^m \qquad (19)$$

where the most recent L measurements are averaged. Using the EWMA filter in the feedback loop, the controller parameters K_C and τ_I along with the filter parameter θ were optimized by minimizing the same objective function when the tank was subjected to the identical feed flow disturbances and controller setpoint changes. Results obtained for the same 100 Gaussian random noise sequences are presented in the third row of Case I of Table 1. The results for the MA filter are also presented in Case I of Table 1. Compared to the DDR filter, the optimal controller gain and integral time were increased. The cost function of the controller performance using the EWMA and MA filters were $\bar{\Phi}_{EWMA} = 9.45 \pm 0.02$ and $\bar{\Phi}_{MA} = 9.08 \pm 0.02$ respectively, corresponding to similar reductions of 33% and 35% respectively. These reductions were significantly smaller than the 46% reduction achieved with the DDR filter.

The true values of controlled and manipulated variables for the storage tank without filter, with DDR filter, and with EWMA filter are presented in Figure 3. This figure shows that, with optimal filters, the controlled variable displayed faster response to external disturbances and controller setpoint changes and had smaller deviations from the target when the external disturbances occurred. However, with the DDR filter, the controlled variables displayed smallest deviations and smallest settling time. The high-frequency variation of the manipulated variable was reduced by all filters, but significantly more by the DDR filter.

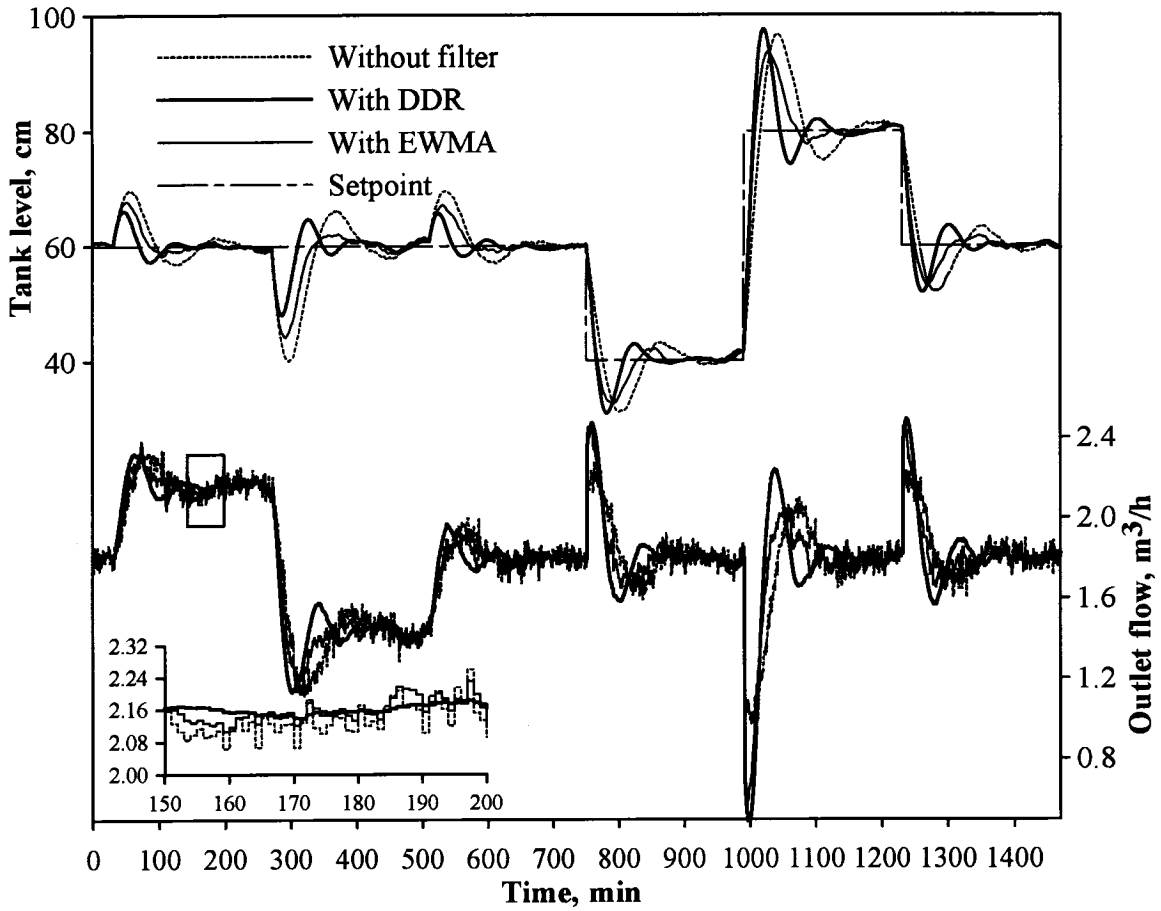


Figure 3 Performance of the storage tank level controller, with and without embedded DDR and EWMA filters, when subjected to a series of step changes in the feed flow rate followed by controller setpoint changes.

The DDR filter performed better than the EWMA and MA filters because it incorporated intrinsic, dynamic process models to provide a better estimate of the controlled variable. It is noted that the measured disturbance variable, the feed flow rate, was used in Equation (17) to calculate the predicted tank level in the DDR filter. The impact of the measurement error of the disturbance variable on the performance of the DDR filter was also examined. After doubling the variance of the noise affecting the measurements of the feed flow rate, the controller and the DDR filter were retuned by minimizing the same objective function under the same process conditions. For this case, the optimal filter gain was $\bar{k} = 0.08 \pm 0.005$ and the cost function of the controller performance was $\bar{\Phi}_{DDR} = 7.60 \pm 0.01$. Compared to the results in Case I of Table 1, these results indicate

that, with an increase in the noise level of the disturbance variable, the gain of the DDR filter increased, indicating a decrease in the confidence put on the model due to the increased input noise to the model. On the other hand, the cost function of controller performance remained essentially unchanged with the increase of noise level in the disturbance variable. The impact of measurement noise in the disturbance variable on the performance of DDR filter was negligible, because of inertia of the process.

The objective function of controller performance depends on the weighting factors applied to the ISE and ISDU terms (Equation (15)). Results presented above were obtained for $\alpha = 0.3$ and $\beta = 0.7$. In practice, the weighting factors for controller performance are chosen by system expert to obtain the desired closed-loop response. To determine the impact of these weighting factors on the performance of the controller and the filters, the above analysis was repeated with $\alpha = 0.7$ and $\beta = 0.3$. The values of the optimal cost function for controller performance without filter and with DDR, EWMA and MA filters were $\bar{\Phi}_{\text{No filter}} = 16.30 \pm 0.03$ $\bar{\Phi}_{\text{DDR}} = 9.45 \pm 0.01$, $\bar{\Phi}_{\text{EWMA}} = 11.30 \pm 0.02$ and $\bar{\Phi}_{\text{MA}} = 11.05 \pm 0.02$, respectively. Nevertheless, all values were significantly higher than the original weightings which placed most weight on the ISDU. This is not surprising since ISE values were considerably greater than ISDU values. The presence of DDR, EWMA and MA filters resulted in 42%, 31% and 32% reductions in the cost function, respectively. These performance improvements are nearly identical to Case I of Table 1 and it was concluded that although the weighting factors did cause a significant increase in the controller performance criteria, they did not have a major impact on the percentage improvements when using filters.

The enhancement of controller performance is due to the noise reduction in the controlled variable by the filters and resulting reduction in unnecessary controller moves. The impact of the noise level of the controlled variable on the performance of controller and filters was examined by performing the above analysis with half and twice the noise magnitude on the measurements of the tank level. Results of this analysis are presented as Cases II and III of Table 1. In the absence of a filter, a larger noise level affecting the

controlled variable resulted in a significantly lower controller gain and a significant increase in the cost function for controller performance. Adding any of the three filters led to a more aggressive controller and a lower cost function for controller performance. As expected, the gain of the DDR filter decreased with an increase of the noise level of the controlled variable, indicating that more confidence was put on model predicted values and less confidence on actual measurements. It is interesting to observe that the value of the objective function with the DDR filter is nearly independent on the level of noise whereas it varied greatly for the unfiltered data. For the EWMA filter, the optimal parameter θ increased significantly with the increase of noise level, meaning that more weight is put on past measurements. For the MA filter, the length of the averaging window remained essentially the same for the three levels of noise. For use of both EWMA and MA filters, the controller performance was affected by the noise level. The advantages of using the DDR filter increase remarkedly with higher levels of noise.

3.2 Distillation Column

For simple processes like the storage tank, it is easy to develop phenomenological dynamic models for the DDR filter. However, for a complex process, it becomes impractical to use fundamental models. Therefore, empirical input-output models are normally used in the DDR filter. In this study, a distillation column used to separate benzene and toluene was employed to demonstrate the strategies of developing DDR filters for complex processes and to determine the effectiveness of the DDR filter when using approximate model predictions.

3.2.1 Feedback Control

A schematic diagram of the control system for the distillation column is presented in Figure 4. Two feedback PI controllers, TIC-D and TIC-B, are used to control the top and bottom temperatures by manipulating the reflux flow rate and the reboiler heat duty, respectively. Feedback PI controllers, LIC-D and LIC-B, are used to control liquid levels in the reflux drum and column base by manipulating the distillate flow rate and the bottom product flow rate, respectively. The discrete sampling time is 30 s for all control loops. The external disturbances are the feed flow rate and feed composition. The

nominal steady-state values for all measurements and their typical noise levels are listed in Table 2. A simulator for the dynamics of this column, based on rigorous models (i.e., mass and heat balance, thermodynamics and tray hydraulics), was developed for this study (Bai, 2003).

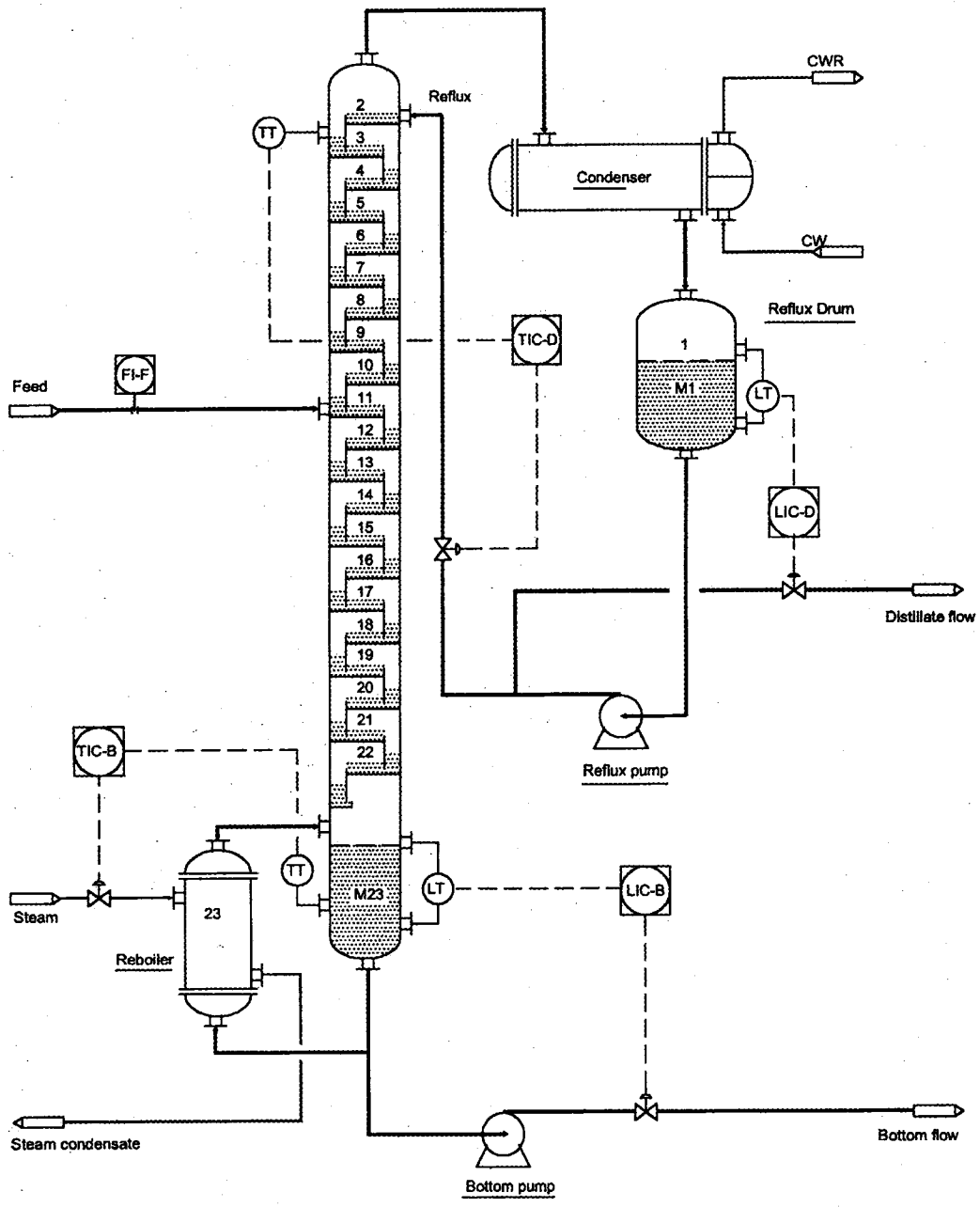


Figure 4 Schematic diagram of the binary distillation column.

Table 2 Nominal steady-state values and noise levels of measured variables for the distillation column.

Variable	Units	Steady-state values	Standard deviation, σ
Feed flow , F	kmol/h	100.0	0.50
Top temperature, T_D	°C	84.2	0.25
Bottom temperature, T_B	°C	117.4	0.25
Reflux drum level, H_D	m	0.50	0.02
Column base level, H_B	m	0.70	0.02

For the four PI controllers, tuning parameters were obtained by minimizing the overall objective function of the control system

$$\Phi(\mathbf{K}_C, \boldsymbol{\tau}_I) = \sum_{i=1}^{4}\left[\alpha_i \sum_{t=0}^{t_s}(x_{i,t}^{*} - x_{i,t})^2 \Delta t + \beta_i \sum_{t=0}^{t_s}(u_{i,t} - u_{i,t-1})^2 \Delta t\right] \tag{20}$$

when the column was subjected to a series of step changes in feed flow rate having magnitudes of 20%, -40% and 20% of steady-state value, respectively. In Equation (20), $x_{i,t}$ represents the true value of controlled variable i, $x_{i,t}^{*}$ represents the setpoint for controlled variable i, and $u_{i,t}$ represents the manipulated variable for control loop i. The weighting factors, α_i, for all the ISE terms were set to 0.15 while the weighting factors, β_i, for all the ISDU terms were set to 0.1. The optimal controller parameters, obtained for ten different Gaussian noise sequences, are presented as Case I of Table 3. The associated optimal values for ISE and ISDU for each controller, as well as the overall cost function of the control system are presented as Case I of Table 4.

Table 3 Optimal controller Tuning Parameters for the Distillation Column (Values listed are sample means plus and minus one standard errors based on 10 random noise sequences).

Case	Filter	LIC-D		TIC-D		LIC-B		TIC-B	
		K_C $kmol.h^{-1}.m^{-1}$	τ_I min	K_C $kmol.h^{-1}.°C^{-1}$	τ_I min	K_C $kmol.h^{-1}.m^{-1}$	τ_I min	K_C $MJ.h^{-1}.°C^{-1}$	τ_I min
I	No	-41.7±0.3	15.0	-2.2±0.03	5.0	-75.8±0.4	5.0	44.6±0.5	5.0
II	DDR	-92.6±0.5	15.0	-5.0±0.03	5.0	-172.8±1.0	5.0	144.8±1.4	5.0
III	EWMA	-159.1±6.3	15.0	-5.0±0.30	5.0	-181.0±3.1	5.0	143.6±2.5	5.0
IV	No	-35.9±0.1	15.0	-1.9±0.03	5.0	-47.9±0.4	5.0	23.3±0.4	5.0
V	DDR	-81.6±0.5	15.0	-5.5±0.06	5.0	-110.2±1.1	5.0	104.2±0.8	5.0
VI	EWMA	-123.7±2.7	15.0	-3.8±0.10	5.0	-108.5±0.8	5.0	145.3±7.3	5.0
VII	DDR	-79.0±1.4	15.0	-7.4±0.06	5.0	-10.5.2±2.7	5.0	91.2±0.2	5.0
VIII	EWMA	-116.7±9.2	15.0	-8.0±0.1	5.0	-138.3±3.2	5.0	70.9±1.1	5.0

Cases I, II, III: Feedback Control; Cases IV, V, VI: Feedforward/feedback control; Cases VII, VIII: Setpoint changes.

In order to improve the controller performance, a DDR filter was developed to reduce noise propagation throughout the control loops for the distillation column. The first step in developing the DDR filter was to identify empirical process models. The open-loop process reaction curve with measurement noise for a pair of input/output variables with a 20% step change in the input was simulated. Then, pure-integrator-plus-dead-time or first-order-plus-dead-time models were used to approximate the process dynamics around the steady state. The model parameters, process static gains, time constants and dead times, were obtained by fitting the data using the least-squares criterion. For the multiple-input, multiple-output (MIMO) system of the distillation column, the superposition property of the step responses for each output variable was assumed. As a result, the overall effects of the inputs on the outputs are additive, and the Laplace domain dynamic models were given by

Table 4 Comparison of controller performance without and with filters for the distillation column

(Values listed are sample means plus and minus one standard errors based on 10 random noise sequences).

Case	Filter	LIC-D α=0.15, β=0.1		TIC-D α=0.15, β=0.1		LIC-B α=0.15, β=0.1		TIC-B α=0.15, β=0.1		Φ
		ISE	ISDU	ISE	ISDU	ISE	ISDU	ISE	ISDU	
I	No	4.80±0.07	6.46±0.08	2.03±0.06	3.98±0.09	4.98±0.17	6.68±0.05	5.49±0.07	3.31±0.05	4.64±0.03
II	DDR	0.80±0.01	0.94±0.01	0.54±0.01	0.71±0.01	0.73±0.03	1.11±0.01	1.44±0.02	1.28±0.01	0.93±0.00
III	EWMA	1.07±0.03	1.21±0.03	1.00±0.07	0.74±0.05	1.89±0.11	2.36±0.05	1.56±0.03	1.08±0.02	1.37±0.01
IV	No	4.07±0.07	4.72±0.06	1.54±0.03	3.07±0.07	1.89±0.14	2.54±0.05	2.37±0.03	0.86±0.03	2.60±0.03
V	DDR	0.82±0.01	0.88±0.01	0.65±0.01	0.67±0.02	0.57±0.03	0.44±0.01	0.68±0.01	0.44±0.01	0.65±0.00
VI	EWMA	0.94±0.02	0.77±0.04	1.08±0.03	0.40±0.05	0.79±0.05	0.62±0.02	0.47±0.02	0.23±0.02	0.69±0.00
VII	DDR	0.37±0.01	0.33±0.01	0.60±0.02	0.54±0.02	0.15±0.02	0.14±0.01	4.01±0.03	2.04±0.01	1.07±0.01
VIII	EWMA	0.58±0.02	0.48±0.04	0.66±0.01	0.83±0.04	0.38±0.07	0.37±0.02	3.47±0.05	1.26±0.03	1.06±0.01

$$\begin{bmatrix} H_D(s) \\ T_D(s) \\ H_B(s) \\ T_B(s) \end{bmatrix} = \mathbf{G}_U(s) \begin{bmatrix} D(s) \\ R(s) \\ B(s) \\ Q(s) \end{bmatrix} + \mathbf{G}_D(s) \begin{bmatrix} F(s) \\ X(s) \end{bmatrix} \tag{21}$$

where H_D is the reflux drum liquid level, T_D is the top temperature, H_B is the column base liquid level, T_B is the column base temperature, D is the distillate flow rate, R is the reflux flow rate, B is the bottom flow rate, Q is the reboiler heat duty, F is the feed flow rate, and X is the benzene mole fraction in the feed. All variables are in deviation form. In Equation (21), $\mathbf{G}_U(s)$ is the matrix of transfer functions related to the manipulated variables, and $\mathbf{G}_D(s)$ is the matrix of transfer functions related to external disturbances.

Discrete forms (see appendix A) of the dynamic models were used to predict the values of the four controlled variables at each sampling time. Consequently, the DDR filter for the distillation column can be written as

$$\begin{bmatrix} H^f_{D,t} \\ T^f_{D,t} \\ H^f_{B,t} \\ T^f_{B,t} \end{bmatrix} = \begin{bmatrix} \hat{H}_{D,t} \\ \hat{T}_{D,t} \\ \hat{H}_{B,t} \\ \hat{T}_{B,t} \end{bmatrix} + \mathbf{K} \left\{ \begin{bmatrix} H^m_{D,t} \\ T^m_{D,t} \\ H^m_{B,t} \\ T^m_{B,t} \end{bmatrix} - \begin{bmatrix} \hat{H}_{D,t} \\ \hat{T}_{D,t} \\ \hat{H}_{B,t} \\ \hat{T}_{B,t} \end{bmatrix} \right\} \tag{22}$$

where $\mathbf{K}$ is the gain matrix of the DDR filter. For simplicity, we assumed $\mathbf{K}$ to be diagonal, and its diagonal elements were treated as tuning parameters. For the distillation column, the DDR filter was first tuned by minimizing the mean squared differences between the reconciled and the true values. In other words, the diagonal elements of the matrix $\mathbf{K}$ were obtained by minimizing

$$\Psi(\mathbf{K}) = \sum_{i=1}^{4} \left(\frac{1}{t_s \sigma_i^2} \sum_{t=0}^{t_s} (\hat{x}_{i,t} - x_{i,t})^2 \right) \tag{23}$$

where σ_i^2 is the variance of the raw measurements for variable i. The optimal DDR gain matrix was obtained for ten different Gaussian random noise sequences, and their average values and standard errors of the diagonal elements were found to be

$$\bar{\mathbf{K}} = \begin{bmatrix} 0.21 \pm 0.003 & 0 & 0 & 0 \\ 0 & 0.24 \pm 0.003 & 0 & 0 \\ 0 & 0 & 0.21 \pm 0.003 & 0 \\ 0 & 0 & 0 & 0.23 \pm 0.003 \end{bmatrix} \tag{24}$$

Using the DDR gain matrix of Equation (24), the DDR filter was embedded in the feedback loops, and the controller parameters were obtained by minimizing the objective function given by Equation (20) when the distillation column was submitted to the same sequence of feed flow rate disturbances. Optimal values of the controller parameters from ten different Gaussian random noise sequences are presented as Case II of Table 3. The controller gains obtained with the DDR filter increased by factors of 2 to 3 compared to those in Case I, while the integral times remained unchanged. Values of the ISE and ISDU obtained with the DDR filter are presented as Case II of Table 4. With those more aggressive controllers, the ISE and ISDU values were reduced significantly by a factors of 2.5 to 7.0, whereas the overall cost function of the control system was reduced by a factor of 5.0 compared to the results obtained without a filter. The raw measurements, the filtered and true values for both the controlled and manipulated variables are presented in Figure 5. The reduced noise level of the filtered values compared to the raw measurements was significant and the filtered values were close to their true values. In addition, the high-frequency oscillations of the control moves were significantly reduced by the DDR filter.

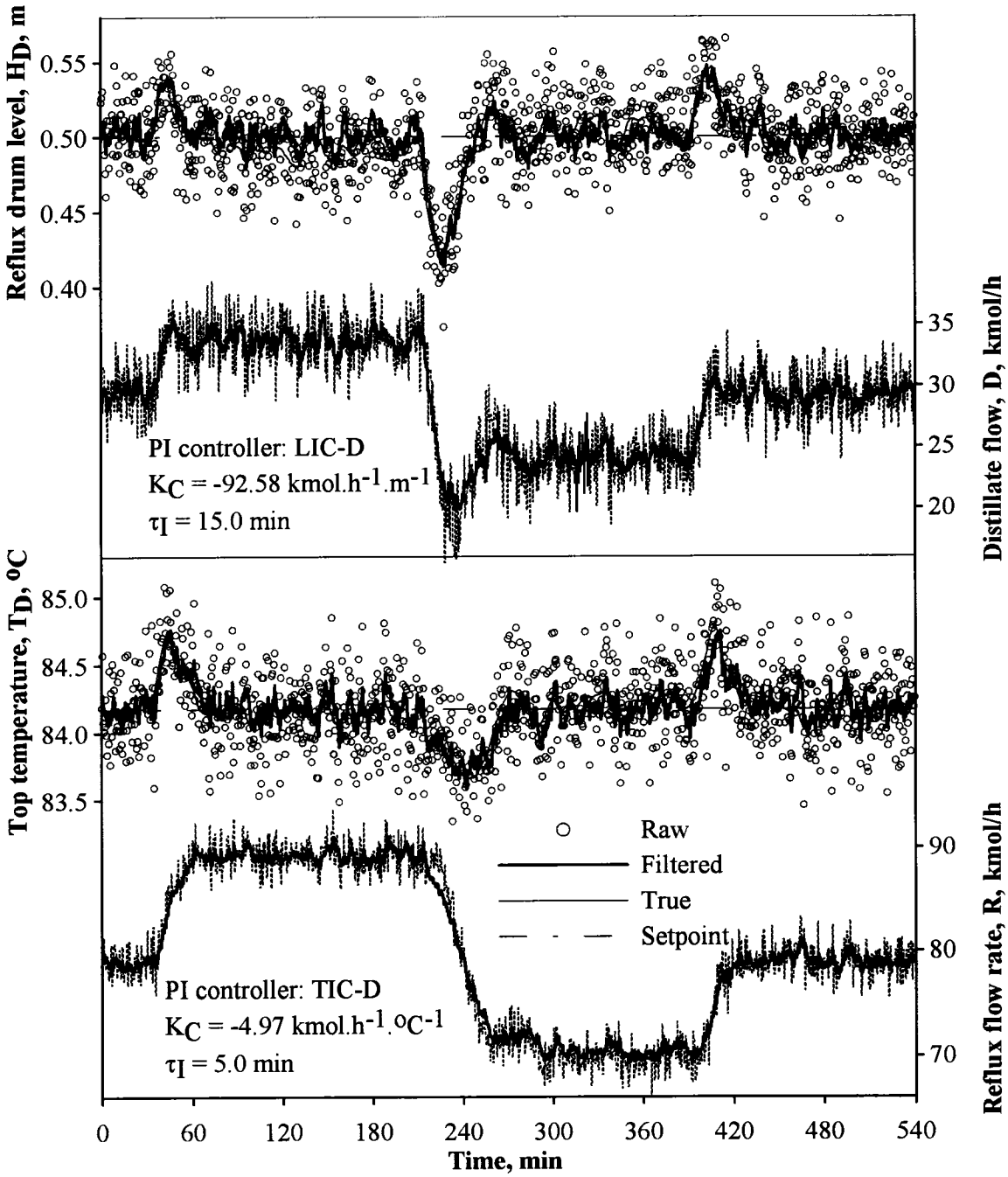


Figure 5(a) Performance of feedback controllers LIC-D and TIC-D with embedded DDR filter in the distillation column for a series of step changes in feed flow rate that are 20%, -40% and 20% of the nominal steady-state value. For the manipulated variables, the dashed line represents the control moves without DDR filter.

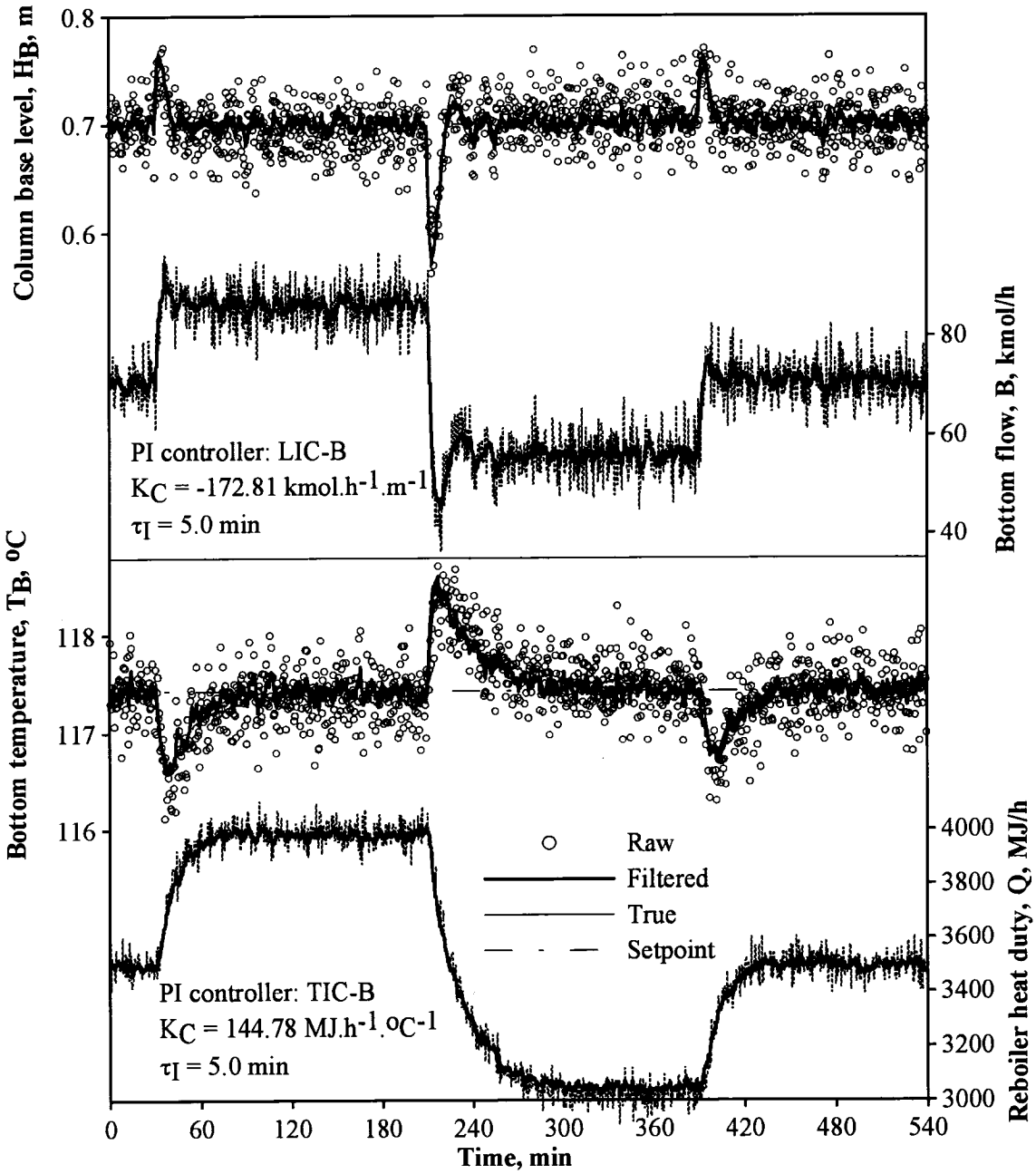


Figure 5(b) Performance of feedback controllers LIC-B and TIC-B with embedded DDR filter in the distillation column for a series of step changes in feed flow rate that are 20%, -40% and 20% of the nominal steady-state value. For the manipulated variables, the dashed line represents the control moves without DDR filter.

The performance of the DDR filter was compared to that of EWMA filters. The tuning parameters for the controllers and EWMA filters were simultaneously optimized by minimizing the controller cost function of Equation (20). Results for the optimal controller parameters obtained for the same ten Gaussian random noise sequences used for Cases I and II are presented as Case III of Table 3. Compared to Case II (with the DDR filter), the controller gain for the two liquid levels increased, while the controller gains for the two temperatures displayed no significant changes. The optimal parameters for the EWMA filters associated with controllers LIC-D, TIC-D, LIC-B and TIC-B were 0.86±0.004, 0.77±0.008, 0.73±0.004 and 0.79±0.005, respectively. The resulting values of ISE and ISDU for each controller using the EWMA filters are presented as Case III of Table 4. All mean values of the ISE and ISDU obtained for the EWMA filters, except the ISDU value of the bottom temperature, increased compared to the case when the DDR filter was used. The overall cost function of the control system increased from $\bar{\Phi}_{DDR} = 0.93 \pm 0.003$ to $\bar{\Phi}_{EWMA} = 1.37 \pm 0.012$. However, this still corresponds to a 70% reduction in the cost function with EWMA filters, compared to Case I; an 80% reduction was obtained with the DDR filter. The true values of controlled and manipulated variables for the distillation column, without filter, with DDR filter and with EWMA filters, are presented in Figure 6. The responses of the column were relatively slow without filters. It took about 210 minutes for the column to reach steady state following the –40% step change in the feed flow rate. The dynamic response of the column became faster when any one filter was implemented. For example, it took approximately 150 minutes for the column to reach steady state following the –40% step change in the feed flow rate. The use of EWMA filters resulted in more oscillatory control than with DDR filter except for the bottom temperature control loop. The DDR filter performed better than the EWMA filters in improving the controller performance. The better performance of the DDR filter was attributed to the fact that the DDR filter is able to anticipate process dynamics because process dynamic models are an integral part of the filter.

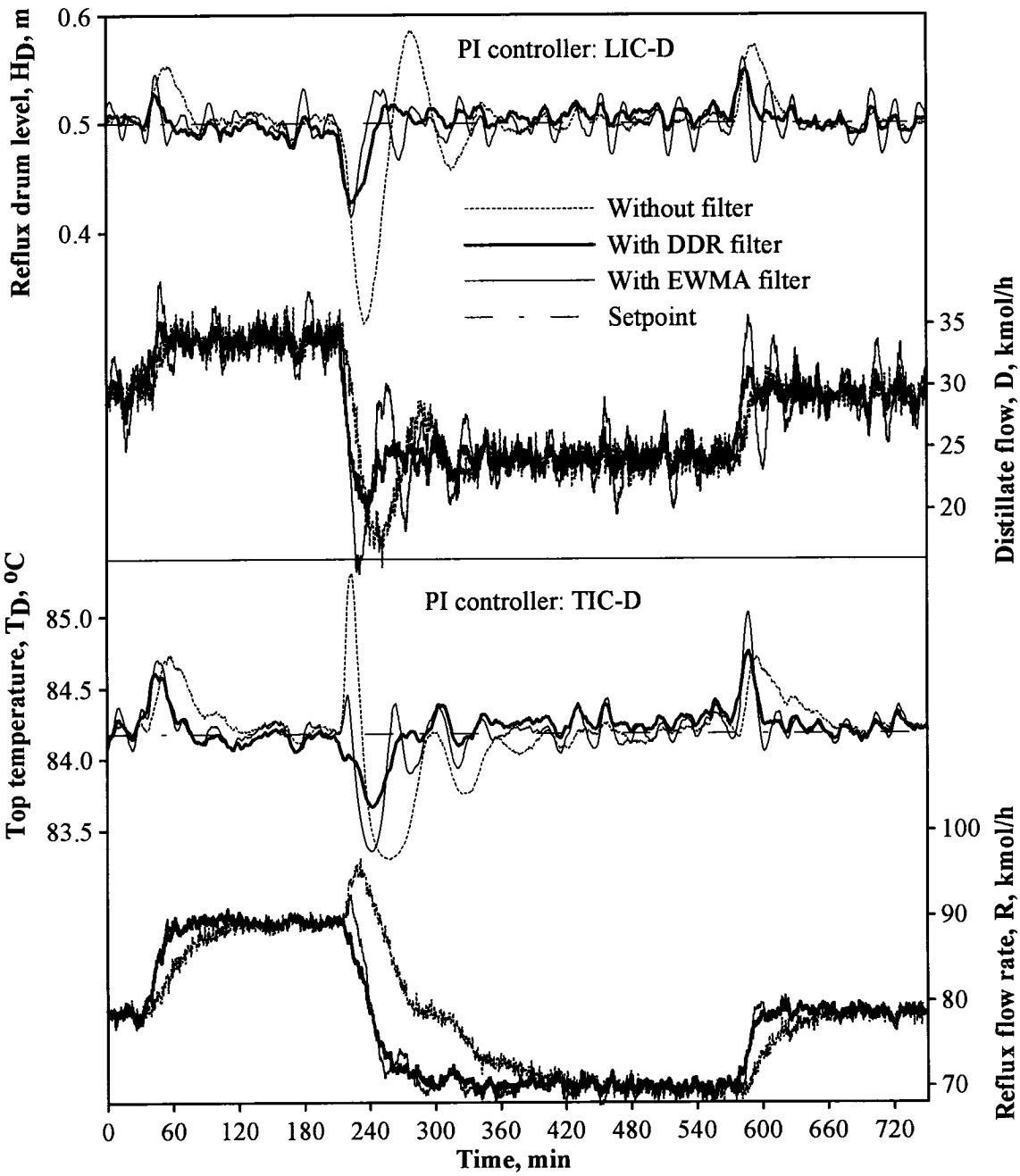


Figure 6(a) Comparison of feedback controller performance for LIC-D and TIC-D with and without embedded filters inside feedback loops for a series of step changes in feed flow rate that are 20%, -40% and 20% of nominal steady-state value.

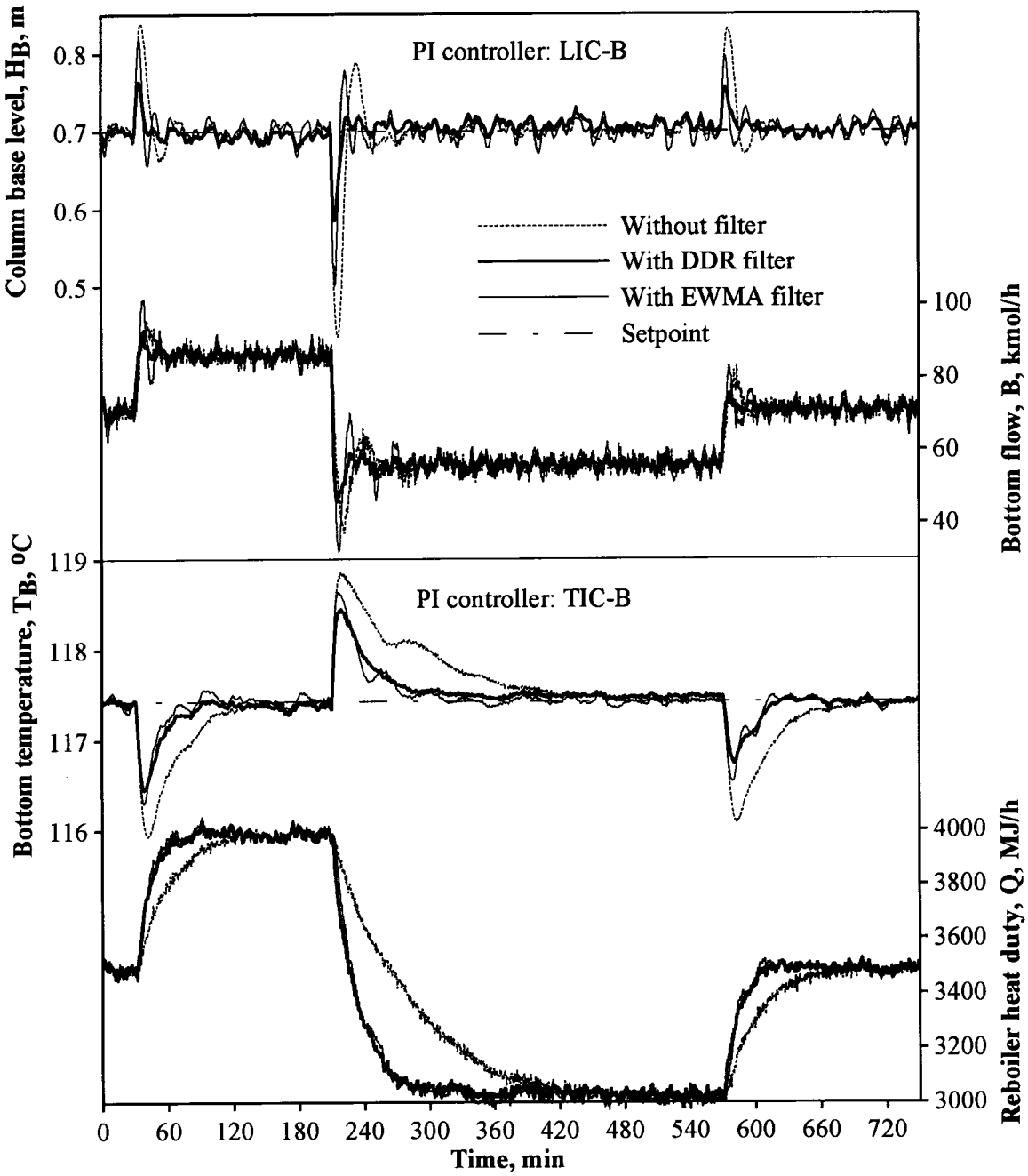


Figure 6(b) Comparison of feedback controller performance for LIC-B and TIC-B with and without embedded filters inside feedback loops for a series of step changes in feed flow rate that are 20%, -40% and 20% of nominal steady-state value.

3.2.2 Feedforward/Feedback Control

Since the DDR filter incorporated information about the disturbance while the EWMA did not. It was therefore decided to add feedforward control action, in addition to

the feedback action, to the two control loops at the bottom of the column and to compare the performance of both filters. Using the empirical model consisting of Equations (A-5) and (A-6) (see Appendix A), a feedforward controller for the column base level was obtained by the following equation to compensate for disturbances in the feed flow rate

$$B_t = 0.69F_t + 0.35F_{t-1} \tag{25}$$

and similarly for the bottom temperature, the feedforward controller was determined as

$$Q_t = 12.5F_{t-1} \tag{26}$$

These two loops were chosen because the feed flow rate had a greater impact on these two controlled variables. With the feedforward/feedback controllers and without filters, the tuning parameters for the feedback controllers were re-optimized by minimizing the objective function of Equation (20) subject to the same series of feed flow disturbances. No tuning was performed for the feedforward controllers. Results of the optimal feedback controller parameters are presented as Case IV of Table 3, and the associated ISE and ISDU values are presented as Case IV of Table 4. Compared to Case I where only feedback controllers were employed, the gains of all controllers decreased, and more importantly for the two loops with feedforward actions, the base level and bottom temperature controllers. The values of ISE and ISDU for all control loops were reduced significantly with the feedforward/feedback control scheme. Adding a feedforward action to the two bottom control loops also had a beneficial effect on the top control loops. The corresponding overall cost function of the control system was $\bar{\Phi}_{\text{Feedforward/feedback}} = 2.60 \pm 0.03$, resulting in a 44% reduction compared to $\bar{\Phi}_{\text{Feedback}} = 4.64 \pm 0.03$.

The DDR filter for the distillation column was re-tuned for the feedforward/feedback control system by minimizing the same objective function defined in Equation (23), and the optimal gain matrix of the DDR filter was

$$\bar{\mathbf{K}} = \begin{bmatrix} 0.24 \pm 0.003 & 0 & 0 & 0 \\ 0 & 0.20 \pm 0.003 & 0 & 0 \\ 0 & 0 & 0.21 \pm 0.003 & 0 \\ 0 & 0 & 0 & 0.20 \pm 0.003 \end{bmatrix} \quad (27)$$

Equation (27) indicates the DDR filter gain changed slightly compared to Equation (24) for the case of feedback control. Embedding the DDR filter inside the feedback loops, the feedback controller parameters were re-optimized under the same series of feed flow disturbances. Results of the optimal controller parameters obtained for the ten Gaussian random noise sequences are presented as Case V of Table 3 and results for the associated ISE, ISDU and the overall cost function of the control system are presented as Case V of Table 4. Use of the DDR filter increased the gain of the feedback controllers and resulted in significantly smaller values of ISE and ISDU for all controllers. The overall cost function of the feedforward/feedback control system was reduced from $\bar{\Phi}_{\text{No filter}} = 2.60 \pm 0.03$ to $\bar{\Phi}_{\text{DDR}} = 0.65 \pm 0.003$, corresponding to a 75% reduction.

The performance of the DDR filter for the feedforward/feedback control scheme was also compared to that of EWMA filters. Using EWMA filters in the feedback loops, the optimal parameters of EWMA filters along with the feedback controllers were simultaneously optimized under the same conditions. For this case, on average the optimal EWMA parameters were 0.86±0.001, 0.79±0.015, 0.74±0.004 and 0.90±0.003, respectively. The feedback controller parameters are presented as Case VI of Table 3, and the resulting ISE, ISDU and overall cost function of the control system are presented as Case VI of Table 4. Compared to Case V (with DDR filter), the gains of the controller for the reflux drum level and bottom temperature increased, while it slightly decreased for the top temperature and column base level. In general, the EWMA filters produced smoother responses such that the ISDU were significantly lower with the exception of the base level controller. On the other hand, the ISE values were higher for three control loops. Only the bottom temperature controller led to a decrease in ISE when the EWMA filters were used. The overall cost function of the control system with the EWMA filters

was $\bar{\Phi}_{EWMA} = 0.69 \pm 0.003$, while it was $\bar{\Phi}_{DDR} = 0.65 \pm 0.003$ with the DDR filter. The DDR filter performed slightly better than the EWMA filters for the feedforward/feedback control scheme.

3.2.3 Setpoint Changes

The performance of the DDR and EWMA filters was studied for setpoint changes of some of the controllers. The control loop TIC-B was subjected to a series of setpoint changes, and the filter and controller parameters were optimized. The resulting values of controller parameters as well as ISE, ISDU and overall cost function of control system are presented in Tables 3 and 4 as cases VII and VIII for the DDR and EWMA filters, respectively. The use of a DDR filter resulted in smaller controller gains, ISE and ISDU values than those for EWMA filters, except for the bottom temperature control loop. No significant difference in the overall cost functions of the control system with DDR and EWMA filters was found. The raw, filtered and true values of the bottom temperature are presented in Figure 7. With the EWMA filter, the filtered values were relatively smooth, but were delayed and displayed relatively large deviations from their true values. On the other hand, with the DDR filter, the filtered values performed very well in tracking their true values, but were affected by larger variations. It is interesting to note that despite the delayed filtered values for the EWMA filter, the controller performance for the TIC-B control loop did significantly better than the control loop with a DDR filter. However, these results were obtained under the unrealistic assumption that the true values of the process were available to optimize the filters and controllers. This assumption does not affect the control loop with the DDR since it was able to accurately track the true values. Using the filtered values to tune both the filters and controllers led to values of the objective function equal to 0.98±0.003 and 1.19±0.012 for the DDR and EWMA, respectively. From these results, one can conclude that under ideal, but unrealistic conditions, the DDR and EWMA filters have similar performances. However, under more realistic conditions, the DDR filter is superior.

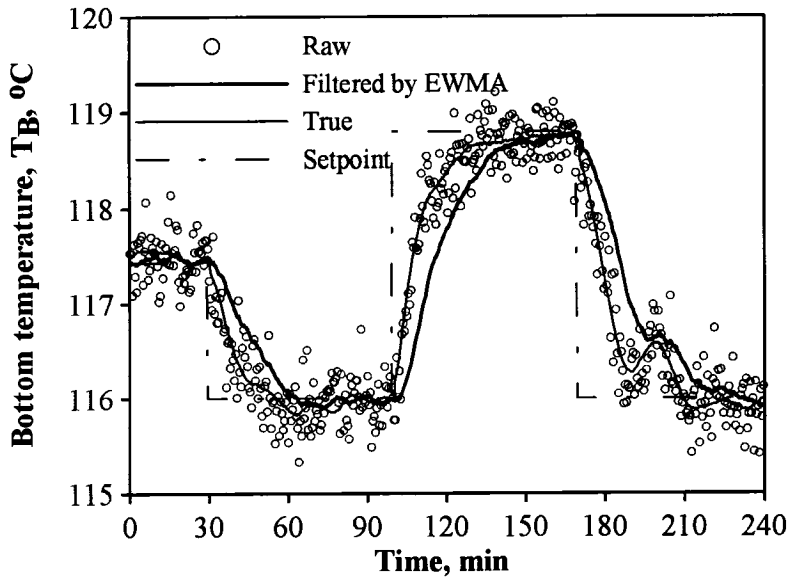


Figure 7(a) Performance of EWMA filter when controller TIC-B had a series of setpoint changes.

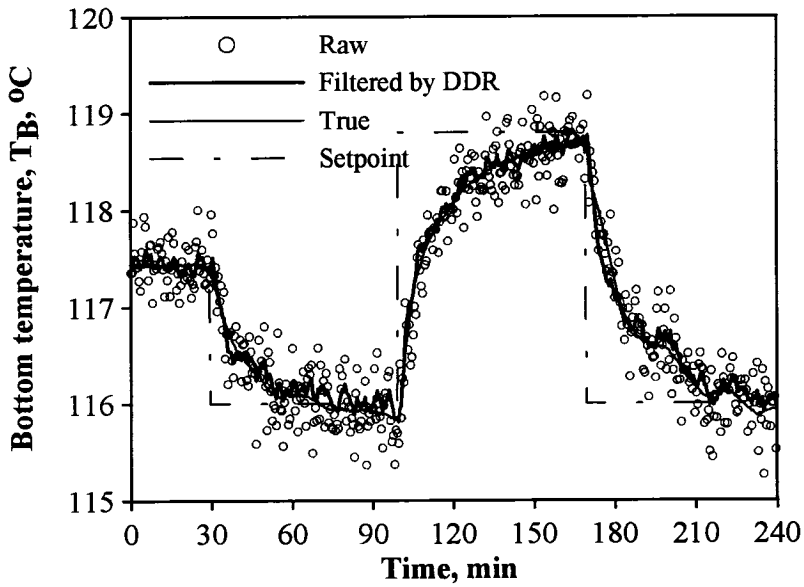


Figure 7(b) Performance of DDR filter when controller TIC-B had a series of setpoint changes.

3.2.4 Model Mismatch

Previous results have shown that DDR filters led to a significant improvement of the controller performance by filtering measurement noise and allowing the use of more aggressive controllers. The DDR filter relies strongly on the accuracy of the process

model to predict the controlled variable in order to provide a good estimate of its true value. In the case of the distillation column, linear models were used to represent the process dynamics and these models are valid in the vicinity of the nominal operating point. For relatively large setpoint changes, the new stationary value predicted by the model may not correspond to the true value of the process. Since the filtered value using the DDR is essentially a compromise between the measured and the predicted values of a controlled variable, the DDR filter will lead to an offset in the controlled variable. The best solution to remedy this problem is to derive a more accurate process model. For instance, such offset was not observed in the response to setpoint changes in the simple example of storage tank process where an accurate process model was used. On the other hand, a EWMA filter would provide a filtered value that approximates the true value provided the mean value recorded by the measuring device are accurate.

4 CONCLUSION

The DDR filter developed in this work was shown to be an effective tool that can be implemented in real-time to reduce the impact of measurement noise before calculating the control action. The application of the DDR filter can result in significantly better feedback controller performance. It allows more aggressive controllers to be used, and at the same time, prevents the manipulated variables from excessive manipulations. The performance of DDR filters is not affected by the noise level in the controlled variable, whereas, the performance of EWMA and MA filters deteriorates with the increase of noise level in the controlled variable. The DDR filters can give better performance than commonly used EWMA and MA filters because they can anticipate the process dynamics using process models. When using DDR filters, the external disturbances have to be measured, and process models have to be identified if fundamental models are unavailable. However, these are not required for the use of EWMA and MA filters.

When available dynamic information is used through a feedforward action, the performance of the feedforward/feedback controllers is enhanced and the filters provide

very similar dynamic performance. Accurate process models play an important role for good performance of DDR filters. Model mismatch is an important factor to be considered for the use of DDR filters. The DDR filters have to be adapted when using linear models for nonlinear processes.

NOMENCLATURE

A : Cross-sectional area of storage tank, (m)

A : Matrix defined in state-space process model

B : Matrix defined in state-space process model

B : Bottom flow rate in distillation column, (kmol/h)

C : Matrix defined in state-space measurement model

d: Process time delay

D : Distillate flow rate, (kmol/h)

f : Functional vector

F_i : Measured feed flow rate entering storage tank, (m^3/h)

F_o : Outlet flow rate leaving storage tank, (m^3/h)

F : Feed flow rate to distillation column, (kmol/h)

$\mathbf{G}_U$: Matrix of Laplace transfer functions for manipulated variables

$\mathbf{G}_D$: Matrix of Laplace transfer functions for external disturbances

h : Storage tank level, (cm)

H_D : Reflux drum level of distillation column, (m)

H_B : Distillation column base level, (m)

K_C : Controller gain

K : Matrix of DDR filter gain

L : Window length of MA filter

P : Matrix used to calculate Kalman gain

$\mathbf{P}^-$: Matrix used to calculate Kalman gain

Q : Reboiler heat duty of distillation column, (MJ/h)

R : Reflux flow rate returning to distillation column, (kmol/h)

R : Covariance matrix of model prediction error

$\mathbf{S}$: Covariance matrix of process noise

t_S: Total number of process sampling time

T_D: Top temperature of distillation column, (°C)

T_B: Base temperature of distillation column, (°C)

$\mathbf{u}$: Vector of inputs including manipulated and disturbance variables

$\mathbf{V}$: Covariance matrix of the measurement errors

$\mathbf{w}$: Vector of process noise

X: Feed composition of benzene, (mole fraction)

$\mathbf{x}$: Vector of true values of process output variables

$\hat{\mathbf{x}}$: Vector of estimates (reconciled) for process output variables

$\mathbf{y}$: Vector of measured values for output variables

$\hat{\mathbf{y}}$: Vector of model predictions

Δt: Sampling time interval

Greek letters

α: Weighting factor for ISE

β: Weighting factor for ISDU

ε: Vector of measurement noise

δ: Vector of model prediction error

υ^2: Variance of model predictions

σ^2: Variance of measurements

κ: Matrix of Kalman gain

Φ: Cost function for controller performance

θ: Tuning parameter in EWMA filter

τ_I: Integral time constant for PI controller

Superscripts, subscripts and overscripts

t: Sampling time

*: Setpoint of controller

^: Estimated

f: Filtered

m: Measured

$\bar{-}$: Average

Acronyms

DDR: Dynamic data reconciliation

EWMA: Exponentially weighted moving average

ISDU: Integral of squared differences of the manipulated variable

ISE: Integral of squared errors of controlled variable

MA: Moving average

REFERENCES

Abu-el-zeet, Z.H., P.D. Roberts and V.M. Becerra, "Enhancing model predictive control by dynamic data reconciliation", *AIChE J.*, **48**, 324-333 (2002).

Albuquerque, J. and L.T. Biegler, "Data reconciliation and gross error detection for dynamic systems", *AIChE J.*, **42**, 2841-2856 (1996).

Bai, S., "Assessment of controller performance with embedded data reconciliation", Master thesis, University of Ottawa, Ottawa, Canada (2003).

Binder, T., L. Blank, W. Dahmen and W. Marquardt, "On the regularization of dynamic data reconciliation problems", *J. Process Control*, **12**, 557-567 (2002).

Liebman, M.J., T.F. Edgar and L.S. Lasdon, "Efficient data reconciliation and estimation for dynamic process using nonlinear programming techniques", *Computers Chem. Engng*, **16**, 963-986 (1992).

Makni, S., D. Hodouin and C. Bazin, "A recursive model imbalance method incorporating a model of flow rate dynamics for online material balance of complex flowsheets", *Minerals Engineering*, **8**, 753-766 (1995).

Ramamurthi, Y., P.B. Sistu and B.W. Bequette, "Control-relevant dynamic data reconciliation and parameter estimation", *Computers Chem. Engng*, **17**, 41-59 (1993).

Romagnoli, J.A. and M.C. Sanchez, "*Data processing and reconciliation for chemical process operations*", Academic Press, San Diego (2000).

Roffel, B. and P. Chin, "*Computer control in the process industries*", Lewis Publishers, Chelsea, Michigan (1987).

Wilson, D.I.; M. Agarwal and D.W.T. Rippin, "Experiences implementing the extended Kalman filter on an industrial batch reactor", *Computer Chem. Engng*, **22**, 1653-1672 (1998).

APPENDIX A

In Equation (21), the matrices of the transfer functions $\mathbf{G}_U(s)$ and $\mathbf{G}_D(s)$ are given by

$$\mathbf{G}_U(s)=\begin{bmatrix} \frac{-0.1087}{s} & \frac{-0.1087}{s} & 0 & \frac{0.0038e^{-10s}}{s} \\ 0 & \frac{-0.0621e^{-14s}}{176s+1} & 0 & \frac{0.0297e^{-120s}}{s} \\ 0 & \frac{0.1228e^{-30s}}{s} & \frac{-0.1382}{s} & \frac{-0.0037e^{-10s}}{s} \\ 0 & \frac{-1.8344e^{-218s}}{3672s+1} & 0 & \frac{0.0031e^{-28s}}{126s+1} \end{bmatrix} \tag{A-1}$$

$$\mathbf{G}_D(s)=\begin{bmatrix} 0 & \frac{-9.0e^{-20s}}{s} \\ \frac{-0.0426e^{-20s}}{s} & \frac{-18.88e^{-95s}}{372s+1} \\ \frac{0.1429e^{-10s}}{s} & \frac{9.1e^{-40s}}{s} \\ \frac{-0.1335e^{-60s}}{513s+1} & \frac{-114.7e^{-150s}}{917s+1} \end{bmatrix} \tag{A-2}$$

The empirical models of Equations (21) were discretized by z-transformation. The discretized and simplified models are given by

$$\begin{aligned} H_{D,t}+H_{D,t-1}=&-9.058\times10^{-4}D_{t-1}-9.058\times10^{-4}R_{t-1}+2.112\times10^{-5}Q_{t-1} \\ &+1.06\times10^{-5}Q_{t-2}-2.5\times10^{-2}X_{t-1}-5.0\times10^{-2}X_{t-2} \end{aligned} \tag{A-3}$$

$$\begin{aligned} T_{D,t}-0.9414T_{D,t-1}=&-5.64\times10^{-3}R_{t-1}-4.332\times10^{-3}R_{t-2}+2.5\times10^{-4}Q_{t-5} \\ &-1.184\times10^{-4}F_{t-1}-2.368\times10^{-4}F_{t-2}-1.227X_{t-4}-0.233X_{t-5} \end{aligned} \tag{A-4}$$

$$H_{B,t} - H_{B,t-1} = 1.023\times10^{-3}R_{t-2} - 1.152\times10^{-3}B_{t-1} - 2.056\times10^{-5}Q_{t-1} - 1.028\times10^{-5}Q_{t-2} + 7.942\times10^{-4}F_{t-1} + 3.971\times10^{-4}F_{t-2} + 5.06\times10^{-2}X_{t-2} + 2.53\times10^{-3}X_{t-3} \quad \text{(A-5)}$$

$$T_{B,t} - 0.9228T_{B,t-1} = -0.011R_{t-8} - 0.00385R_{t-9} + 4.867\times10^{-5}Q_{t-1} + 6.084\times10^{-4}Q_{t-2} - 7.583\times10^{-3}F_{t-3} - 3.69X_{t-6} \quad \text{(A-6)}$$

where all the variables are in deviation forms.

CHAPTER IV

Dynamic Data Reconciliation: Alternative to Kalman Filter

Shuanghua Bai, Jules Thibault and David D. McLean*

Department of Chemical Engineering

University of Ottawa

Ottawa, Ontario, Canada K1N 6N5

Published in **Journal of Process Control,**

Volume 16, 485-498, 2006.

* Corresponding author, Tel: 613-562-5800 ext. 6110

Email: mclean@genie.uottawa.ca

Preface

Chapter 4 rigorously derives the DDR algorithm based on statistical properties of measurements and models using Bayesian arguments. The DDR algorithm has a convenient predictor-corrector format akin to the Kalman filter. The relationships between the DDR and the Kalman filters are identified and evaluated. Modifications to the DDR algorithm and the Kalman filter are introduced to allow applications to situations involving autocorrelated measurement noise.

ABSTRACT

Process measurements are often corrupted with varying degrees of noise. Measurement noise undermines the performance of process monitoring and control systems. To reduce the impact of measurement noise, exponentially-weighted moving average and moving average filters are commonly used. These filters have good performance for processes under steady state or with slow dynamics. For processes with significant dynamics, more sophisticated filters, such as model-based filters, have to be used. The Kalman filter is a well known model-based filter that has been widely used in the aerospace industry. This paper discusses another model-based filter, the dynamic data reconciliation (DDR) filter. Both the Kalman and the DDR filters adhere to the same basic principle of using information from both measurements and models to provide a more reliable representation of the current state of the process. However, the DDR filter can more easily incorporate a wide variety of model structures and is easier to understand and implement. Simulation results for a binary distillation column with four controlled variables showed that the DDR filters had equivalent performance to the Kalman filter in dealing with both white and autocorrelated noise.

Keywords: Measurement noise, Dynamic data reconciliation, Kalman filter

1 INTRODUCTION

Sampled process data are affected by measurement noise. Noisy measurements not only mask the true values of process variables, but, when used to calculate control moves, they may significantly deteriorate controller performance. In such cases the manipulated variables can display excessive adjustments even becoming saturated, and possibly upsets in other process variables. As a result, the controllers must be detuned to regain acceptable performance. However, detuned controllers lead to more sluggish response of controlled variables. In order to reduce the impact of measurement noise, filters such as exponentially-weighted moving average (EWMA) and moving average (MA) filters have been widely applied to measurement signals before they are transmitted to controllers. The EWMA and MA filters employ current and past measurements to estimate the current state of measured variables. These filters are acceptable for steady-state operation or processes with slow dynamics. Although these filters can effectively reduce the effects of measurement noise, they may introduce unacceptable time delays for processes having significant dynamics.

For such processes, it is beneficial to use more advanced filters. The well-known Kalman filter uses underlying dynamic models to estimate the current state of process variables, and has acquired a reputation as a panacea for process state estimation and prediction. Although applications of Kalman filters in tracking moving objects such as aircraft and missiles have been common, their applications in chemical engineering are relatively infrequent [1, 2]. As Brosilow and Joseph [3] pointed out, the difficulties in implementing Kalman filters in the chemical industries are associated with identifying reliable dynamic models for the process as well as specifying the process noise terms required in the Kalman filter.

Recently, Bai et al. [4] proposed an alternative model-based approach, based on dynamic data reconciliation (DDR), to attenuate measurement noise inside control loops. The DDR filter, in a predictor-corrector format, used dynamic process models as redundant information to complement that in the process measurements. At each sampling time, the

filtered data were obtained by reconciling values of model predictions and raw measurements. It was demonstrated that the DDR filter was an effective tool to attenuate propagation of measurement noise inside control loops. It yielded better performance than EWMA and MA filters for dynamic processes when sufficient redundancy was available. Due to noise reduction, the DDR filter allowed more aggressive controller tuning and, as a result, controller performance was significantly enhanced. In this work, we compare its structure and performance with that of the Kalman filter and extend the form for both filters to deal with autocorrelated measurements in the context of multivariable control of a binary distillation column.

We first provide a more rigorous development of the DDR objective function using Bayesian principles, along with the predictor-corrector form of the DDR filter and expressions for estimating the covariance of the filter predictions. In addition, we show, under typical assumptions regarding the error structure of the models and measurements, that the DDR filter predictions are unbiased and that their variance is always smaller than that of predictions using either the measurements or the model on their own. The filter form is then modified to handle cases involving autocorrelated measurements.

A brief overview of the Kalman filter is provided in Section 3 along with its modification to deal with autocorrelated measurements. In Section 4, closed-loop performances of the use of the DDR and Kalman filters are compared, via simulation for control of a binary distillation column under different noise structures. Conclusions are summarized in Section 5.

2 DYNAMIC DATA RECONCILIATION

Plant measurements are stochastic variables and plant models display some degree of uncertainty. The problem is to estimate the state of the dynamic process given the information provided by measurements and plant models. Consider a process with M state variables that are monitored and/or controlled, subject to random variation. The true values of the M process variables at time t can be represented by a M×1 vector, $\mathbf{x}_t$. The

measured values of the M process variables, $\mathbf{y}_t$, are assumed to be adequately described by the additive noise model

$$\mathbf{y}_t = \mathbf{x}_t + \boldsymbol{\varepsilon}_t \tag{1}$$

where $\boldsymbol{\varepsilon}_t$ is a M×1 vector of the measurement noise associated with each of the M measured values at time t. The goal is to estimate $\mathbf{x}_t$ based on the available information in the measurements, $\mathbf{y}_t$, $\mathbf{y}_{t-1}$, ..., $\mathbf{y}_1$. The resulting estimate of $\mathbf{x}_t$, denoted as $\hat{\mathbf{x}}_t$, can be expressed as

$$\hat{\mathbf{x}}_t = \varphi(\mathbf{y}_t, \mathbf{y}_{t-1}, \ldots, \mathbf{y}_1) \tag{2}$$

where φ represents the functional form of the estimator. This function can be linear or nonlinear. However, given only Equation (1) and the current and past measurements, $\hat{\mathbf{x}}_t$ cannot be computed. Other information beyond the current and past measurements is required. Indeed, information about $\mathbf{x}_t$ or $\boldsymbol{\varepsilon}_t$ is needed in the estimation of $\mathbf{x}_t$.

Assuming that $\boldsymbol{\varepsilon}_t$ is normally distributed ($\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \mathbf{V})$) for all t and where $\mathbf{V}$ is the M×M covariance matrix of $\boldsymbol{\varepsilon}_t$, the probability density function of $\boldsymbol{\varepsilon}_t$ can be written as

$$f(\boldsymbol{\varepsilon}_t) = \frac{1}{(2\pi)^{M/2} |\mathbf{V}|^{1/2}} \exp\left(-\frac{1}{2} \boldsymbol{\varepsilon}_t^T \mathbf{V}^{-1} \boldsymbol{\varepsilon}_t \right) \tag{3}$$

where $|\mathbf{V}|$ is the determinant of $\mathbf{V}$ and $\boldsymbol{\varepsilon}_t^T$ is the transpose of $\boldsymbol{\varepsilon}_t$. From Equation (1), and assuming $\mathbf{x}_t$ and $\boldsymbol{\varepsilon}_t$ are independent, it follows that the conditional density function of measurements $\mathbf{y}_t$, given $\mathbf{x}_t$, is

$$f(\mathbf{y}_t | \mathbf{x}_t) = f(\mathbf{y}_t - \mathbf{x}_t) = f(\boldsymbol{\varepsilon}_t) \tag{4}$$

Since $\varepsilon_t \sim N(\mathbf{0}, \mathbf{V})$, $f(\mathbf{y}_t \mid \mathbf{x}_t)$ is Gaussian with mean $\mathbf{x}_t$ and covariance $\mathbf{V}$, and the probability density function for $\mathbf{y}_t$ can be interpreted as the likelihood of $\mathbf{x}_t$ given the measurements, $\mathbf{y}_t$, denoted as

$$L(\mathbf{x}_t \mid \mathbf{y}_t) = \frac{1}{(2\pi)^{M/2} \mid \mathbf{V} \mid^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{y}_t - \mathbf{x}_t)^T \mathbf{V}^{-1} (\mathbf{y}_t - \mathbf{x}_t) \right] \tag{5}$$

Now the problem is to find the most likely value of $\mathbf{x}_t$ (i.e., the value that maximizes $L(\mathbf{x}_t \mid \mathbf{y}_t)$). These estimates are known as maximum likelihood (ML) estimates. From Equation (5), it is clear that the maximum occurs when $\mathbf{y}_t = \mathbf{x}_t$. Therefore, the ML estimates are $\hat{\mathbf{x}}_t = \mathbf{y}_t$, which means that the optimal estimates of the true values of process variables are simply the raw measurements when the only available information on measurement noise is $\varepsilon_t \sim N(\mathbf{0}, \mathbf{V})$.

Fortunately, in many cases, prior information about $\mathbf{x}_t$ is available. One common source of information about $\mathbf{x}_t$ can come from a wide variety of process models which can be phenomenological or empirical, discrete or continuous, linear or nonlinear. For example, $\mathbf{x}_t$ can be estimated from model predicted values, $\hat{\mathbf{y}}_t$ (e.g., a one-step-ahead prediction). The information about the measurement noise, ε_t, can be combined with the information about the model predictions, $\hat{\mathbf{y}}_t$ of $\mathbf{x}_t$, to give better estimates of $\mathbf{x}_t$. Because process models can rarely be considered to be perfect, model predicted values also contain some degree of error. Therefore, we assume the model predictions, $\hat{\mathbf{y}}_t$, can be expressed in an additive noise model,

$$\hat{\mathbf{y}}_t = \mathbf{x}_t + \delta_t \tag{6}$$

where δ_t represents the vector of model prediction error which is a complex function of numerous factors, such as incorrect model structure, uncertainty in model parameters, unaccounted model inputs as well as inaccurate measured input variables. For simplicity, we assume $\delta_t \sim N(\mathbf{0}, \mathbf{R})$ where $\mathbf{R}$ is the covariance matrix of the prediction error. The problem is now to find the best estimates of $\mathbf{x}_t$ given both the measured data and the predicted values. Using Bayes' formula [5] we can write

$$f(\mathbf{x}_t \mid \mathbf{y}_t) \propto L(\mathbf{x}_t \mid \mathbf{y}_t) f(\mathbf{x}_t) \tag{7}$$

where $f(\mathbf{x}_t \mid \mathbf{y}_t)$ is the posterior distribution of $\mathbf{x}_t$ given the measured data, $L(\mathbf{x}_t \mid \mathbf{y}_t)$ is the likelihood function and, $f(\mathbf{x}_t)$ is the prior distribution which expresses our knowledge about $\mathbf{x}_t$ prior to having the data. If we now add independent information about $\mathbf{x}_t$, based on model predictions, $\hat{\mathbf{y}}_t$, we can apply Bayes' formula again, giving

$$f(\mathbf{x}_t \mid \hat{\mathbf{y}}_t, \mathbf{y}_t) \propto L(\mathbf{x}_t \mid \hat{\mathbf{y}}_t) f(\mathbf{x}_t \mid \mathbf{y}_t) \propto L(\mathbf{x}_t \mid \hat{\mathbf{y}}_t) L(\mathbf{x}_t \mid \mathbf{y}_t) f(\mathbf{x}_t) \tag{8}$$

where $f(\mathbf{x}_t \mid \hat{\mathbf{y}}_t, \mathbf{y}_t)$ represents our posterior knowledge of $\mathbf{x}_t$ based on both $\mathbf{y}_t$ and $\hat{\mathbf{y}}_t$.

Based on our assumptions about the measured data and the model predictions, it follows that

$$L(\mathbf{x}_t \mid \mathbf{y}_t) = \frac{1}{(2\pi)^{M/2} \mid \mathbf{V} \mid^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{y}_t - \mathbf{x}_t)^T \mathbf{V}^{-1} (\mathbf{y}_t - \mathbf{x}_t) \right] \tag{9}$$

$$L(\mathbf{x}_t \mid \hat{\mathbf{y}}_t) = \frac{1}{(2\pi)^{M/2} \mid \mathbf{R} \mid^{1/2}} \exp\left[-\frac{1}{2} (\hat{\mathbf{y}}_t - \mathbf{x}_t)^T \mathbf{R}^{-1} (\hat{\mathbf{y}}_t - \mathbf{x}_t) \right] \tag{10}$$

If we also assume that our initial prior information is noninformative (i.e., $f(\mathbf{x}_t)$ is constant over the range of $\mathbf{x}_t$ values of interest) then our posterior distribution

becomes

$$f(\mathbf{x}_t \mid \mathbf{y}_t, \hat{\mathbf{y}}_t) = \frac{k}{(2\pi)^M \mid \mathbf{VR} \mid^{1/2}} \exp\left\{-\frac{1}{2}\left[(\mathbf{y}_t - \mathbf{x}_t)^T \mathbf{V}^{-1}(\mathbf{y}_t - \mathbf{x}_t) + (\hat{\mathbf{y}}_t - \mathbf{x}_t)^T \mathbf{R}^{-1}(\hat{\mathbf{y}}_t - \mathbf{x}_t)\right]\right\} \tag{11}$$

where k is a constant. We select the value of $\mathbf{x}_t$ which maximizes Equation (11) as our best estimate of the true value of the vector of process states. We refer to these estimates as maximum a posteriori (MAP) estimates and denote them as $\hat{\mathbf{x}}_t$.

The maximization of Equation (11) is equivalent to minimizing the positive argument of the exponential in Equation (11). The MAP estimate, $\hat{\mathbf{x}}_t$, consequently can be obtained by minimizing

$$J(\hat{\mathbf{x}}_t) = \frac{1}{2}\left[(\mathbf{y}_t - \mathbf{x}_t)^T \mathbf{V}^{-1}(\mathbf{y}_t - \mathbf{x}_t) + (\hat{\mathbf{y}}_t - \mathbf{x}_t)^T \mathbf{R}^{-1}(\hat{\mathbf{y}}_t - \mathbf{x}_t)\right] \tag{12}$$

2.1 Predictor-corrector algorithm for DDR

Solving Equation (12) yields the predictor-corrector form of the DDR algorithm (see Appendix A) given as

$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}}_t + \mathbf{K}(\mathbf{y}_t - \hat{\mathbf{y}}_t) \tag{13}$$

where

$$\mathbf{K} = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1} = (\mathbf{I} + \mathbf{VR}^{-1})^{-1} \tag{14}$$

and $\mathbf{I}$ is the identity matrix. Equations (13) and (14) are the expressions of the DDR filter.

Equation (13) indicates that $\hat{\mathbf{x}}_t$ is composed of two terms that are respectively the model predicted values, $\hat{\mathbf{y}}_t$, plus the measurement corrections $\mathbf{K}(\mathbf{y}_t - \hat{\mathbf{y}}_t)$. $\mathbf{K}$ in Equation (13) is the gain of the DDR filter. From Equation (14), it is seen that $\mathbf{K} \to \mathbf{0}$ when measurement errors are significantly larger than model prediction errors; consequently $\hat{\mathbf{x}}_t \to \hat{\mathbf{y}}_t$. On

the other hand, when the model predictions have significant errors, $\mathbf{K} \to \mathbf{I}$, and then $\hat{\mathbf{x}}_t \to \mathbf{y}_t$.

2.2 DDR estimation error

The estimation error, ξ_t, of the DDR filter is given by

$$\xi_t = \hat{\mathbf{x}}_t - \mathbf{x}_t \tag{15}$$

Putting Equation (13) into (15) and using Equations (1) and (6) results in

$$\xi_t = \delta_t + \mathbf{K}(\varepsilon_t - \delta_t) \tag{16}$$

Taking expectations of both sides of Equation (16) yields

$$E[\xi_t] = \mathbf{K}E[\varepsilon_t] + (\mathbf{I} - \mathbf{K})E[\delta_t] \tag{17}$$

Since $E[\varepsilon_t] = \mathbf{0}$ and $E[\delta_t] = \mathbf{0}$, $E[\xi_t] = \mathbf{0}$. Therefore, $\hat{\mathbf{x}}_t$ is an unbiased estimator. From Equation (16), the covariance matrix of the estimation error, or the covariance matrix of the reconciled values by DDR, can be obtained as

$$\mathrm{Cov}(\xi_t) = \mathbf{K}^{\mathrm{T}}\mathrm{Cov}(\varepsilon_t)\mathbf{K} + (\mathbf{I} - \mathbf{K})^{\mathrm{T}}\mathrm{Cov}(\delta_t)(\mathbf{I} - \mathbf{K}) \tag{18}$$

Putting $\mathrm{Cov}(\varepsilon_t) = \mathbf{V}$, $\mathrm{Cov}(\delta_t) = \mathbf{R}$ and $\mathbf{K} = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}$ into Equation (18) and rearranging yields

$$\mathrm{Cov}(\xi_t) = \mathrm{Cov}(\hat{\mathbf{x}}_t) = \left(\mathbf{V}^{-1} + \mathbf{R}^{-1}\right)^{-1} \tag{19}$$

Equation (19) indicates that the covariance matrix of the estimation error of the DDR filter for the M process variables is a function of $\mathbf{V}$ and $\mathbf{R}$. Because both $\mathbf{V}$ and $\mathbf{R}$ are

positive definite symmetric matrices, the elements in $\mathrm{Cov}(\xi_t)$ are less than those in $\mathbf{V}$ and $\mathbf{R}$, meaning that combination of the measured values and model predicted values results in more precise estimates than using only one of them.

2.3 Modified DDR algorithm for autocorrelated measurements

The DDR filter algorithm derived in Section 2.1 is based on the assumption that the measurement error is white Gaussian noise. For the case where the noise is autocorrelated, Equation (13) has to be modified to incorporate the correlation. Accordingly, we can write the autocorrelated measurement noise for all measurements in the form of a weighted sum of previous values driven by white Gaussian noise

$$\boldsymbol{\varepsilon}_t = \sum_{i=1}^{q} \boldsymbol{\Phi}_i \boldsymbol{\varepsilon}_{t-i} + \boldsymbol{\alpha}_t \tag{20}$$

where $\boldsymbol{\Phi}_i$ is a $M \times M$ matrix of coefficients, $\boldsymbol{\alpha}_t$ is a $M \times 1$ vector of white Gaussian noise, and q is the order of the noise model. The measurement noise at each time t-1, $\boldsymbol{\varepsilon}_{t-1}$, can be estimated by

$$\hat{\boldsymbol{\varepsilon}}_{t-1} = \mathbf{y}_{t-1} - \hat{\mathbf{x}}_{t-1} \tag{21}$$

Using Equations (20) and (21), the estimated measurement noise at time t can be obtained by the one-step-ahead prediction

$$\hat{\boldsymbol{\varepsilon}}_t = \sum_{i=1}^{q} \boldsymbol{\Phi}_i \hat{\boldsymbol{\varepsilon}}_{t-i} \tag{22}$$

After the realization of the measurements at time t, $\hat{\boldsymbol{\varepsilon}}_t$ is subtracted from $\mathbf{y}_t$ and, under ideal conditions (model mismatch for the process model and the measurement error model is zero), the pre-whitened measurements, $(\mathbf{y}_t - \hat{\boldsymbol{\varepsilon}}_t)$, only contain white Gaussian noise. Substituting $(\mathbf{y}_t - \hat{\boldsymbol{\varepsilon}}_t)$ for $\mathbf{y}_t$ in Equation (13), the reconciled data vector $\hat{\mathbf{x}}_t$ is

calculated. Repeating the above procedure continues the dynamic data reconciliation algorithm. Information flow to deal with the autocorrelated measurement noise is schematically illustrated in Figure 1. It is worth noting that the filter gain matrix, $\mathbf{K}$, is calculated based on the pre-whitened measurements. However, in practice, since $(\mathbf{y}_t - \hat{\boldsymbol{\varepsilon}}_t)$ may not represent the fully pre-whitened measurements due to potential inadequacies of the process and the measurement error models, the covariance matrix of $(\mathbf{y}_t - \hat{\boldsymbol{\varepsilon}}_t)$ is inflated and, as a result, the gain, $\mathbf{K}$, will need to be detuned.

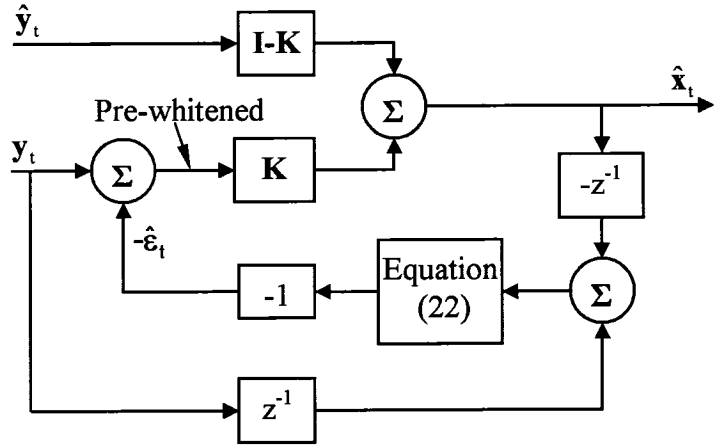


Figure 1 Information flow diagram to deal with autocorrelated noise using DDR filter. z^{-1} represents the backshift operator such that $z^{-1}y_t = y_{t-1}$.

3 KALMAN FILTER

The Kalman filter also employs information from both measurements and process models. Dynamic state-space models, assumed linear and stochastic, are used to describe the dynamics of the process, and additive noise models are used to represent raw measurements. The process state/measurement models used in the Kalman filter are written as

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} + \mathbf{w}_{t-1}, \tag{23}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \boldsymbol{\varepsilon}_t \tag{24}$$

where **A**, **B**, and **C** are deterministic matrices having appropriate dimensions. The input vector $\mathbf{u}_{t-1}$ containing manipulated and disturbance variables can also embrace values of inputs at several previous time steps (e.g., $\mathbf{B}_1\mathbf{u}_{t-2}$, $\mathbf{B}_2\mathbf{u}_{t-3}$, ...). $\mathbf{w}_t$ denotes process model noise assumed to be white Gaussian noise, $\mathbf{w}_t \sim \mathrm{N}(\mathbf{0}, \mathbf{S})$, accounting for: (i) improper model structure (e.g., lower order approximation for a higher order process); (ii) inaccurate values for elements in matrices **A** and **B**; (iii) inaccurate implementation of inputs $\mathbf{u}_{t-1}$; (iv) other unaccounted disturbances to the process. $\boldsymbol{\varepsilon}_t$ is the measurement noise, also assumed to have white Gaussian form. $\mathbf{w}_t$ and $\boldsymbol{\varepsilon}_t$ are assumed independent.

3.1 Kalman filter under white noise

For the process defined by Equations (23-24), it is desired to find an estimate of the state vector $\mathbf{x}_t$ based on known quantities $\mathbf{y}_t$ and $\mathbf{u}_{t-1}$ at time t. The Kalman filter provides an estimator having minimum prediction variance. Detailed derivation of the Kalman filter has been well documented in literature [6]. Practical implementation of the Kalman filter consists of the following steps:

1. *Filter initialization.* At time t = 0, initialize the filter with

$$\hat{\mathbf{x}}_0 = \text{guess of } \hat{\mathbf{x}}_0, \tag{25}$$

$$\mathbf{P}_0 = \text{guess of covariance of estimation errors, } \mathrm{E}\left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^{\mathrm{T}}\right]. \tag{26}$$

Set t = 1, start the Kalman filter recursive calculations.

2. *Model prediction.* At time t, calculate the model prediction with

$$\hat{\mathbf{y}}_t = \mathbf{A}\hat{\mathbf{x}}_{t-1} + \mathbf{B}\mathbf{u}_{t-1}, \tag{27}$$

and the covariance matrix of the model predictions with

$$\mathbf{P}_t^- = \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^{\mathrm{T}} + \mathbf{S}. \tag{28}$$

3. *Measurement correction.* At time t, after the measurements $\mathbf{y}_t$ are available, compute the a posteriori quantities

$$\mathbf{K}_t = \mathbf{P}_t^-\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\mathbf{P}_t^-\mathbf{C}^{\mathrm{T}} + \mathbf{V}\right)^{-1}, \tag{29}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}}_t + \mathbf{K}_t\left(\mathbf{y}_t - \mathbf{C}\hat{\mathbf{y}}_t\right), \tag{30}$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t \mathbf{C} \mathbf{P}_t^- \tag{31}$$

where $\mathbf{P}_t$ is the covariance matrix of estimation error at time t, and $\mathbf{K}_t$ is known as the Kalman gain.

4. *Time increment*. Increment time index and repeat calculations starting at step 2.

Because covariance matrices $\mathbf{P}_t^-$ and $\mathbf{P}_t$ and the Kalman gain $\mathbf{K}_t$ need to be updated at each sampling time, the Kalman filter is a time-variant system. However, for the process defined by Equations (23-24), matrices $\mathbf{P}_t^-$, $\mathbf{P}_t$ and $\mathbf{K}_t$ reach a steady state and take on constant values. At this point, the Kalman filter becomes time-invariant, and the Kalman filter is known as the steady-state Kalman filter. It is noted that the recursive calculations of the matrices $\mathbf{P}_t^-$, $\mathbf{P}_t$ and $\mathbf{K}_t$ are independent of the measurement realizations $\mathbf{y}_t$. Therefore, they can be computed offline without actually making any measurements.

Assuming all state variables of interest are measured, $\mathbf{C} = \mathbf{I}$. For this case the Kalman gain, Equation (29), reduces to

$$\mathbf{K}_t = \mathbf{P}_t^- \left(\mathbf{P}_t^- + \mathbf{V}\right)^{-1} = \left(\mathbf{I} + \mathbf{V}\mathbf{P}_t^{-1}\right)^{-1} \tag{32}$$

Because $\mathbf{P}_t^-$ is the covariance matrix of model predictions, it is consequently equivalent to $\mathbf{R}$ used in the DDR filter. Replacing $\mathbf{P}_t^-$ by $\mathbf{R}$ in Equation (32) shows that the gain of the Kalman filter is equivalent to the gain for the DDR filter (see Equation (14)) under such restrictions. It can also be shown that $\mathbf{P}_t$ in Equation (31) is equivalent to the covariance of the reconciled values in the DDR filter in Equation (19). The Kalman filter is indeed a subset of the DDR filter since it only uses process state-space models and thus the DDR filter can be viewed as a more generalized model-based filter.

The Kalman filter employs linear state/measurement models. For many chemical engineering processes, especially dynamic processes, where values of the process variables change considerably, this assumption is unlikely to be valid. To overcome this

problem, the process model is linearized at each time step and the usual Kalman algorithm is used; this is known as the Extended Kalman filter (EKF). A considerable increase in complexity is introduced in this case. The matrices **A** and **C** become time-dependent as do matrices $\mathbf{P}_t^-$, $\mathbf{P}_t$ and $\mathbf{K}_t$. As a result, they must be calculated at every time step (rather than the single off-line calculation required for linear models). On the other hand, the DDR filter places no limits on the model form, but it may require numerical solution of the nonlinear model equations at each time step.

3.2 Kalman filter under autocorrelated measurement noise

The derivation of the Kalman filter is based on the assumption that the measurement error is white Gaussian noise. However, when the measurement noise is autocorrelated, the Kalman filter has to be augmented by incorporating the autocorrelation information into the structure of the process state/measurement models. This starts with transformation of the models of the autocorrelated measurement errors to yield a state-space format that can be combined with the process state/measurement models to formulate new state-space models with increased dimension. Then, an augmented Kalman filter is applied. Details of the mathematical manipulations to formulate the new state-space models are given in Appendix B.

4 SIMULATION EXAMPLES

The performance of the DDR and the Kalman filters in attenuating measurement noise was evaluated by simulating a multiple-loop control of a binary distillation column.

4.1 The distillation column

The distillation column, presented in Figure 2, has four PI control loops. Controllers TIC-D and TIC-B are used to control the top and bottom temperatures by manipulating the reflux flow rate and the flow of steam to the reboiler, respectively. Controllers LIC-D and LIC-B are used to control the reflux drum and column base liquid levels by manipulating the distillate flow rate and the bottom product flow rate, respectively. External disturbances to the column were made by changing the feed flow

rate. A sampling period of 30 seconds and a zero-order hold were used. Nominal steady-state values for all measurements and their Gaussian noise levels are listed in Table 1. The configuration of each control loop with embedded filter is depicted in Figure 3.

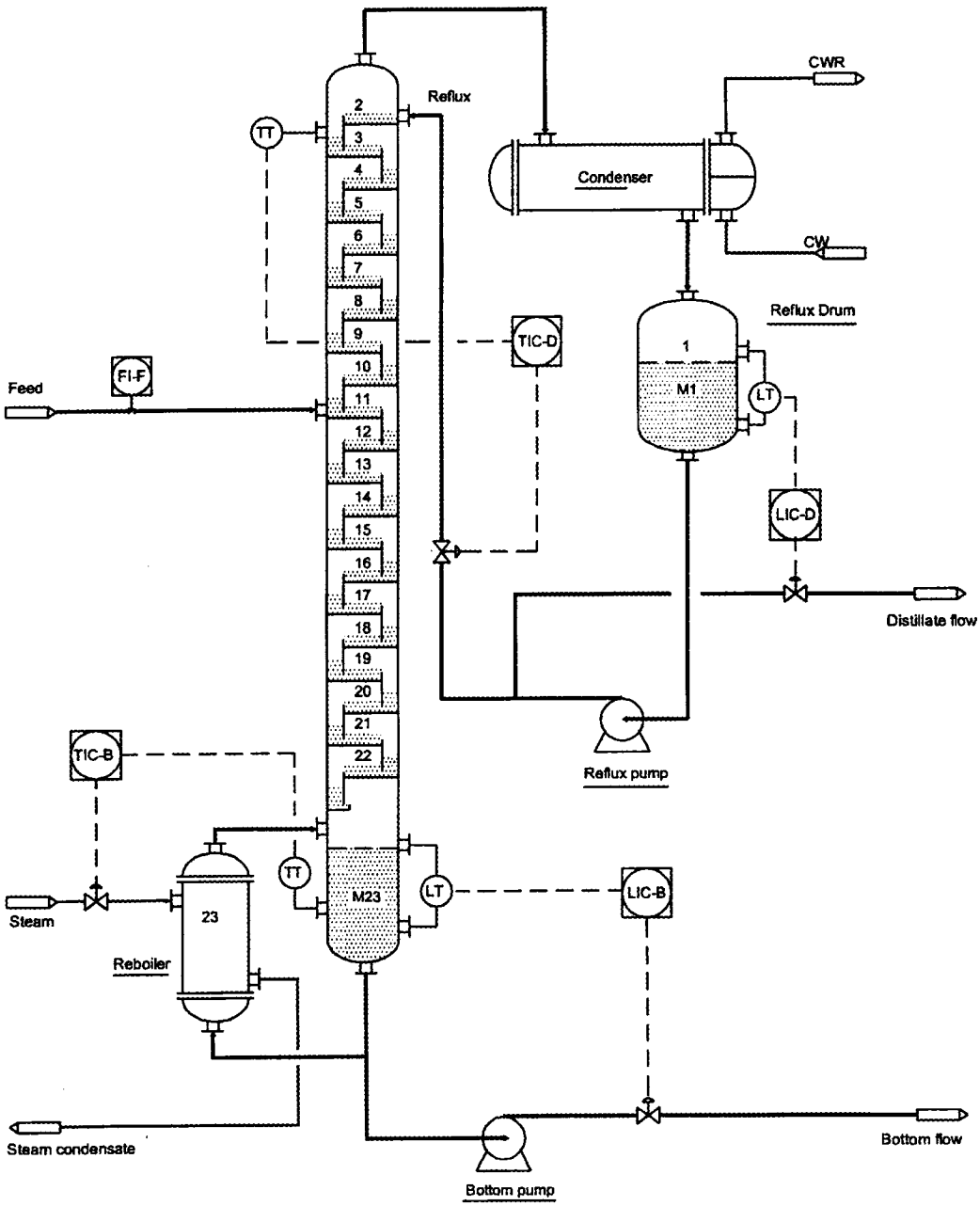


Figure 2 Schematic diagram of the distillation column.

Table 1 Nominal steady-state values and noise levels of measured variables

Measured variable	Units	Steady-state value	Standard deviation
Feed flow, F	kmol/h	100.0	2.0
Reflux drum level, H_D	cm	50.0	2.0
Top temperature, T_D	°C	84.2	0.25
Column base level, H_B	cm	70.0	2.0
Bottom temperature, T_B	°C	117.4	0.25

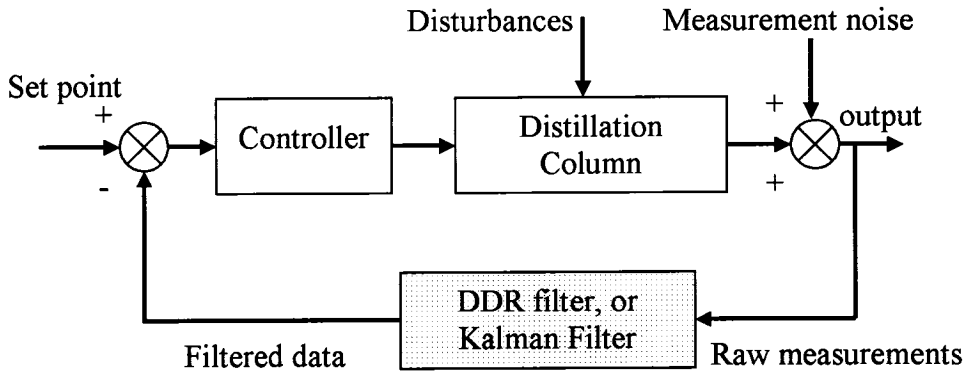


Figure 3 Schematic diagram of a control loop with filters.

A dynamic distillation simulator developed by Bai [7] was used in this study. This simulator is based on rigorous distillation models (mass and heat balances, vapor-liquid equilibria, and tray hydraulics). For each controlled variable, an empirical input-output model was identified around the process nominal steady state using step responses as described by Marlin [8]. These models were either first-order-plus-dead-time models or pure-integrator-plus-dead-time models. The discrete version of these models was obtained by z-transformation and is presented in Appendix C. One-step-ahead prediction errors of these models were computed by comparing true values of process variables to model predictions under nominal white noise conditions as listed in Table 1. The covariance matrix of model prediction errors was evaluated as

$$\mathrm{Cov}[\delta_t] = \mathbf{R} = \begin{array}{c} \begin{matrix} H_D & T_D & H_B & T_B \end{matrix} \\ \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 0.013 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0.014 \end{bmatrix} \end{array} \tag{33}$$

The cross-correlation and autocorrelation of the model predictions were found to be negligible and set to zero. It is important to note that, in practice, the exact values of the covariance matrix of model predictions may be difficult to determine. As a result, elements of **R** could be treated as tuning parameters. This is demonstrated in Section 4.3.

4.2 White noise case study

Since the measurement error for each of the four controlled variables was white Gaussian noise, without cross correlation, the measurement covariance matrix based on standard deviations given in Table 1 was

$$\mathrm{Cov}[\varepsilon_t] = \mathbf{V} = \begin{array}{c} \begin{matrix} H_D & T_D & H_B & T_B \end{matrix} \\ \begin{bmatrix} 4.0 & 0 & 0 & 0 \\ 0 & 0.0625 & 0 & 0 \\ 0 & 0 & 4.0 & 0 \\ 0 & 0 & 0 & 0.0625 \end{bmatrix} \end{array} \tag{34}$$

The DDR filter gain calculated by $\mathbf{K} = (\mathbf{I} + \mathbf{V}\mathbf{R}^{-1})^{-1}$ was

$$\mathbf{K}_{DDR} = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.172 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.183 \end{bmatrix} \tag{35}$$

Using these values for $\mathbf{K}_{DDR}$, closed-loop performance of the DDR filter was evaluated for a 20% step increase in the feed flow rate. Results for the raw measurements, filtered values and their true values for the four controlled variables are presented in Figure 4. Initially under steady state and without disturbance for the first 60 minutes, the four controlled variables displayed variations around their setpoints because of the internal propagation of measurement noise inside the control loops. With the external disturbance in the feed flow rate, the dynamic responses of the four controlled variables were

significant, and they returned to steady state after about 240 minutes under closed-loop conditions. Figure 4 shows that the data filtered by the DDR filter were less noisy than the raw measurements, and performed well in tracking the true values of controlled variables.

Next, a Kalman filter was tested to deal with the same sequence of white noise. As stated in Section 3.1, using the values of **R** in Equation (33) for $\mathbf{P}_t^-$ in Equation (32), the Kalman gain is equivalent to the DDR gain, Equation (35). Thus, knowing the Kalman gain, by trial-and-error it is possible to calculate the value of **S** defined in the Kalman filter, given by

$$\mathrm{Cov}(\mathbf{w}_t) = \mathbf{S} = \begin{bmatrix} 0.1999 & 0 & 0 & 0 \\ 0 & 0.0035 & 0 & 0 \\ 0 & 0 & 0.1999 & 0 \\ 0 & 0 & 0 & 0.0043 \end{bmatrix} \tag{36}$$

As a result, the DDR and Kalman filters were equivalent to each other and, of course yielded the same results.

In practice the values of **S** may not be known. To test the sensitivity of the Kalman filter to the values of **S**, the values of **R,** given in Equation (33), were substituted for **S** in Equation (36). Assuming the initial value, $\mathbf{P}_0 = \mathbf{I}$, the Kalman gain was calculated. After 11 iterations, the Kalman gain reached the steady state given by

$$\mathbf{K}_{\mathrm{Kalman}} = \begin{bmatrix} 0.390 & 0 & 0 & 0 \\ 0 & 0.336 & 0 & 0 \\ 0 & 0 & 0.390 & 0 \\ 0 & 0 & 0 & 0.339 \end{bmatrix} \tag{37}$$

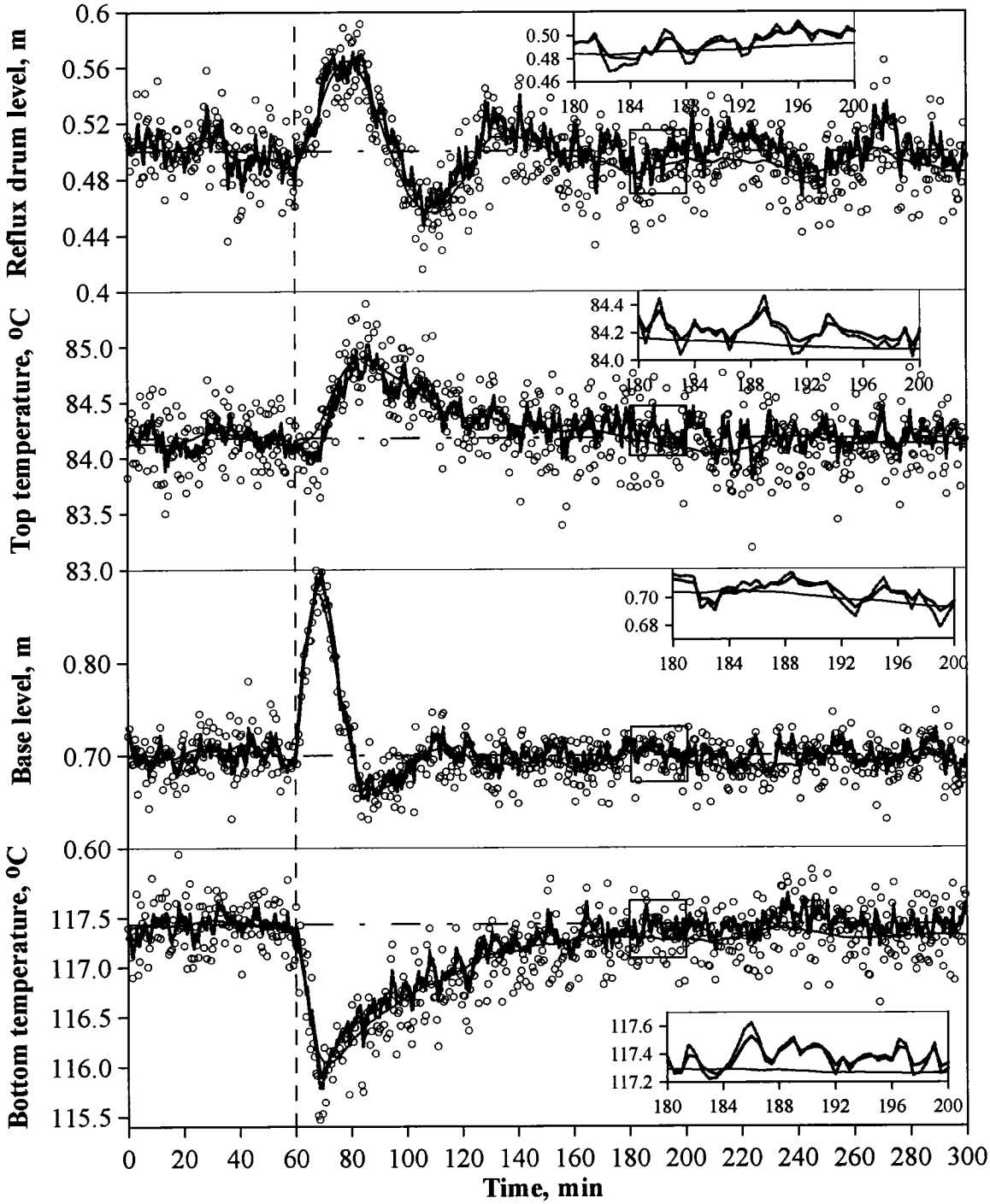


Figure 4 Performance of the DDR and Kalman filters to reduce white Gaussian noise for a 20% step increase in the feed flow rate at t = 60 min. Inserted expanded graphs more clearly show results for the DDR and Kalman filters for purposes of comparison. o Raw; ------ Kalman filter; —— DDR filter; — True; — - — Setpoint

The closed-loop performance of the Kalman filter was evaluated for the same feed flow rate disturbance. The simulation results are also presented in Figure 4. This figure shows that the filtered data obtained by the Kalman filter were very close to those obtained by the DDR filter. The data produced by the Kalman filter also displayed much less variability and tracked the true values of the controlled process variables well. The addition of either a DDR or Kalman filter smoothed the raw measurements, thereby reducing the propagation of measurement noise through the control loops. The differences shown in Figure 4 did not appear to have any practical significance.

In order to quantitatively evaluate the performance the two filters, the standardized mean squared error (MSE) of the filtered data for each variable defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{t=0}^{N}\left(\frac{\hat{x}_t - x_t}{\sigma}\right)^2 \tag{38}$$

where σ is the standard deviation of the nominal white measurement noise listed in Table 1, was introduced. The MSE values were calculated for the two filters for the same feed flow rate disturbance and identical white noise sequence for each simulation. One hundred and ten simulation runs were performed each with a different noise sequence. The differences in MSE values between the two filters, $(\mathrm{MSE}_{\mathrm{DDR}} - \mathrm{MSE}_{\mathrm{Kalman}})$, for each noise sequence were determined and histograms of these differences were created and are presented in Figure 5. Negative values of the difference indicate the DDR filter performed better, whereas positive values indicate the Kalman filter performed better. The mean of the differences, the standard deviation and the standard error of the mean evaluated for each variable are also presented in Figure 5. At the nominal variance level, it shows that the Kalman filter gave significantly larger MSE values for all controlled variables. This is the result of using an inappropriate value for **S.**

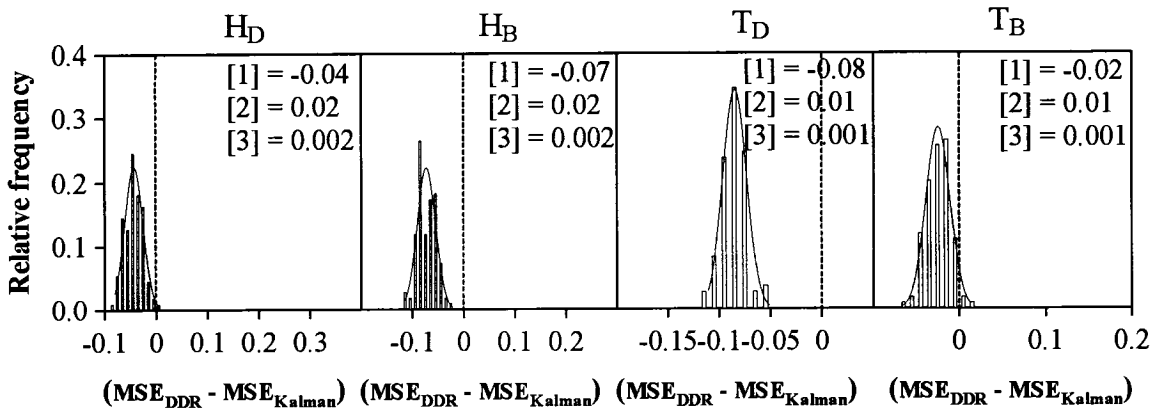


Figure 5 Histograms of the differences of MSE values obtained by DDR and Kalman filters for one hundred and ten white Gaussian noise sequences. [1] = Mean of the differences. [2] = Standard deviations of the differences. [3] = Standard deviation of the mean of the differences.

The effect of the magnitude of the measurement noise on the performance of the DDR filter was also examined. When the nominal variances of the measurements were doubled, the gain matrix of the DDR was re-calculated. For the same process disturbance, the MSE values for one hundred and ten white Gaussian noise sequences were evaluated for each controlled variable and their mean values and standard deviations are presented in Table 2. For comparison, the mean values and standard deviations for the nominal measurement variances are also presented in Table 2. The mean MSE value increased by 50% for the top temperature, while the mean MSE values for the other controlled variables approximately doubled. However, for the doubled measurement variances, using these variances to scale the MSE values, it is clear the mean MSE values for the four controlled variables reduce to 0.25, 0.08, 0.20 and 0.18, respectively. Compared to the case of the nominal measurement variances, the effectiveness of the DDR filter for noise reduction remained relatively unchanged even though the measurement noise level was doubled.

Table 2 Sample mean and standard deviations of the MSE values for each controlled variable obtained by the DDR filter.

Controlled variable	Nominal measurement variance		Doubled measurement variance	
	$\overline{MSE}$	σ_{MSE}	$\overline{MSE}$	σ_{MSE}
Reflux drum level	0.21	0.029	0.50	0.079
Top temperature	0.11	0.017	0.16	0.034
Column base level	0.19	0.026	0.41	0.077
Bottom temperature	0.19	0.023	0.36	0.046

The white noise case study demonstrates that both the DDR and Kalman filters are effective tools to reduce white noise propagation through feedback control loops. The gain matrix of the DDR is directly calculated by covariance matrices of model predictions **R** and measurements **V**, whereas the gain matrix of the Kalman filter is calculated iteratively using covariance matrices of process noise **S** and measurements **V** with an initial guess of the covariance matrix of estimation error. If both matrices **R** and **S** were known precisely, using a DDR or a Kalman filter should make no difference.

4.3 Treating R as tuning parameters in DDR

Often, neither the covariance matrix of model predictions used in the DDR filter, nor the covariance matrix of process model noise used in the Kalman filter can be easily evaluated. To deal with this problem, the elements in **R** or **S** can be treated as tuning parameters in the implementation of the DDR and Kalman filters.

To investigate this approach, the diagonal elements of matrix **R** were treated as tuning parameters in the DDR filter for the nominal white noise case. Matrix **R** in Equation (33) was multiplied by a factor of magnitude c. Matrix $c\mathbf{R}$ was used to calculate the DDR filter gain matrices, and then the DDR filters were implemented inside the control loops. For each DDR filter, the MSE values for each of the controlled variables were evaluated when the column was subjected to the same feed flow disturbance under the nominal

white noise. The MSE values for each variable are shown in Figure 6 as a function of the logarithm of the factor c, along with the values of the filter gain. The minimal MSE values for the four variables are at $c = 1$, which represents the optimal gain given by Equation (35). By decreasing the factor c, the gain of the filters decreased, meaning that the filter was placing more weight on model predictions. For higher values of c, corresponding to higher gains, a greater confidence was placed on the measured data. In both cases, moving away from the optimal filter gain, the MSE values increased as shown in Figure 6. This study shows that degradation of the DDR filter may occur if the DDR filter is poorly tuned. In addition, it is important to comment that, the performance of the DDR filter relies on the accuracy of the process models used to predict the true value of process variables. In this case study, linear models were used to represent the process dynamics in the vicinity of the nominal operating point. If process state shifts or operating point changes, the increased model mismatch of the linear models will deteriorate the DDR performance. To remedy these problems, process models have to be adapted or more accurate process models such as nonlinear models have to be used.

4.4 Autocorrelated noise case study

In practice, measurement noise is frequently autocorrelated to some degree due, for example, to the dynamics of measuring devices. In this case study, the performance of both DDR and Kalman filters was tested in the presence of autocorrelated noise. The measurement noise for each of the four controlled variables following a first-order autoregressive model was taken as the case study

$$\varepsilon_t = \lambda \varepsilon_{t-1} + \alpha_t \tag{39}$$

where λ is the autocorrelation coefficient and α_t is a white Gaussian noise whose standard deviation is identical to the one given in Table 1.

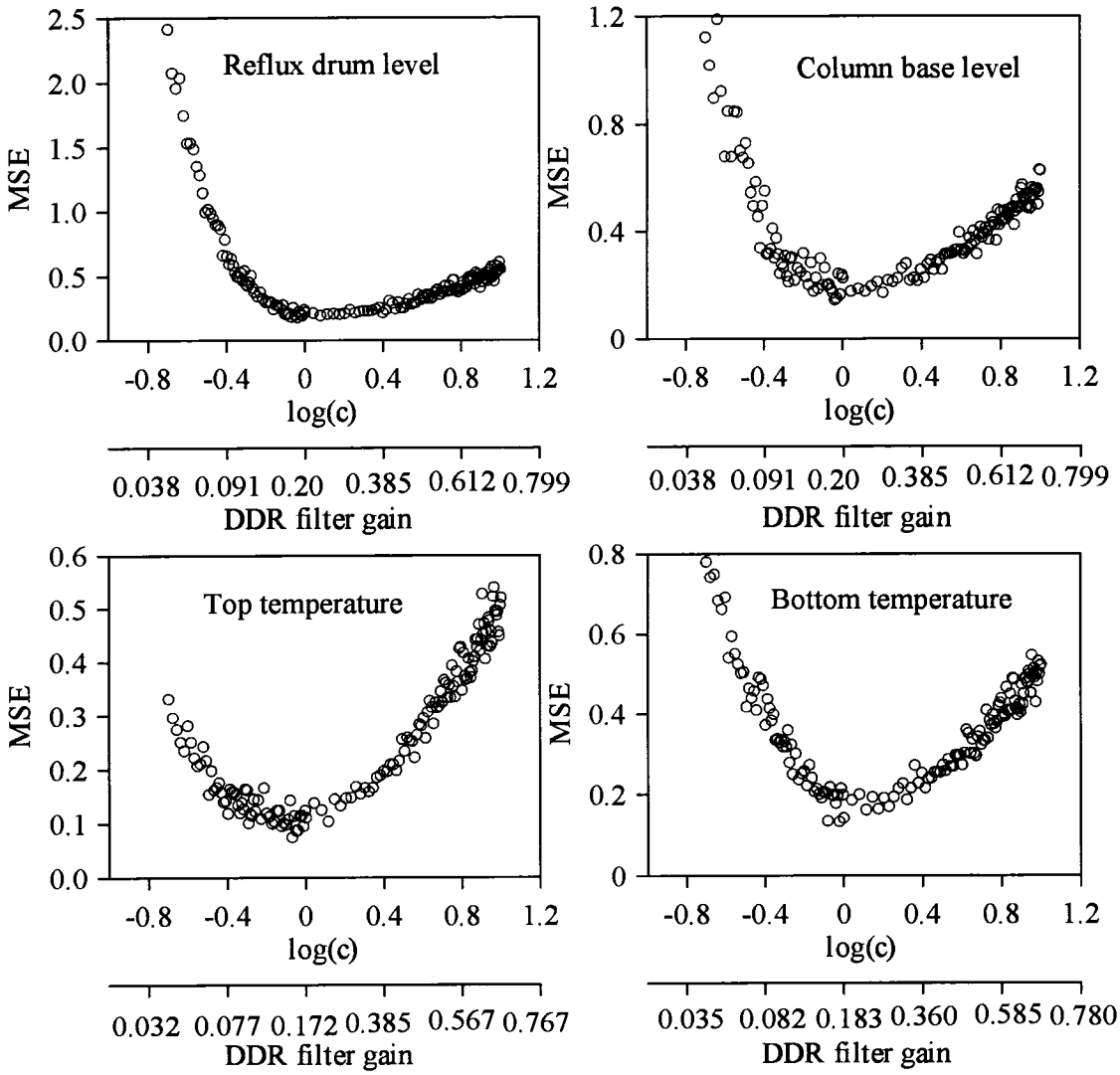


Figure 6 MSE values of filtered data using DDR filter with the variance of model predictions as tuning parameters for nominal white noise case. c is the multiplier of the matrix **R** in Equation (33).

4.4.1 DDR and augmented Kalman filters

Using $\lambda = 0.8$ as an illustrative example, the autocorrelation information given by Equation (39) was incorporated in the DDR filter by evaluating intermediate variables of noise terms as illustrated in Figure 1. For this case, Equation (35) is the filter gain matrix. The raw measurements, filtered values and true values of the four controlled variables are presented in Figure 7. It shows the controlled variables displayed larger variations around

their setpoints compared to the white noise case study, due to the propagation of the autocorrelated noise throughout the feedback loops. Next, to consider the autocorrelated measurement noise using the Kalman filter, the process state/measurement models for the column were augmented to incorporate the information of the autocorrelated measurements noise using Equation (39). The covariance matrix of process noise of Equation (36) was used to calculate the augmented Kalman filter gain. After 80 iterative calculations, the steady-state value for the augmented Kalman filter gain was

$$\mathbf{K}_{\text{Kalman}} = \left[\begin{array}{cccc} 0.154 & 0 & 0 & 0 \\ 0 & 0.082 & 0 & 0 \\ 0 & 0 & 0.154 & 0 \\ 0 & 0 & 0 & 0.084 \\ \hline 0.464 & 0 & 0 & 0 \\ 0 & 0.538 & 0 & 0 \\ 0 & 0 & 0.464 & 0 \\ 0 & 0 & 0 & 0.539 \end{array}\right] \tag{40}$$

Compared to the DDR filter, the dimension of the Kalman filter gain was doubled. For the same feed flow disturbance, the closed-loop performance of the augmented Kalman filter was tested and the simulation results for the filtered data were also plotted in Figure 7. These results show that the augmented Kalman filter performed well and the two filters appeared to display no practically significant difference in their performance.

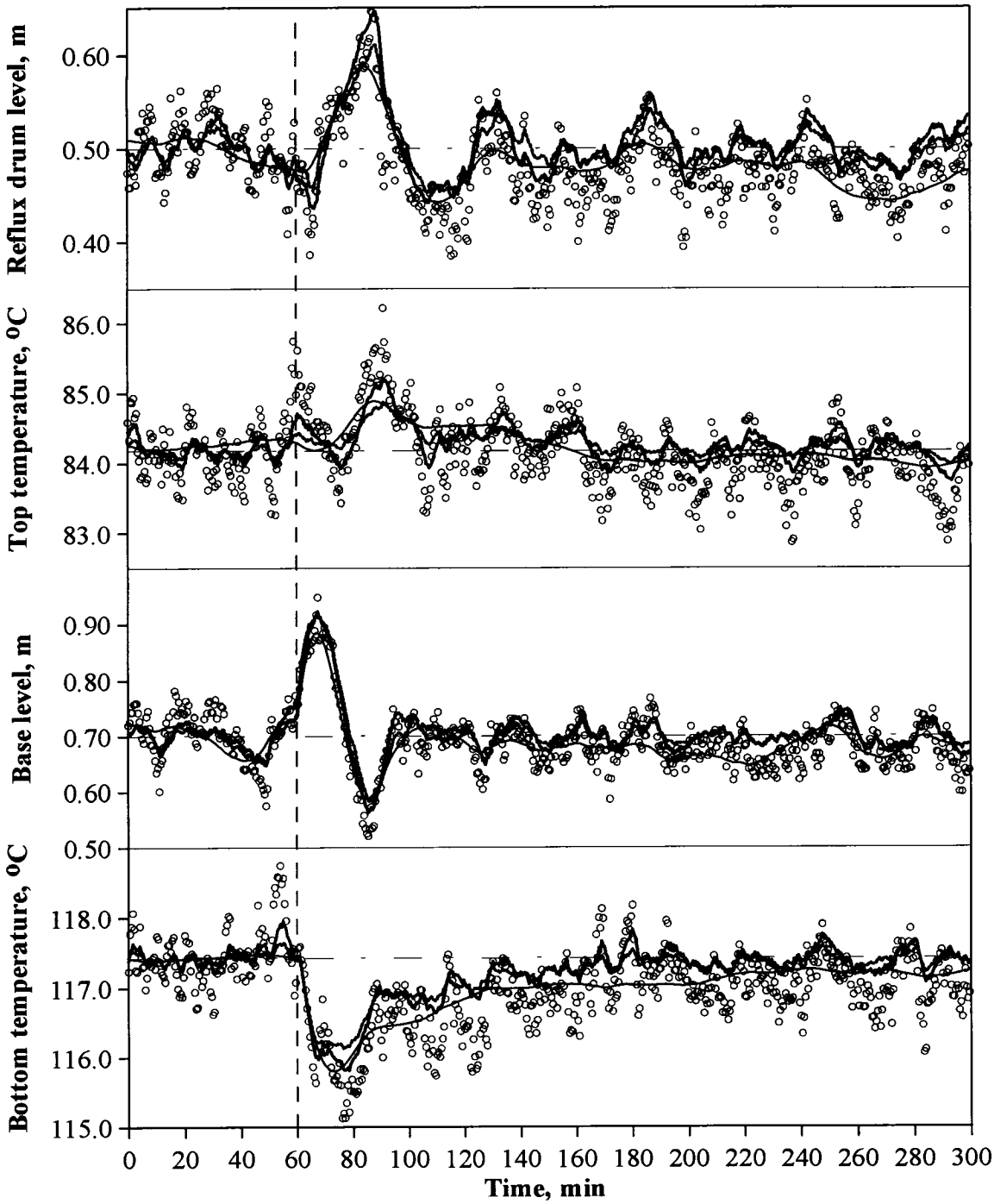


Figure 7 Performance of the DDR and augmented Kalman filters for autocorrelated noise (λ=0.8) for a 20% step change in feed flow rate at t = 60 min. o Raw; ------ Kalman filter; —— DDR filter; — True; — - — Setpoint.

To more rigorously evaluate the performance of the two filters for the autocorrelated noise cases, the MSE values of filtered data for each controlled variable obtained by the DDR filter and the augmented Kalman filter were calculated. One hundred and ten simulation runs were performed each with a different autocorrelated noise sequence. The histograms of the differences of the MSE values are presented in the first row of Figure 8. It shows the DDR filter performed better for the reflux drum level, column base level and bottom temperature, while the Kalman filter performed better for the top temperature.

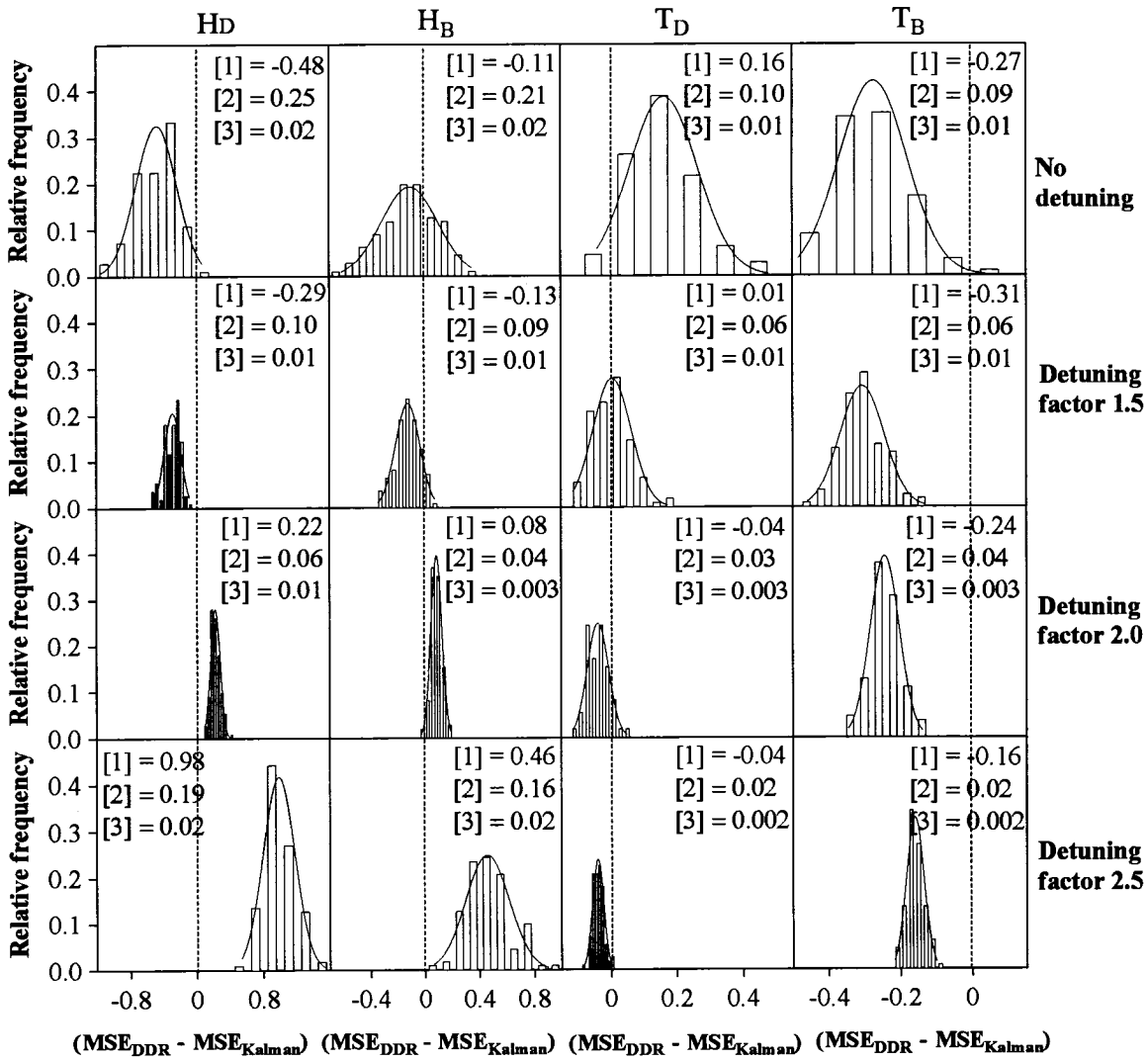


Figure 8 Histograms of differences of MSE values obtained by DDR and augmented Kalman filters for autocorrelated noise cases ($\lambda = 0.8$). [1] = Mean of the differences. [2] = Standard deviations of the differences. [3] = Standard deviation of the mean of the differences.

As described in Section 2, the gain of the DDR filter needs to be detuned in dealing with autocorrelated measurement noise as a result of process model mismatch in predicting values of the measurement errors. The impact of detuning on the performance of DDR filter was then examined. The DDR filter gain was detuned by factors of 1.5, 2.0 and 2.5, respectively. Then, the performance of the detuned DDR filters was compared to that of the augmented Kalman filter. The results are also presented in the Figure 8. It shows that, when the detuning factor was set at 1.5, the DDR performed better than the Kalman for all controlled variables. However, when the detuning factor was increased to 2.0 and 2.5, the augmented Kalman filter performed better for the two liquid levels, while the DDR filter performed better for both the top and bottom temperatures. These results indicate the need for proper tuning of the DDR filter to achieve its best performance in dealing with auto correlated measurements. We expect that tuning of the Kalman filter will also be beneficial in these cases.

4.4.2 Shortcut DDR and Kalman filters

It is important to recall that, in the implementations of the DDR and augmented Kalman filters for autocorrelated measurements, the noise terms had to be evaluated as intermediate variables. As a result, the dimensionality of the filters was increased. In addition, the DDR filter has to be detuned after its initial design and no general rule for detuning exists. Therefore, a more direct approach to implement the DDR and Kalman filters for the autocorrelated noise case is now examined.

From Equation (39), the variance of the autocorrelated measurements can be easily evaluated [9] as

$$\mathrm{Var}[\varepsilon_t] = \frac{\mathrm{Var}[\alpha_t]}{1-\lambda^2} \tag{41}$$

Thus, for $\lambda = 0.8$ as the illustrative example, the covariance matrix of the measurements was

$$\mathrm{Cov}[\varepsilon_t] = \mathbf{V} = \begin{bmatrix} 11.11 & 0 & 0 & 0 \\ 0 & 0.174 & 0 & 0 \\ 0 & 0 & 11.11 & 0 \\ 0 & 0 & 0 & 0.174 \end{bmatrix} \tag{42}$$

Using Equations (33) and (42) results in a shortcut DDR filter with gain matrix

$$\mathbf{K}_{DDR} = (\mathbf{I} + \mathbf{VR}^{-1})^{-1} = \begin{bmatrix} 0.083 & 0 & 0 & 0 \\ 0 & 0.070 & 0 & 0 \\ 0 & 0 & 0.083 & 0 \\ 0 & 0 & 0 & 0.075 \end{bmatrix} \tag{43}$$

Compared to the filter derived for the white noise case study (Equation (35)), the shortcut filter gain for the autocorrelated noise is much smaller, indicating that a greater confidence is placed on model-predicted values than on measured values when calculating the filtered data. The closed-loop performance of the shortcut DDR filter was evaluated for the identical one hundred and ten autocorrelated noise sequences. The differences of the MSE values for each variable obtained by the shortcut and the complete DDR filters (i.e., $\Delta\mathrm{MSE} = (\mathrm{MSE}_{\mathrm{shortcut}} - \mathrm{MSE}_{\mathrm{complete}})$) were calculated and their histograms are presented in the top row of Figure 9, along with pertinent statistics. The shortcut DDR filter performed better for the top temperature, while the complete DDR filter performed better for the reflux drum level. The performance of the shortcut and the complete DDR filters showed no significant differences for the column base level and bottom temperature. This example indicates that the shortcut DDR filter provides a much easier approach to deal with autocorrelated measurement noise and it has comparable performance relative to the complete DDR filter.

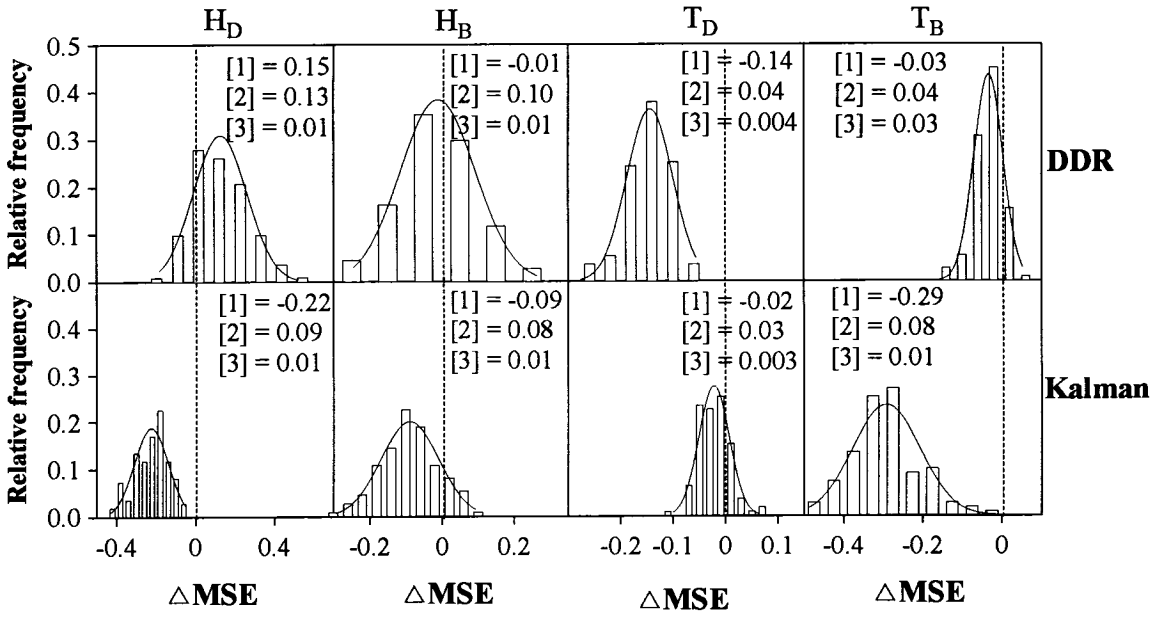


Figure 9 Histograms of the differences of MSE values obtained by shortcut DDR and complete DDR filters (the top row), $(\mathrm{MSE}_{\mathrm{shortcut}} - \mathrm{MSE}_{\mathrm{complete}})$, and shortcut Kalman and augmented Kalman filters (the bottom row), $(\mathrm{MSE}_{\mathrm{shortcut}} - \mathrm{MSE}_{\mathrm{augmented}})$, for autocorrelated noise ($\lambda = 0.8$). [1] = Mean of the differences. [2] = Standard deviations of the differences. [3] = Standard deviation of the mean of the differences.

Akin to the DDR filter, a shortcut Kalman filter was considered. Using Equation (42) as the covariance matrix of measurements in the Kalman filter, after 44 iterative calculations, the steady-state shortcut Kalman filter gain was

$$\mathbf{K}_{\mathrm{Kalman}} = \begin{bmatrix} 0.077 & 0 & 0 & 0 \\ 0 & 0.045 & 0 & 0 \\ 0 & 0 & 0.077 & 0 \\ 0 & 0 & 0 & 0.097 \end{bmatrix} \tag{44}$$

The closed-loop performance of the shortcut Kalman filter was compared to that of the augmented Kalman filter for the identical one hundred and ten autocorrelated noise sequences. Differences of the MSE values, (i.e., $\Delta\mathrm{MSE} = (\mathrm{MSE}_{\mathrm{shortcut}} - \mathrm{MSE}_{\mathrm{augmented}})$) were calculated; their histograms along with pertinent statistics are presented in the

bottom row of Figure 9. It indicates that the concise shortcut Kalman filter resulted in better performance for all variables. This case study indicates that the shortcut DDR and Kalman filters have equivalent or better performance than their complete formats for nearly all controlled variables. The better performance results from the ability of the shortcut formats to overcome the process model prediction errors occurring in the evaluation of the intermediate variables in their complete formats. The two shortcut filters can be reliably and more easily used to deal with autocorrelated noise.

5 CONCLUSION

Using a dynamic data reconciliation filter is an effective approach to reduce measurement noise inside control loops. It uses information from both measured values and model predicted values, such that the filtered data better represent the current state of the process. In this article we have developed a convenient predictor-corrector form for the DDR filter, along with expressions for estimating the covariance of the filter predictions. Modifications of both filters have been introduced to allow application to situations involving autocorrelated measurement noise. In cases where the process can be adequately described by linear models with white Gaussian measurement and model noise, the DDR and Kalman filters are essentially equivalent, differing only in the information required and the manner of calculating the filter gain. When the process is nonlinear, DDR can be applied directly using the nonlinear models, albeit numerical solution of the model equations is required, while an extended Kalman filter is required. In cases involving autocorrelated measurements, the modified DDR filter has a potential computational advantage over the augmented Kalman filter. For cases involving autocorrelated measurement noise, an effective shortcut procedure was developed to simplify implementation. In all but one case, the shortcut method was shown to provide equivalent or better performance than use of the complete format in our simulations.

The effectiveness of both filters as part of a feedback control system for a binary distillation column has been demonstrated via simulation. Differences in performance were observed due to differences in the calculation of the filter gains. In practice the

covariance information required for setting up the filter may not be easy to obtain. Consequently, it is recommended to assume that the covariance matrices are diagonal and to use the diagonal elements as tuning parameters to achieve acceptable performance. Simulations showed that this procedure was effective.

NOMENCLATURE

$\mathbf{A}$: Matrix of coefficients in process state models

$\mathbf{B}$: Matrix of coefficients in process state models

$\mathbf{C}$: Matrix of coefficients in measurement models

F : Flow rate of column feed stream, (kmol/h)

H_B : Column base level, (m)

H_D : Reflux drum level, (m)

$\mathbf{K}_t$: Matrix of Kalman gain

$\mathbf{K}$: Matrix of DDR gain

MSE : Mean square errors

$\mathbf{P}_t^-$: Covariance matrix of prior estimation error

$\mathbf{P}_t$: Covariance matrix of posterior estimation error

q : Order of measurement noise model.

$\mathbf{R}$: Covariance matrix of model prediction error

$\mathbf{S}$: Covariance matrix of process noise

T_B : Column bottom temperature, (°C)

T_D : Column top temperature, (°C)

$\mathbf{u}_{t-1}$: Vector of manipulated and disturbance variables

$\mathbf{V}$: Covariance matrix of the measurement noise

$\mathbf{w}_t$: Vector of process noise defined in Kalman filter

$\mathbf{x}_t$: Vector of the true values of process variables at time t

$\hat{\mathbf{x}}_t$: Vector of reconciled values of the measured process variables

$\mathbf{y}_t$: Vector of measured values of process variables

$\hat{\mathbf{y}}_t$: Vector of model predicted values

Creek letters

ε_t: Vector of random variables representing measurement noise

$\hat{\varepsilon}_t$: One-step-ahead prediction for measurement error

δ_t: Vector of random variables representing model prediction errors

ξ_t: Estimation error by the DDR algorithm

φ: Functional form of process estimator

Φ_i: Matrix of coefficients in measurement noise model

α_t: Vector of white Gaussian noise

σ: Standard deviation of the raw measurements

λ: Autocorrelation coefficient

REFERENCES

[1] B. Roffel, P. Chin, *Computer Control in the Process Industries*, Lewis Publishers, Chelsea, Michigan, 1987.

[2] D.I. Wilson, M. Agarwal, D.W.T. Rippin, Experiences implementing the extended Kalman filter on an industrial batch reactor, *Computers Chem. Engng*, **22** (1998), 1653-1672.

[3] C. Brosilow, B. Joseph, *Techniques of Model-Based Control*, Prentice Hall, New Jersey, 2002.

[4] S. Bai, D.D. McLean, J. Thibault, Enhancing controller performance via dynamic data reconciliation, *Can. J. Chem. Eng.*, **83** (2005), 515-526.

[5] G.E.P. Box, G.C. Tiao, *Bayesian Inference in Statistical Analysis*, Wiley Classical Library Ed., Toronto, 1992.

[6] E.W. Kamen, J.K. Su, *Introduction to Optimal Estimation*, Springer, London, 1999.

[7] S. Bai, *Assessment of Controller Performance with Embedded Data Reconciliation*, M.A.Sc. thesis, University of Ottawa, Canada, 2003.

[8] T.E. Marlin, *Process Control, Design Processes and Control Systems for Dynamic Performance*, 2nd Ed., McGraw-Hill, Boston, 2000.

[9] G.E.P. Box, G.M. Jenkins, *Time Series Analysis, Forecasting and Control*, Revised Edition, Holden-Day, San Francisco, 1976.

APPENDIX A

Taking the derivative of the objective function given by Equation (12) with respect to $\mathbf{x}_t$, and setting it equal to zero yields

$$\frac{\partial J}{\partial \mathbf{x}_t} = \frac{1}{2}\left[\mathbf{V}^{-1}(\mathbf{y}_t - \hat{\mathbf{x}}_t) + \mathbf{R}^{-1}(\hat{\mathbf{y}}_t - \hat{\mathbf{x}}_t)\right] = \mathbf{0} \tag{A1}$$

Solving Equation (A1) yields

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}(\mathbf{V}^{-1}\mathbf{y}_t + \mathbf{R}^{-1}\hat{\mathbf{y}}_t) \tag{A2}$$

Expanding Equation (A2) gives

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\mathbf{y}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{R}^{-1}\hat{\mathbf{y}}_t \tag{A3}$$

Adding and subtracting $(\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\hat{\mathbf{y}}_t$ to the right-hand side of Equation (A3) gives

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\mathbf{y}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\left[(\mathbf{V}^{-1} + \mathbf{R}^{-1}) - \mathbf{V}^{-1}\right]\hat{\mathbf{y}}_t$$

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\mathbf{y}_t + \hat{\mathbf{y}}_t - (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\hat{\mathbf{y}}_t$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}(\mathbf{y}_t - \hat{\mathbf{y}}_t)$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}}_t + \mathbf{K}(\mathbf{y}_t - \hat{\mathbf{y}}_t) \tag{A4}$$

where,

$$\mathbf{K} = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1} = (\mathbf{I} + \mathbf{V}\mathbf{R}^{-1})^{-1} \tag{A5}$$

Equations (A4) and (A5) express the predictor-corrector form of the DDR algorithm.

APPENDIX B

Consider the noise term ε_M of the measurements y_M in Equation (24) to be autocorrelated, and that it can be described by a linear model driven by white Gaussian noise, α_t. The transfer function of ε_M in the z-domain can be written as

$$G(z) = \frac{\varepsilon_M(z)}{\alpha(z)} = D + \frac{\Phi(z)}{\Upsilon(z)} \tag{B1}$$

where D is a constant and $\Phi(z)$ and $\Upsilon(z)$ are two polynomials in z given by

$$\Phi(z) = \sum_{i=0}^{H} b_i z^i , \tag{B2}$$

$$\Upsilon(z) = z^L + \sum_{i=0}^{L-1} a_i z^i \tag{B3}$$

with $H < L$. The model of ε_M given by Equation (B1) can be transformed into a state-space representation as

$$\mathbf{x}_{\varepsilon,t} = \mathbf{A}_\varepsilon \mathbf{x}_{\varepsilon,t-1} + \boldsymbol{\Gamma}_\varepsilon \alpha_{t-1} \tag{B4}$$

$$\varepsilon_{M,t} = \mathbf{C}_\varepsilon \mathbf{x}_{\varepsilon,t} + D\alpha_t \tag{B5}$$

where $\mathbf{x}_{\varepsilon,t}$ represents state variables of the noise. Matrices $\mathbf{A}_\varepsilon$, $\boldsymbol{\Gamma}_\varepsilon$ and $\mathbf{C}_\varepsilon$ can be represented in the control canonical form

$$\mathbf{A}_\varepsilon = \begin{bmatrix} 0 & 1 & 0 & .. & 0 \\ 0 & 0 & 1 & .. & 0 \\ : & : & : & : & : \\ -a_0 & -a_1 & -a_2 & .. & -a_{L-1} \end{bmatrix}, \quad \boldsymbol{\Gamma}_\varepsilon = \begin{bmatrix} 0 \\ 0 \\ : \\ 1 \end{bmatrix}, \tag{B6}$$

$$\mathbf{C}_\varepsilon = \begin{bmatrix} b_0 & b_1 & \ldots & b_H & 0 & \ldots & 0 \end{bmatrix}. \tag{B7}$$

It is now possible to define a new state vector comprised of the model and the noise state variables

$$\mathbf{x}_t^* = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{\varepsilon,t} \end{bmatrix}$$

The combination of Equations (23), (24), (B4) and (B5) gives a new process state/measurement model that accounts for the autocorrelated nature of measurements

$$\mathbf{x}_t^* = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_\varepsilon \end{bmatrix} \mathbf{x}_{t-1}^* + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u}_{t-1} + \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_\varepsilon \end{bmatrix} \mathbf{w}_{t-1}^* \tag{B8}$$

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{C} & \mathbf{E} \end{bmatrix} \mathbf{x}_t^* + \varepsilon_t^* \tag{B9}$$

where $\mathbf{w}_{t-1}^* = \begin{bmatrix} \mathbf{w}_{t-1} \\ \alpha_{t-1} \end{bmatrix}$, $\mathbf{E} = \begin{bmatrix} \mathbf{0} \\ \mathbf{C}_\varepsilon \end{bmatrix}$, and $\varepsilon_t^* = \begin{bmatrix} \varepsilon_{1,t} & \varepsilon_{2,t} & \ldots & \varepsilon_{M-1,t} & D\alpha_t \end{bmatrix}^T$. Since both $\mathbf{w}_t^*$ and ε_t^* are white noise in the new process state/measurement models, the augmented Kalman filter can then be implemented by steps described by Equations (25-31).

For example, assume a state-space model for a two-output one-input system is given by

$$\begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} = \begin{bmatrix} 0.7 & 0 \\ 0 & 0.6 \end{bmatrix} \begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u_{t-1} + \begin{bmatrix} \delta_{1,t-1} \\ \delta_{2,t-1} \end{bmatrix} \tag{B10}$$

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{bmatrix} \tag{B11}$$

where $\delta_{1,t}$ and $\delta_{2,t}$ are white noise. The measurement noise affecting the two outputs is a white Gaussian noise $\varepsilon_{1,t}$ and an autocorrelated noise $\varepsilon_{2,t}$. The autocorrelated noise is given by the following model

$$\varepsilon_{2,t} = 0.8\varepsilon_{2,t-1} + 0.1\varepsilon_{2,t-2} + \alpha_t \tag{B12}$$

where α_t is white Gaussian. The transfer function of Equation (B12) can be written as

$$G(z) = \frac{\varepsilon_{2,t}}{\alpha_t} = \frac{1}{1 - 0.8z^{-1} - 0.1z^{-2}} = \frac{z^2}{z^2 - 0.8z - 0.1} \tag{B13}$$

The transfer function of (B13) can be reorganized as

$$G(z) = D + \frac{0.8z + 0.1}{z^2 - 0.8z - 0.1} \tag{B14}$$

where D=1. Following the procedures described by Equations (B1-B7), the coefficients are $b_0 = 0.1$, $b_1 = 0.8$; $a_0 = -0.1$, $a_1 = -0.8$. The matrices are consequently

$$\mathbf{A}_\varepsilon = \begin{bmatrix} 0 & 1 \\ 0.1 & 0.8 \end{bmatrix}; \quad \mathbf{\Gamma}_\varepsilon = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad \mathbf{C}_\varepsilon = \begin{bmatrix} 0.1 & 0.8 \end{bmatrix}.$$

Therefore, the state-space model for the autocorrelated measurement noise is given by

$$\begin{bmatrix} x_{\varepsilon 1,t} \\ x_{\varepsilon 2,t} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} x_{\varepsilon 1,t-1} \\ x_{\varepsilon 2,t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \alpha_{t-1} \tag{B15}$$

$$\varepsilon_{2,t} = \begin{bmatrix} 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} x_{\varepsilon 1,t} \\ x_{\varepsilon 2,t} \end{bmatrix} + \alpha_t \tag{B16}$$

Combining the two state-space models, (B10), (B11), (B15) and (B16), gives the new state-space model as

$$\begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{\varepsilon 1,t} \\ x_{\varepsilon 2,t} \end{bmatrix} = \begin{bmatrix} 0.7 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \\ x_{\varepsilon 1,t-1} \\ x_{\varepsilon 2,t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \\ 0 \\ 0 \end{bmatrix} u_{t-1} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_{1,t-1} \\ \delta_{2,t-1} \\ \alpha_{t-1} \end{bmatrix}$$

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{\varepsilon 1,t} \\ x_{\varepsilon 2,t} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \alpha_t \end{bmatrix}$$

where both $\varepsilon_{1,t}$ and α_t are white.

APPENDIX C

The linear empirical input-output models identified for the distillation column were:

$$H_{D,t} - H_{D,t-1} = -9.058\times10^{-4}D_{t-1} - 9.058\times10^{-4}R_{t-1} + 2.112\times10^{-5}Q_{t-1} + 1.06\times10^{-5}Q_{t-2} \quad \text{(C1)}$$

$$T_{D,t} - 0.9414T_{D,t-1} = -5.64\times10^{-3}R_{t-1} - 4.332\times10^{-3}R_{t-2} + 2.5\times10^{-4}Q_{t-5} - 1.184\times10^{-4}F_{t-1} - 2.368\times10^{-4}F_{t-2} \quad \text{(C2)}$$

$$H_{B,t} - H_{B,t-1} = 1.023\times10^{-3}R_{t-2} - 1.152\times10^{-3}B_{t-1} - 2.056\times10^{-5}Q_{t-1} - 1.028\times10^{-5}Q_{t-2} + 7.942\times10^{-4}F_{t-1} + 3.971\times10^{-4}F_{t-2} \quad \text{(C3)}$$

$$T_{B,t} - 0.9228T_{B,t-1} = -0.011R_{t-8} - 0.00385R_{t-9} + 4.867\times10^{-5}Q_{t-1} + 6.084\times10^{-4}Q_{t-2} - 7.583\times10^{-3}F_{t-3} \quad \text{(C4)}$$

where $H_{D,t}$ is the liquid level of reflux drum at sampling time t, m; $T_{D,t}$ the top temperature of the distillation column, °C; $H_{B,t}$ the liquid level of column base, m; $T_{B,t}$ the bottom temperature of the distillation column, °C; D_t the distillate flow rate, kmol/h; R_t the reflux flow rate, kmol/h; B_t the bottom product flow rate, kmol/h; Q_t the reboiler heat duty, MJ/h; and F_t the feed flow rate, kmol/h. All the variables are in their deviation forms.

CHAPTER V

Impact of Model Structure on the Performance of Dynamic Data Reconciliation

Shuanghua Bai, David D. McLean and Jules Thibault*

Department of Chemical Engineering

University of Ottawa
Ottawa, Ontario, Canada K1N 6N5

Submitted to **Computers & Chemical Engineering**, revised version after reviewers' comments.

* Corresponding author. Tel: 613-562-5800 ext. 6094
Email: thibault@genie.uottawa.ca

Preface

Process dynamic models, as an integral part of the DDR algorithm, provide redundant information to improve the estimates of the state variables obtained from measurements alone. The degree of improvement will depend upon the accuracy and precision of the model predictions. Phenomenological models are often unavailable or too complex for use in DDR and a variety of simpler black-box models could potentially be used. Chapter 5 investigates the impact of the structures of black-box models on the performance of DDR.

ABSTRACT

Dynamic Data Reconciliation (DDR) is a technique used to estimate the true values of process variables when plant measurements are corrupted by measurement noise. DDR integrates information from both measurements and process models such that the reconciled values become more reliable and better represent the current state of the process. Process models play a key role in the performance of DDR. Empirical or black-box models are identified and used in the DDR when phenomenological models are unavailable, impractical to obtain, or whose solutions require excessive computation time for real-time applications. Black-box models usually have higher degree of uncertainty and use a wide variety of structures. This article examines the impact of model structure on the performance of DDR, and more importantly, on the performance of controllers when the DDR is embedded inside feedback control loops. Simulation results of a binary distillation column demonstrated that the model structure can have a major impact on the performance of DDR. The DDR using simple linear models can successfully attenuate the noise propagation; however, further significant improvement of DDR performance can be achieved if more advanced models such as nonlinear models are used.

Keywords: Dynamic data reconciliation, Black-box models, Model structure, DDR performance, Controller performance

1 INTRODUCTION

Modern chemical plants employ a large quantity of sampled data for online process monitoring, control, optimization and management decision making. Unfortunately, the sampled measurements are usually corrupted by measurement noise. Information quality, abstracted from the noisy measurements, is reduced and, as a result, the performance of process control and optimization strategies, for example, is often affected. Data reconciliation is a technique used to reduce the impact of measurement noise by integrating information from both measurements and process models. In the past decade, the benefits of applying data reconciliation to steady-state processes have triggered some researchers to extend this technique to dynamic processes (e.g., Darouach and Zasadzinski, 1991; Liebman et al., 1992; Ramamurthi et al., 1993; Albuquerque and Biegler, 1996; Bagajewicz and Jiang, 1997; Binder et al., 2002). Mathematically, the estimates of measured and unmeasured process variables or model parameters by dynamic data reconciliation (DDR) are obtained by solving the constrained least-squares optimization problem,

$$\text{Minimize} \quad J(\hat{\mathbf{x}}_t, \hat{\mathbf{z}}_t) = \sum_{t=0}^{N} \left[\left(\mathbf{y}_t - \hat{\mathbf{x}}_t \right)^T \mathbf{V}^{-1} \left(\mathbf{y}_t - \hat{\mathbf{x}}_t \right) \right] \tag{1}$$

$$\text{subject to} \quad \mathbf{f}\left[\frac{d\hat{\mathbf{x}}}{dt}, \hat{\mathbf{x}}, \hat{\mathbf{z}} \right] = \mathbf{0}$$

$$\mathbf{g}\left(\hat{\mathbf{x}}, \hat{\mathbf{z}} \right) = \mathbf{0}$$

$$\mathbf{h}\left(\hat{\mathbf{x}}, \hat{\mathbf{z}} \right) \geq \mathbf{0}$$

where $\mathbf{y}_t$ is a vector of measured values of process variables at time t , $\hat{\mathbf{x}}_t$ is a vector of reconciled values of the measured process variables, $\hat{\mathbf{z}}_t$ is a vector of estimates for unmeasured process variables and/or model parameters. $\mathbf{V}$ is a covariance matrix of the measurement noise, $\mathbf{f}$ represents a functional vector of dynamic process models, $\mathbf{g}$ and $\mathbf{h}$ represent functional vectors of process equality and inequality constraints. N represents

the number of sampling periods representing the window over which the dynamic process is considered for each evaluation.

Although solutions to the optimization problem of Equation (1) for dynamic data reconciliation have been well established in the literature, the DDR problem formulated by Equation (1) suffers from several shortcomings. First, it assumes the process models represent the true dynamics of the process, and the reconciled data satisfy exactly the models despite model uncertainties. Second, first-principle models, such as reaction rate equations, must be available for the DDR. Unfortunately, such phenomenological models are rarely available for chemical plants. Third, online computation time is large in solving the dynamic optimization problem.

Accordingly, Hodouin and Makni (1996) used empirical dynamic models coupled with steady-state mass conservation equations for online reconciliation of mineral processing plant data. Bai et al. (2005) employed linear empirical (black-box) models for DDR when phenomenological models were too cumbersome to obtain for the data reconciliation and the performance of data reconciliation was compared to that of exponentially weighted moving average (EWMA) and moving average (MA) filters. Unlike phenomenological models, black-box models have a higher degree of uncertainty or model mismatch in approximating a real plant and, as a result, process data need to be reconciled by finding a judicious compromise between the measured and model predicted values. In addition, black-box models usually display a wide variety of structures which can be linear or nonlinear. This work examines the impact of model structures on the performance of DDR. In particular, the performance of DDR algorithms embedded inside feedback loops is studied. Section 2 presents the DDR algorithm based on statistical properties of both measurement and model errors. Section 3 describes different model structures that are tailored for the DDR algorithm. In Section 4, implementations of the DDR strategies are demonstrated through a simulation example consisting of a binary distillation column with four PI control loops. Section 5 summarizes this paper and presents some conclusions.

2 DYNAMIC DATA RECONCILIATION

For a plant with M measured variables, without systematic bias or serial correlation, at sampling time, t, it is assumed that the M measured values are given in the additive noise form

$$\mathbf{y}_t = \mathbf{x}_t + \boldsymbol{\varepsilon}_t \tag{2}$$

where $\mathbf{y}_t$ is a M×1 vector of the measured values for the M variables, $\mathbf{x}_t$ is a M×1 vector of the true values for the M variables, and $\boldsymbol{\varepsilon}_t$ is a M×1 vector of random variables representing measurement noise, assumed to be Gaussian white noise, i.e. $\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \mathbf{V})$. Based on Equation (2), $E(\mathbf{y}_t) = \mathbf{x}_t$, simply shows that the measurements are unbiased estimates of the true values. In order to obtain more precise knowledge about the current state of the process, prior information is required in addition to the raw measured values. This prior information can originate from previously identified dynamic process models; then $\mathbf{x}_t$ can be explicitly estimated from model predictions (e.g., one-step-ahead predictions), $\hat{\mathbf{y}}_t$. The information from the model predictions can be combined with the information from the measurements to give better estimates of $\mathbf{x}_t$.

Because no mathematical models are perfect, model predictions contain some degree of model mismatch. It is assumed that the model predictions, $\hat{\mathbf{y}}_t$, can be written in the additive noise form

$$\hat{\mathbf{y}}_t = \mathbf{x}_t + \boldsymbol{\delta}_t \tag{3}$$

where $\boldsymbol{\delta}_t$ is a M×1 vector of random variables representing model prediction errors, which are complex functions of factors such as inappropriate model forms, uncertainties in model parameters and inaccurate measured input variables. For simplicity, $\boldsymbol{\delta}_t$ is

assumed to be Gaussian white noise, i.e. $\delta_t \sim N(\mathbf{0}, \mathbf{R})$. At sampling time t, $\mathbf{y}_t$ and $\hat{\mathbf{y}}_t$ are known. The most likely values of $\mathbf{x}_t$, in other words, the "best" estimates of $\mathbf{x}_t$, can be obtained by simultaneously minimizing the weighted sum of squared measurement and model errors such that

$$\text{Minimize } J(\hat{\mathbf{x}}_t) = \frac{1}{2}\left[(\mathbf{y}_t - \hat{\mathbf{x}}_t)^T \mathbf{V}^{-1}(\mathbf{y}_t - \hat{\mathbf{x}}_t) + (\hat{\mathbf{y}}_t - \hat{\mathbf{x}}_t)^T \mathbf{R}^{-1}(\hat{\mathbf{y}}_t - \hat{\mathbf{x}}_t)\right] \tag{4}$$

Solving the optimization problem and rearranging the result give (see Appendix A)

$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}}_t + \mathbf{K}(\mathbf{y}_t - \hat{\mathbf{y}}_t) \tag{5}$$

where $\mathbf{K} = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1} = (\mathbf{I} + \mathbf{V}\mathbf{R}^{-1})^{-1}$, and $\mathbf{I}$ is the identity matrix.

The estimation error by the DDR algorithm is given by

$$\xi_t = \hat{\mathbf{x}}_t - \mathbf{x}_t \tag{6}$$

Putting Equation (5) into (6) and using Equations (2) and (3) give

$$\xi_t = \delta_t + \mathbf{K}(\varepsilon_t - \delta_t) \tag{7}$$

Taking expectations of both sides of Equation (7) yields

$$E[\xi_t] = \mathbf{K}E[\varepsilon_t] + (\mathbf{I} - \mathbf{K})E[\delta_t] \tag{8}$$

Since $E[\varepsilon_t] = \mathbf{0}$ and $E[\delta_t] = \mathbf{0}$, thus $E[\xi_t] = \mathbf{0}$. Therefore the DDR algorithm is an unbiased estimator. From Equation (7), the covariance matrix of the estimation error can be given by

$$V(\xi_t) = \mathbf{K}^T V(\varepsilon_t)\mathbf{K} + (\mathbf{I}-\mathbf{K})^T V(\delta_t)(\mathbf{I}-\mathbf{K}) \tag{9}$$

Putting $V(\varepsilon_t) = \mathbf{V}$, $V(\delta_t) = \mathbf{R}$ and $\mathbf{K} = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}$ into Equation (9) and rearranging yield

$$V(\xi_t) = \left(\mathbf{V}^{-1} + \mathbf{R}^{-1}\right)^{-1} \tag{10}$$

Because both **V** and **R** are symmetrical and positive definite matrices, the elements of $V(\xi_t)$ are smaller than those of **V** and **R**, meaning that the combination of the measured and model predicted values results in more precise estimates than using only one of them. For a single variable, the variance of the reconciled values as a function of the variance of its measurements and the variance of model predictions are plotted in Figure 1. It shows that the variance of reconciled values increases with the increase of both variances of measurements and model predictions. However, it is always less than both variances individually.

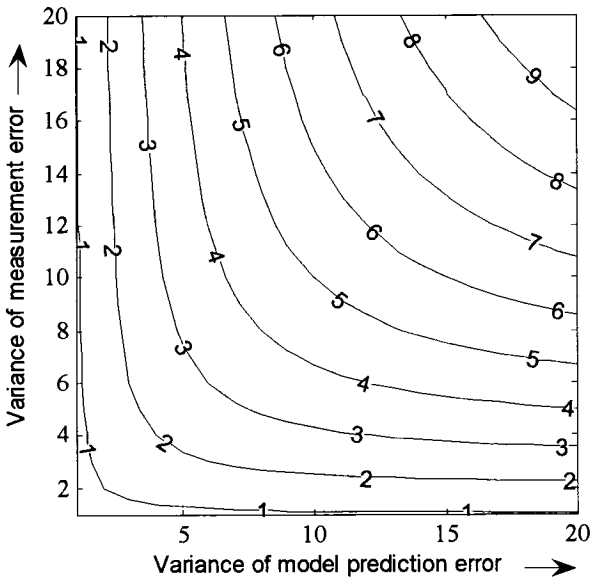


Figure 1 Contour plot of the variance of reconciled data as a function of variances of raw measurements and model predictions.

It should be mentioned that the model prediction errors are assumed to be Gaussian white noise in the derivations of the DDR algorithm. Nevertheless, this is rarely the case and statistical properties of the model errors, such as its covariance matrix **R**, are difficult to evaluate in practice. Consequently, elements in **R** are usually treated as tuning parameters. In other words, the gain matrix of the DDR can be treated as tuning parameters. Moreover, the DDR algorithm assumes both model predictions and measurements are within their variable lower and upper bounds. It cannot handle the situation where the model predictions and measurements are outside of their constraints.

Equation (5) indicates that the estimates (or reconciled values) of the current state of the process variables are comprised of two terms that are respectively the model predictions, $\hat{\mathbf{y}}_t$, and the measurement corrections, $\mathbf{K}(\mathbf{y}_t - \hat{\mathbf{y}}_t)$. The predictor-corrector form of the DDR has the same form as the Kalman filter (KF). Therefore, it is interesting to compare the two algorithms. For cases where a process can be described by a linear state-space model driven by white Gaussian noise and all state variables of interest are measured, it can be shown that the KF gain matrix reduces to the DDR gain matrix and the covariance matrix of filtered values obtained by the KF leads to Equation (10). Because the DDR algorithm can use a wide variety of process models that can be phenomenological or empirical, dynamic or static, continuous or discrete, linear or nonlinear, whereas the use of the KF is limited to process state-space models, the KF can be seen as a subset of the DDR algorithm. The DDR can therefore be viewed as a more generalized model-based filter.

3 STRUCTURES OF BLACK-BOX MODELS

The DDR algorithm requires process models to provide redundant information on the measured values. Reliable models undoubtedly play a predominant role in the estimation of the true values of process variables using DDR algorithms. Black-box models are certainly the prime choice to encapsulate the dynamic behavior of a process when phenomenological models are unavailable or impractical to use or derive.

3.1 Linear models

Linear models, such as first-order-plus-dead-time (FOPDT) or pure-integrator-plus-dead-time (PIPDT) models, are the most commonly used models in chemical engineering as they are usually able to adequately capture the underlying dynamic behavior of a process in the vicinity of the operating point with a relatively small number of parameters. The FOPDT models in both the Laplace and z domains for a single-input single-output (SISO) process are given by

$$\hat{y}(s) = \frac{K_P\, e^{-\theta s}}{\tau s + 1} u(s) \quad ; \quad \hat{y}_t = \frac{(b_0 - b_1 z^{-1}) z^{-p-1}}{1 - a_1 z^{-1}} u_t \tag{11}$$

where u is the input variable, K_P the gain of the process, θ the dead time and τ the time constant. In the discretized form, the model parameter $p = \mathrm{INT}(\theta / \Delta t)$ with Δt the sampling time interval. Other parameters are related to the Laplace domain equation as follows: $a_1 = e^{-\Delta t/\tau}$, $b_0 = K_P(1 - e^{-(1-\lambda)\Delta t/\tau})$ and $b_1 = K_P(e^{-\Delta t/\tau} - e^{-(1-\lambda)\Delta t/\tau})$ where λ is the remainder of the division, $\lambda = \mathrm{REM}(\theta / \Delta t)$. Similarly, the PIPDT models are given by

$$\hat{y}(s) = \frac{K_P\, e^{-\theta s}}{s} u(s) \quad ; \quad \hat{y}_t = \frac{(b_0 - b_1 z^{-1}) z^{-p-1}}{1 - z^{-1}} u_t \tag{12}$$

where $b_0 = K_P(1 - \lambda)\Delta t$ and $b_1 = -K_P \lambda \Delta t$. The FOPDT and PIPDT models are usually identified using a process reaction curve method (Marlin, 2000). For multiple-input multiple-output (MIMO) processes, the linear superposition property of the step responses is often assumed for each output variable.

In addition to these two simple models, more complex linear models, such as high-order auto-regressive with exogenous inputs (ARX), auto-regressive moving average with exogenous inputs (ARMAX), or Box-Jenkins models, can also be considered. Detailed descriptions and techniques for identification of these linear models can be found in Ljung (1999).

3.2 Nonlinear models

A linear model often provides a good representation of the plant in the close vicinity of its nominal operation point. However, for highly nonlinear processes, linear models are unable to provide accurate predictions over a wide range of the operating region. Therefore, nonlinear models are required. Although several structures of nonlinear models have been proposed in the literature (e.g., Henson and Seborg, 1997), feedforward neural networks have the advantage of providing more representative nonlinear models that can more accurately capture the underlying dynamics of the process (Zhu, 2001).

The procedure to identify a neural network model is identical to that of the identification of linear models. A neural network is a black-box model that can be used to map past inputs and outputs of the process into the current output. A schematic representation of a neural network used as a one-step-ahead predictor is shown in Figure 2. It consists of an input layer, an output layer and one (or more) hidden layers. The neurons in the input layer simply store and redistribute the past inputs and outputs of the process to the first hidden layer. The neurons, except the bias, in the hidden and output layers perform a nonlinear transformation of weighted sum of neuron outputs in the previous layer. Mathematically, the output of a neuron *i* in the hidden layer is given by

$$v_i = f\left(c_i + \sum_{j=1}^{r} W_{i,j} y_{t-j} + \sum_{k=0}^{s} w_{i,k} u_{t-d-k} \right) \tag{13}$$

where f represents a differentiable nonlinear function (e.g., sigmoid), $W_{i,j}$, $w_{i,k}$ and c_i are the weights associated with the connections for the past outputs, the past inputs and the bias, respectively, r is the order of the autoregressive term, s is the order of the moving average term and d is the time delay for the process input variable. The values of r, s and d are problem dependent and determined experimentally. The output of the neural network, as a one-step-ahead predictor, is given by

$$\hat{y}_t = g\left(\varphi + \sum_{i=1}^{q} \Theta_i \nu_i\right) \tag{14}$$

where Θ_i and φ are the associated weights to each of the neuron connections and the bias respectively. q is the number of neurons in the hidden layer, which must be determined by trial and error. The function, g, can be either a linear or nonlinear function.

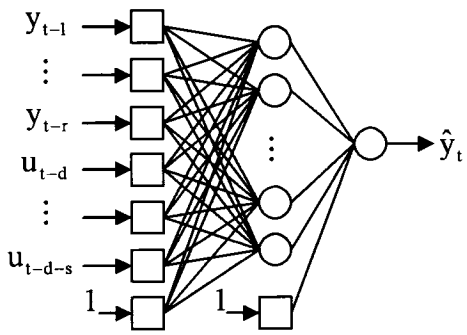


Figure 2 Architecture of feedforward neural networks for model identification.

A neural network has to be trained to produce the desired outputs corresponding to associated inputs. In training a neural network, the mean squared prediction error

$$J(\mathbf{W},\mathbf{w},\mathbf{c},\Theta,\varphi) = \frac{1}{L}\sum_{t=0}^{L}(y_t - \hat{y}_t)^2 \tag{15}$$

is minimized using alternative nonlinear optimization algorithm where L is the total number of data points used for training. Among training algorithms, backpropagation is

the simplest method. However, more advanced training algorithms such as quasi-Newton or Levenberg-Marquardt methods are most often used for faster convergence. The extension of neural networks for MIMO process model identification is straightforward.

The training of the neural network can be performed in two manners: (i) using only measured input and output variables as inputs to the neural network as presented in Figure 2 and, (ii) using the measured input variables and time-delayed predicted outputs as inputs to the neural network. The first approach rarely leads to a robust predictor, especially for the case where the sampling period is short, because the predicted output is always very close to the most recent past measurement. The second approach provides a accurate predictor because the network is able to perform a multi-step ahead prediction given future inputs (Thibault, 1991).

4 SIMULATION EXAMPLES

The DDR strategies were implemented in closed-loop control of a binary (benzene/toluene) distillation column, shown in Figure 3, via simulation. The distillation column has four PI control loops. Controllers TIC-D and TIC-B are used to control column top and bottom temperatures by manipulating reflux flow rate and flow of steam to the reboiler, respectively. Controllers LIC-D and LIC-B are used to control reflux drum and column base liquid levels by manipulating distillate flow rate and bottom product flow rate, respectively. A sampling period of 30 seconds is used. The dynamic distillation simulator was based on rigorous distillation models, (i.e., mass and heat balances, vapor-liquid equilibrium, and tray hydraulics). The nominal steady-state values for all measurements and their noise levels are listed in Table 1.

Table 1 Nominal steady-state values and noise levels of measured variables

Measured variable	Units	Steady-state values	Standard deviation
Reflux drum level, H_D	cm	50.0	2.0
Column top temperature, T_D	°C	84.2	0.25
Column base level, H_B	cm	70.0	2.0
Column bottom temperature, T_B	°C	117.4	0.25

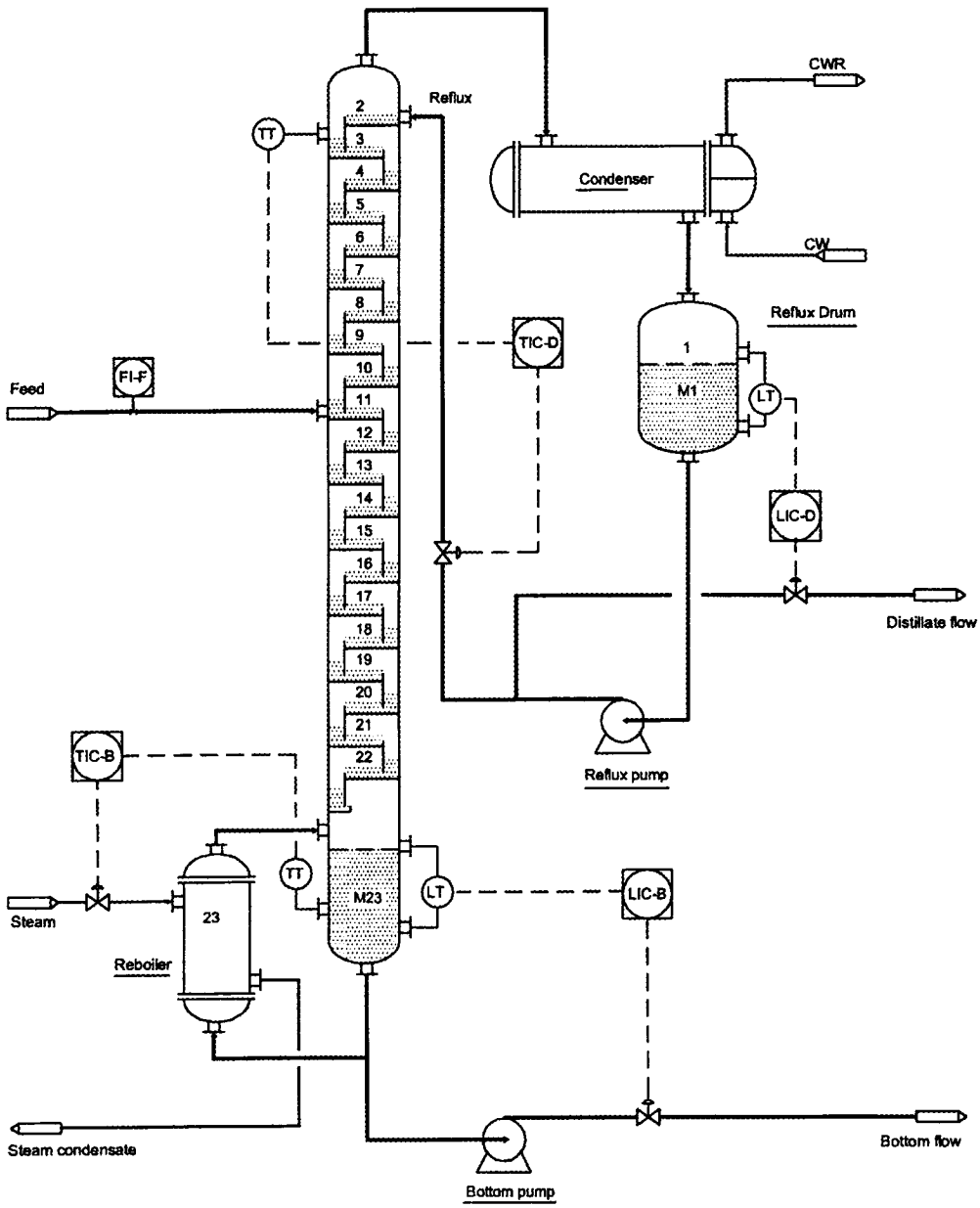


Figure 3 Schematic diagram of the distillation column.

The tuning parameters of the four PI controllers were obtained by minimizing the overall cost function of the control system

$$\Phi(\mathbf{K}_C, \boldsymbol{\tau}_I) = \sum_{i=1}^{4}\left[\alpha_i \sum_{t=0}^{N}(x_{i,t}^{*} - x_{i,t})^2 \Delta t \; + \; \beta_i \sum_{t=0}^{N}(u_{i,t} - u_{i,t-1})^2 \Delta t\right] \qquad (16)$$

when the distillation column was disturbed by a series of controller setpoint changes as listed in Table 2. In Equation (16), $x_{i,t}$ represents the true value of controlled variable i, $x^*_{i,t}$ the setpoint for controlled variable i, and $u_{i,t}$ the manipulated variable for control loop i. The weighting factors, α_i, for all the ISE terms were set to 0.15 while the weighting factors, β_i, for all the ISDU terms were set to 0.1. The selection of the weighting factors for ISE and ISDU is a trade-off between the dynamic response of the controlled and manipulated variables. The optimization was carried out using a quasi-Newton method with simple lower and upper bounds. The optimal controller parameters, obtained for ten different Gaussian noise sequences, are presented as Case I of Table 3. The integral times for the controllers reached their lower bounds which were set to 15 min for the LIC-D loop and 5 min for the other three loops. The resulting values of ISE and ISDU for each controller, as well as the overall cost function of the control system are presented as Case I of Table 4.

Table 2 Controller setpoint changes in the distillation column.

Time, min	Setpoint changes of controllers			
	LIC-D (cm)	TIC-D (°C)	LIC-B (cm)	TIC-B (°C)
0	50.0	84.2	70.0	117.4
90	70.0	84.2	70.0	115.0
180	70.0	88.0	70.0	115.0
270	70.0	88.0	50.0	118.0
360	40.0	88.0	50.0	115.0
450	70.0	88.0	50.0	117.4
540	70.0	84.2	70.0	117.4
630	70.0	88.0	70.0	117.4
720	50.0	88.0	40.0	117.4

Table 3 Optimal controller tuning parameters for the distillation column (Values listed for Cases I~IV are sample means plus and minus one standard error based on 10 random noise sequences)

Case	DDR	LIC-D		TIC-D		LIC-B		TIC-B	
		K_C $kmol.h^{-1}.m^{-1}$	τ_I min	K_C $kmol.h^{-1}.°C^{-1}$	τ_I min	K_C $kmol.h^{-1}.m^{-1}$	τ_I min	K_C $MJ.h^{-1}.°C^{-1}$	τ_I min
I	No	-41.05±0.14	15.0	-2.01±0.01	5.0	-51.02±0.48	5.0	46.40±0.31	5.0
II	Linear	-65.97±0.19	15.0	-2.65±0.02	5.0	-92.90±0.42	5.0	65.78±0.51	5.0
III	Adaptive Linear	-65.52±0.10	15.0	-2.79±0.01	5.0	-92.55±0.26	5.0	74.62±0.45	5.0
IV	Nonlinear	-66.34±0.14	15.0	-3.35±0.02	5.0	-94.00±0.32	5.0	90.99±0.32	5.0
V	"Perfect"	-66.61	15.0	-3.55	5.0	-95.50	5.0	95.99	5.0

Table 4 Comparison of controller performance without and with different DDR strategies in the distillation column (Values listed for Cases I~IV are sample means plus and minus one standard error based on 10 random noise sequences)

Case	DDR	**LIC-D** α=0.15, β=0.1		**TIC-D** α=0.15, β=0.1		**LIC-B** α=0.15, β=0.1		**TIC-B** α=0.15, β=0.1		Φ
		ISE	ISDU	ISE	ISDU	ISE	ISDU	ISE	ISDU	
I	No	12.67±0.06	10.15±0.08	9.53±0.08	6.07±0.06	5.52±0.09	4.49±0.06	11.00±0.08	5.59±0.06	8.44±0.03
II	Linear	8.11±0.07	5.67±0.04	8.11±0.06	5.08±0.05	3.12±0.02	2.43±0.02	9.07±0.06	4.78±0.02	6.06±0.03
III	Adaptive Linear	8.00±0.05	5.61±0.02	7.27±0.06	4.44±0.05	3.10±0.04	2.41±0.02	7.64±0.05	4.28±0.03	5.58±0.01
IV	Nonlinear	8.15±0.05	5.78±0.02	6.50±0.06	3.96±0.04	3.22±0.03	2.52±0.02	6.13±0.02	3.62±0.02	5.18±0.01
V	"Perfect"	7.69	5.30	5.87	3.87	2.79	2.27	5.64	3.57	4.80

Because the presence of measurement noise in the controlled variables inevitably deteriorates the controller performance, three different DDR algorithms were developed for the distillation column. The DDR algorithms, as filters, were embedded inside feedback control loops to reduce the noise propagations as depicted in Figure 4, where the raw measurements were first reconciled before they were used to calculate the control moves.

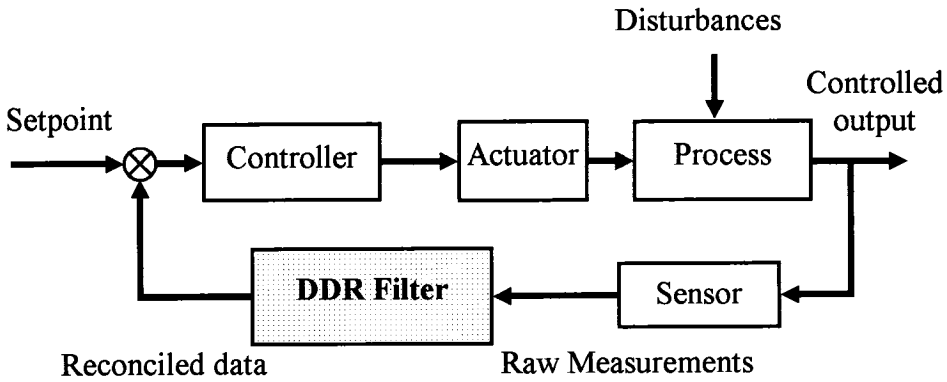


Figure 4 Scheme of DDR algorithm embedded inside feedback loops.

4.1 Linear DDR

To implement a linear DDR into the control system, FOPDT or PIPDT models (see Appendix B) around the nominal steady-state were first identified for each controlled variable. The DDR algorithm for the distillation column can be written in the compact form

$$\begin{bmatrix} H^r_{D,t} \\ T^r_{D,t} \\ H^r_{B,t} \\ T^r_{B,t} \end{bmatrix} = \begin{bmatrix} \hat{H}_{D,t} \\ \hat{T}_{D,t} \\ \hat{H}_{B,t} \\ \hat{T}_{B,t} \end{bmatrix} + \mathbf{K} \left\{ \begin{bmatrix} H^m_{D,t} \\ T^m_{D,t} \\ H^m_{B,t} \\ T^m_{B,t} \end{bmatrix} - \begin{bmatrix} \hat{H}_{D,t} \\ \hat{T}_{D,t} \\ \hat{H}_{B,t} \\ \hat{T}_{B,t} \end{bmatrix} \right\} \tag{17}$$

where H_D is the reflux drum liquid level, T_D the top temperature, H_B the column base liquid level and T_B the column base temperature. The superscript "r" represents the reconciled values, "m" the measured values, and "$\wedge$" the model predicted values.

K is the gain matrix of the DDR treated as tuning parameters and assumed diagonal in this case study. The diagonal elements in **K** were obtained by minimizing normalized mean squared errors between the reconciled and the true values when the column was submitted to the series of controller setpoint changes listed in Table 2, i.e.

$$\text{Minimize } \Psi(\mathbf{K}) = \sum_{i=1}^{4}\left[\frac{1}{N}\sum_{t=0}^{N}\left(\frac{\hat{x}_{i,t} - x_{i,t}}{\sigma_i}\right)^2\right] \tag{18}$$

where the weighting factor σ_i is the standard deviation of the raw measurements for variable i used to normalize the values of MSE. It is important to note that the true values of controlled variables are exactly known in the simulated case studies. With the true values, it is possible to compare without any prejudice the various algorithms and to assess their performance. However, for implementation of the DDR in a real plant, one cannot tune the DDR gain using the criterion of Equation (18). However, this does not impede the use of DDR algorithms because the calculations of reconciled values using the DDR do not require the true values. In practice, statistics of residuals between the reconciled and the measured values could be used in hypothesis tests to determine the DDR gain. Being white noise with mean zero and variance close to that of the raw measurements for the residuals would be a sign of good performance of the DDR algorithms.

The optimal linear DDR gain matrix was obtained for ten different Gaussian random noise sequences, and the average values plus and minus one standard errors of all diagonal elements of **K** were

$$\bar{\mathbf{K}} = \begin{array}{c} \begin{matrix} \quad H_D & \quad T_D & \quad H_B & \quad T_B \end{matrix} \\ \begin{bmatrix} 0.20 \pm 0.003 & 0 & 0 & 0 \\ 0 & 0.67 \pm 0.003 & 0 & 0 \\ 0 & 0 & 0.20 \pm 0.003 & 0 \\ 0 & 0 & 0 & 0.66 \pm 0.005 \end{bmatrix} \end{array} \tag{19}$$

The values of the DDR gain in Equation (19) indicate that higher confidence was put on the model predicted values in reconciling the variables of reflux drum level and base level, whereas, higher confidence was given to the raw measurements in reconciling the top and bottom temperatures. With the DDR in the control loops, the controller parameters were re-optimized by minimizing the objective function of Equation (16) with the identical series of controller setpoint changes. The optimal values of the controller parameters for the same ten Gaussian noise sequences are presented as Case II of Table 3. The controller gains for LIC-D, TIC-D, LIC-B and TIC-B were increased by factors of 1.6, 1.3, 1.8 and 1.4 respectively, while the integral times remained unchanged, meaning that the controllers became more aggressive with the embedded linear DDR algorithm. The corresponding values of ISE and ISDU for each controller, as well as the overall cost function of the control system, are presented as Case II of Table 4. Compared to Case I where no DDR was used, the values of ISE and ISDU for the two liquid level controllers were reduced by magnitudes of 36~46%, while the values of ISE and ISDU for the two temperature controllers were reduced by magnitudes of 15~18%, and the overall cost function of the control system was reduced by 28%

Table 4 shows that the controller performance was significantly enhanced by the DDR, because the DDR was able to attenuate the measurement noise, resulting in improved information quality for the controllers. The normalized mean squared errors of the reconciled data, i.e. the ratio of the variance of the reconciled data to that of measurements for each controlled variable, $(MSE)_i / \sigma_i^2$, are presented in Table 5. The DDR using the linear models yielded relatively larger $(MSE)_i / \sigma_i^2$ values for the two temperatures than for the two liquid levels. These values are consistent with the DDR gains in Equation (19). A larger DDR gain coincides with a larger value of $(MSE)_i / \sigma_i^2$.

Table 5 Values of $(MSE)_i / \sigma_i^2$ for each controlled variable (Values listed are sample means plus and minus one standard error based on 10 random noise sequences).

DDR	$(MSE)_i / \sigma_i^2$ ± standard error			
	H_D	T_D	H_B	T_B
Linear	0.13±0.005	0.50±0.013	0.13±0.004	0.48±0.011
Adaptive Linear	0.13±0.005	0.42±0.006	0.13±0.003	0.35±0.009
Nonlinear	0.12±0.004	0.16±0.003	0.13±0.03	0.12±0.003

4.2 Adaptive linear DDR

The above results showed that the linear DDR performed better for the two liquid levels than for the two temperatures. The performance of the DDR relies on the accuracy of the associated process models and it is believed that the dynamic responses of the two temperatures are too nonlinear to be adequately represented by simple linear models. To investigate the dynamic features of the two temperatures, a series of FOPDT models for the top and bottom temperatures relating to the reflux flow rate and the reboiler heat duty, respectively, were identified over a wide range of operating points. The static gains and time constants of the FOPDT models are presented in Figures 5. These figures clearly show that the model parameters change considerably with a change of operating point. The dynamic behavior of the two temperatures is nonlinear. The nonlinearity of the two temperatures lies in the fact that the top temperature could never decrease below the boiling point of pure benzene, and the bottom temperature could never exceed the boiling point of pure toluene under the given column pressure. Consequently, adaptive linear models were developed to improve the DDR performance for the two temperatures.

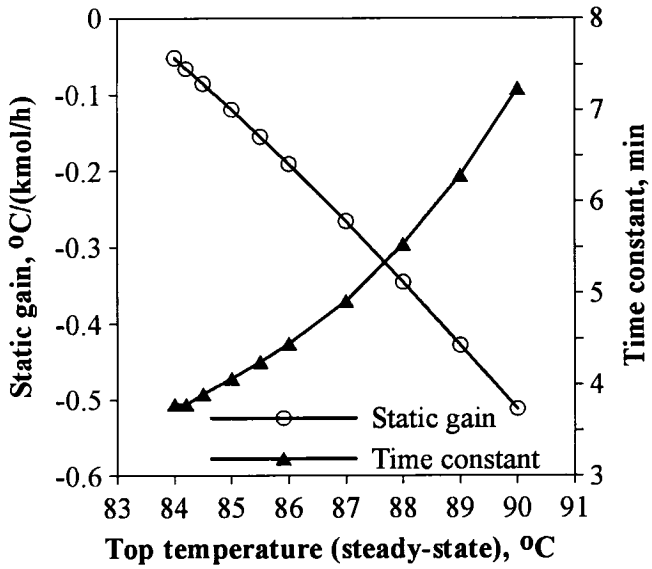


Figure 5(a) Static gain and time constant of FOPDT models for column top temperature over a range of operating points.

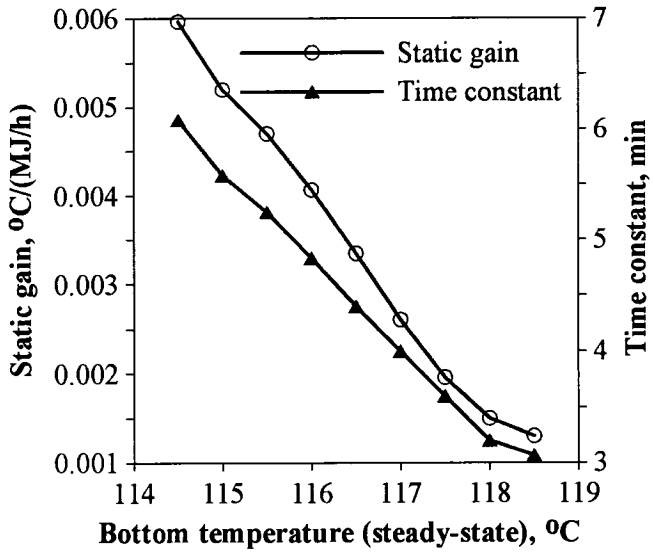


Figure 5(b) Static gain and time constant of FOPDT models for bottom temperature over a range of operating point.

At each operating point, the FOPDT models for the top and bottom temperatures presented in Figures 5(a) and 5(b) were discretized by z-transformation, resulting in a series of discrete models with different values of coefficients a_1, b_0 and b_1 (see Equation (11)). Next, fitted straight lines were used to approximate these coefficients as functions of the operating temperatures. For the top temperature, the adaptive

linear model representing the collections of these linear models relating to the reflux flow rate was

$$\hat{T}'_{D,t} = \frac{(-0.0048T_{D,t-1} + 0.395)z^{-1}}{1-(0.0098T_{D,t-1} + 0.0471)z^{-1}} R'_t \tag{20}$$

where the model predicted top temperature, $\hat{T}'_{D,t}$, and the reflux flow rate, R'_t, are in their deviation forms. For the bottom temperature, the adaptive linear model relating to the reboiler heat duty was

$$\hat{T}'_{B,t} = \frac{(-0.0838T_{B,t-1} + 10.112)\times 10^{-3} z^{-1}}{1-(-0.022T_{B,t-1} + 3.453)z^{-1}} Q'_t \tag{21}$$

where the model predicted bottom temperature, $\hat{T}'_{B,t}$, and the reboiler heat duty, Q'_t, are in their deviation forms.

Using the adaptive linear models, the gain matrix of the DDR was re-tuned by minimizing the objective function of Equation (18) under the same conditions of controller setpoint changes. The average values plus and minus one standard errors of the diagonal elements of the gain matrix for the same ten Gaussian random noise sequences were

$$\bar{\mathbf{K}} = \begin{bmatrix} 0.20\pm0.003 & 0 & 0 & 0 \\ 0 & 0.56\pm0.003 & 0 & 0 \\ 0 & 0 & 0.20\pm0.003 & 0 \\ 0 & 0 & 0 & 0.51\pm0.005 \end{bmatrix} \tag{22}$$

The gains of the adaptive linear DDR for the top and bottom temperatures were decreased, compared to the case of linear DDR, indicating more confidence was put on model predicted values in the adaptive linear DDR algorithm. The gains of the

DDR for the two liquid levels were unaffected since process models for the two liquid levels used in the DDR were unchanged. The $(MSE)_i / \sigma_i^2$ value for each variable is presented in the second row of Table 5. The $(MSE)_i / \sigma_i^2$ values for the two temperatures decreased, meaning that the adaptive linear DDR resulted in more accurate estimates than the linear DDR. With this new DDR algorithm embedded in feedback loops, the four controllers were re-tuned by minimizing the overall cost function under the same conditions. Results of the controller parameters for the ten Gaussian random noise sequences are presented in Case III of Table 3, and the corresponding ISE and ISDU values for each controller are presented in Case III of Table 4. The controllers for the two liquid levels were essentially unaffected, while the controllers for the two temperatures became more aggressive, compared to the case where linear DDR was used. The values of ISE and ISDU for the two temperatures were reduced by 10~16%, and the overall cost function of the control system was reduced by 8%.

4.3 Nonlinear DDR

Both the gain of the adaptive linear DDR and the $(MSE)_i / \sigma_i^2$ values in Table 5 indicated more comprehensive models are required to further improve the DDR performance for the two temperature control loops. Consequently, to more adequately capture the nonlinear dynamics of the two temperatures, the use of neural network models was explored. Under open loop conditions, the distillation column was excited by a series of random step changes in the input variables. Using this data set, a neural model having eight neurons in the input layer (including bias neuron), eight neurons in the hidden layer and one neuron in the output layer was developed for the top temperature. This neural network can be expressed as

$$\hat{T}_{D,t} = \mathbb{N}(\hat{T}_{D,t-1}, R_{t-1}, \ldots, R_{t-3}, Q_{t-1}, \ldots, Q_{t-3}) \tag{23}$$

where $\mathbb{N}$ represents the nonlinear transformations. Another neural model having six neurons (including the bias neuron) in the input layer, eight neurons in the hidden

layer and one neuron in the output layer was also developed for the bottom temperature. This neural network can be expressed as

$$\hat{T}_{B,t} = \mathbb{N}(\hat{T}_{B,t-1}, R_{t-6}, R_{t-7}, Q_{t-1}, Q_{t-2}) \tag{24}$$

Figure 6 presents the plots of the true and the model predicted values of the two temperatures. The first 750 data points were used for training whereas the remaining 300 points were used for validation. This figure shows the neural models performed well in capturing the underlying dynamics despite the high nonlinearity of the process.

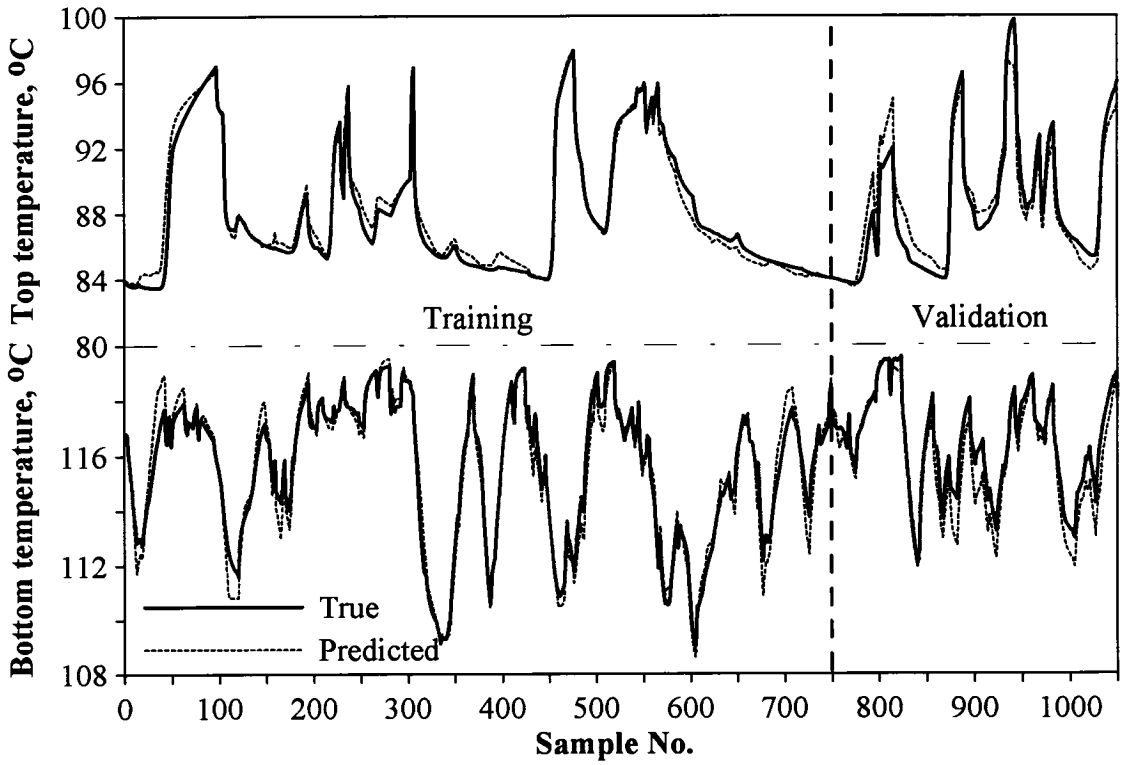


Figure 6 True values and predicted values for the top and bottom temperatures in network training and validation.

Using the neural models for the two temperatures, the DDR gain matrix was retuned under the same conditions. The gain matrix for the nonlinear DDR obtained for the same ten Gaussian noise sequences was

$$\bar{\mathbf{K}} = \begin{bmatrix} 0.20 \pm 0.003 & 0 & 0 & 0 \\ 0 & 0.24 \pm 0.003 & 0 & 0 \\ 0 & 0 & 0.20 \pm 0.003 & 0 \\ 0 & 0 & 0 & 0.20 \pm 0.003 \end{bmatrix} \quad (22)$$

Compared to the gain matrix of the adaptive linear DDR, the nonlinear DDR gain for the two temperatures was further reduced, meaning that more confidence was put on the model predictions in the nonlinear DDR. The corresponding $(MSE)_i / \sigma_i^2$ values of the reconciled data are presented in the third row of Table 5. The $(MSE)_i / \sigma_i^2$ values for the top and bottom temperatures decreased significantly, meaning that the nonlinear DDR resulted in better estimates of these two controlled variables than the adaptive linear DDR.

Embedded with the nonlinear DDR, the controller parameters were re-optimized again under the same process perturbations. Results of the controller parameters, the associated ISE and ISDU values, as well as the overall cost function of the control system for the same ten Gaussian noise sequences are presented in Case IV of Tables 3 and 4, respectively. Compared to Case III, the controllers of the top and bottom temperatures were changed considerably, while the controllers for the two liquid levels were affected only slightly. The gains of the controllers for the two temperatures increased, and the resulting ISE and ISDU values for the top and bottom temperatures decreased significantly. The overall cost function of the control system was reduced from $\bar{\Phi} = 5.58 \pm 0.01$ to $\bar{\Phi} = 5.18 \pm 0.01$, a 7% reduction with the nonlinear DDR compared to the adaptive linear DDR.

The true values, the reconciled values by the nonlinear DDR and the raw measurements for the four controlled variables, along with the control moves for the sequences of controller setpoint changes are presented in Figure 7. These figures show that the reconciled values were less noisy than the raw measurements and performed very well in tracking the true values of the controlled variables so that the differences between the reconciled and the true values are very hard to discern in the

figures. In addition, the control moves displayed relatively smaller variations with the embedded nonlinear DDR.

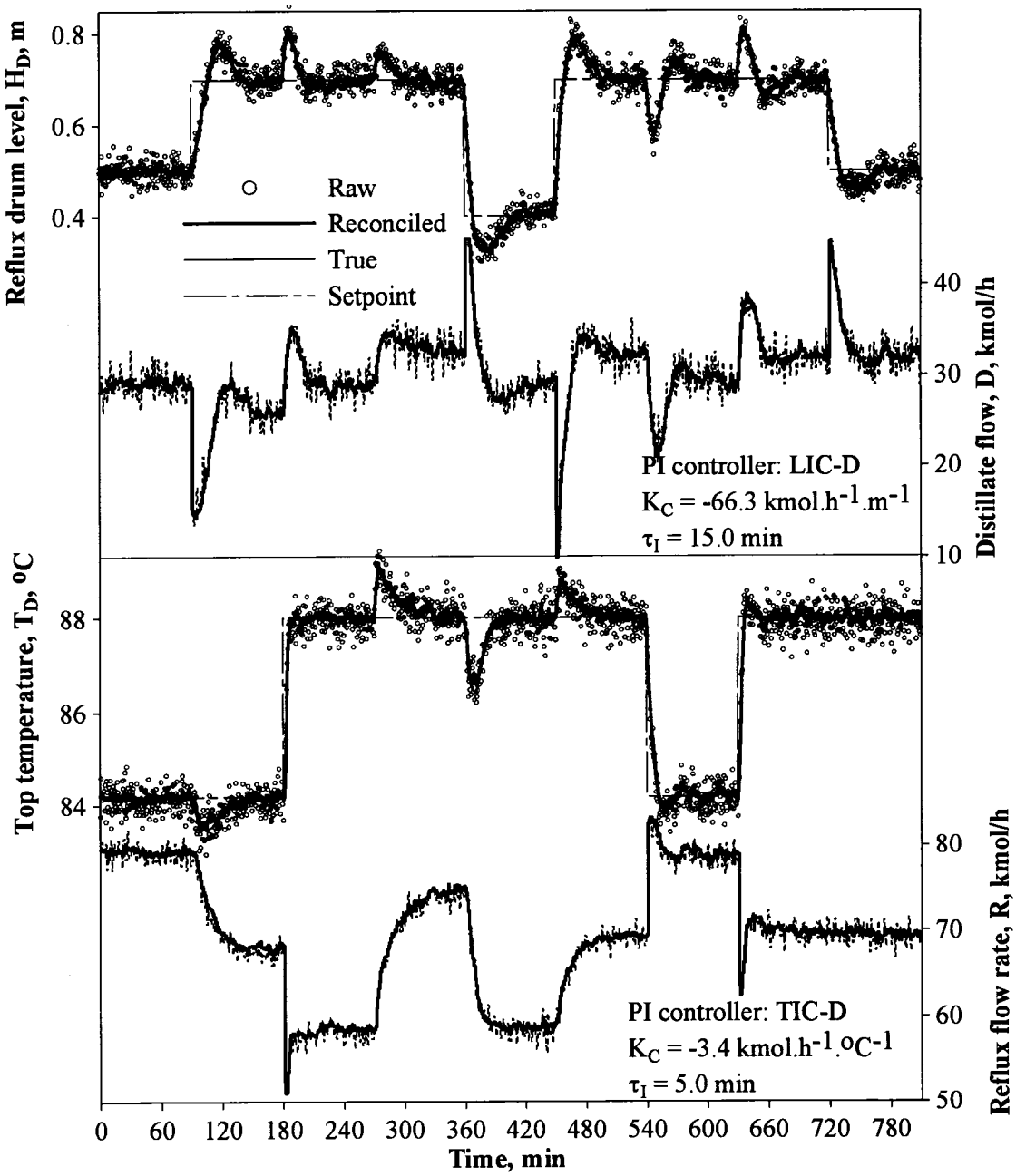


Figure 7(a) Closed-loop performance of nonlinear DDR and controllers LIC-D and TIC-D when the distillation column submitted to a series of controller setpoint changes. The dashed line for the manipulated variables represents the control moves without DDR.

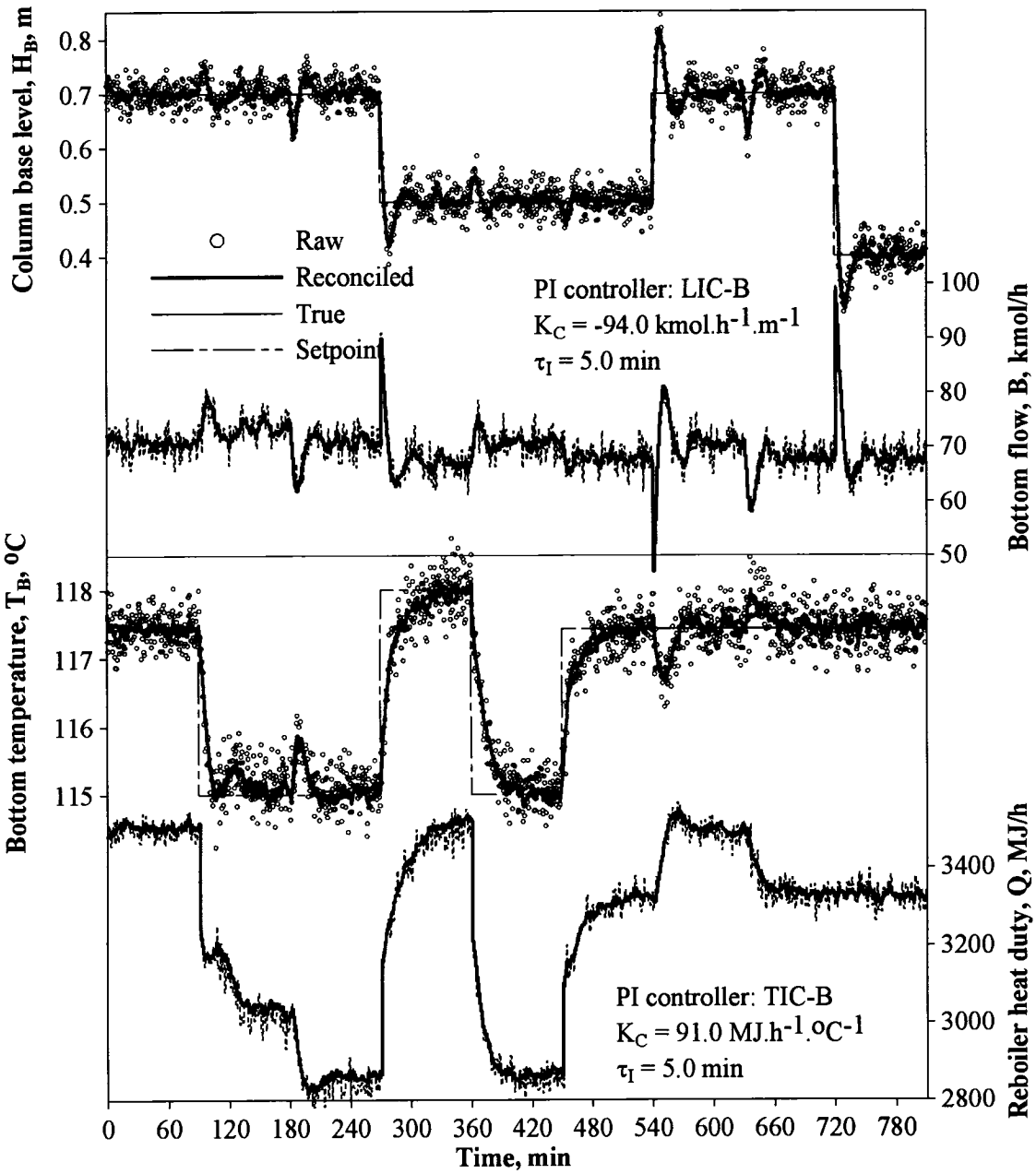


Figure 7(b) Closed-loop performance of nonlinear DDR and controllers LIC-B and TIC-B when the distillation column submitted to a series of controller setpoint changes. The dashed line for the manipulated variables represents the control moves without DDR.

The above case studies indicate a better model can improve the performance of DDR, causally resulting in better control of the process. For the distillation column, the performance of the control system was studied when "perfect" models were assumed

and used in the DDR so that the measurement noise was completely eliminated by the DDR. In other words, the performance of the control system was investigated without measurement noise. In this case, results of the optimum controller parameters are presented in Case V of Table 3 and the corresponding ISE and ISDU values as well as the overall cost function are presented in Case V of Table 4. Under the ideal conditions, as expected, the gains of the controllers, except for the reflux drum level, increased significantly. The values of ISE and ISDU for each controller decreased. The overall cost function of the control system reduced from $\bar{\Phi} = 5.18 \pm 0.01$ to $\Phi = 4.80$, leaving a margin of 8% to be improved between the nonlinear DDR and the "perfect" DDR. The overall cost function with the "perfect" DDR represents the ideal targets of the control system. The margins between the achieved overall cost functions without DDR, with linear DDR, adaptive linear DDR and nonlinear DDR to the ideal targets are presented in Figure 8. This figure shows that even with the linear DDR, the margin was remarkably reduced, a 65% reduction, compared to the case without DDR. However, the gap was further reduced by 79% and 90% respectively with the adaptive linear DDR and the nonlinear DDR.

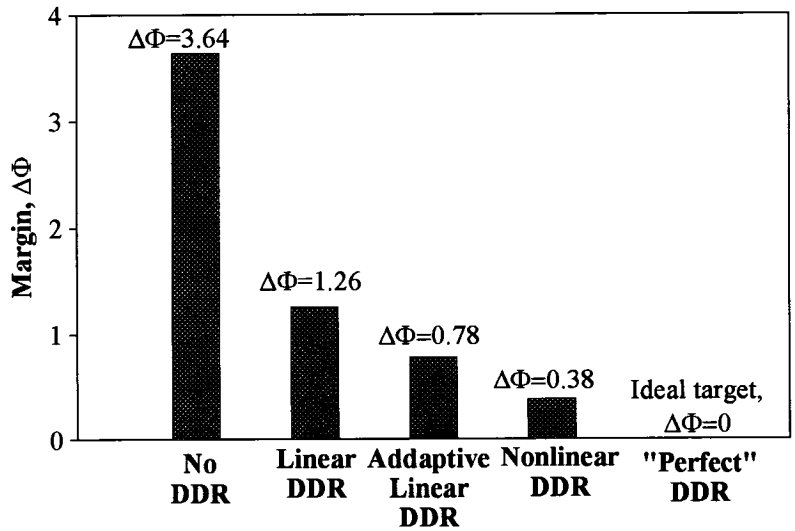


Figure 8 Margins between the achieved overall cost functions to their ideal target.

5 CONCLUSION

Dynamic data reconciliation was found to provide an effective approach to estimate the current state of a dynamic process by integrating information from both measurements and underlying process models so that controller performance can be significantly improved. When phenomenological models are unavailable for a process, black-box models such as linear or neural network models can be identified and used in the DDR. The structures of black-box models have considerable impact on the performance of the DDR. Indeed, a DDR using simple linear models can successfully attenuate the measurement noise. Further improvement of the DDR performance can be achieved using more comprehensive models that can more efficiently capture the underlying dynamics of the process. The use of simple linear models is first recommended, and then more advanced models can be used if additional noise reduction is required. The trade-off between the noise reduction and the engineering effort in model identification will invariably depend on the judgment of an expert in a plant.

NOMENCLATURE

B : Bottom flow rate, (kmol/h)

c_i : Network connection weights for bias neuron

d : Time delay for input variable

D : Distillate flow rate, (kmol/h)

H_B : Column base level, (m)

H_D : Reflux drum level, (m)

K : Matrix of DDR gain

K_C : Controller gain

K_P : Gain of the process

L : Total number of data points used for network training

N : Number of sampling periods

Q : Reboiler heat duty, (MJ/h)

q : Number of neurons in the hidden layer

r : Order of the autoregressive term

R : Reflux flow rate, (kmol/h)

$\mathbf{R}$: Covariance matrix of model prediction error

s : Order of the moving average term

T_B : Column bottom temperature, (°C)

T_D : Column top temperature, (°C)

t_s : Number of sampling periods

u : Process input variable

$\mathbf{V}$: Covariance matrix of the measurement noise

w : Network connection weights for past process inputs

W : Network connection weights for past process outputs

$\mathbf{x}_t$: Vector of the true values of process variables at time t

$\hat{\mathbf{x}}_t$: Vector of reconciled values of the measured process variables

$\mathbf{y}_t$: Vector of measured values of process variables

$\hat{\mathbf{y}}_t$: Vector of model predicted values

$\hat{\mathbf{z}}_t$: Vector of estimates for unmeasured process variables

Creek letters

ε_t : Vector of random variables representing measurement noise

δ_t : Vector of random variables representing model prediction errors

ξ_t : Estimation error by the DDR algorithm

θ : Dead time

τ : Time constant

Θ : Network connection weights between neurons

φ : Network connection weights between neurons and bias

α_i : Weighting factor ISE in controller performance objective function

β_i : Weighting factor for ISDU in controller performance objective function

τ_I : Controller time constant

σ: Standard deviation of the raw measurements

Φ: Overall cost function of the control system

Acronym

DDR: Dynamic data reconciliation

ISDU: Integral of squared differences of manipulated variable

ISE: Integral of squared errors for controlled variable

MSE: Mean squared errors between reconciled and the true values

MIMO: Multi-input multi-output

FOPDT: First order plus dead time

PIPDT: Pure integrator plus dead time

SISO: Single-input single-output

REFERENCES

Albuquerque, J., and Biegler, L. (1996). Data reconciliation and gross error detection for dynamic systems. *AIChE J.*, **42**, 2841-2856.

Bagajewicz, M., and Jiang, Q. (1997). Integral approach to plant linear dynamic reconciliation. *AIChE J.*, **43**, 2546-2558.

Bai, S., McLean, D.D., and Thibault, J. (2005). Enhancing controller performance via dynamic data reconciliation. *Can. J. Chem. Eng.*, **83**, 515-526.

Binder, T., Blank, L., Dahmen, W., and Marquardt, W. (2002). On the regularization of dynamic data reconciliation problems. *J. Process Control*, **12**, 557-567.

Darouach, M., and Zasadzinski, M. (1991). Data reconciliation in generalized linear dynamic systems. *AIChE J.*, **37**, 193-201.

Henson, M.A., and Seborg, D.E. (1997). *Nonlinear Process Control.* Prentice Hall, New Jersey.

Hodouin, D., and Makni, S. (1996). Real-time reconciliation of mineral processing plant using bilinear material balance equations coupled to empirical dynamic models. *Int. J. Miner. Process.*, **48**, 245-264.

Liebman, M.J., Edgar T.F., and Lasdon L.S. (1992). Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers Chem. Engng*, **16**, 963-986.

Ljung, L. (1999). *System identification, theory for the user*. 2nd Ed., Prentice Hall, Upper Saddle River, NJ.

Marlin, T. E. (2000). *Process control, designing processes and control system for dynamic performance*. 2nd ed., McGraw Hill, Boston.

Ramamurthi, Y., Sistu, P.B., and Bequette B.W. (1993). Control relevant dynamic data reconciliation and parameter estimation. *Computers Chem. Engng*, **17**, 41-59.

Thibault, J. (1991). Feedforward neural networks for the identification of dynamic processes. *Chem. Eng. Comm.*, **105**, 109-128.

Zhu, Y. (2001). *Multivariable system identification for process control*. Elsevier Science, Amsterdam.

APPENDIX A

Taking partial derivatives of the objective function of Equation (4) with respect to $\hat{\mathbf{x}}_t$, and setting it to zero yields

$$\frac{\partial J}{\partial \hat{\mathbf{x}}_t} = \frac{1}{2}\left[\mathbf{V}^{-1}(\mathbf{y}_t - \hat{\mathbf{x}}_t) + \mathbf{R}^{-1}(\hat{\mathbf{y}}_t - \hat{\mathbf{x}}_t)\right] = \mathbf{0} \tag{A.1}$$

Solving Equation (A.1) yields

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}(\mathbf{V}^{-1}\mathbf{y}_t + \mathbf{R}^{-1}\hat{\mathbf{y}}_t) \tag{A.2}$$

Expanding Equation (A.2) gives

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\mathbf{y}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{R}^{-1}\hat{\mathbf{y}}_t \tag{A.3}$$

Adding and subtracting the term $(\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\hat{\mathbf{y}}_t$ in the right-hand side of Equation (A.3) gives

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\mathbf{y}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\left[(\mathbf{V}^{-1} + \mathbf{R}^{-1}) - \mathbf{V}^{-1}\right]\hat{\mathbf{y}}_t$$

$$\hat{\mathbf{x}}_t = (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\mathbf{y}_t + \hat{\mathbf{y}}_t - (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}\hat{\mathbf{y}}_t$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}}_t + (\mathbf{V}^{-1} + \mathbf{R}^{-1})^{-1}\mathbf{V}^{-1}(\mathbf{y}_t - \hat{\mathbf{y}}_t) \tag{A.4}$$

APPENDIX B

The linear black-box models around the nominal steady-state operating point for the distillation column in the s-domain were given by

$$\begin{bmatrix} \hat{H}_D'(s) \\ \hat{T}_D'(s) \\ \hat{H}_B'(s) \\ \hat{T}_B'(s) \end{bmatrix} = \begin{bmatrix} \frac{-0.1087}{s} & \frac{-0.1087}{s} & 0 & \frac{0.0038e^{-10s}}{s} \\ 0 & \frac{-0.06596}{227s+1} & 0 & \frac{0.0297e^{-120s}}{s} \\ 0 & \frac{0.1228e^{-30s}}{s} & \frac{-0.1382}{s} & \frac{-0.0037e^{-10s}}{s} \\ 0 & \frac{-1.8344e^{-218s}}{3672s+1} & 0 & \frac{0.001958}{221s+1} \end{bmatrix} \begin{bmatrix} D'(s) \\ R'(s) \\ B'(s) \\ Q'(s) \end{bmatrix} \quad \text{(B.1)}$$

where H_D' is the reflux drum liquid level, T_D' the top temperature, H_B' the column base liquid level, T_B' the column base temperature, D' the distillate flow rate, R' the reflux flow rate, B' the bottom flow rate, and Q' the reboiler heat duty. All variables are in deviation form. Equations (B.1) were discretized by z-transform, and then the discrete models were used in the DDR.

CHAPTER VI

Use of an Autoassociative Neural Network for Dynamic Data Reconciliation

Shuanghua Bai, Jules Thibault* and David D. McLean

Department of Chemical Engineering

University of Ottawa,
Ottawa, Ontario, Canada K1N 6N5

Published in **16th IFAC World Congress,**
Prague, Czech Republic, July 4-8, 2005 (refereed conference).

* Corresponding author. Tel: 1-613-562-5800 ext. 6094
Email: thibault@genie.uottawa.ca

Preface

The DDR algorithms developed in Chapters 2-5 require calculating model predictions (e.g., one-step-ahead predictions) at each sampling time, or solving a dynamic optimization problem for cases where model predictions cannot be explicitly calculated. For complex processes, this may result in excessive computation time. To remedy this problem, a method to train an autoassociative neural network (AANN) to perform the DDR is developed in Chapter 6.

ABSTRACT

The technique of dynamic data reconciliation has been previously studied in the literature and shown to be an effective tool to better estimate the true values of process variables by using information from both measured values and process models. Real-time implementation of dynamic data reconciliation involves solving complex optimization problem, leading to large computation time. This paper presents a study on the use of a dynamic **A**utoassociative **N**eural **N**etwork (AANN) for dynamic data reconciliation. Once trained, the AANN can be directly used for online signal validation. Closed-loop performance of the AANN for both linear and nonlinear processes was evaluated using simulations of two storage tank processes. The AANN provided accurate estimates of measured values for the two processes studied in this investigation. *Copyright © 2005 IFAC*

Keywords: Measurement noise, data reconciliation, dynamic neural network, controller performance.

1 INTRODUCTION

Signal validation plays an important role in real-time plant operation since measured signals of process variables are often contaminated by measurement errors that include random and nonrandom errors. The random error is often referred to as measurement noise that can be approximated by a Gaussian distribution. On the other hand, the nonrandom errors include measurement outliers and biases. When the corrupted measurements are used for process monitoring and control, the knowledge of the true state of the process is inaccurate and the performance of controllers may be deteriorated. Therefore, it is imperative to validate the measured signals prior to their use as inputs to the controllers or in management system. The estimation of system variables is traditionally performed using filters such as exponentially weighted moving average (EWMA) or moving average (MA) filters. These filters use measurement temporal redundancy, meaning that past measurements are used to estimate the current state of the process. The simple EWMA or MA filter performs well for steady-state or slow dynamic processes. However, for processes having significant dynamics and for the purpose of detecting nonrandom measurement errors, model-based filters are preferable. The model-based filters employ process dynamic models, such that both measurement temporal and spatial redundancies are used to estimate the current state of the dynamic system. A well known model-based filter is the Kalman filter that employs stochastic linear state-space process models and measurement models. The most attractive advantage of the Kalman filter lies in its optimal estimation in the sense of minimum mean squared prediction errors (Kamen and Su, 1999). However, the optimality of the Kalman filter requires two restrictive prerequisites, linear state-space models and independent Gaussian white noise for both process models and measurements. In its implementation, the Kalman filter is commonly tuned by adjusting the process model and measurement noise covariances and treating them as design parameters. For nonlinear processes, Extended Kalman Filters (EKF) are used whereby nonlinear process models are linearized at each sampling time. However, in chemical engineering, the applications of EKF have met some problems, such as divergent and unreliable results and difficulty in tuning the filter (Wilson et al., 1998).

An alternative approach to the Kalman filter for real-time plant signal validation is to use Dynamic Data Reconciliation (DDR) technique. Data reconciliation integrates information originating from measurements and process models to provide more accurate estimates of process variables. More importantly, the reconciled data are consistent with relationships that exist between process variables, such as mass and heat balances. Data reconciliation techniques were initially developed for steady-state processes in order to calculate process mass and heat balances (Kuehn and Davidson, 1961; Mah and Stanley, 1976; Crowe et al., 1983; and Crowe, 1986). Recent work (e.g., Darouach and Zasadzinski, 1991; Liebman et al., 1992; Albuquerque and Biegler, 1996; Hodouin and Makni, 1996; Bagajewicz and Jiang, 1997) extended the concept of data reconciliation for dynamic processes. In the application of the dynamic data reconciliation techniques, measured variables are optimally adjusted to satisfy exactly process models and constraints. In practice, however, both measurements and process models are prone to errors such that the following objective function is minimized to perform DDR.

$$\text{Minimize } J(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) = (\mathbf{y}_t - \hat{\mathbf{y}}_t)^T \mathbf{V}^{-1} (\mathbf{y}_t - \hat{\mathbf{y}}_t) + \mathbf{f}^T(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) \Omega^{-1} \mathbf{f}(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t) \qquad (1)$$

$$\text{subject to } \mathbf{y}_{t,l} \le \hat{\mathbf{y}}_t \le \mathbf{y}_{t,u}$$

$$\mathbf{z}_{t,l} \le \hat{\mathbf{z}}_t \le \mathbf{z}_{t,u}$$

where $\mathbf{y}_t$ is a vector of measured values of process variables at time t , $\hat{\mathbf{y}}_t$ is a vector of reconciled values of the measured process variables, $\mathbf{V}$ is a covariance matrix of the measurement error. $\mathbf{f}(\hat{\mathbf{y}}_t, \hat{\mathbf{z}}_t)$ is a functional vector of process algebraic or discretized differential equations, $\hat{\mathbf{z}}_t$ is a vector of unmeasured variables or model parameters simultaneously estimated by the DDR algorithm. $\mathbf{y}_{t,l}$, $\mathbf{z}_{t,l}$, $\mathbf{y}_{t,u}$ and $\mathbf{z}_{t,u}$ are vectors of lower and upper bounds. Ω is a covariance matrix of model errors whose elements are often treated as tuning parameters due to the difficulty to determine them.

The benefits of using DDR for real-time process monitoring and control have been reported. Ramamurthi et al. (1993) proposed that DDR led to better closed-loop

performance of nonlinear predictive controller. Soderstrom et al. (2000) implemented a real-time dynamic data reconciliation strategy to improve inventory calculations for a diluent plant. Abu-el-zeet et al. (2002) claimed that DDR, in conjunction with systematic bias detection, enhanced a model predictive control scheme. Bai et al. (2004) applied a data reconciliation filter embedded in PI feedback control loops for the control of a binary distillation column, and demonstrated that the controller performance was significantly improved by the DDR. The benefits of using DDR for real-time signal validation include improving controller performance, reducing the variability of controlled variable at process nominal steady-state, estimating unmeasured process variables, as well as detecting measurement biases.

The DDR algorithm requires nonlinear programming at each sampling time such that long computer time may be needed for complex processes. To reduce the computation, this paper proposes to train an **a**utoassociative **n**eural **n**etwork (AANN) to directly perform data reconciliation. Once trained, the neural network can perform data reconciliation without any iteration, and the neural DDR becomes more suitable for real-time applications.

2 AUTOASSOCIATIVE NEURAL NETWORK

2.1 Architecture of AANN

An AANN is a network having similar architecture as a conventional feedforward neural network composed of an input layer, hidden layers and an output layer as illustrated in Figure 1. The input and output layers have the same number of neurons determined by the nature of problem being solved. The first and third hidden layers, called mapping and demapping layers respectively, contain a relatively larger number of neurons. The second hidden layer, called bottleneck layer, contains less neurons. The AANN is trained to reproduce its inputs as its outputs. The key feature of AANN is to perform data compression by the bottleneck layer. The input layer, mapping and bottleneck layers compress the input information to a lower dimension, and then the demapping and output layers regenerate the main underlining features of the original

information of the inputs. The mapping/demapping process enables the network to represent the input information in a compressed form that can often reveal the essence of the data. For training AANNs, the input and the target vectors presented to the network are identical, and the mean of squared errors between the network outputs and its inputs are minimized. After successful training, the AANN can be regarded as a filter that can be used for filtering random noise and detecting fault sensors for steady-state or stationary processes (Kramer, 1992).

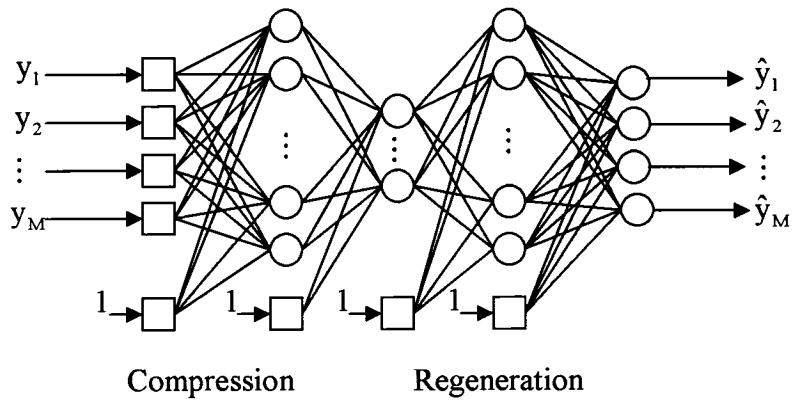


Figure 1 Architecture of an AANN for steady-state processes.

Du et al. (1997) used an autoassociative neural network for nonlinear steady-state data reconciliation. They developed a mass-balance-related AANN scheme to rectify flow rates and mass compositions. The mass balance of a system was incorporated directly into the objective function such that the redundant information for the measurements was taken into account in training the network. With the redundant information provided by the process models, the performance of the AANN is expected to perform significantly better than that of an AANN trained using only the measurements for measurement noise reduction, systematic bias detection, and estimation of unmeasured process variables.

2.2 Dynamic AANN for DDR

For dynamic data reconciliation, the static structure of the feedforward AANN of Figure 1 must be modified to encapsulate the dynamics of the process. The AANN architecture of Figure 2 has been selected. To capture the dynamics of the process, the reconciled output $\hat{\mathbf{y}}_t$, delayed a number of times, is fed back to the input layer. This

AANN incorporates both temporal and spatial patterns. In Figure 2, D represents the required number of time delays for the process output variables, and $\mathbf{u}_{t-d}$, ..., $\mathbf{u}_{t-d-I}$ represent the inputs with time delay of d, d+1, ..., d+I.

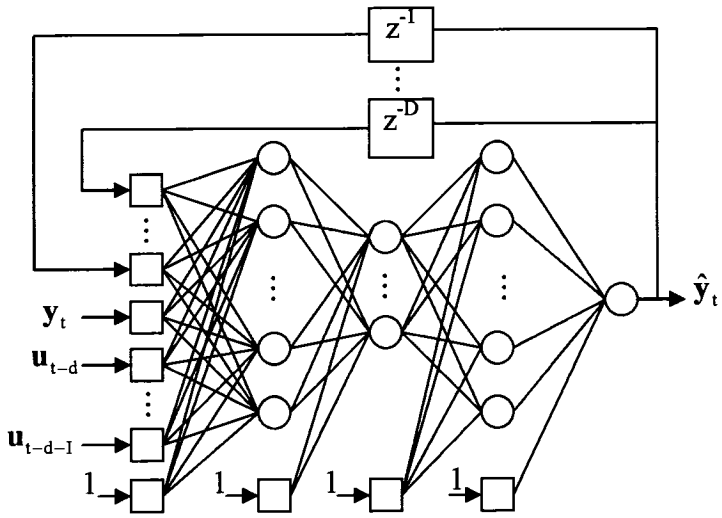


Figure 2 Architecture of an AANN for dynamic processes.

It is important to note that the number of neurons in the input and output layers of the dynamic AANN are not necessarily identical as they are in the static AANN, but their number is problem-dependent. In addition, the number of neurons in each hidden layer is determined experimentally.

Prior information relating the input and output variables of the process is used to train the AANN for DDR. The objective function to train the network can be written as

$$\text{Minimize } J = \frac{1}{N}\sum_{t=0}^{N}\Big[(\mathbf{y}_t-\hat{\mathbf{y}}_t)^T\mathbf{V}^{-1}(\mathbf{y}_t-\hat{\mathbf{y}}_t)+\mathbf{f}^T(\hat{\mathbf{y}}_t,\hat{\mathbf{y}}_{t-1},\ldots,\mathbf{u}_{t-d},\mathbf{u}_{t-d-1},\ldots)\boldsymbol{\Omega}^{-1} \mathbf{f}(\hat{\mathbf{y}}_t,\hat{\mathbf{y}}_{t-1},\ldots,\mathbf{u}_{t-d},\mathbf{u}_{t-d-1},\ldots)\Big] \quad (2)$$

where $\mathbf{f}(\hat{\mathbf{y}}_t,\hat{\mathbf{y}}_{t-1},\ldots,\mathbf{u}_{t-d},\mathbf{u}_{t-d-1},\ldots)$ is a functional vector of process dynamic models. In the first iteration in training of the dynamic AANN, the output vector fed back to the input layer is not known, but can be assigned the raw measurements. Then, the network is trained until satisfactory convergence criterion is met. After the first iteration, the vectors

of the network outputs are fed back as the inputs, and then the network is trained again. After several recurrent iterations, the feedback vectors and the objective function will not change, indicating the training of the dynamic network has been completed. After the successful training, the dynamic AANN can be incorporated to the plant for real-time data reconciliation.

3 EXAMPLES

To illustrate the use of autoassociative neural networks for dynamic data reconciliation, two examples were investigated. One example is a cylindrical storage tank process. The other is a spherical storage tank process. Both processes are controlled by conventional PI controllers. The developed dynamic AANNs were used as DDR filters to provide better estimates of the controlled variable for the controller. The performance of the AANNs was examined with closed loop control of the processes as illustrated by Figure 3.

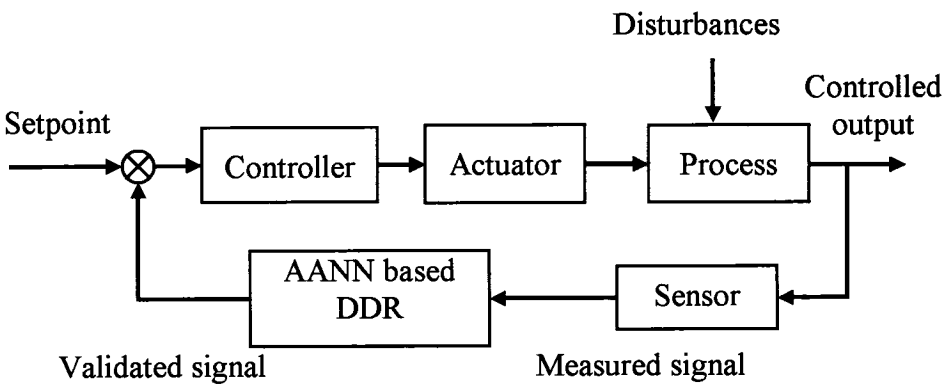


Figure 3 Scheme of implementing an AANN for signal validation in a control loop.

3.1 Cylindrical storage tank process

The schematic diagram of the cylindrical storage tank process is shown in Figure 4. A PI controller was used to regulate the liquid level of the tank by manipulating the outlet flow. The feed flow to the tank was measured but not controlled. The sampling interval was 1 min. The nominal feed flow to the tank was 1.8 m^3/h. Random Gaussian white noise with standard deviations σ_{Fi}=0.09 m^3/h for the feed flow and σ_H=0.1 m for

the tank level corrupted the measurements. The dynamics of the measuring device and control valve were neglected. The model of the process is given by the mass balance

$$A\frac{dH}{dt} = F_i - F_o \tag{3}$$

where H is the liquid height in the tank, F_i and F_o are the feed and outlet flow rates, and A is the cross-sectional area of the tank.

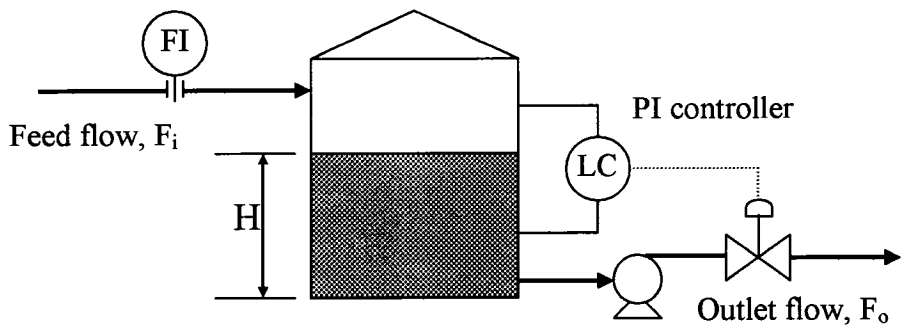


Figure 4 Cylindrical storage tank process.

Under open-loop conditions and with measurement noise, the tank was perturbed by random step changes in the feed and outlet flow rates. 1000 samples were simulated. 90% of the samples were used to train an AANN and the remaining 10% was used for network validation. An AANN having the structure [5, 6, 3, 6, 2] determined experimentally, shown in Figure 5, was trained by minimizing the objective function

$$J = \sum_{p=1}^{N}\left\{\frac{1}{\sigma_{F_i}^2}\left(F_{i,t-1} - \hat{F}_{i,t-1}\right)^2 + \frac{1}{\sigma_H^2}\left(H_t - \hat{H}_t\right)^2 + \frac{1}{\sigma_{Model}^2}\left[\hat{H}_t - \hat{H}_{t-1} - \frac{\Delta t}{A}\left(\hat{F}_{i,t-1} - F_{o,t-1}\right)\right]^2\right\} \tag{4}$$

where the discretized model of Equation (3) was incorporated into the objective function and Δt represents the sampling time interval.

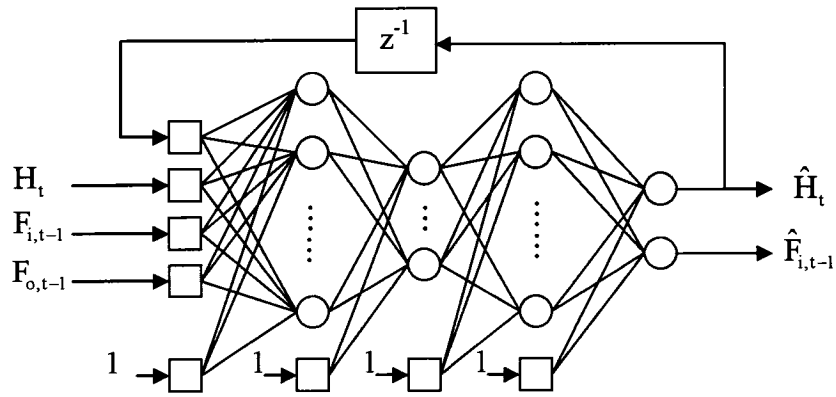


Figure 5 Structure of dynamic AANN for the cylindrical storage tank process.

Because the variance of the model error was not known, the value of σ^2_{Model} in the objective function of Equation (4) was treated as a tuning parameter. For a value of σ^2_{Model}, the network was trained completely, and then tested using the validation data sets. The variances of the reconciled tank level $\hat{H}_t$ and the reconciled feed flow $\hat{F}_{i,t-1}$, evaluated using the validation data set, are presented in Figure 6. The variance of $\hat{H}_t$ decreased and then increased with an increase of σ^2_{Model}. When σ^2_{Model} was set to small values, the model mismatch distorted the reconciled tank level, so that the variance of $\hat{H}_t$ increased dramatically. On the other hand, when σ^2_{Model} was set to larger values, the measurements were not constrained severely such that the variance of $\hat{H}_t$ approached the variance of the raw values. However, the variance of the reconciled feed flow was not affected by the change of σ^2_{Model}, and nearly matched the variance of raw measurements, because the noise in the disturbances was damped by the inertia of the process and the model had nearly no constraint on it.

The optimal value for the tuning parameter was $\sigma^2_{Model}=9.0\times10^{-4}$, and the corresponding variance of the reconciled tank level evaluated was $\hat{\sigma}^2_{\hat{H}_t} = 9.72\times10^{-4}$. Given $\sigma^2_{H_t} = 1.0\times10^{-2}$, then $\hat{\sigma}^2_{\hat{H}_t} / \sigma^2_{H_t} = 0.097$ indicates the AANN performed very well in estimating the true values of the tank level. Using the optimal value of σ^2_{Model} to train the network, results of

the raw, reconciled and true values of the tank level for network training and validation for the last 200 data sets are presented in Figure 7.

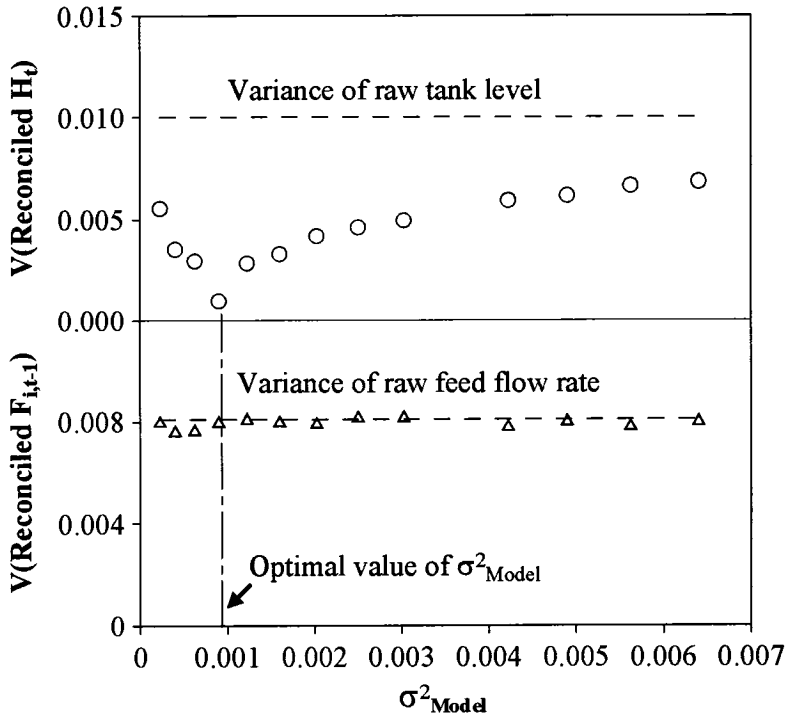


Figure 6 Variance of reconciled tank level and feed flow as a function of σ^2_{Model}.

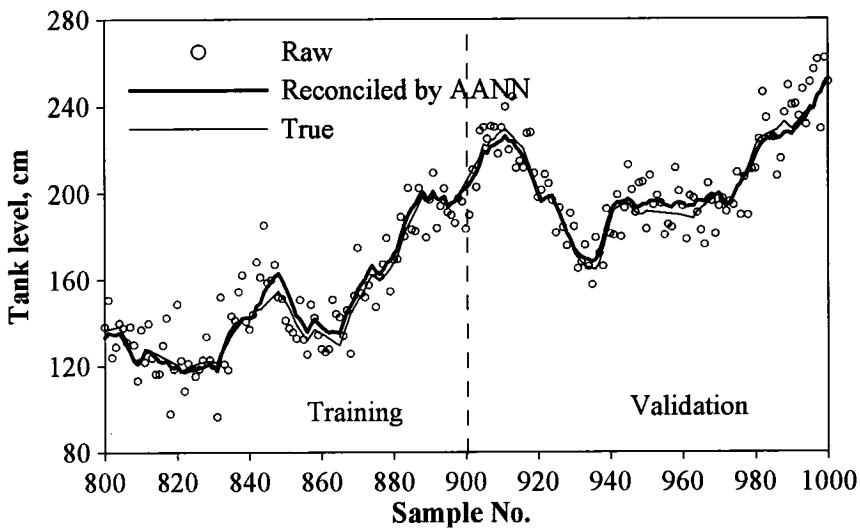


Figure 7 Raw, reconciled and true values for cylindrical tank level in network training and validation.

The AANN was then embedded inside the feedback loop before the controller calculates the control moves. Closed-loop performance of the AANN was examined for controller

setpoint changes. Results of raw, reconciled and true values for the controlled variables as well as the control moves with and without the AANN, are presented in Figure 8. The reconciled values for the tank level were less noisy than the raw measurements and close to their true values. In addition, the saturated high-frequency oscillations of the control moves were significantly reduced with the embedded AANN filter. It is worth noting that the reconciled tank level displayed some degree of deviations from the true values after the change of setpoint. This is because the saturated manipulations resulted in less accurate predictions of the AANN.

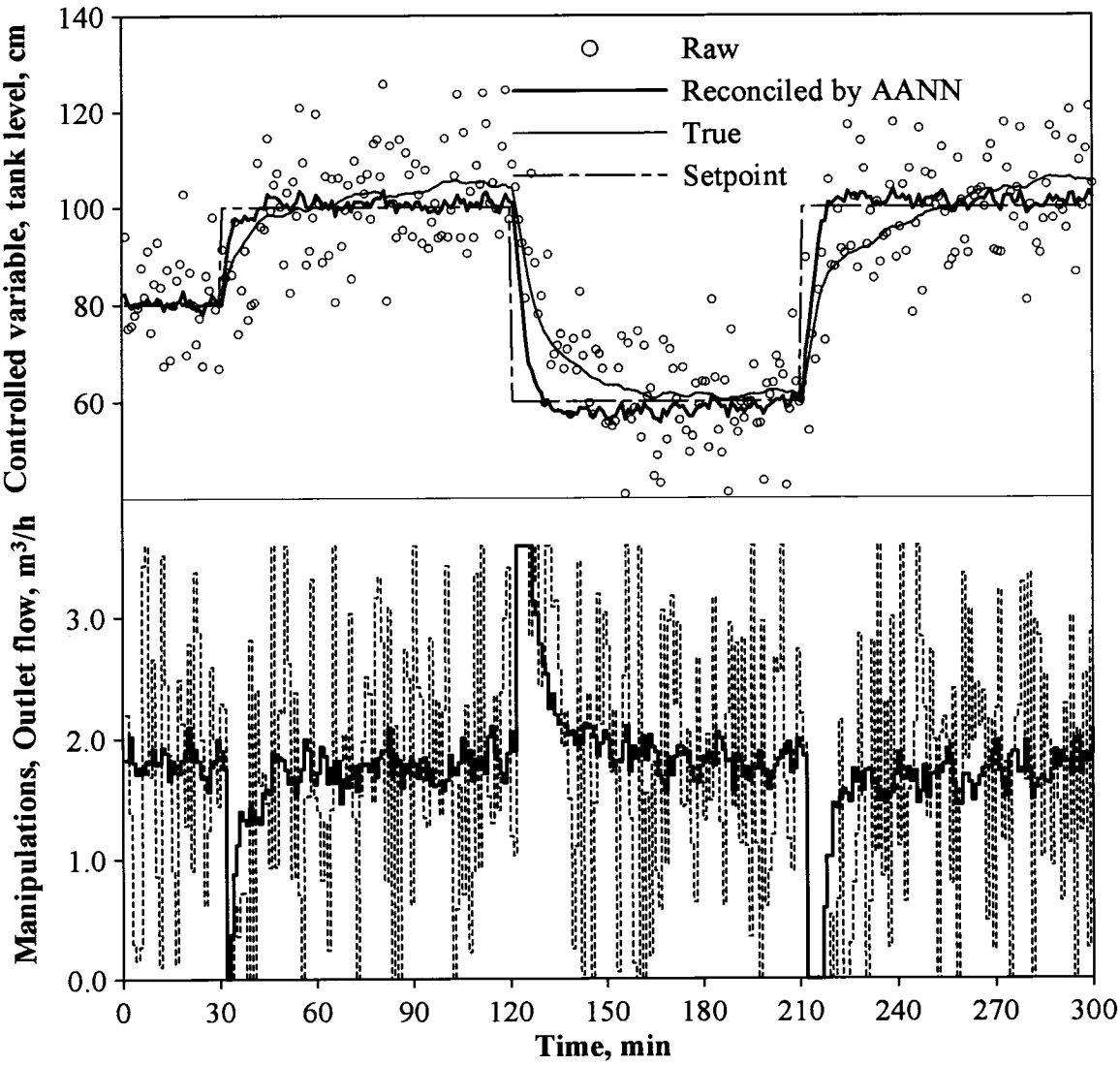


Figure 8 Closed-loop performance of AANN for setpoint changes. Dashed line in manipulated variable represents control moves without AANN.

3.2 Spherical storage tank process

The spherical storage tank process, studied in this work, is shown in Figure 9. The radius of the tank was R = 5.0 m. The nominal feed flow to the tank was 60.0 m^3/h. A PI controller was used to control the tank level by manipulating the outlet flow rate. The sampling time was 1 min. The nominal measurement noise level for the feed flow rate was σ_{F_i}=0.5 m^3/h, and σ_H=0.1 m for the liquid level. The process model for the spherical tank is the mass balance

$$(2\pi RH - \pi H^2)\frac{dH}{dt} = F_i - F_o \tag{5}$$

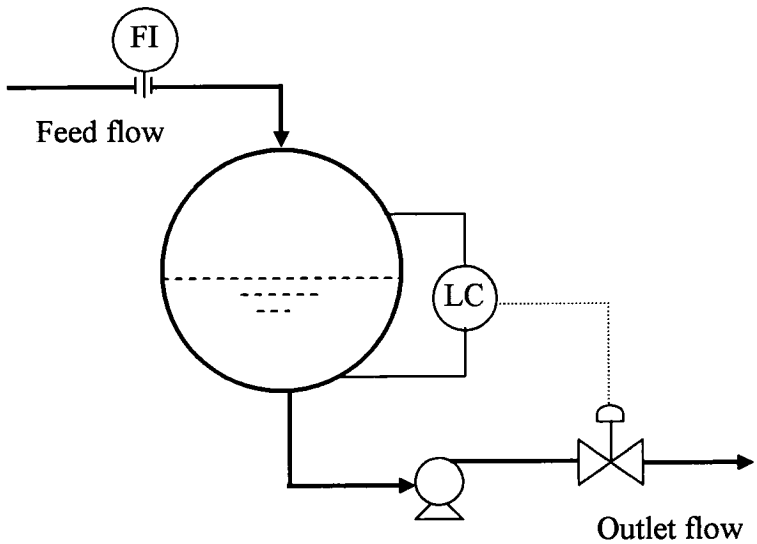


Figure 9 Spherical storage tank process.

The tank, in open-loop, was disturbed by a series of random step changes in the feed flow rate as well as in the outlet flow rate. An AANN selected to have the structure [5, 14, 7, 14, 2] was trained by minimizing the objective function

$$J = \sum_{p=1}^{N}\left\{\frac{1}{\sigma_{F_i}^2}\left(F_{i,t-1} - \hat{F}_{i,t-1}\right)^2 + \frac{1}{\sigma_H^2}\left(H_t - \hat{H}_t\right)^2 + \frac{1}{\sigma_{Model}^2}\left[\left(2\pi R\hat{H}_{t-1} - \pi\hat{H}_{t-1}^2\right)\left(\hat{H}_t - \hat{H}_{t-1}\right) - \Delta t\left(\hat{F}_{i,t-1} - F_{o,t-1}\right)\right]^2\right\} \tag{6}$$

where the disretized model of Equation (5) was incorporated into the objective function to train the network. The "best" value of σ^2_{Model} used to train the network was found to be 0.2, and after successfully training it resulted in the variance of the reconciled liquid level for the validation data sets was $\hat{\sigma}^2_{\hat{H}_t} = 2.83\times10^{-4}$ and the variance of reconciled feed flow was $\hat{\sigma}^2_{\hat{F}_{i,t-1}} = 0.26$. Compared to $\sigma^2_{H_t} = 1.0\times10^{-2}$ and $\sigma^2_{F_{i,t-1}} = 0.25$, the network filtered almost completely the noise for the controlled variable, but for the disturbance variable, the variance of the reconciled values was not affected. The raw measurements, reconciled and true values for the tank level for network training and validation for the last 400 data points are presented in Figure 10. It shows, the noise in the controlled variable was significantly reduced and the reconciled values by the AANN performed well in tracking their true values.

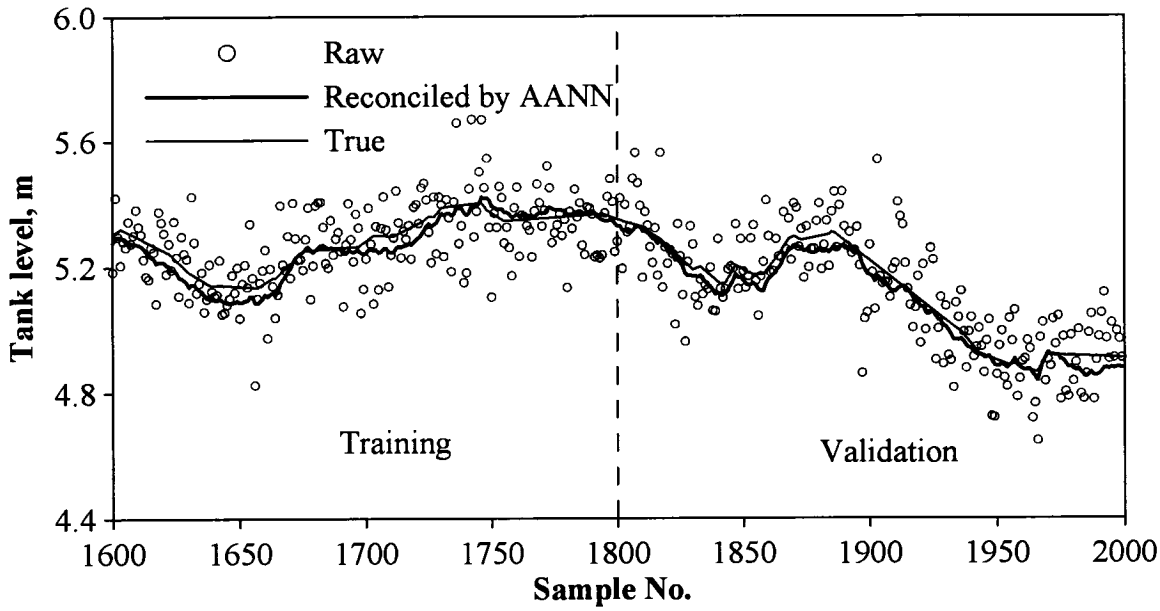


Figure 10 Raw, reconciled and true values of spherical tank level for network training and validation.

The closed-loop performance of the AANN for the spherical tank was examined for step changes in the feed flow rate, having magnitudes of -33% and 66% of steady state value, and for control setpoint changes. Results of raw, reconciled and true values for the controlled and manipulated variables for the load changes are presented in Figures 11 and

results for the controller setpoint changes are presented in Figure 12. For both cases, the AANN performed very well in tracking the true values of the controlled variable even if the process had significant dynamic changes and the process is nonlinear. In addition, the high-frequency oscillations of control moves were eliminated. Due to the significant reduction of noise propagation inside the control loop, the controller was allowed to be tuned more aggressively, such that faster dynamic response of the process was expected for load and setpoint changes. As a consequence, the performance of the controller was improved by the embedded AANN.

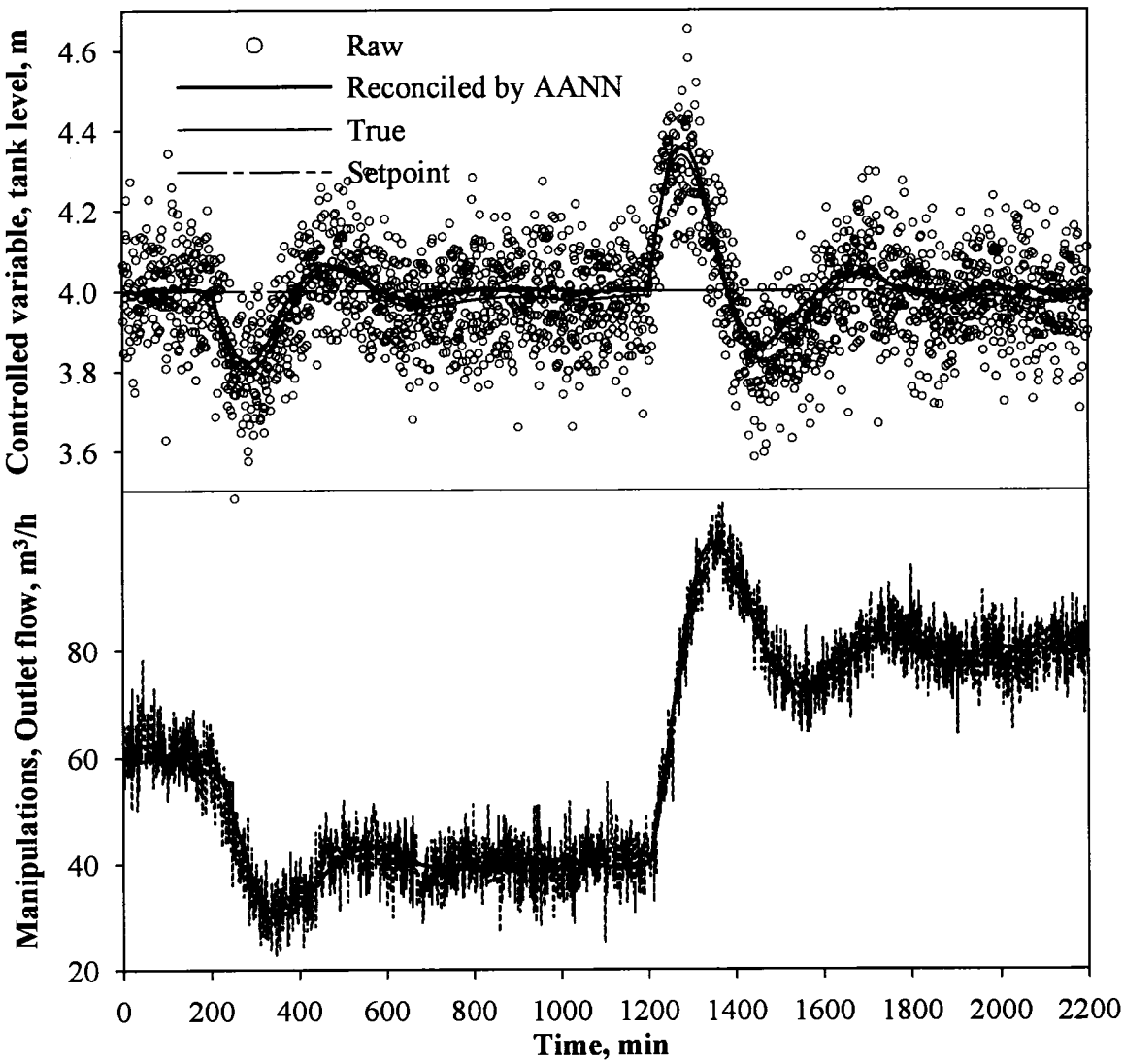


Figure 11 Raw, reconciled and true values for the controlled and manipulated variables for the spherical tank for step changes in the feed flow rate. Dashed line in manipulated variable represents control moves without AANN.

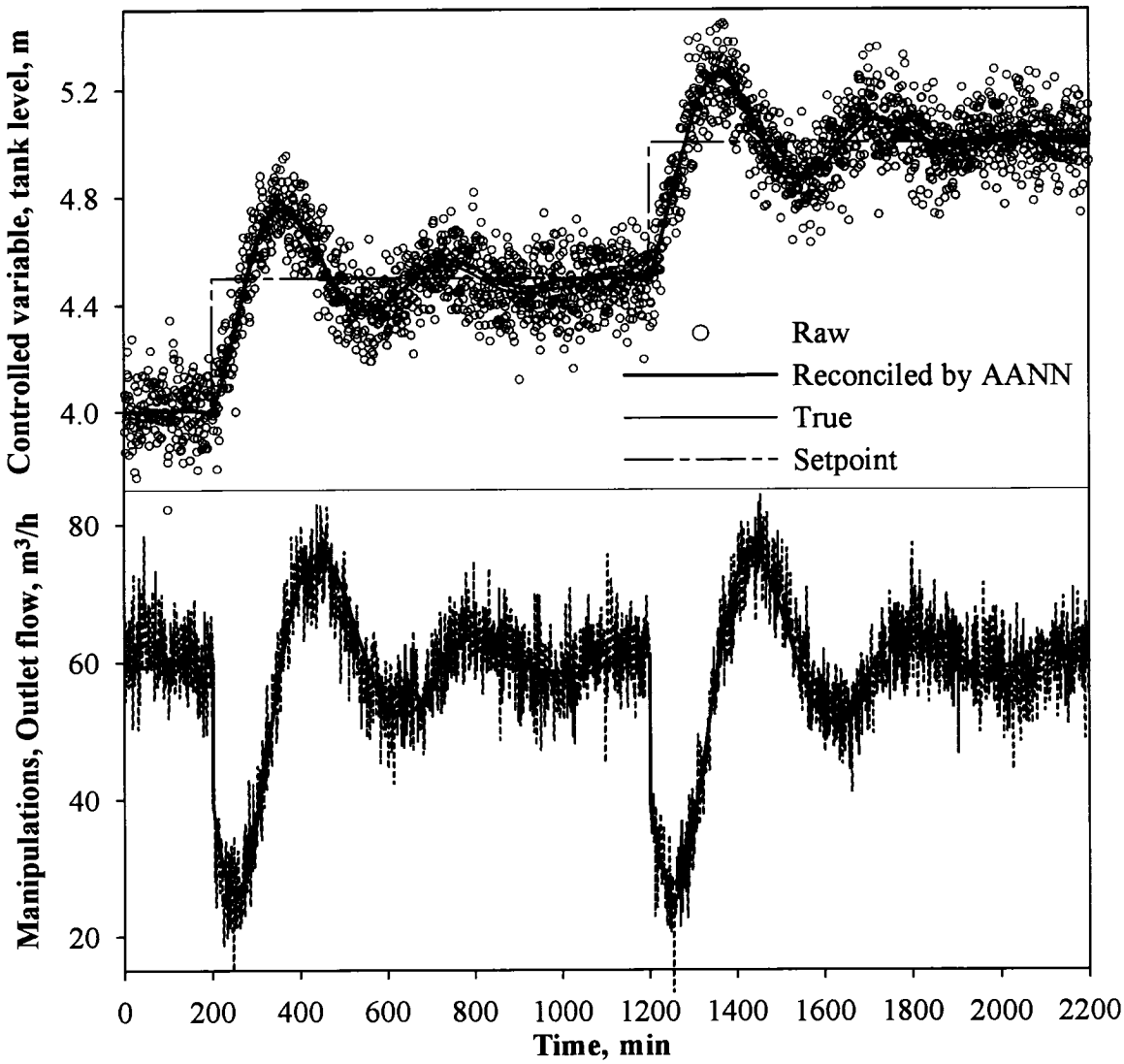


Figure 12 Raw, filtered and true values for the controlled and manipulated variable for setpoint changes. Dashed line in manipulated variable represents control moves without AANN.

4 CONCLUSION

The dynamic AANN presented in this work has shown to be an alternative approach for real-time signal validation for dynamic processes. Because it is trained by incorporating intrinsic process models, the performance of the dynamic AANN is more effective for estimation of the current state of the process. Although implementation of

AANN filters for complex processes (e.g., multivariable systems) and comparisons for the AANN to DDR using linear or nonlinear models are required for further demonstration, we believe that the use of this technique can greatly improve plant monitoring as well as controller performance.

NOMENCLATURE

A : Cross-sectional area of tank, (m^2)

D : Number of time delays for process output variables

d : Number of time delay for process input variables

F_i : Tank feed flow rate, (m^3/h)

F_o : Tank outlet flow rate, (m^3/h)

H : Tank liquid level, (m)

N : Number of data points used for network training

$\boldsymbol{\Omega}$: Covariance matrix of model errors

R : Radius of spherical tank, (m)

$\mathbf{u}_t$: Vector of process input variables

$\mathbf{V}$: Covariance matrix of measurement errors

$\mathbf{y}_t$: Vector of measured values of process variables at time t

$\hat{\mathbf{y}}_t$: Vector of reconciled values for measured process variables

$\hat{\mathbf{z}}_t$: Vector of estimated unmeasured process variables model parameters

σ^2 : Variance

Δt : Sampling time interval

Acronyms

AANN: Autoassociative neural network

DDR: Dynamic data reconciliation

EKF: Extended Kalman filter

EWMA: Exponentially weighted moving average

MA: Moving average

REFERENCES

Abu-el-zeet, Z., P.D. Roberts and V.M. Becerra (2002). Enhancing model predictive control using dynamic data reconciliation. *AIChE J.*, **48**, 324-333.

Albuquerque, J. and L. Biegler (1996). Data reconciliation and gross error detection for dynamic systems. *AIChE J.*, **42**, 2841-2856.

Bagajewicz, M. and Q. Jiang (1997). Integral approach to plant linear dynamic reconciliation. *AIChE J.*, **43**, 2546-2558.

Bai, S., J. Thibault and D.D. McLean (2005). Closed-loop data reconciliation for the control of a binary distillation column, *Chem. Eng. Comm.*, **192**, 1444-1467.

Crowe, C.M., Y.A.G Campos and A. Hrymak (1983). Reconciliation of process flow rates by matrix projection. Part I: Linear case. *AIChE J.*, **29**, 881-888.

Crowe, C.M. (1986). Reconciliation of process flow rates by matrix projection. Part II: Nonlinear case. *AIChE J.*, **32**, 616-623.

Darouach, M. and M. Zasadzinski (1991). Data reconciliation in generalized linear dynamic systems. *AIChE J.*, **37**, 193-201.

Du, Y., D. Hodouin and J. Thibault (1997). Use of novel autoassociative neural network for nonlinear steady-state data reconciliation, *AIChE J.*, **43**, 1785-1796.

Hodouin, D. and S. Makni (1996). Real-time reconciliation of mineral processing plant data using bilinear material balance equations coupled to empirical dynamic models. *Int. J. Miner. Process.*, **48**, 245-264.

Kamen, E.W. and Su, J.K. (1999). *Introduction to Optimal Estimation*, Springer, London.

Kramer, M.A. (1992). Autoassociative neural networks. *Computers Chem. Engng*, **16**, 313-328.

Kuehn, D.R. and H. Davidson (1961). Computer control, II Mathematics of control. *Chemical Engineering Progress*, **57**, 44.

Liebman, M.J., T.F. Edgar and L.S. Lasdon (1992). Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers Chem. Engng*, **16**, 963-986.

Mah, R.S.H., G. Stanley and D. Downing (1976). Reconciliation and rectification of process flow and inventory data. *Ind. Eng. Chem. Proc. Des.* Dev., **15**, 175-183.

Ramamurthi, Y., P.B. Sistu and B.W. Bequette (1993). Control relevant dynamic data reconciliation and parameter estimation. *Computers Chem. Engng*, **17**, 41-59.

Soderstrom, T.A., T.F. Edgar, L.P. Russo and R.E. Young (2000). Industrial application of a large-scale dynamic data reconciliation strategy. *Ind. Eng. Chem. Res.* **39**, 1683-1693.

Wilson, D.I., M. Agarwal and D.W.T. Rippin (1998). Experience implementing the extended Kalman filter on an industrial batch reactor. *Computers Chem. Engng*, **22**, 1653-1672.

CHAPTER VII

Autoassociative Neural Networks for Robust Dynamic Data Reconciliation

Shuanghua Bai, David D. McLean and Jules Thibault*

Department of Chemical Engineering

University of Ottawa

Ottawa, Canada K1N 6N5

Submitted to **AIChE Journal**.

* Corresponding author. Tel: 613-562-5800 ext. 6094

Email: thibault@genie.uottawa.ca

Preface

The AANN-based DDR algorithm developed in Chapter 6 is evaluated for the control of a linear process, the cylindrical storage tank, and a nonlinear process, the spherical storage tank. The storage tanks have multiple inputs and a single output. Phenomenological models (mass balances) for the two processes are used to train the AANNs to perform DDR. Unfortunately, such phenomenological models are often unavailable for complex chemical processes. Consequently, Chapter 7 explores the methodologies of AANN-based DDR using empirical process models to describe the distillation process. Moreover, strategies for implementing AANN-based DDR to multiple input and multiple output processes are demonstrated.

ABSTRACT

Reliable estimation of dynamic process variables for plant monitoring and control is an important topic that has been studied extensively. The Kalman filter has often been used and has acquired an enviable reputation. However, the use of the Kalman filter suffers from two restrictive conditions: it employs state-space models and it has to be tuned online in order to achieve its best performance. Recently, alternative methodologies based on dynamic data reconciliation have been proposed. Although the approach of dynamic data reconciliation can incorporate any form of model, it involves online optimization which may require long computation time for complex systems. This paper explores a new methodology based on a combination of autoassociative neural networks and dynamic data reconciliation. It overcomes the need for online tuning, as required by the Kalman filter, and online optimization as required by conventional dynamic data reconciliation methods. Simulation examples of a distillation column demonstrate that the autoassociative neural network based dynamic data reconciliation approach is capable of effectively attenuating measurement noise and is robust to changes of the noise level in plant measurements.

Keywords: Process variable estimation, Autoassociative neural network, Dynamic data reconciliation, Kalman filter.

1 INTRODUCTION

A critical concern in plant operation is the estimation of the current state of a dynamic process. Unfortunately, process measurements often contain some degree of inaccurate information, such as measurement noise. The presence of measurement noise not only prevents plant operators from identifying true values of process variables, but also inhibits the performance of automatic control systems. To cope with this problem, the model-based Kalman filter has been most widely used (Kamen & Su, 1999). The Kalman filter employs stochastic state-space process and measurement models. In its discrete form it is represented by

$$\mathbf{x}_t = \mathbf{f}\left(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}\right) + \mathbf{w}_{t-1} \tag{1}$$

$$\mathbf{y}_t = \mathbf{g}\left(\mathbf{x}_t\right) + \boldsymbol{\varepsilon}_t \tag{2}$$

where $\mathbf{x}_t$ is a vector of process state variables at time t, $\mathbf{y}_t$ is a vector of measured process variables, and $\mathbf{u}_{t-1}$ is a vector of process input variables. $\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ is a functional vector representing process models. $\mathbf{g}(\mathbf{x}_t)$ is a functional vector representing measurement models that reduces to $\mathbf{x}_t$ if all the state variables are directly measured. $\mathbf{w}_{t-1}$ is a vector of random variables denoting white Gaussian process model noise, i.e., $\mathbf{w}_{t-1} \sim \mathrm{N}(\mathbf{0}, \mathbf{S})$. $\boldsymbol{\varepsilon}_t$ is a vector of random variables representing white Gaussian measurement noise, i.e., $\boldsymbol{\varepsilon}_t \sim \mathrm{N}(\mathbf{0}, \mathbf{V})$. For the problem of process estimation defined by Equations (1) and (2), the optimal estimates, $\hat{\mathbf{x}}_t$, using the Kalman filter are given by

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t\left[\mathbf{y}_t - \mathbf{C}_t\mathbf{g}(\hat{\mathbf{x}}_t^-)\right] \tag{3}$$

where $\hat{\mathbf{x}}_t^-$ is the a priori estimate (i.e., the vector of model predicted values) given by

$$\hat{\mathbf{x}}_t^- = \mathbf{f}\left(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}\right), \tag{4}$$

$\mathbf{K}_t$ is the gain of the Kalman filter, recursively calculated by

$$\mathbf{P}_t^- = \mathbf{A}_t \mathbf{P}_{t-1} \mathbf{A}_t^T + \mathbf{S} \tag{5}$$

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{C}_t^T \left(\mathbf{C}_t \mathbf{P}_t^- \mathbf{C}_t^T + \mathbf{V} \right)^{-1} \tag{6}$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t \mathbf{C}_t \mathbf{P}_t^- \tag{7}$$

In Equations (5~7), $\mathbf{A}_t$ is a Jacobian matrix obtained by linearizing $\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ with respect to $\mathbf{x}_{t-1}$ at the realization $\mathbf{x}_{t-1} = \hat{\mathbf{x}}_{t-1}$, $\mathbf{C}_t$ is a Jacobian matrix obtained by linearizing $\mathbf{g}(\mathbf{x}_t)$ at the realization $\mathbf{x}_t = \hat{\mathbf{x}}_t^-$. $\mathbf{P}_t^-$ and $\mathbf{P}_t$ are the covariance matrices of a priori and a posteriori estimation errors with initial values assumed for $\mathbf{P}_t$ (e.g., $\mathbf{P}_0 = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix).

The Kalman filter is a time-variant system. However, when $\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ and $\mathbf{g}(\mathbf{x}_t)$ are linear in terms of the state variables, the gain of the filter $\mathbf{K}_t$ reaches a constant value. At this point the Kalman filter becomes time-invariant. The most attractive advantage of the Kalman filter lies in its optimality in the sense of providing minimum variance estimates. However, the optimality of the Kalman filter requires two restrictive prerequisites: linear state-space models and independent white Gaussian noise for both process and measurements. Usually, it is difficult to determine the statistical properties of the process model noise, $\mathbf{w}_{t-1}$. As a result, elements in the covariance matrix of $\mathbf{w}_{t-1}$ are commonly viewed as tuning parameters rather than measurable constants in the implementation of the Kalman filters (Wilson et al., 1998).

In addition to the Kalman filter, another approach, dynamic data reconciliation (DDR) has been developed in recent years for process variable estimation (e.g., Liebman et al., 1992; Ramamurthi et al., 1993; Albuquerque and Biegler, 1996; Hodouin and Makni, 1996; Bagajewicz and Jiang, 1997; Binder et al., 2002). Mathematically, the problem of

process estimation by the DDR was initially formulated as the following constrained least-squares optimization problem

$$\text{Minimize } J(\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_t) = \sum_{i=0}^{t} \left[\left(\mathbf{y}_i - \hat{\mathbf{x}}_i\right)^T \mathbf{V}^{-1} \left(\mathbf{y}_i - \hat{\mathbf{x}}_i\right) \right] \tag{8}$$

$$\text{subject to } \mathbf{f}\left[\frac{d\hat{\mathbf{x}}}{dt}, \hat{\mathbf{x}}\right] = \mathbf{0}$$

where $\mathbf{f}$ represents a functional vector of constraints consisting of dynamic models. To solve the optimization problem for data reconciliation, the differential equations were discretized or integrated and then the DDR problem was converted to minimize the objective function constrained by algebraic equations. Subsequently, the optimization was solved analytically or by nonlinear programming (NLP). The dimensionality of the dynamic optimization problem defined by Equation (8) increases linearly with the number of sampling periods since all measurements from time zero are considered. To reduce the magnitude of the online optimization problem, a moving window in the time horizon, [t-H, t], has normally been used. At time t, only the measurements within the moving window were reconciled, and only the estimates for the current time were used for process monitoring and/or control. A critical assumption in DDR, as formulated by Equation (8), is that the models represent the true dynamics of the process and the estimates (reconciled values) exactly satisfy the model constraints. However, no mathematical model perfectly represents a real plant; consequently model mismatch should be taken into account in the data reconciliation algorithm. Furthermore, large-scale online optimization using NLP may require excessively long computation time.

Accordingly, Bai et al. (2005) proposed using autoassociative neural networks (AANN) to perform DDR. An AANN for a dynamic process was employed and trained offline. The training objective function consisted of minimizing simultaneously the weighted sum of squared measurement and model errors. Dynamic process models with various structures can be encapsulated into the training objective function. An AANN can be directly implemented online to perform dynamic data reconciliation after its offline

training. It neither requires online tuning nor explicit calculation of model predictions as required in the Kalman filter, and it does not require online dynamic optimization as in the conventional DDR algorithm.

In Section 2, the general concept of dynamic data reconciliation for process estimation is presented considering errors that inevitably arise in both process measurements and models. Then an AANN-based algorithm is developed to perform dynamic data reconciliation. The performance of the AANN-based algorithm is ten evaluated and compared to that of Kalman filter via simulation of the multivariable control of a binary distillation column in Section 3. Section 4 summarizes this paper and draws some conclusions.

2 AANN-BASED DDR ALGORITHM

2.1 Formulation of the dynamic data reconciliation problem

In view of the uncertainties in both the process model and process measurements, a compromise is achieved by simultaneously minimizing the weighted sum of squared errors of process measurements and model, namely,

$$\text{Minimize } J(\hat{\mathbf{x}}_t) = (\mathbf{y}_t - \hat{\mathbf{x}}_t)^T \mathbf{V}^{-1} (\mathbf{y}_t - \hat{\mathbf{x}}_t) + \mathbf{f}^T (\hat{\mathbf{x}}_t) \Omega^{-1} \mathbf{f} (\hat{\mathbf{x}}_t) \qquad (9)$$

where $f(\hat{\mathbf{x}}_i)$ is a functional vector of process models in discrete form, and Ω is the covariance matrix of model errors. The dynamic data reconciliation problem formulated by Equation (9) requires online optimization at each sampling time. For complex processes, a large computation time may be needed. To overcome this problem, this paper suggests using an autoassociative neural network trained with respect to Equation (9) to perform the data reconciliation.

2.2 Development of AANN-based DDR algorithm

An AANN is a network whose structure is similar to a conventional feedforward neural network. It normally consists of an input layer, three hidden layers and an output

layer. The input and the output layers have an identical number of neurons (excluding the bias in the input layer). The first and third hidden layers, named mapping and demapping layers respectively, contain a relatively larger number of neurons. The second hidden layer, referred to as the bottleneck layer, usually contains a lower number of neurons. The transfer functions for the mapping and demapping layers are nonlinear functions (e.g., sigmoid functions), whereas, the transfer function for the bottleneck and output layers can be linear or nonlinear. The key feature of AANN is its data compression/regeneration via the bottleneck layer. The input, mapping and bottleneck layers compress the input information to a lower dimension, and then the demapping and output layers recover the main underlying features of the original information. The compression/regeneration process enables the network to represent the input information in a compressed form so that it can often reveal the essence of the data. An AANN has to be trained in order to produce its desired outputs. In network training, the input and the target vectors presented to the network are identical, and the objective function consisting of the mean squared errors between the network outputs and its inputs is minimized. After successful training, an AANN can be regarded as a filter used to perform steady-state data reconciliation (Kramer, 1992; Du et al., 1997).

For dynamic data reconciliation, a similar architecture, shown in Figure 1, can be employed to take advantage of the noise reduction characteristic generated by the compression/regeneration of the AANN. In the dynamic case, the neural network is not fully autoasscociative since additional past inputs of the process, $\mathbf{u}_{t-d},\ldots,\mathbf{u}_{t-d-I}$, necessary to capture its dynamic behavior, are required such that the vector of inputs is not identical to the vector of outputs. However, the neural network of Figure 1 is autoassociative with respect to the output variables.

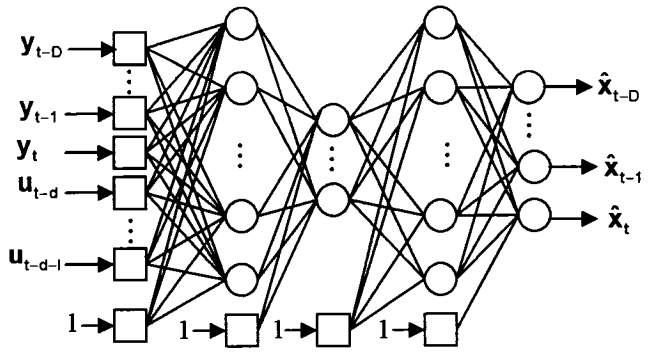


Figure 1 Architecture of feedforward AANN for a dynamic process.

Process models are encapsulated within the structure of the network to perform data reconciliation. The number of neurons in the input and output layers is determined by the structure of the process models. In Figure 1, D represents the order of the models, while d, ..., d+I represent the time delays of the input variables. However, the number of neurons in the hidden layers has to be determined by trial-and-error. The objective function to train the dynamic AANN to perform data reconciliation can be written as

$$\text{Minimize } J(\theta) = \frac{1}{N}\sum_{t=D}^{N}\Big[\left(\mathbf{y}_{t-D}-\hat{\mathbf{x}}_{t-D}\right)^T \mathbf{V}^{-1}\left(\mathbf{y}_{t-D}-\hat{\mathbf{x}}_{t-D}\right)+\ldots+\left(\mathbf{y}_t-\hat{\mathbf{x}}_t\right)^T \mathbf{V}^{-1}\left(\mathbf{y}_t-\hat{\mathbf{x}}_t\right)+ \mathbf{f}^T\left(\hat{\mathbf{x}}_t,\ldots,\hat{\mathbf{x}}_{t-D},\mathbf{u}_{t-d},\ldots,\mathbf{u}_{t-d-I}\right)\boldsymbol{\Omega}^{-1}\mathbf{f}\left(\hat{\mathbf{x}}_t,\ldots,\hat{\mathbf{x}}_{t-D},\mathbf{u}_{t-d},\ldots,\mathbf{u}_{t-d-I}\right)\Big] \tag{10}$$

where θ is the vector composed of all connection weights (parameters) between the neurons in the network, and N is the total number of data points in the training horizon. Among the training algorithms, backpropagation is the simplest method. However, more advanced training algorithms such as quasi-Newton and Levenberg-Marquardt methods are most often used for faster convergence. Following the training, the performance of the network is validated using a fresh data set. Then the AANN can be applied online to directly perform data reconciliation.

In the AANN-based DDR algorithm formulated by Equation (10), at each sampling time, t, the raw measurements within the time window [t-D, t] are reconciled. However, only the reconciled values for the current time $\hat{\mathbf{x}}_t$ are retained for process monitoring and/or

control purposes. Other reconciled values for past measurements are discarded. As a result, the dimensionality of the data reconciliation problem becomes large. To overcome this problem, the architecture of the dynamic AANN is modified as presented in Figure 2. The output of the network, $\hat{\mathbf{x}}_t$, delayed a number of times, is fed back to the input layer such that both process temporal and spatial patterns are incorporated in the structure of the recurrent AANN. For this case, the neural network is autoassociative only for the current output variables, and obviously only the current measurements are reconciled at each sampling time.

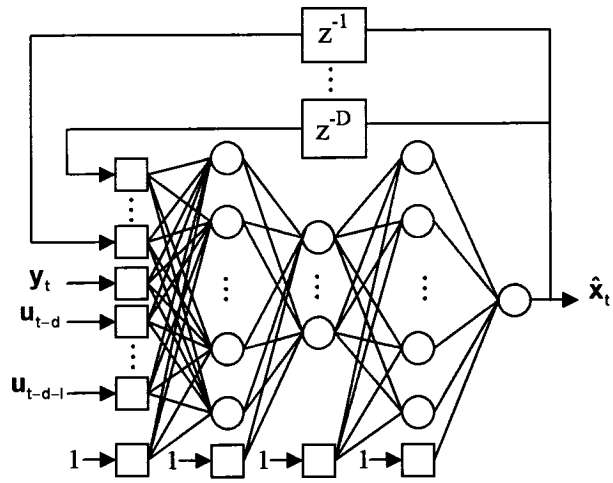


Figure 2 Architecture of recurrent AANN for a dynamic process (z^{-1} denotes the backshift operator such that $z^{-1}\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1}$).

By means of the recurrent architecture of the AANN, the training objective function for data reconciliation becomes

$$\text{Minimize } J(\theta) = \frac{1}{N}\sum_{t=D}^{N}\Big[\left(\mathbf{y}_t - \hat{\mathbf{x}}_t\right)^T \mathbf{V}^{-1}\left(\mathbf{y}_t - \hat{\mathbf{x}}_t\right) + \mathbf{f}^T\left(\hat{\mathbf{x}}_t,\ldots,\hat{\mathbf{x}}_{t-D},\mathbf{u}_{t-d},\ldots,\mathbf{u}_{t-d-l}\right)\boldsymbol{\Omega}^{-1}\mathbf{f}\left(\hat{\mathbf{x}}_t,\ldots,\hat{\mathbf{x}}_{t-D},\mathbf{u}_{t-d},\ldots,\mathbf{u}_{t-d-l}\right)\Big] \tag{11}$$

In the network training, it is important to note that the output vector, $\hat{\mathbf{x}}_t$, fed back to the input layer is not known initially, i.e., the values of $\hat{\mathbf{x}}_{t-D},\ldots,\hat{\mathbf{x}}_{t-1}$ as inputs to the network are unknown. Accordingly, an iterative training methodology is developed to manage this

problem. To begin the training, the values for $\hat{\mathbf{x}}_{t-D},\ldots,\hat{\mathbf{x}}_{t-1}$ are assigned the values of their associated raw measurements, $\mathbf{y}_{t-D},\ldots,\mathbf{y}_{t-1}$. Then, the network is trained with respect to Equation (11) until satisfactory convergence is met. After the first iteration, the network output, $\hat{\mathbf{x}}_t$, is delayed a number of times and fed back as inputs $\hat{\mathbf{x}}_{t-D},\ldots,\hat{\mathbf{x}}_{t-1}$, and then the network is trained again. After several iterations, the feedback vectors and the objective function will not change, indicating the training of the neural network has been completed. Comparing the two training objective functions for the feedforward and recurrent AANNs, it appears that in the feedforward AANN, past and current measurements are reconciled, providing more degrees of freedom in minimizing the objective function such that process models impose less restriction on the raw measurements. On the contrary, in the recurrent AANN, only the current measurements are reconciled so that the raw measurements can be constrained more tightly by the process models. The recurrent structure of the AANN presented in Figure 2 is expected to be more effective for dynamic data reconciliation than the feedforward structure presented in Figure 1. This statement is verified in Section 3.1.

The covariance matrix of the model residuals, $\boldsymbol{\Omega}$, in the training objective function is often difficult to determine. Consequently, $\boldsymbol{\Omega}$ is considered diagonal and the elements in $\boldsymbol{\Omega}$ are treated as tuning parameters in network training. If elements of $\boldsymbol{\Omega}$ are given larger numbers relative to the variances of the raw measurements, meaning that more confidence is put on the raw measurements, the outputs of the network (i.e., the reconciled values) will be close to the raw measurements. On the other hand, if the elements of $\boldsymbol{\Omega}$ are given relatively smaller values, then more confidence is given to the models, and, as a result, model mismatch may bias the reconciled values. The matrix $\boldsymbol{\Omega}$ provides a measure for offline tuning for the AANN to perform dynamic data reconciliation. The AANN-based DDR provides the advantage that it does not need online tuning as required in the Kalman filter. Furthermore, the use of AANN-based DDR is straightforward without explicitly calculating model predictions or performing online dynamic optimization, because information about process dynamics has been encoded into the neural network during its offline training.

3 AN ILLUSTRATIVE EXAMPLE

The methodologies of the AANN-based DDR algorithm developed in this work were evaluated through simulation of multivariable control of a binary (benzene/toluene) distillation column. The distillation column having four PI control loops is presented in Figure 3. Controllers TIC-D and TIC-B are used to control the top and bottom temperatures by manipulating the reflux flow rate and the flow of steam to the reboiler, respectively. Controllers LIC-D and LIC-B are used to control the reflux drum and column base liquid levels by manipulating the distillate flow rate and the bottom product flow rate, respectively. A sampling time of 30 s was used for all the measurements. White Gaussian noise with standard deviation, σ, presented in Table 1, was added to the true values of the four controlled variables to simulate their measured values. The dynamic distillation simulator, based on rigorous distillation models, (i.e., mass and heat balances, vapor-liquid equilibrium, and tray hydraulics) was developed by Bai (2003).

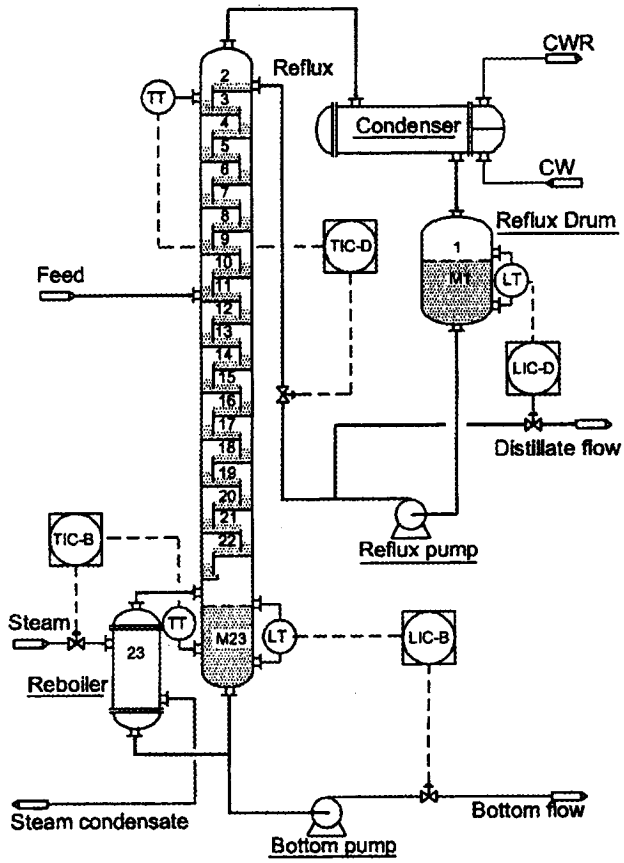


Figure 3 Control scheme of a binary distillation column.

Table 1 Nominal noise levels of controlled variables for the distillation column

Measured variables	Units	Standard deviation, σ
Reflux drum level, H_D	m	0.02
Top temperature, T_D	°C	0.25
Column base level, H_B	m	0.02
Bottom temperature, T_B	°C	0.25

3.1 Offline training AANNs

3.1.1 Dynamic models

Because of the complexity of the distillation process, it is impractical to develop phenomenological models for use in data reconciliation. Consequently, input-output empirical models were identified for the four controlled variables. It was found that the dynamic behavior of the two liquid levels can be adequately described by linear models, whereas nonlinear models were required for the two temperatures. The nonlinearity of the two temperatures lies in the fact that the top temperature cannot decrease below the boiling point of pure benzene, and the bottom temperature cannot exceed the boiling point of pure toluene under the column pressures. The linear models for the two liquid levels were given in terms of deviation variables (denoted by primes) by

$$H'_{D,t} = H'_{D,t-1} - 9.058\times10^{-4}D'_{t-1} - 9.058\times10^{-4}R'_{t-1} + 2.112\times10^{-5}Q'_{t-1} + 1.06\times10^{-5}Q'_{t-2} \quad (12)$$

$$H'_{B,t} = H'_{B,t-1} + 1.023\times10^{-3}R'_{t-2} - 1.152\times10^{-3}B'_{t-1} - 2.056\times10^{-5}Q'_{t-1} - 1.028\times10^{-5}Q'_{t-2} \quad (13)$$

where $H_{D,t}$ is the reflux drum level, $H_{B,t}$ is the column base level, D is the distillate flow rate, R is the reflux flow rate, B is the bottom flow rate, and Q is the reboiler heat duty. A feedforward neural network model, illustrated in Figure 4(a), was developed for the top temperature, $T_{D,t}$. This neural network model can be concisely expressed as

$$T_{D,t} = f_{NN}(T_{D,t-1}, R_{t-1}, R_{t-2}, R_{t-3}, Q_{t-1}, Q_{t-2}, Q_{t-3}) \quad (14)$$

where f_{NN} represents the neural model used to predict the column top temperature at time t. A similar feedforward neural network, illustrated in Figure 4(b), was developed for the column bottom temperature, $T_{B,t}$. It can be expressed as

$$T_{B,t} = f_{NN}(T_{B,t-1}, R_{t-6}, R_{t-7}, Q_{t-1}, Q_{t-2}) \quad (15)$$

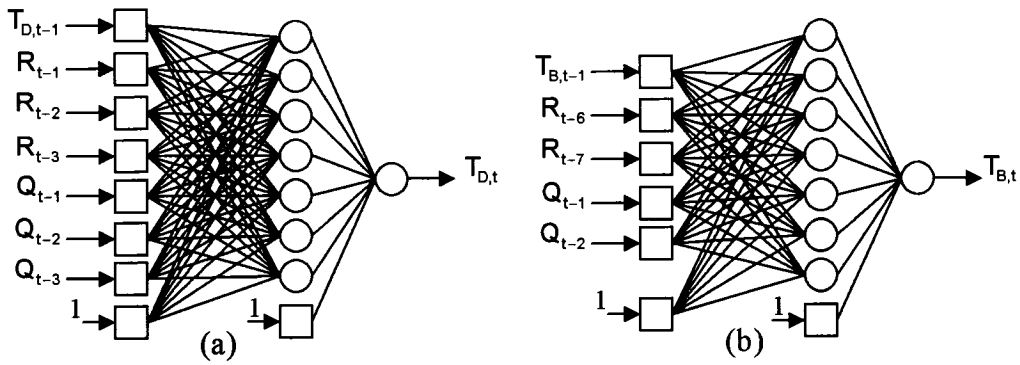


Figure 4 Feedforward neural network models for the top (a) and bottom (b) temperatures in the distillation column.

3.1.2 Structures of AANNs

For the distillation column having multiple inputs and multiple outputs, it is possible to develop an AANN that contains all the variables collectively to perform dynamic data reconciliation. However, this approach results in an extremely large network that is difficult to handle. Consequently, individual AANN was developed for each controlled variable. The use of the recurrent AANN was considered. The architectures of the AANNs for the reflux drum level and top temperature are presented in Figure 5, where the superscript "m" for the controlled variables in the inputs of the networks denotes the raw measurements and "r" in outputs of the networks denotes the reconciled values. Akin to the reflux drum level and top temperature, the AANNs for the column base level and bottom temperature have similar architectures.

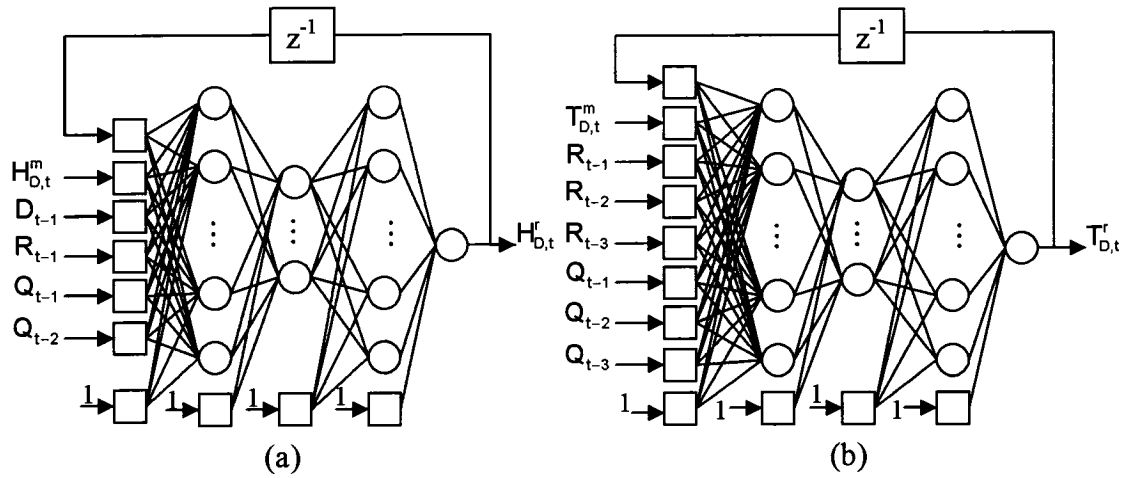


Figure 5 Architectures of recurrent AANNs for dynamic data reconciliation for (a) the reflux drum level and (b) top temperature.

The cross correlations of the model errors for the four controlled variables were found to be negligible. Therefore, the four objective functions corresponding to the four controlled variables used to train the four AANNs are given by the following equations:

$$J(\theta)=\frac{1}{N}\sum_{t=2}^{N}\left\{\frac{1}{\sigma^2_{H_D}}\left(H^m_{D,t}-H^r_{D,t}\right)^2+\frac{1}{\sigma^2_{H_D,R}}\left[H^r_{D,t}-H^r_{D,t-1}+9.058\times10^{-4}(D_{t-1}-D_0)\right.\right.$$
$$\left.\left.+9.058\times10^{-4}\left(R_{t-1}-R_0\right)-2.112\times10^{-5}\left(Q_{t-1}-Q_0\right)-1.06\times10^{-5}\left(Q_{t-2}-Q_0\right)\right]^2\right\} \quad (16)$$

$$J(\theta)=\frac{1}{N}\sum_{t=3}^{N}\left\{\frac{1}{\sigma^2_{T_D}}\left(T^m_{D,t}-T^r_{D,t}\right)^2+\frac{1}{\sigma^2_{T_D,R}}\right.$$
$$\left.\left[T^r_{D,t}-f_{NN}\left(T^r_{D,t-1},R_{t-1},R_{t-2},R_{t-3},Q_{t-1},Q_{t-2},Q_{t-3}\right)\right]^2\right\} \quad (17)$$

$$J(\theta)=\frac{1}{N}\sum_{t=2}^{N}\left\{\frac{1}{\sigma^2_{H_B}}\left(H^m_{B,t}-H^r_{B,t}\right)^2+\frac{1}{\sigma^2_{H_B,R}}\left[H^r_{B,t}-H^r_{B,t-1}-1.023\times10^{-3}(R_{t-2}-R_0)\right.\right.$$
$$\left.\left.+1.152\times10^{-3}\left(B_{t-1}-B_0\right)+2.056\times10^{-5}\left(Q_{t-1}-Q_0\right)+1.028\times10^{-5}\left(Q_{t-2}-Q_0\right)\right]^2\right\} \quad (18)$$

$$J(\theta)=\frac{1}{N}\sum_{t=7}^{N}\left\{\frac{1}{\sigma^2_{T_B}}\left(T^m_{B,t}-T^r_{B,t}\right)^2+\frac{1}{\sigma^2_{T_B,R}}\left[T^r_{B,t}-f_{NN}\left(T^r_{B,t-1},R_{t-6},R_{t-7},Q_{t-1},Q_{t-2}\right)\right]^2\right\} \quad (19)$$

where D_0, R_0, B_0, and Q_0 are steady-state values for the distillate flow rate, reflux flow rate, bottom flow rate and reboiler heat duty, respectively. $\sigma^2_{H_D}$, $\sigma^2_{T_D}$, $\sigma^2_{H_B}$ and $\sigma^2_{T_B}$ are the associated variances of the raw measurements. $\sigma^2_{H_D,R}$, $\sigma^2_{T_D,R}$, $\sigma^2_{H_B,R}$ and $\sigma^2_{T_B,R}$ are the associated variances of the model errors.

3.1.3 Data sets

Under open loop operation, the distillation column was excited by a series of random step changes in the four manipulated variables. For the distillate flow rate and bottom flow rate, the random step changes between two consecutive sampling periods had a switching probability of 0.2, while for the reflux flow rate and reboiler heat duty, a switching probability of 0.1 was used. Further, to effectively capture the nonlinearity of the process, the amplitudes of the random step changes had a uniform distribution around the nominal steady state. 1450 data points for the responses of the four controlled variables along with the variations of the four manipulated variables were obtained. 80% of the data points were used for network training and the remaining 20% were used for network validation.

3.1.4 Network training

From Figure 5 and Equations (16-19), it can be seen that, for each AANN, the number of neurons in the input layer was determined by the structure of the dynamic model encapsulated in the training objective function. The number of neurons in the three hidden layers for each AANN was determined by trail-and-error. The same number of neurons was assigned to the mapping and demapping layers. The training of the neural networks was carried out using a quasi-Newton method.

To assess the performance of each AANN, the mean squared error (MSE) between the reconciled values and the true values from the validation data set was used as the objective criterion. The MSE is defined as

$$\mathrm{MSE} = \frac{1}{K}\sum_{t=0}^{K}(\hat{x}_t - x_t)^2 \qquad (20)$$

where K is the total number of data points in the validation data set. It is very important to note that the calculation of the MSE criterion is only achievable in the simulated case study because the true values of the controlled variables are exactly known in the simulation, so that it is possible to compare the various algorithms and assess their performance. However, for implementation in a real plant, one could not assess the reconciliation performance using the criterion of Equation (20), but it does not impede the use of the AANN-based DDR since the training objective functions (see Equations (16-19)) do not use these true values. In practice, location and dispersion statistics of residuals between the reconciled and the measured values could be used in hypothesis tests as criteria to evaluate the reconciliation performance. Being white noise with mean zero and variance close to that of the raw measurements for the residuals would be a sign of good performance of the AANN-based DDR algorithm.

A group of AANNs was selected in the trial-and-error examination for each controlled variable. For the column base level and bottom temperature, the AANNs are presented in Table 2. The number of neurons in the bottleneck layer was chosen from 2 to 4, while the number of neurons in the mapping and demapping layers was chosen from 3 to 8. The number of neurons in each layer included a bias neuron, except for the output layer. For each AANN, it was completely trained and validated twenty times, each time with the network having different initial weights. The MSE value standardized by the associated variance of the raw measurements, MSE/σ^2, was calculated for each trial. The mean of the MSE/σ^2 values and its standard error are also presented in Table 2. For each group of AANNs having the same number of neurons in the bottleneck layer, the mean of the MSE/σ^2 values decreased, and then increased slightly, with the increased number of neurons in the mapping and demapping layers. With smaller numbers of neurons in the mapping and demapping layers, the structures of the AANNs were insufficient for data reconciliation and consequently resulted in larger MSE/σ^2 values. On the other hand, having larger numbers of neurons in the mapping and demapping layers, the AANNs were over parameterized and consequently the noise attenuation performed by the AANNs deteriorated. Similar results were obtained for the reflux drum level and top temperature. The AANNs having the structures as {7, 8, 3, 8, 1}, {9, 8, 2, 8, 1}, {7, 6, 3,

6, 1} and {7, 6, 4, 6, 1} for the reflux drum level, top temperature, column base level and bottom temperature, respectively, were selected for online implementation.

Table 2 Performance of AANNs for the column base level and bottom temperature (results are based on 20 training/validation trials)

Number of neurons in each layer	Total number of neurons	Total number of weights	$\overline{MSE/\sigma^2}$ ±standard error	
			H_B	T_B
{7, 3, 2, 3, 1}	16	24	0.921±0.046	0.284±0.005
{7, 4, 2, 4, 1}	18	35	0.802±0.043	0.275±0.005
{7, 5, 2, 5, 1}	20	46	0.779±0.039	0.269±0.002
{7, 6, 2, 6, 1}	22	57	0.630±0.026	0.273±0.007
{7, 7, 2, 7, 1}	24	68	0.653±0.017	0.276±0.003
{7, 8, 2, 8, 1}	26	79	0.657±0.023	0.281±0.006
{7, 3, 3, 3, 1}	17	29	0.896±0.030	0.283±0.005
{7, 4, 3, 4, 1}	19	42	0.741±0.056	0.275±0.005
{7, 5, 3, 5, 1}	21	55	0.622±0.040	0.269±0.001
{7, 6, 3, 6, 1}	23	68	0.610±0.053	0.270±0.001
{7, 7, 3, 7, 1}	25	81	0.674±0.043	0.271±0.001
{7, 8, 3, 8, 1}	27	94	0.671±0.031	0.272±0.001
{7, 3, 4, 3, 1}	18	34	0.882±0.032	0.286±0.007
{7, 4, 4, 4, 1}	20	49	0.709±0.051	0.275±0.005
{7, 5, 4, 5, 1}	22	64	0.672±0.048	0.272±0.002
{7, 6, 4, 6, 1}	24	79	0.612±0.031	0.269±0.001
{7, 7, 4, 7, 1}	26	94	0.647±0.041	0.270±0.001
{7, 8, 4, 8, 1}	28	109	0.662±0.027	0.271±0.001

As mentioned in Section 2.2, the variance of model errors provides a measure for offline tuning an AANN to perform dynamic data reconciliation. Next, the performance of the recurrent AANN selected for each controlled variable as a function of the variance of model error in the training objective function was investigated. Given a value for the

variance of the model error, the AANN was trained and then validated. The standardized MSE values as a function of the tuning parameter are plotted in Figure 6. The minimum MSE/σ^2 values for the four controlled variables were found at $\sigma^2_{H_D,R} = 2.5\times10^{-3}$, $\sigma^2_{T_D,R} = 0.36$, $\sigma^2_{H_B,R} = 1.8\times10^{-4}$ and $\sigma^2_{T_B,R} = 0.22$, respectively, which represent the optimal tuning parameters for the AANN-based DDR algorithm. Increasing the variance of the model error means that more confidence level was put on the raw measurements. On the other hand, decreasing the variance of the model error means that more confidence was put on the process model. As shown in Figure 6, if the variance of the model error in the training objective function assumed a different value of its optimal value, a larger MSE/σ^2 value resulted.

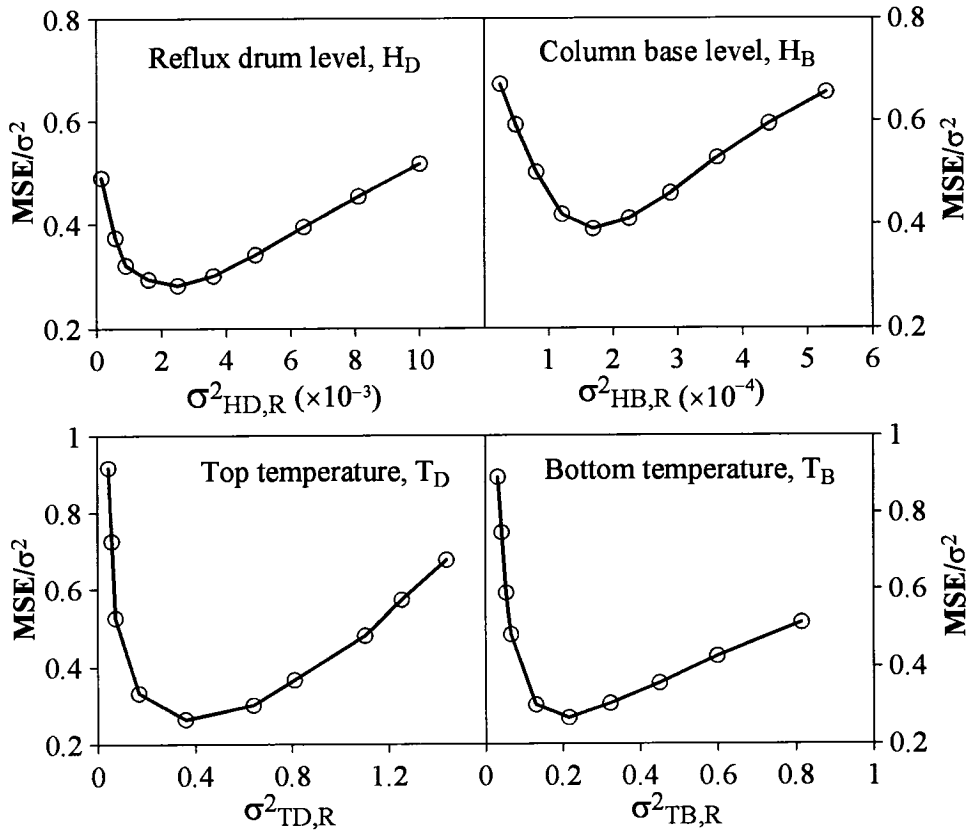


Figure 6 Values of MSE/σ^2 as a function of the variances of model errors in the training objective functions of recurrent AANNs.

Using the optimal variances of model errors to train the AANNs, representative samples of raw, reconciled and true values for the four controlled variables in the network training

and validation are presented in Figure 7. This figure shows that the reconciled values were very close to the true values. The noise contained in the measurements was significantly reduced after being reconciled by the neural network. Using the representative data shown in Figure 7, the residuals between the reconciled and raw measurements in the validation data set were calculated for each variable. The mean values of the residuals evaluated for the reflux drum level, top temperature, base level and bottom temperature were -0.002 m, 0.088 °C, -0.004 m and -0.032 °C, respectively. These results corroborate the results shown in Figure 7, indicating that the reconciled values are unbiased based on the analysis of the residuals.

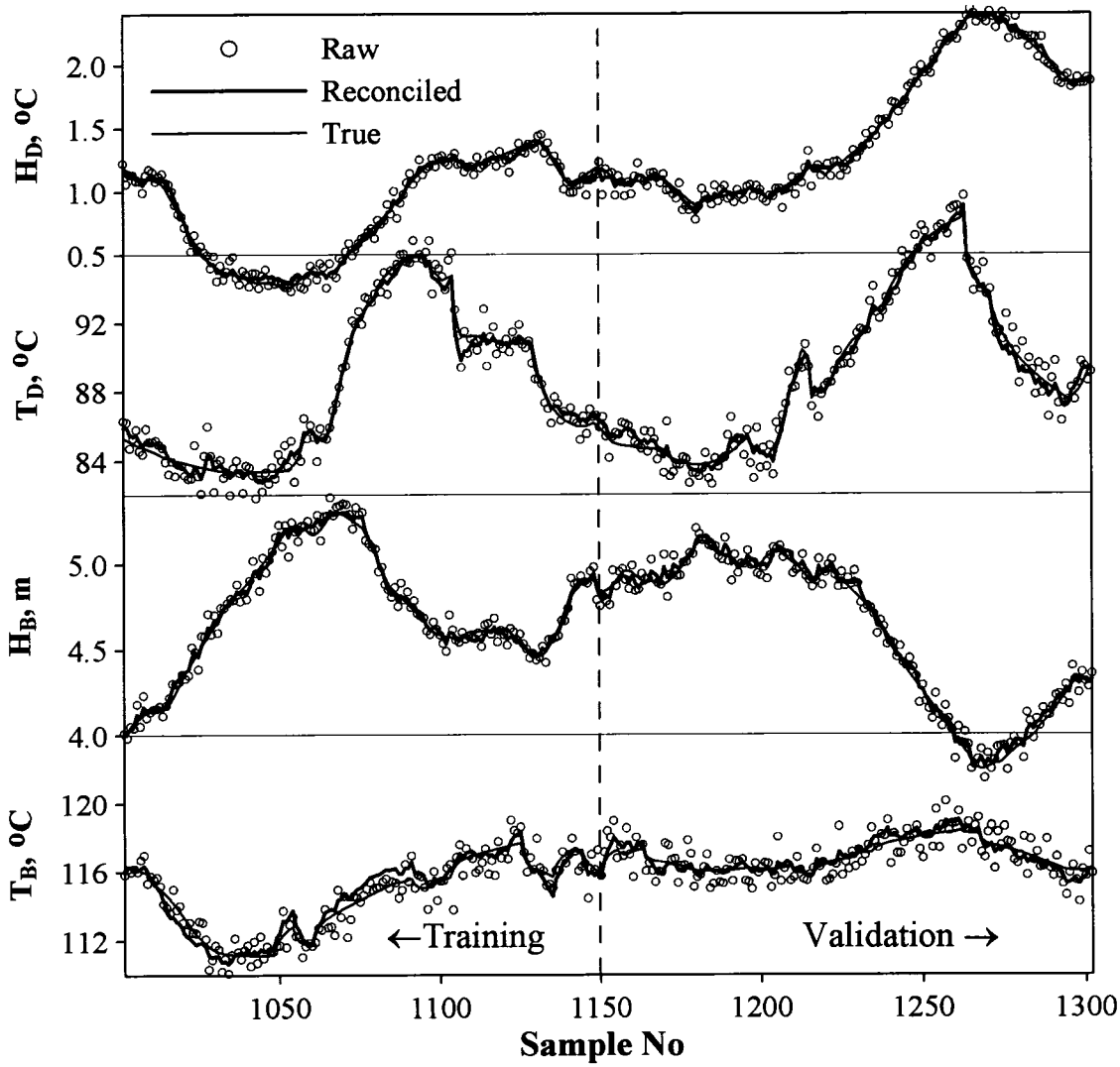


Figure 7 Representative samples of raw, reconciled and true values for the four controlled variables in network training and validation.

3.1.5 Comparison of recurrent AANN to feedforward AANN

As stated in Section 2.2, the AANNs having recurrent structures are expected to have better performance than those having feedforward structures. This hypothesis was verified by examining the values of MSE/σ^2 obtained using each structure. For each controlled variable, a feedfoward AANN was developed. It had the same number of neurons in each layer as in the recurrent AANN, except that it was necessary that the network output layer had two neurons for the reconciled values at time t and t-1 (see Figure 2 for the structure of a feedforward AANN). The feedforward AANN as well as the recurrent AANN were trained and validated twenty times, each time with different initial weights. The average values of MSE/σ^2 and their standard errors obtained by the AANNs are presented in Table 3. On average, the values of MSE/σ^2 achieved by the feedforward AANNs were 1.5-2 times larger than those achieved by the recurrent AANNs. The recurrent AANNs performed significantly better than the feedforward AANNs for dynamic data reconciliation.

Table 3 Comparison of performance of recurrent and feedforward AANNs (Results are based on 20 training/validation trials)

AANN	$\overline{MSE/\sigma^2}$ ± standard error			
	H_D	T_D	H_B	T_B
Feedforward	0.44±0.01	0.49±0.01	0.61±0.01	0.56±0.01
Recurrent	0.27±0.02	0.25±0.01	0.40±0.05	0.27±0.01

3.2 Online implementations of AANNs

Following the offline training, the recurrent AANNs were implanted inside PI feedback control loops in the distillation column as illustrated in Figure 8, where the raw measurements were first reconciled using the AANNs, and then the outputs of the AANNs were used by the controllers to calculate control moves.

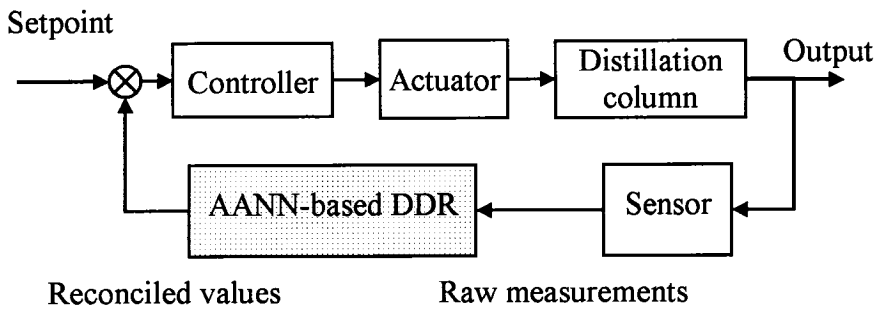


Figure 8 Feedback control scheme using AANN-based DDR.

The closed-loop performance of the AANN-based DDR for each controlled variable was first evaluated for the nominal noise levels listed in Table 1. In order to obtain significant dynamic change in the process, the setpoint of the bottom temperature was lowered by 2.5 °C at 30 min and returned to its nominal setpoint at 210 min, whereas the setpoint of the top temperature was increased by 3.0 °C at 120 min and returned to its nominal value at 300 min. The raw measurements, reconciled and true values for the four controlled variables along with the control moves are presented in Figure 9. A setpoint change in either of the two temperatures had an impact on the other three controlled variables due to inherent interactions among the four control loops. Despite the complex dynamics, the AANN-based DDR performed very well in tracking the true values. The reconciled data were significantly less noisy than the raw measurements. Moreover, the high-frequency oscillations of the control moves were also reduced.

To quantitatively assess the performance of AANN-based DDR, the MSE/σ^2 value for each controlled variable was evaluated for 20 simulation runs, each run with a different white noise sequence. The mean of the MSE/σ^2 values and its standard error for each controlled variable are presented in the first row of Table 4. Compared to the variances of the raw measurements, the MSE values for the reconciled data were much smaller, indicating that noise reduction for the four controlled variables was significant.

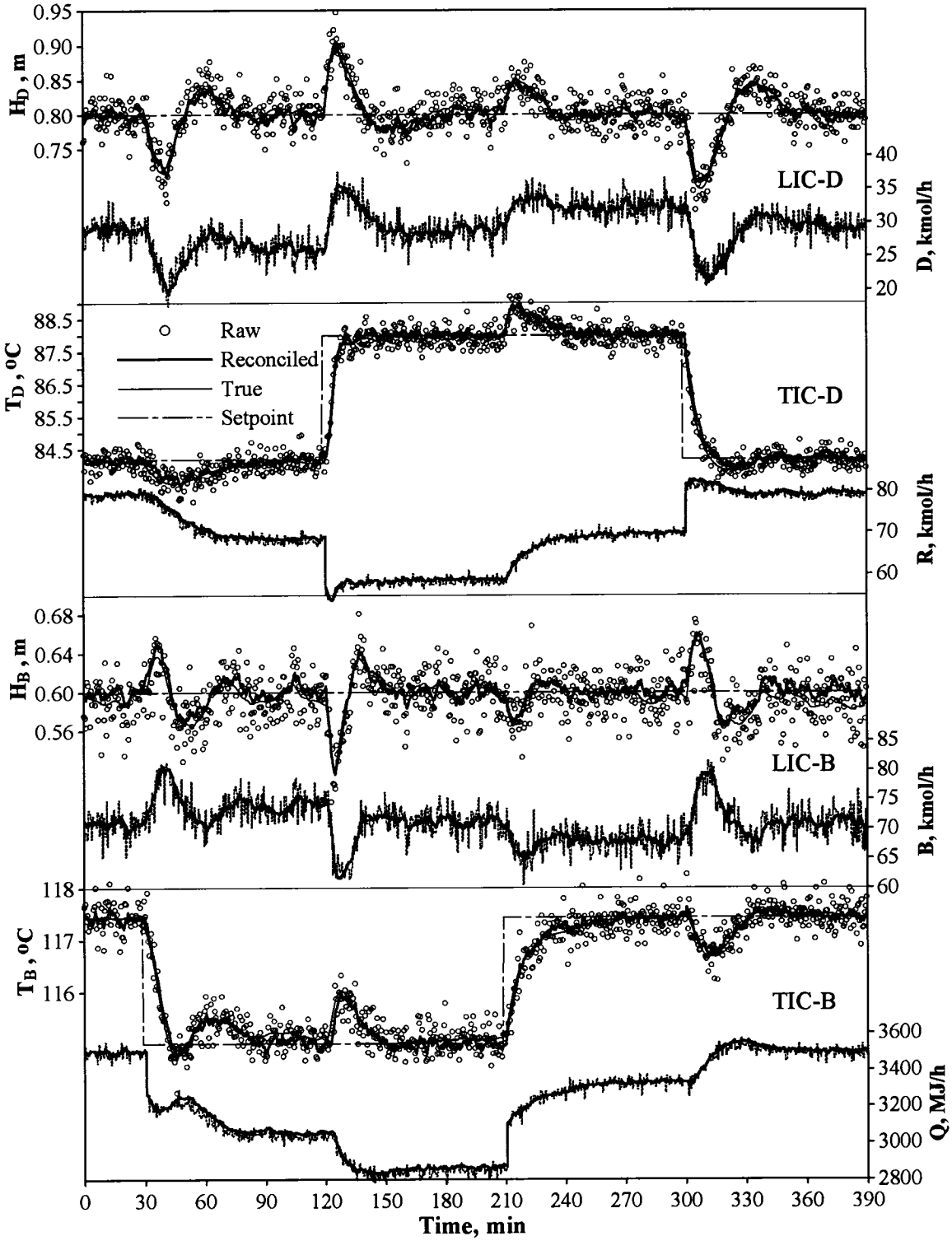


Figure 9 Closed-loop performance of AANN-based DDR for measurements corrupted by the nominal noise level. The thinner dashed lines for the manipulated variables represent control moves without the AANN-based DDR.

Table 4 MSE/σ^2 values of reconciled data for the four controlled variables achieved by the AANN-based DDR (Results are based on 20 white noise sequences)

Noise level	$\overline{MSE_i/\sigma_i^2} \pm$ standard error			
	H_D	T_D	H_B	T_B
Nominal, $\sigma_1=\sigma$	0.169±0.006	0.245±0.006	0.182±0.006	0.163±0.004
Doubled, $\sigma_2=2\sigma$	0.121±0.003	0.152±0.004	0.110±0.004	0.128±0.003
Tripled, $\sigma_3=3\sigma$	0.112±0.003	0.134±0.004	0.097±0.003	0.121±0.003

Without re-tuning the AANNs, the performance of the identical AANNs was evaluated when the noise level of the measurements was twice and thrice the nominal noise level. For the identical sequences of top and bottom temperature setpoint changes, the process was simulated 20 times for each noise level, each time with a different white noise sequence. The mean of the MSE/σ^2 values and its standard error for each controlled variable for the two noise level cases are presented in the second and third rows of Table 4, respectively. On average, the MSE/σ^2 values decreased despite the increase of the noise level, indicating the performance of the AANN-based DDR for noise reduction became more effective, thereby demonstrating robustness to the change in the magnitude of the measurement noise. For thrice the nominal noise level, the raw measurements, reconciled and true values for the four controlled variables along with the control moves are presented in Figure 10. The dynamic responses of the four controlled variables without data reconciliation were considerably masked by the measurement noise. The noise-to-signal ratios of the two liquid levels were very high such that it was difficult to observe any dynamic trend. However, with the AANN-based DDR, the dynamic responses were clearly observed and the high magnitudes of oscillations of the control moves were significantly reduced. The benefits of using the AANN-based DDR inside the control loops became more obvious.

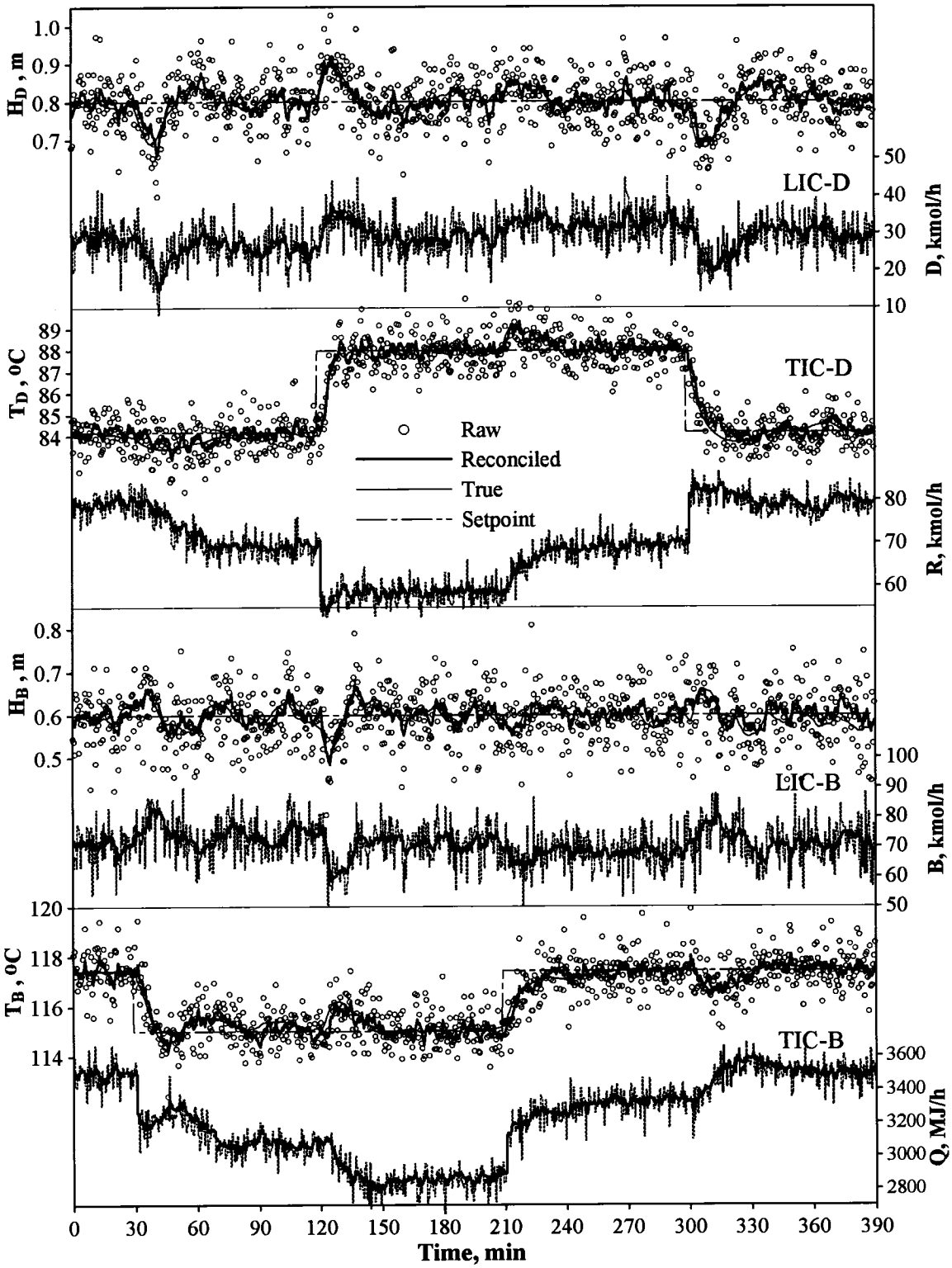


Figure 10 Closed-loop performance of AANN-based DDR for measurements corrupted by thrice the nominal noise level. The thinner dashed lines for the manipulated variables represent control moves without the AANN-based DDR.

3.3 Comparison to Kalman filter

The performance of the AANN-based DDR to attenuate the impact of measurement noise embedded inside the control loops was also compared to that of an extended Kalman filter (EKF). To use the Kalman filter, process models, Equations (12-15), were directly employed. Since all the four controlled variables were directly measured, the measurement model was

$$\mathbf{y}_t = \mathbf{x}_t + \boldsymbol{\varepsilon}_t \tag{21}$$

Because the covariance matrix of the process model noise, $\mathbf{S}$, required by the Kalman filter, was difficult to determine, $\mathbf{S}$ was assumed diagonal with elements treated as tuning parameters in calculating the Kalman gain, $\mathbf{K}_t$. In addition, the two neural models for the top and bottom temperatures were linearized at each sampling time in order to calculate the gain of the Kalman filter.

The performance of the Kalman filter was evaluated when the distillation column was subjected to the same sequence of temperature setpoint changes and for the three noise levels that were used to assess the AANN-based DDR. The MSE values standardized by their respective variances of raw measurements, MSE/σ^2, for each controlled variable as a function of the tuning parameters are plotted in Figure 11. It can be observed that, using the Kalman filter for each noise level, the optimal tuning parameters were obtained for minimum MSE/σ^2 values. Increasing or decreasing the tuning parameters away from their optimal values, the values of MSE/σ^2 increased rapidly. When the noise level of the measurements changed, the Kalman filter was re-tuned to regain its optimality, otherwise sub-optimal performance resulted.

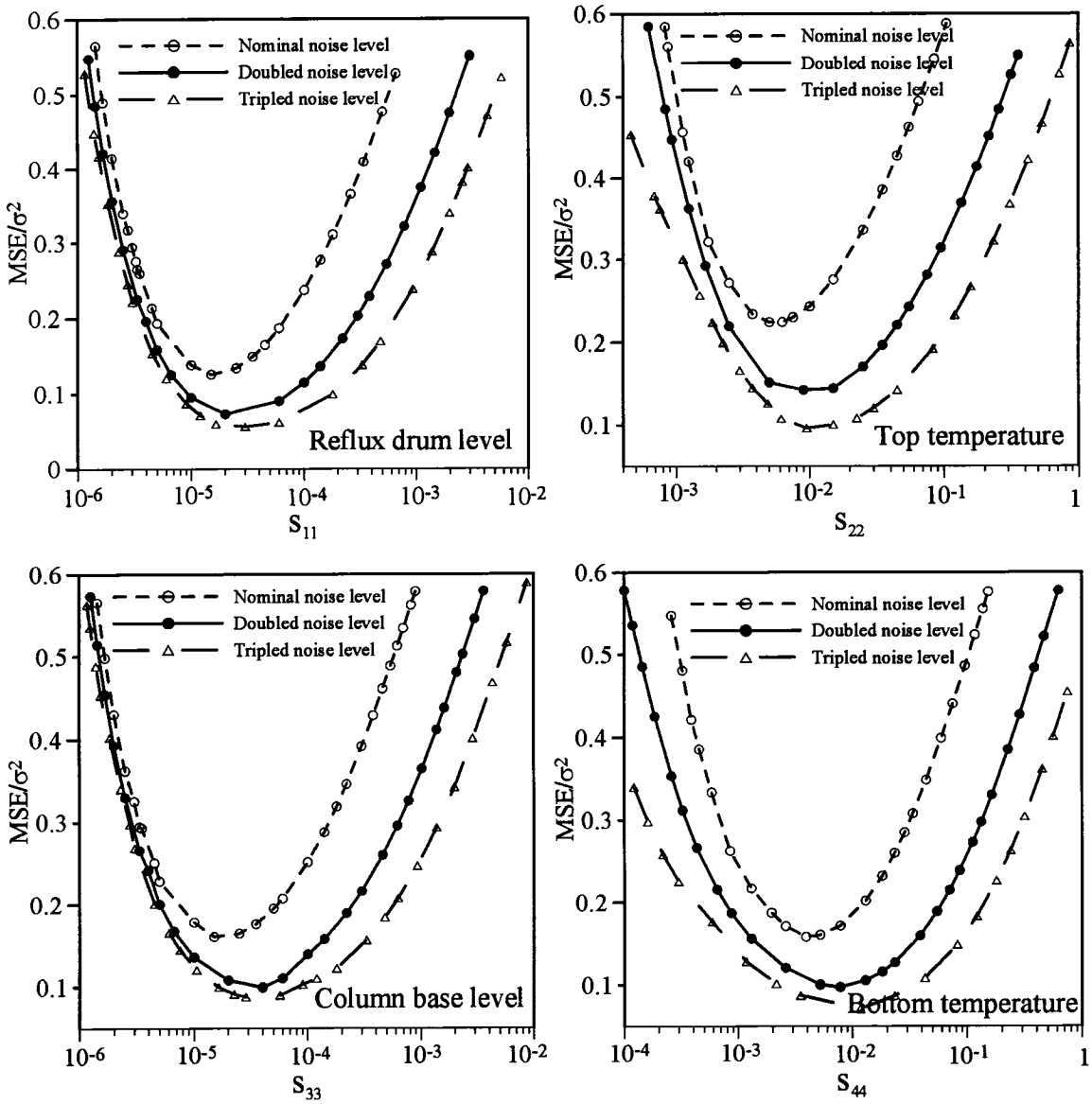


Figure 11 Performance of Kalman filter for the four controlled variables for nominal, twice and thrice the nominal noise level as a function of variances of process model noise.

The performance of the extended Kalman filter using its optimal tuning parameters was compared to that of the AANN-based DDR when the measurements were corrupted by the three noise levels, respectively. Results of the mean of the MSE/σ^2 values and its standard error obtained from twenty white noise sequences for each controlled variable and for each noise level are presented in Table 5. Compared to the values shown in Table 4, the extended Kalman filter performed better than the AANN-based DDR in the vicinity

of the optimal regions. The better performance of the extended Kalman filter is attributed to the fact that process models directly participated in the data reconciliation, whereas, process models indirectly participated in the AANN-based DDR. Information about the process dynamics contained in the models was partly lost during offline training of the AANNs.

Table 5 MSE/σ^2 values of reconciled data for the four controlled variables achieved by the extended Kalman filter (Results are based on 20 white noise sequences)

Noise level	$\overline{MSE_i/\sigma_i^2}$ ± standard error			
	H_D	T_D	H_B	T_B
Nominal, $\sigma_1=\sigma$	0.161±0.005	0.188±0.005	0.151±0.005	0.153±0.004
Doubled, $\sigma_2=2\sigma$	0.111±0.004	0.113±0.004	0.104±0.004	0.091±0.003
Tripled, $\sigma_3=3\sigma$	0.086±0.004	0.085±0.003	0.083±0.004	0.067±0.002

4 CONCLUSION

This paper developed an AANN-based dynamic data reconciliation algorithm, and its performance for process estimation was quantitatively assessed when it was embedded inside feedback control loops. Simulation results from the distillation column demonstrated the AANN-based DDR can efficiently attenuate the impact of measurement noise and consequently resulted in improved process monitoring and/or control. Two AANN structures for dynamic data reconciliation, namely a feedforward AANN and a recurrent AANN, were proposed. The performance of the recurrent AANN proved more effective than that of the feedforward AANN. Models with various structures, e.g., linear or nonlinear, can be encapsulated into the training objective function of an AANN to perform dynamic data reconciliation. The covariance matrix of the model residuals provides a measure for offline tuning for the AANN-based DDR. After offline successful training, the AANN can be directly implemented online to perform DDR, without explicitly calculating model predictions as required in the Kalman filter, and without

resorting to an optimization algorithm as required in the conventional DDR algorithm. Both the AANN-based DDR and the Kalman filter are robust to the changes of noise level in plant measurements. The Kalman filter has better performance than the AANN-based DDR at the vicinity of its optimal region. However, with adequate offline training, the AANN-based DDR is easier to implement online.

NOMENCLATURE

$\mathbf{A}_t$: Jacobian matrix obtained by linearizing process models

B : Bottom flow rate

$\mathbf{C}_t$: Jacobian matrix obtained by linearizing measurement models

D : Distillate flow rate

f_{NN} : Neural model

$H_{D,t}$: Reflux drum level at time t

$H_{B,t}$: Column base level at time t

$\mathbf{I}$: Identity matrix

$\mathbf{K}_t$: Matrix of Kalman filter gain

N : Total number of data points used to train neural networks

$\mathbf{P}_t^-$: Covariance matrix of a priori estimation error

$\mathbf{P}_t$: Covariance matrix of a posteriori estimation error

Q : Reboiler heat duty

R : Reflux flow rate

$\mathbf{S}$: Covariance matrix of process model noise

$T_{B,t}$: Column bottom temperature

$T_{D,t}$: Column top temperature

$\mathbf{u}_t$: Vector of process input variables at time t

$\mathbf{V}$: Covariance matrix of the measurement noise

$\mathbf{w}_t$: Vector of random variables denoting white Gaussian process model noise

$\mathbf{x}_t$: Vector of the true values of process variables at time t

$\hat{\mathbf{x}}_t$: Vector of estimates of true values of process variables

$\mathbf{y}_t$: Vector of measured values of process variables

$\hat{\mathbf{y}}_t$: Vector of model predicted values

Creek letters

ε_t: Vector of random variables representing white Gaussian measurement noise

θ: Vector of connection weights for neural networks

Ω: Covariance matrix of model error

σ: Standard deviation of the raw measurements

Acronyms

AANN: Autoassociative neural network

DDR: Dynamic data reconciliation

EKF: Extended Kalman filter

MSE: Mean squared errors between reconciled and the true values

NLP: Nonlinear programming

REFERENCES

Albuquerque, J. and L. Biegler (1996). Data reconciliation and gross error detection for dynamic systems. *AIChE J.*, **42**, 2841-2856.

Bagajewicz, M. and Q. Jiang (1997). Integral approach to plant linear dynamic reconciliation. *AIChE J.*, **43**, 2546-2558.

Bai, S. (2003). *Assessment of controller performance via embedded data reconciliation.* M.A.Sc. thesis, University of Ottawa, Canada.

Bai, S., J. Thibault and D.D. McLean (2005). Use of autoassociative neural network for dynamic data reconciliation. *Proceedings of 16th IFAC World Congress*, Prague, Czech Republic, July 4-8, 2005.

Binder, T., L. Blank, W. Dahmen and W. Marquardt (2002). On the regularization of dynamic data reconciliation problems. *J. Process Control*, **12**, 557-567.

Du, Y., D. Hodouin and J. Thibault (1997). Use of novel autoassociative neural network for nonlinear steady-state data reconciliation, *AIChE J.*, **43**, 1785-1796.

Hodouin, D. and S. Makni (1996). Real-time reconciliation of mineral processing plant data using bilinear material balance equations coupled to empirical dynamic models. *Int. J. Miner. Process*, **48**, 245-264.

Kamen, E.W. and J.K. Su (1999). *Introduction to Optimal Estimation*, Springer, London.

Kramer, M.A. (1992). Autoassociative neural networks. *Computers Chem. Engng*, **16**, 313-328.

Liebman, M.J., T.F. Edgar and L.S. Lasdon (1992). Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers Chem. Engng*, **16**, 963-986.

Ramamurthi, Y., P.B. Sistu and B.W. Bequette (1993). Control relevant dynamic data reconciliation and parameter estimation. *Computers Chem. Engng*, **17**, 41-59.

Wilson, D.I., M. Agarwal and D.W.T. Rippin (1998). Experiences implementing the extended Kalman filter on an industrial batch reactor. *Computers Chem. Engng*. **22**, 1653-1672.

CHAPTER VIII

Conclusion, Contributions and Recommendations

1 CONCLUSION

The DDR algorithms

In summary, three **d**ynamic **d**ata **r**econciliation (DDR) algorithms were studied in this work, namely, a **n**onlinear **p**rogramming (NLP) based DDR, a predictor-corrector based DDR, and an **a**uto**a**ssociative **n**eural **n**etwork (AANN) based DDR. The NLP-based DDR algorithm produces the reconciled values by minimizing a least squares objective function consisting of the weighted sum of squared measurement and model errors. Online optimization using nonlinear programming techniques is required at each sampling time to minimize the objective function. The NLP-based DDR algorithm can be schematically represented by Figure 1. The noisy measurements characterized by larger variance are processed by the NLP-based DDR. The noise level in the measured process variables is attenuated after the reconciliation procedure.

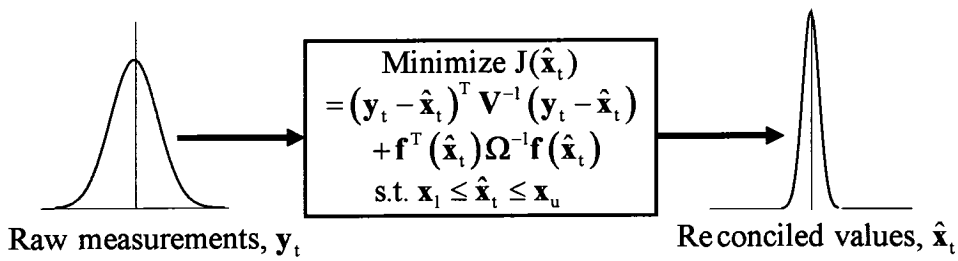


Figure 1 Scheme of NLP-based DDR algorithm.

The predictor-corrector-based DDR algorithm is derived based on Bayesian arguments when process models can be used to explicitly calculate model predictions, i.e., $\hat{\mathbf{y}}_t = \mathbf{g}(\hat{\mathbf{x}}_{t-1}, \hat{\mathbf{x}}_{t-2}, \ldots \mathbf{u}_{t-d-1}, \mathbf{u}_{t-d-2}, \ldots)$. The predictor-corrector based DDR algorithm is schematically presented in Figure 2. As shown, although errors arise in both measured and model predicted values, the reconciled values produced by the DDR algorithm exhibit more precise estimates than either the measured or the model predicted values.

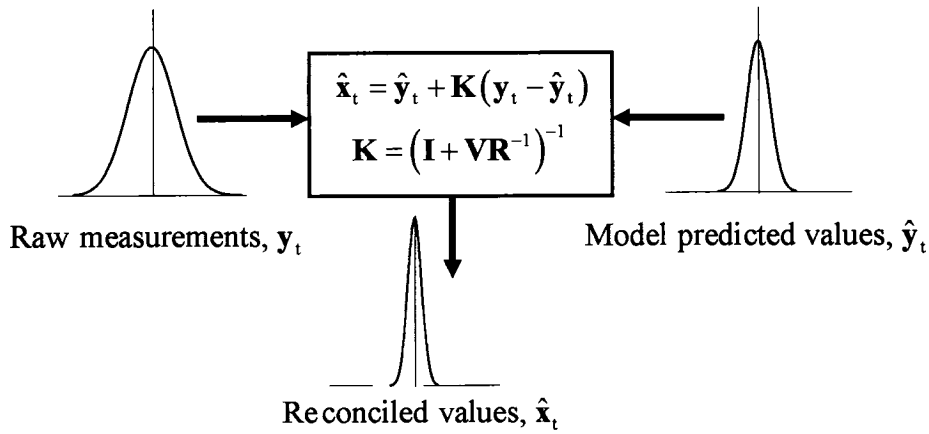


Figure 2 Scheme of predictor-corrector-based DDR algorithm.

The predictor-corrector-based DDR algorithm requires explicit calculation of model predictions at each sampling time and online tuning for the filter gain, whereas, the NLP-based DDR algorithm calls for online optimization. For complex processes, large computation time may be needed. To overcome this problem, the AANN-based DDR algorithm was developed. The AANN-based DDR algorithm is an autoassociative neural network trained offline with respect to the formulation of dynamic data reconciliation (see Equation (9) in Chapter VII). After successful offline training, the AANN-based DDR algorithm provides the advantage that it can be directly implemented online to perform DDR. Schematically, the AANN-based DDR algorithm is presented in Figure 3. The raw measurements are processed through the autoassociative neural network to yield the reconciled values. Again, the noise level contained in the reconciled values is considerably reduced relatively to that in the raw measurements. With this approach, the online tuning and optimization are eliminated.

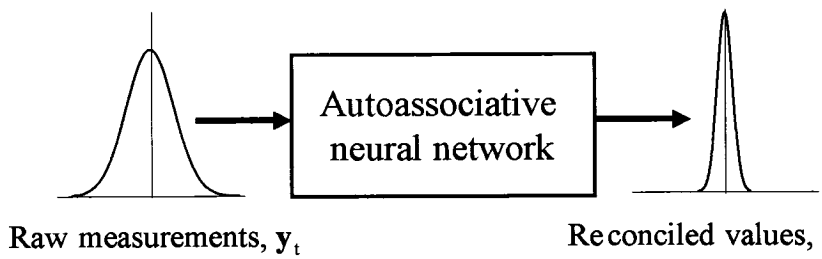


Figure 3 Scheme of AANN-based DDR algorithm.

Comparison to the Kalman filter

The predictor-corrector-based DDR algorithm was shown to be of the same form as the Kalman filter. It is interesting to compare the two algorithms. For cases where a process is described by linear models driven by white Gaussian noise and all state variables of interest are directly measured, the Kalman filter recursively calculates the covariance matrix of model predictions as

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\mathbf{P}_t^- \tag{1}$$

Putting the Kalman filter gain $\mathbf{K}_t = \mathbf{P}_t^-\left(\mathbf{P}_t^- + \mathbf{V}\right)^{-1}$ into Equation (1) yields

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{P}_t^-\left(\mathbf{P}_t^- + \mathbf{V}\right)^{-1}\mathbf{P}_t^- \tag{2}$$

Let the first term on the right-hand-side of Equation (2), $\mathbf{P}_t^-$, be multiplied by a term, $\left(\mathbf{P}_t^- + \mathbf{V}\right)^{-1}\left(\mathbf{P}_t^- + \mathbf{V}\right) = \mathbf{I}$, then

$$\mathbf{P}_t = \mathbf{P}_t^-\left(\mathbf{P}_t^- + \mathbf{V}\right)^{-1}\left(\mathbf{P}_t^- + \mathbf{V}\right) - \mathbf{P}_t^-\left(\mathbf{P}_t^- + \mathbf{V}\right)^{-1}\mathbf{P}_t^-$$

$$\mathbf{P}_t = \mathbf{P}_t^-\left(\mathbf{P}_t^- + \mathbf{V}\right)^{-1}\left[\left(\mathbf{P}_t^- + \mathbf{V}\right) - \mathbf{P}_t^-\right]$$

$$\mathbf{P}_t = \mathbf{P}_t^-\left(\mathbf{P}_t^- + \mathbf{V}\right)^{-1}\mathbf{V} \tag{3}$$

Applying the formula of matrix inverse, $\left(\mathbf{AB}\right)^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$, Equation (3) becomes

$$\mathbf{P}_t = \left[\left(\mathbf{P}_t^- + \mathbf{V}\right)\mathbf{P}_t^{-1}\right]^{-1}\mathbf{V}$$

$$\mathbf{P}_t = \left[\mathbf{I} + \mathbf{V}\mathbf{P}_t^{-1}\right]^{-1}\mathbf{V}$$

$$\mathbf{P}_t = \left[\mathbf{V}^{-1}\left(\mathbf{I} + \mathbf{V}\mathbf{P}_t^{-1}\right)\right]^{-1}$$

$$\mathbf{P}_t = \left(\mathbf{V}^{-1} + \mathbf{P}_t^{-1}\right)^{-1} \tag{4}$$

Equation (4) is exactly the covariance matrix of reconciled values obtained using the DDR algorithm shown in Equation (19) in Chapter IV by substituting $\mathbf{R}$ for $\mathbf{P}_t^-$. This means that estimates obtained using the DDR and Kalman filter are equivalent and it can also be shown that both the DDR and Kalman filter are minimum variance estimators,

provided that the covariance matrices of model prediction in the DDR and the covariance matrix of process model noise defined in the Kalman filter are accurately known. In practice, implementations of DDR are easier and more straightforward than those using the Kalman filter, since they directly evaluate the model predictions. In addition, the DDR can employ a wide variety of process models that can be phenomenological and empirical, dynamic and static, continuous and discrete, linear and nonlinear, whereas, the use of the Kalman filter is limited to process state-space models. The DDR filter can be viewed as a more generalized model-based filter with the Kalman filter being a subset.

Comparison with the EWMA and/or MA filter

The predictor-corrector-based DDR algorithm can be rewritten as

$$\hat{\mathbf{x}}_t = (\mathbf{I} - \mathbf{K})\hat{\mathbf{y}}_t + \mathbf{K}\mathbf{y}_t \tag{5}$$

where $\mathbf{K}$, the gain of the DDR algorithm, ranges from $\mathbf{0}$ to $\mathbf{I}$, neglecting cross-correlations for measurements and models. With this form, the reconciled values can be interpreted as being composed of two terms, a contribution of the model predicted value and a contribution of the measured value. Recall the form of the **e**xponentially **w**eighted **m**oving **a**verage (EWMA) filter

$$\hat{\mathbf{x}}_t = (\mathbf{I} - \Psi)\hat{\mathbf{x}}_{t-1} + \Psi\mathbf{y}_t \tag{6}$$

where Ψ, the tuning parameter of the filter, also ranges from $\mathbf{0}$ to $\mathbf{I}$. It is apparent that the DDR and EWMA filters have similar structure, except that in the DDR algorithm the model predicted values $\hat{\mathbf{y}}_t$ are used, whereas in the EWMA filter the averaged past measurements $\hat{\mathbf{x}}_{t-1}$ are used. The DDR makes use of both process temporal and spatial redundancy while only process temporal redundancy is used in the EWMA filter. It is not surprising that the DDR performs better than the EWMA filter for estimation of dynamic processes, because of its ability to anticipate process dynamics using process dynamic models.

Impact of model structure

Some important issues need to be emphasized for the use of the DDR algorithms. Process models provide redundant information for the estimation of process variables in

the DDR. When phenomenological models are unavailable or impractical to use, empirical (black-box) models need to be identified and used in the DDR. Unlike phenomenological models, such as the mass balance, black-box models often have a higher degree of uncertainty, and to choose a particular structure among the wide variety of structures is not a trivial task. Structures of black-box models have a considerable impact on the performance of DDR. A DDR algorithm using simple linear models, such as the widely used first-order-plus-dead-time model, can successfully attenuate the measurement noise. Further improvement of the DDR performance can be achieved using more comprehensive models such as neural network models. The improved performance of the DDR originates from the fact that these models can more efficiently capture the underlying dynamics of the process over a wide range of operating conditions.

It is important to comment that, the DDR algorithms rely strongly on the accuracy of process models to predict the process variables in order to provide a good estimate of their true values. Often, process state shifts or the operating point changes. For those cases, the process models may no longer accurately predict the dynamics of the process. Consequently, the models need to be adapted in order to result in better performance of the DDR algorithm.

Enhanced controller performance

The developed DDR algorithms were evaluated within the structure of feedback control loops. Without filters, the dynamics of controlled variables are often masked by measurement noise and manipulated variables display high-frequency oscillations due to noise propagation inside the feedback loops. In such cases, controllers have to be de-tuned and with de-tuned controllers, the process inevitably behaves sluggishly when subjected to external disturbances or controller setpoint changes. The use of filters, such as the classical digital EWMA or MA filters, enables the controllers to be tuned more aggressively such that the processes can be controlled more tightly. However, the DDR filters perform significantly better than the EWMA and MA filters for improved controller performance, because the DDR can provide more accurate knowledge about the state of the process by integrating information from process dynamic models.

2 CONTRIBUTIONS

The main contributions of this thesis are summarized as follows:

Development of efficient DDR algorithms

Along with the recently proposed NLP-based DDR algorithm (Makni et al., 1995; Hodouin and Makni, 1996), two new algorithms for DDR, namely predictor-corrector-based DDR and AANN-based DDR were developed. The performance of the three algorithms for the estimation of dynamic processes was quantitatively evaluated. The NLP-based algorithm is generally applicable while the predictor-corrector algorithm provides computational benefits when process models can be used to explicitly calculate model predictions and there are no hard constraints on these predictions. The AANN-based algorithm can be generally applied and avoids the online optimization in the NLP-based algorithm. Extensions of the predictor-corrector algorithm to deal with autocorrelated measurements were also developed. This work showed that the new DDR algorithms are theoretically and practically sound.

Impact of model structures on the performance of DDR

Although different model structures have been used in previous studies of DDR, no evaluation of the impact of model structure on the DDR performance has been found. In this thesis, a range of model structures, from phenomenological to black-box models were examined. It is shown that:

- The use of linear models (which are easy to identify and implement) in the DDR can significantly attenuate measurement noise.
- If further attenuation of measurement noise is required, more complex models can be used but with more engineering effort and diminishing returns.
- These findings suggest that, in practice, one should start with simple linear models and move to more complex models only if further improvements are required.

Dynamic data reconciliation versus Kalman filtering

This thesis clarifies the relationships between the DDR algorithm and the Kalman filter. Development of the DDR algorithm via Bayesian arguments showed that the

predictor-corrector form of the DDR algorithm is identical to the Kalman filter under the following assumptions: i) model is in linear state-space form; ii) model noise terms, $\mathbf{R}$ (DDR) and $\mathbf{S}$ (Kalman), must be known and such that the converged values of $\mathbf{P}_t^-$ in the Kalman filter equals $\mathbf{R}$; iii) there are no hard constraints on the state variables.

The DDR algorithm allows any form of model to be used, while the Kalman and extended (for linear approximations of nonlinear models) Kalman filters are restricted to linear state-space models. Further, the NLP-based DDR algorithm can handle hard constraints on the state variables. Overall, the Kalman filter can be regarded as a subset of the DDR algorithm.

Controller performance with embedded DDR algorithms

This thesis demonstrates the improvement in controller performance with DDR algorithms embedded in standard PID feedback control loops. DDR filters resulted in significant improvements over traditionally used EWMA and MA filters for both storage tank and distillation column processes.

Use of AANN-based DDR algorithm

This thesis demonstrates the novel application of AANN for DDR in two storage tanks and a distillation column. Algorithms were developed using a recurrent AANN and a feedforward AANN. Both algorithms were effective but the recurrent form displayed better performance for the distillation process. The AANN-based DDR algorithm was found to possess the following benefits:

- Training of the AANN can be done offline
- After training, the AANN can be directly implemented online without explicit calculations of model predictions (as required by Kalman filter) or online optimizations (as required by NLP-based DDR)
- Similar to other DDR algorithms, the AANN-based DDR is robust to the level of measurement noise.

3 RECOMMENDATIONS

Throughout this work, a number of interesting research topics arose and the most important are presented here as recommendations for future studies. Some initial ideas that would allow carrying on these studies are discussed below.

Dynamic data reconciliation considering cross-correlated measurements

Although the DDR algorithms developed in this work take into account both variances and covariances of measurement errors, the covariances were not added to the measurement errors in the simulation examples performed throughout this work (i.e., the covariance matrix of measurements was diagonal). From a practical point of view, for simplicity, the covariance matrix is often assumed to be diagonal. For steady-state data reconciliation, Hodouin et al. (1998) showed that neglecting the covariances does not dramatically degrade the reliability of reconciled data if the covariances are relatively small. However, considerations of covariances can improve the consistency of reconciled data (Bazin and Hodouin, 2001). Evaluations of the impact of covariances on the results of dynamic data reconciliation are recommended for future studies.

Design of neural controller to simultaneously perform DDR and process control

This work demonstrated that the use of an AANN-based DDR algorithm inside control loops significantly improved the controller performance due to noise attenuations performed by the neural network. It would be interesting to integrate the controllers and the DDR into a neural network that could simultaneously perform process control and noise reduction. The scheme of the neural controller is proposed in Figure 4. A neural network having the similar architecture as the AANN is trained offline. For network training, open loop data sets for manipulated variables and controlled variables corrupted by measurement noise are collected. Then, in network training, the vectors $[\mathbf{y}_{t+H}, \mathbf{y}_t, \mathbf{u}_{t-d}]^T$ are presented to the input layer while the vectors $\mathbf{u}_t$ are presented to the output layer of the network. The training criterion is to minimize the sum of squared differences between the network outputs and the training data $\mathbf{u}_t$. The training is essentially to let the network to learn the inverse dynamics of the process such that the

network can produce appropriate control moves when implemented online as a controller. The number of future time steps, H, as a tuning parameter, is selected such that the neural controller has acceptable online performance for setpoint changes and for external disturbances. In addition, by means of data compression and regeneration performed by the network, the noise contained in the controlled variables can be significantly attenuated by passing it through the network. Consequently, smooth control actions are achieved. For multiple-input multiple-output (MIMO) process, the use of the proposed neural controller also provides the advantage that it can decouple the interactions between controlled variables since it is trained by considering plant-wide dynamics.

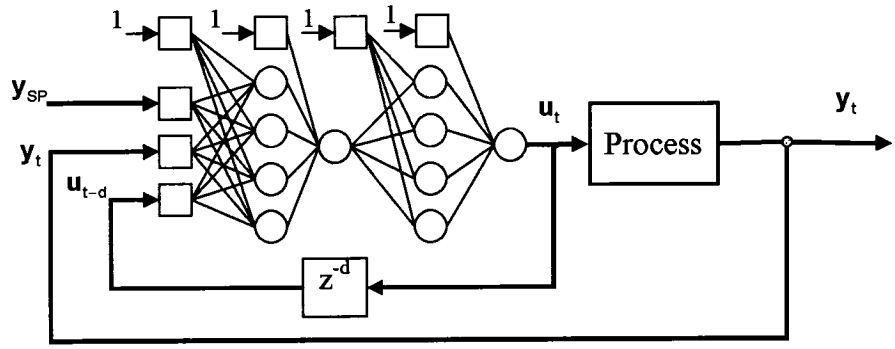


Figure 4 Scheme of a neural controller to simultaneously perform DDR and process control.

Measurement bias detection

The DDR algorithms developed in this work assume measured values are unbiased. However, measurement bias is commonly present in process measurements. Detection of measurement bias is an important issue in data reconciliation. Measurement bias needs to be identified and removed from raw measurements before reconciliation. McBrayer and Edgar (1995) proposed a bias detection method incorporated in a dynamic data reconciliation algorithm using process phenomenological models as constraints. As stated, phenomenological models are often difficult to obtain for a chemical process. On the other hand, empirical models have been widely used in practice. The algorithm of measurement bias detection in dynamic data reconciliation can be formulated as

Minimize $J(\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_{t-1}, \ldots, \hat{\mathbf{x}}_{t-L}, \hat{\beta}) =$

$$\sum_{i=0}^{L}\left\{\left[\left(\mathbf{y}_{t-i}-\hat{\beta}\right)-\hat{\mathbf{x}}_{t-i}\right]^{T}\mathbf{V}^{-1}\left[\left(\mathbf{y}_{t-i}-\hat{\beta}\right)-\hat{\mathbf{x}}_{t-i}\right]+\mathbf{f}^{T}\left(\hat{\mathbf{x}}_{t-i}\right)\Omega^{-1}\mathbf{f}\left(\hat{\mathbf{x}}_{t-i}\right)\right\} \quad (7)$$

where $\hat{\beta}$ is a vector of estimates for the measurement bias assumed constant within the time horizon, [t-L, t]. A moving window is necessarily added to the DDR algorithm in order to effectively detect the true values of the measurement bias, β. The window length L should be an important parameter for the bias detection. The larger the value of L, the more effective the detection algorithm should be, but at the expense of more computation effort in solving the optimization problem of Equation (7).

Fault detection and diagnosis for dynamic processes using neural networks

Fault detection and diagnosis are important issues in plant operation. Process faults involve measurement device malfunction, failure of actuators and other abnormal process conditions such as serious degradations of reaction catalyst. There has been a considerable interest in this field from industrial practitioners and academic researchers. Various detection and diagnosis methods from different perspectives have been proposed in the literature, ranging from analytical methods which are based on process models, to artificial intelligence methods which are based on historical process data (Venkatasubramanian et al., 2003). The use of artificial neural networks has been recognized as an effective approach to detect and diagnose process faults for steady-state processes and has been considerably studied in the literature. Fault detection and diagnosis for dynamic processes using neural networks is an appealing topic recommended for future studies.

Implementation of Dynamic data reconciliation for industrial processes

It is recommended that the models proposed for the autogenous/semi-autogenous (AG/SAG) grinding process presented in Annex A be calibrated using plant data. To implement data reconciliation strategies, these calibrated models could run online. At each sampling time, raw measurements can be reconciled with associated model predicted values using the predictor-correct form of the DDR as the estimates for measured process variables, meanwhile estimates for unmeasured process variables are

just the model predicted values. The applications of the DDR algorithm should significantly improve process monitoring and control strategies.

REFERENCES

Bazin, C. and D. Hodouin (2001). Importance of covariance in mass balance of particle size distribution data. *Minerals Engineering*, **14**, 851-860.

Hodouin, D. and S. Makni (1996). Real-time reconciliation of mineral processing plant data using bilinear material balance equations coupled to empirical dynamic models. *Int. J. Miner. Process*, **48**, 245-264.

Hodouin, D., A. Mirabedini, S. Makni, and C. Bazin (1998). Reconciliation of mineral processing data containing correlated measurement errors. *Int. Miner. Process.* **54**, 201-215.

Makni, S. and D. Hodouin and C. Bazin (1995). A recursive node imbalance method incorporating a model of flowrate dynamics for on-line material balance of complex flowsheets. *Miner. Eng.*, **8**, 753-766.

McBrayer, K.F and T.F Edgar (1995). Bias detection and estimation in dynamic data reconciliation. *J. Process Control*, **5**, 285-289.

Venkatasubramanian, V., R. Rengaswamy, K. Yin and S.N. Kavuri (2003). A review of process fault detection and diagnosis, Part I: Quantitative model-based methods. *Computers Chem. Engng*, **27**, 293-311.

Venkatasubramanian, V., R. Rengaswamy, and S.N. Kavuri (2003). A review of process fault detection and diagnosis, Part II: Qualitative models and search strategies. *Computers Chem. Engng*, **27**, 313-326.

Venkatasubramanian, V., R. Rengaswamy, S.N. Kavuri and K. Yin (2003). A review of process fault detection and diagnosis, Part III: Process history based methods. *Computers Chem. Engng*, **27**, 327-346.

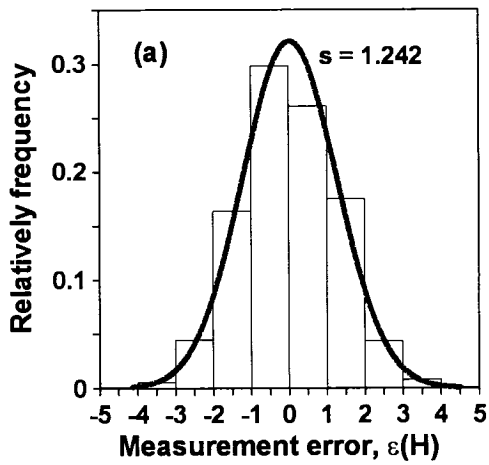
ANNEX A

Histograms of Measurement Errors, Model Prediction Errors and Reconciliation Errors

The DDR algorithms developed in this work are based on the assumption that the measurement noise and model prediction error are normally distributed. If this assumption is valid, it can be shown that the reconciled values are also normally distributed and the reconciliation error is less than both model prediction and the measurement errors (see Equation (10) in Chapter V). In the simulation conducted throughout this work, simulated white Gaussian noise was added to the true values of process variables to represent typical raw measurements. In order to validate the assumption that the model prediction error is also Gaussian noise, the model prediction errors were calculated and their histograms are presented.

Cylindrical storage tank process

For the simple process consisting of the cylindrical storage tank process shown in Figure 1 of Chapter III, the model predictions for the tank level were calculated by the linear model, i.e. the mass balance for the tank (see Equation (17) in Chapter III). For the dynamic case study shown in Figure 2 of Chapter III, the histograms of the measurement noise, model prediction error and reconciliation error for the tank level are presented in Figure 1. It sows both the model prediction and reconciliation errors resemble normal distributions, and their variations are significantly smaller than the raw measurement noise. In addition, as expected, the reconciliation error has a standard deviation that is lower than both measurement and model prediction errors and satisfies the relationship that was established in Equation (10) of Chapter V.



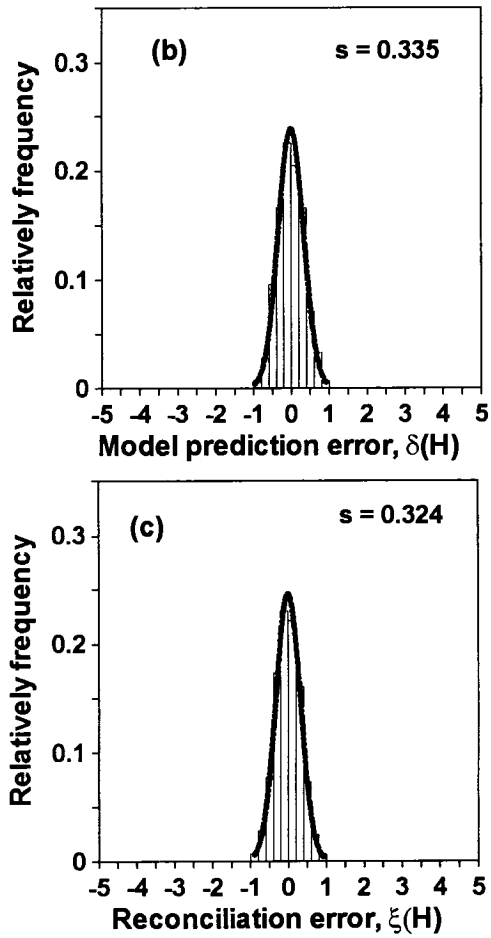


Figure 1 Histograms of (a) measurement noise, (b) model prediction error, and (c) reconciliation error for the cylindrical storage tank level. s: standard deviation of the error.

Distillation column process

The assumption about the structure of model prediction error was also justified for a more complex process, i.e. for the distillation column where black-box models were identified and employed. Two linear models (See Equations (C1) and (C3) in Chapter IV) were used for the two liquid levels, while two nonlinear models (See Equations (23) and (24) in Chapter V) were used for the two temperatures. For the dynamic case study shown in Figures 7(a) and (b) of Chapter V, the histograms of measurement noise, model prediction errors, and reconciliation errors for the reflux drum level, top temperature, column base level and bottom temperature are presented in Figures 2-5, respectively.

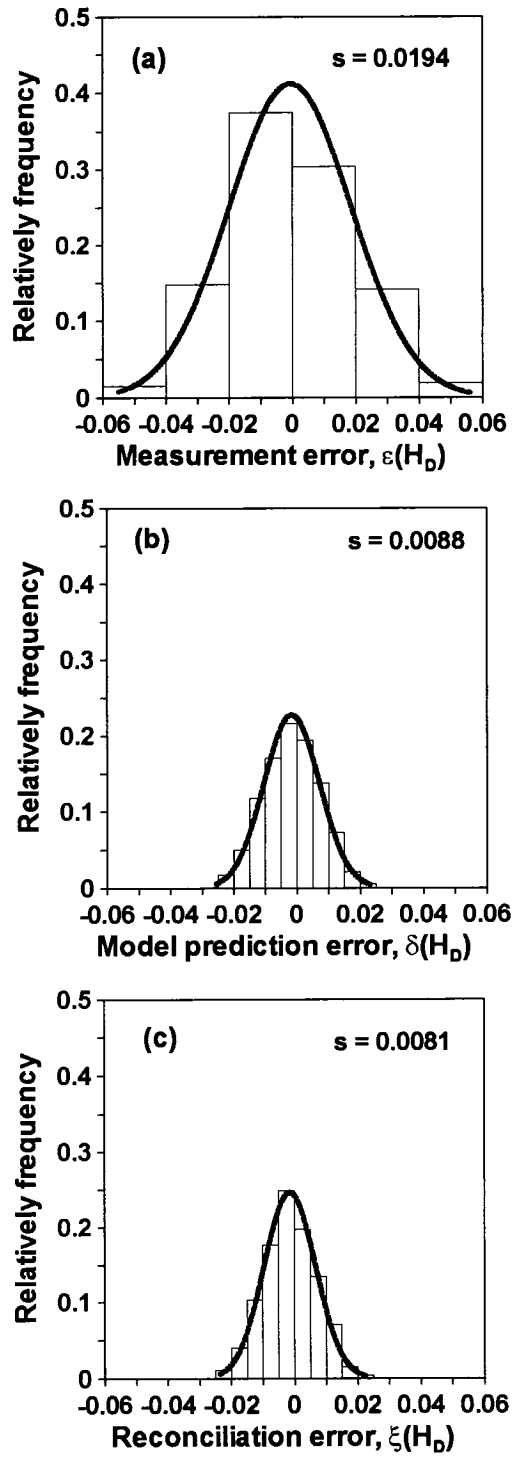


Figure 2 Histograms of (a) measurement noise, (b) mode prediction error and, (c) reconciliation error for the reflux drum level of distillation column. s: standard deviation of the error.

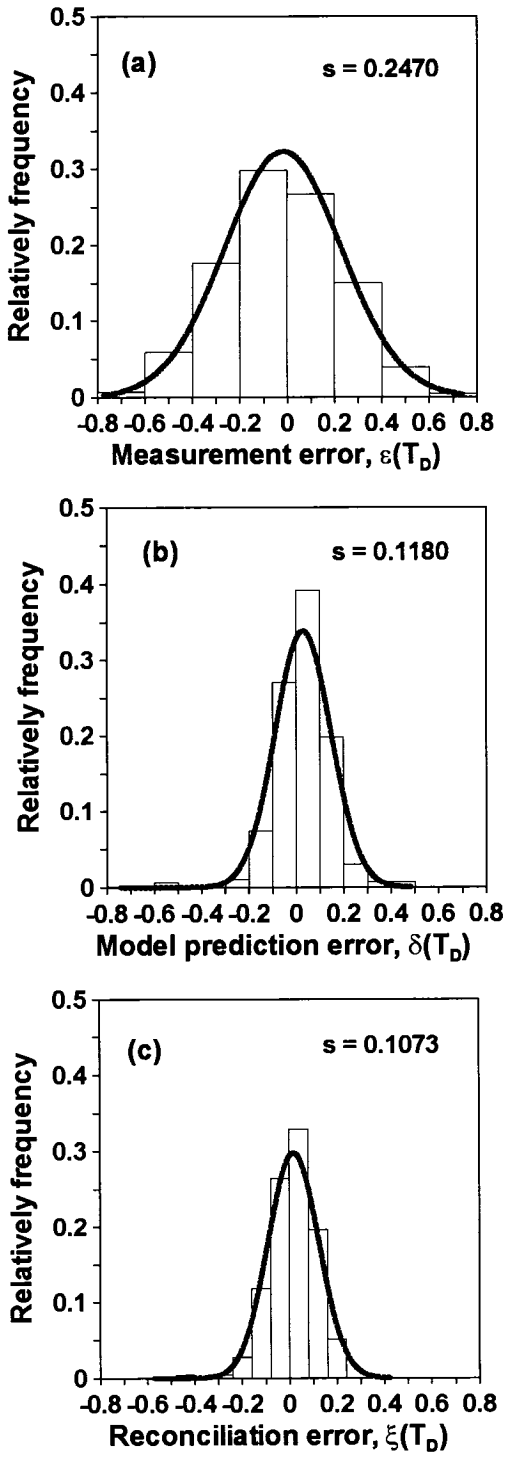


Figure 3 Histograms of (a) measurement noise, (b) mode prediction error and, (c) reconciliation error for the top temperature of distillation column. s: standard deviation of the error.

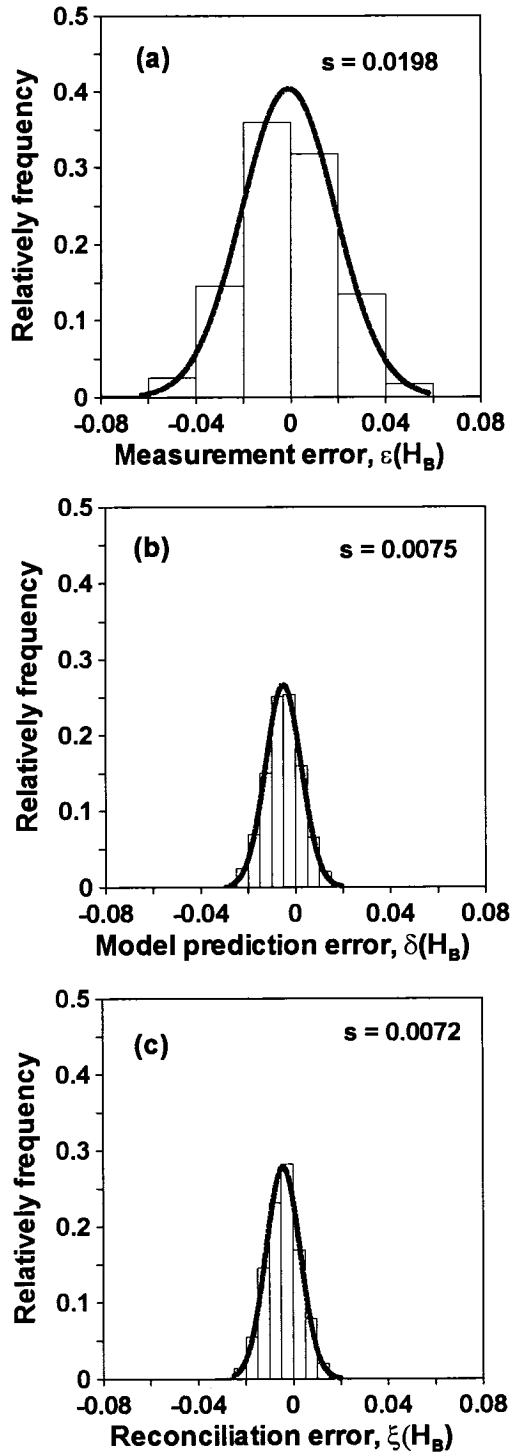


Figure 4 Histograms of (a) measurement noise, (b) mode prediction error and, (c) reconciliation error for the base level of distillation column. s: standard deviation of the error.

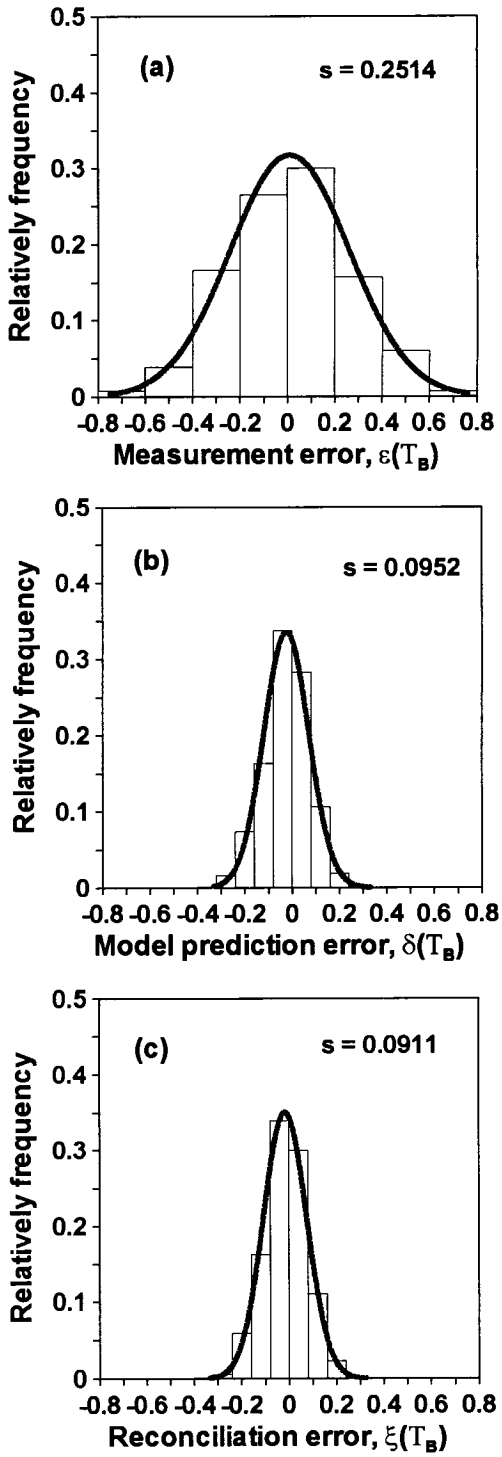


Figure 5 Histograms of (a) measurement noise, (b) mode prediction error and, (c) reconciliation error for the bottom temperature of distillation column. s: standard deviation of the error.

Figures 2-5 show very clearly that both the model prediction errors and reconciliation errors very closely resemble Gaussian distribution, and the reconciliation errors have smaller standard deviations than the model prediction errors and the measurement errors. Furthermore, the standard deviations of the measurement noise, the model prediction errors, and the reconciliation errors statistically satisfy Equation (10) in Chapter V.

It can therefore be concluded that the assumption of normal distribution for the model prediction errors are satisfied for the processes that were studied in this work. As a result, the reconciliation errors were also normally distributed.

ANNEX B

A Simple Model for Dynamic Simulation of Autogenous/Semi-Autogenous Grinding Mills

Shuanghua Bai, Jules Thibault*, and David D. McLean

Department of Chemical Engineering

University of Ottawa

Ottawa, Canada K1N 6N5

Presented at **International Conference on Mineral Process Modeling, Simulation and Control**, Sudbury, ON, Canada, June 6-7, 2006.

* Corresponding author. Tel.: 613-562-5800 ext. 6094
Email: thibault@genie.uottawa.ca

ABSTRACT

In the mineral processing industry, autogenous (AG)/semi-autogenous (SAG) grinding mills play a vital role in production of valuable minerals. Plant operation personnel are constantly seeking to improve the economic performance of the whole plant by trying to achieve a higher productivity and a more consistent product quality while minimizing energy consumption and environmental impact. However, the energy required for grinding may account for up to 40% of the direct operating cost and improving the efficiency of the grinding operation will translate into substantial savings. Process modeling and simulation may provide a cost-effective approach to achieve these objectives. Indeed, process modeling and simulation allow investigation of the dynamic behavior of the grinding process and assessment of different operating strategies under a variety of operating conditions. This paper provides a more parsimonious parametric model for the AG/SAG grinding mill process. Based on the proposed models, the steady state and dynamics of a typical industrial grinding process was studied under various scenarios.

Keywords: AG/SAG grinding mill, modeling, simulation.

1 INTRODUCTION

In mineral processing plants, autogenous (AG)/semi-autogenous (SAG) grinding mills are important unit operations for the production of valuable minerals from mined ore. Crushers are used to break bulky lumps of mined ore into relatively large particles of various sizes, and then these particles are conveyed into the AG/SAG mills for further size reduction in order to obtain the desired particle size distribution. Leaving the grinding process, the ore particles are sent to floatation cells to preferentially recover valuable minerals. One critical objective in the AG/SAG mill operation is therefore to produce an optimal particle-size distribution prior to the floatation operation. This objective must be satisfied because only the minerals embedded in a particular range of particle size, called liberation size, can be effectively recovered. Grinding to a size either larger or smaller than the liberation size reduces the efficiency of recovery and also may cause a pollution control problem (Radhakrishnan, 1999). Another critical goal in the AG/SAG mill operation is to increase plant energy efficiency. It is estimated that the grinding mills take 30% to 70% of total plant power consumption, while power consumption in the mineral processing accounts for half of the operating cost (Nikkhah and Anderson, 2001). It is important to point out, however, that less than 10% of the total power input is actually utilized in the size reduction (Rajamani and Herbst, 1991). As a result, it is essential to efficiently monitor, control and optimize the grinding mill processes, such that the operating cost is minimized, the plant productivity is maximized, and the size specifications for the particles leaving for the floatation operation are satisfied.

One avenue to achieve these objectives is through process modeling and simulation. Indeed, process modeling and simulation not only provide a better understanding of the influence of the various disturbances affecting the performance of the process, but also allow evaluations of different operating and control strategies for the mill. A few steady-state and dynamic models of grinding mill processes have been proposed in the literature (Rajamani and Herbst, 1991; Valery and Morrell, 1995; Morrell et al., 1996; Morrell and Man, 1997; Morrell, 2004). The proposed models are relatively complex and are usually

characterized by a very large number of model parameters composed of breakage rate constants and breakage distribution functions of ore particles. As a result, calibrating these models can be tedious and expensive. In this study, we developed an alternative approach to model the grinding mill process. The proposed model has a significantly lower number of parameters than those published in the literature. The structure of the simplified model for the grinding process is outlined. Then, steady-state conditions and dynamic behaviors of the process were investigated using the proposed models.

2. MODELS FOR THE GRINDING MILL PROCESS

A typical flow diagram of an industrial AG/SAG grinding process is presented in Figure 1. The fresh ore is continuously fed to the grinding mill. The mill mainly consists of a rotating cylinder having a large diameter and a relatively short length. During its rotation, the contents of the mill tumble violently causing breakage of the particles by impact, abrasion and attrition between the particles. In AG mills, only ore particles are present and used as grinding media, whereas, a number of steel balls are added to SAG mills in order to enhance the breakage. Fresh water is continuously added to the mill to create a slurry that is required to allow the solid particles to flow. The load on the mill consisting of the ore particles, steel balls (about 8% of the load, if used) and the water, occupies approximately 30% of mill volume. When the ore particles have been reduced to the desired size, they are able to escape through the openings of the discharge grates along with the water. The discharge slurry, containing about 20% water, is then collected and diluted with additional water in the stirred sump. The slurry level of the sump is controlled by manipulating a variable speed pump on the outlet stream. The diluted ore slurry then enters into the hydrocyclone classifier to separate the ore into finer and coarser particles. The underflow stream of the hydrocyclone containing larger particles is recycled to the mill for further grinding. The overflow stream of the hydrocyclone, containing finer particles, is sent to the flotation cells to recover valuable components.

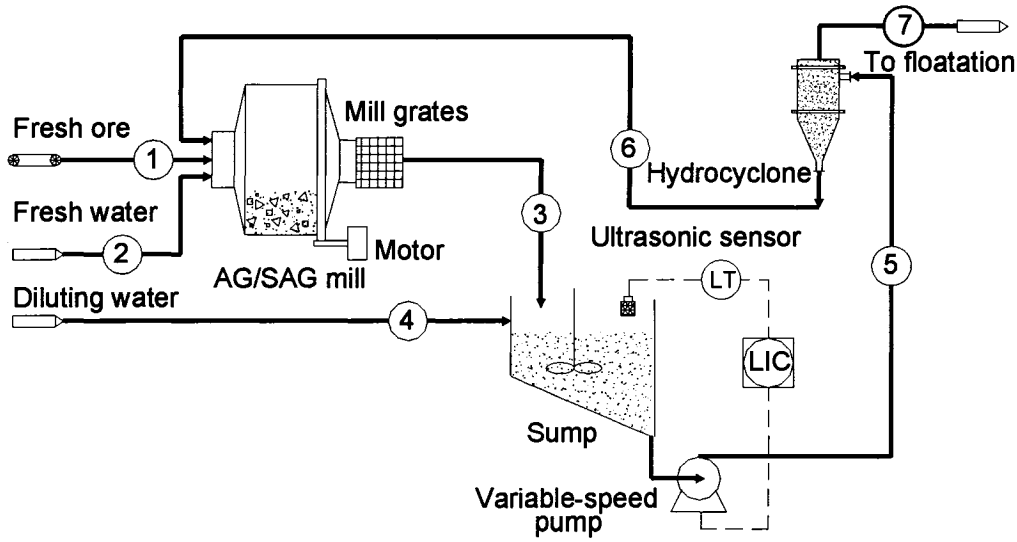


Figure 1 Flow diagram of an AG/SAG grinding mill process.

2.1 Grinding mill model

The particles in AG/SAG mills can be considered to have a continuous particle size distribution. To conveniently model the grinding process, the continuous size distribution is discretized into N particle-size classes. Particle class 1 corresponds to the biggest particles, whereas particle class N contains the finest particles. In the modeling of breakage kinetics for particle-size class i, i = 1, 2, ..., N, two expressions have been typically used in the literature, namely the breakage rate constant and breakage distribution functions (Rajamani and Herbst, 1990; Valery and Morrell, 1995; Morrell et al., 1996; Morrell and Man, 1997; Morrell, 2004). The breakage rate constant is employed to describe how fast the particles of size class i are broken into its descendant smaller size classes i+1, i+2, ..., N. Meanwhile, the breakage distribution functions are the particle fractions for sizes i, i+1, i+2, ..., and N after the breakage event of the particles of size i. The breakage rate constant and the distribution functions are used to calculate the generation rate for particles of size i, g_i. It is defined as the sum of the products of the breakage rate constants, the distribution functions and the mass holdups from size 1 to i, and it is given by

$$g_i = \sum_{j=1}^{i} a_{i,j} r_j m_j \qquad (1)$$

where $a_{i,j}$ is the values of breakage distribution function for particles of size j, r_j is the associated breakage rate constant, and m_j is the mass holdup of particle size j in the mill. To obtain the model parameters, the breakage rate constants are usually calibrated from plant data, while the distribution functions are obtained from a series of drop-weight tests in the laboratory. For particles divided into N particle classes, the parameters of the distribution functions can be represented by a N×N upper-right-hand-triangular matrix. The problem with this approach is the large number of parameters, $[N+N(N-1)/2]$, that need to be identified, especially when the number of particle size classes N is large. In addition, the laboratory tests required to estimate this matrix are not representative of the conditions that prevail in industrial grinding equipment. It is nearly impossible to obtain this information from industrial data unless one accepts to spend significant effort and time to perform numerous tests.

As an alternative to the breakage distribution function, a simplified model to account for the breakage action of the ore particles is proposed in this work. On average, the progressive particle size reduction can be represented by a series of chemical reactions as shown schematically in Figure 2 where each particle size represents a chemical species. With this scheme, the generation of particles of class size i can only result from its parent particles of class i-1. Therefore, as schematically illustrated in Figure 2, the generation rate of particles of size i is simply the product of the breakage rate constant, k_{i-1}, and the associated mass holdup, m_{i-1}. Using this scheme, only the breakage rate constants are involved, and therefore the number of model parameters reduces to N-1.

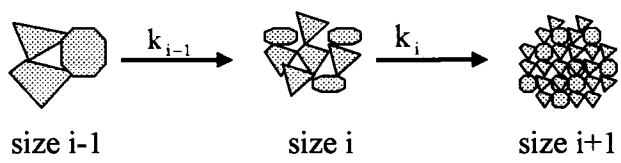


Figure 2 Scheme of breakage model of particles in AG/SAG grinding mills.

Sometimes, density of the ore particles fed to the grinding mill may vary dramatically during the operation, and the breakage dynamics will also vary. To accommodate density changes and their impact on breakage rate, the possibility of having different classes of density has been added to the model. The model is therefore developed for M density classes and each density class is divided into N size classes. Considering the grinding mill process shown in Figure 1, the population mass balance for particles belonging to size i and density j at time t can be written as

For i = 1 and j = 1, 2, ..., M

$$\frac{dm_{1,j}}{dt} = f^{1}_{1,j} + f^{6}_{1,j} - f^{3}_{1,j} - k_{1,j}m_{1,j} \tag{2}$$

For i = 2, 3, ..., N-1 and j = 1, 2, ..., M

$$\frac{dm_{i,j}}{dt} = f^{1}_{i,j} + f^{6}_{i,j} - f^{3}_{i,j} + k_{i-1,j}m_{i-1,j} - k_{i,j}m_{i,j} \tag{3}$$

For i = N and j = 1, 2, ..., M

$$\frac{dm_{N,j}}{dt} = f^{1}_{N,j} + f^{6}_{N,j} - f^{3}_{N,j} + k_{N-1,j}m_{N-1,j} \tag{4}$$

where $m_{i,j}$ is the particle mass holdup for size i and density j in the mill; $f^{p}_{i,j}$ is the mass flow rate of the particles of size i and density j in the p^{th} material stream; $k_{i,j}$ is the associated breakage rate constant.

The breakage rate constant, $k_{i,j}$, is a measure of the friability of the ore for passing from one class size to the next. For a specified mined ore, the values of $k_{i,j}$ depend on numerous factors, such as the ball charge, the charge particle size distribution and the speed of rotation of the mill. The calibration of this model will probably show that the values of $k_{i,j}$ would be higher for larger particles. Further, since any change in the feed particle size distribution will directly affect the charge size distribution, variations in the feed particle size distribution will affect the breakage rate, especially for AG mills, because the particles of the ore are the grinding media. Studies on the impact of mill

speed, fresh ore feed size distribution, ball size and ball charge volume on the breakage rate constants have been presented by Morrell et al. (1996).

In addition to the mass balance of the ore particles, the mass balance of water in the mill is given by

$$\frac{dm_{H_2O}}{dt} = f^2_{H_2O} + f^6_{H_2O} - f^3_{H_2O} \tag{5}$$

where m_{H_2O} is the mass holdup of water in the mill; $f^p_{H_2O}$ represents the water flow rate in the p^{th} material stream. Knowing the mass holdup for each individual particle class and water, the mass holdup for the ore in the mill, M_{ore}, and the total mass holdup, M_{total}, can be calculated as

$$M_{ore} = \sum_{i=1}^{N}\sum_{j=1}^{M} m_{i,j} \tag{6}$$

$$M_{total} = M_{ore} + m_{H_2O} \tag{7}$$

Similarly, the mass flow rate of the ore particles, F^p_{ore}, and the total mass flow rate, F^p, in the p^{th} material stream can be given by

$$F^p_{ore} = \sum_{i=1}^{N}\sum_{j=1}^{M} f^p_{i,j} \tag{8}$$

$$F^p = F^p_{ore} + f^p_{H_2O} \tag{9}$$

Equations (2-9) are the lumped-parameter models with the assumption of perfect mixing for the ore particles and water in the grinding mill.

2.2 Mill discharge rate model

The flow rate of the slurry leaving the mill depends on two mechanisms: the transport to the mill grate and the classification by the discharge grate. The discharge flow rate of particles belonging to size i and density j is proportional to their mass holdup in the mill and depends on the grate openings, i.e.

$$f_{i,j}^{3} = d_i m_{i,j} \tag{10}$$

where d_i is the associated discharge rate function. Although d_i can be calibrated from the measurements of the mass of each particle size i inside the mill and the mass flow rate of the same particle size exiting the mill, to determine accurately this value is not a trivial task. Consequently, Morrell (2004) recommended using Figure 3 to determine the values of d_i. For particle size smaller than ϕ_m (usually 1 mm), the discharge rate function is constant and equal to that of water, d_{max}. This assumes that the slurry containing particles smaller than size ϕ_m behaves like a homogenous fluid. For particle size larger than ϕ_m, the discharge function decreases linearly on a semi-logarithmic scale until it reaches zero at particle sizes larger than the grate port size, ϕ_0. Within the particle size region [ϕ_m, ϕ_0], the probability of particle retention by the grate increases with the increase of the particle size.

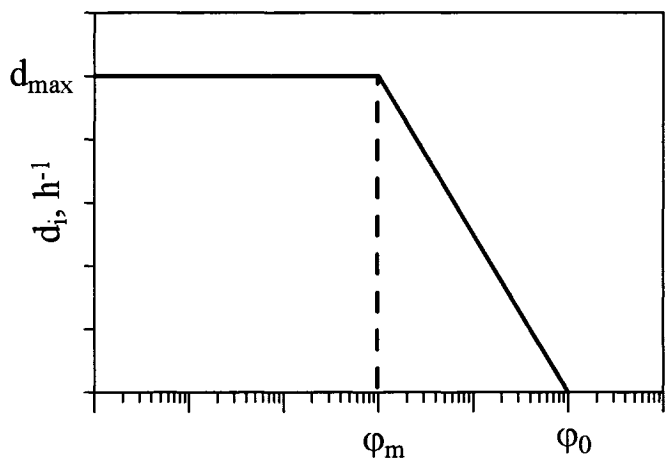


Figure 3 Model of discharge rate function for the AG/SAG mill.

2.3 Sump model

The sump can be modeled as a continuously stirred tank with perfect mixing. Neglecting size changes of ore particles in the sump due to abrasion or attrition caused by the blades, the mass balance for particles belonging to size i and density j can be represented by

$$\frac{ds_{i,j}}{dt} = f^3_{i,j} - f^5_{i,j} \tag{11}$$

where $s_{i,j}$ is the mass holdup in the sump tank for particles of size i and density j. The mass balance for the water around the sump can be given by

$$\frac{ds_{H_2O}}{dt} = f^3_{H_2O} + f^4_{H_2O} - f^5_{H_2O} \tag{12}$$

where s_{H_2O} is the water holdup in the sump. The mass holdup for all the ore particles, S_{ore}, and the total material mass holdup in the sump, S_{total}, can be given by

$$S_{ore} = \sum_{i=1}^{N}\sum_{j=1}^{M} s_{i,j} \tag{13}$$

$$S_{total} = S_{ore} + s_{H_2O} \tag{14}$$

With closed loop of the slurry level controller, the volumetric flow rate of the slurry leaving the sump is determined by a control equation. For a proportional-integral (PI) controller, the controller equation in its discrete form is given by

$$V^5_{slurry,t} = V^5_{slurry,t-1} + K_C\left[(\varepsilon_t - \varepsilon_{t-1}) + \frac{\Delta T}{\tau_I}\varepsilon_t\right] \tag{15}$$

where K_C and τ_I are the controller parameters, ε_t denotes the deviation between the controller setpoint and the slurry level at time step t, and $V^5_{slurry,t}$ represents the control move at time step t, i.e. the volumetric flow rate of the outlet stream of the sump at time t.

2.4 Hydrocyclone model

The residence time of the slurry in the hydrocyclone is much smaller than the residence time of the slurry in the mill and the sump. Therefore, the dynamics of the hydrocyclone can be considered negligible. During the hydrocyclone operation, the centrifugal force induced by the flow causes coarser particles to be discharged with a small amount of water to the underflow stream, which is subsequently recycled to the mill for further grinding. At the same time, the inner spiral of fluid brings the finer particles into the overflow stream. It is known that developing a phenomenological model based on fluid flow mechanics for the hydrocyclone process is difficult. Consequently, an empirical model having eight parameters has been used in the literature (e.g., Rajamani and Herbst, 1990). Obviously, estimation of an eight-parameter model is not trivial. In this work, a two-parameter model for the hydroyclone process is proposed. Akin to the slurry discharge flow leaving the grinding mill, the mass flow rate of particles belonging to size i and density j in the overflow stream is proportional to the flow rate in the feed stream, i.e.

$$f_{i,j}^{7} = p_{i,j} f_{i,j}^{5} \tag{16}$$

where $p_{i,j}$ is the probability that particles of size i and density j leave with the overflow stream. $p_{i,j}$ is a nonlinear function of particle size. A sigmoid function was used in this investigation to model $p_{i,j}$ as

$$p_{i,j} = \frac{1}{1+e^{a_j \log_{10}(\phi_{i,j})+b_j}} \tag{17}$$

where a_j and b_j are the model parameters for density class j and they are determined by measuring the mass fractions of the particle classes in the overflow stream and the feed stream to the hydrocyclone. Figure 4 illustrates the probability functions for a variety of density classes. For a particular density class, the probability of smaller particles (less

than 0.1 mm) leaving with the overflow stream is close to 1. The probability for medium particles (0.3-3 mm) decreases very rapidly with an increase in particle size, and it reaches zero for particles larger than 10 mm. For a particular size class, the probability of particles leaving with the overflow stream is lower for higher density particles than for lower density particles, since higher density particles are more easily expelled to the underflow stream under centrifugal forces. The model parameters, a_j and b_j, are functions of the water concentration in the feed stream to the hydrocyclone. For finer particles, their probability of leaving with the overflow stream increases with higher water concentration. On the other hand, for coarser particles, the probability to leave with the underflow stream increases with higher water concentration.

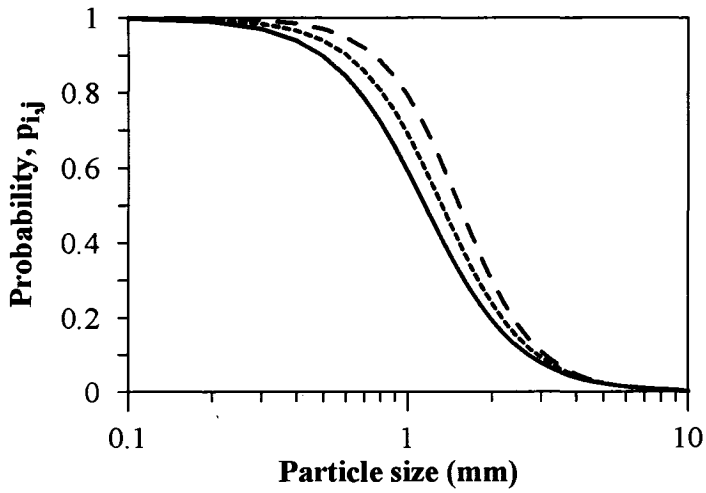


Figure 4 Possible probability functions for particles to leave with the overflow stream. ——: density class j = 1; -----: j = 2; — — —: j = 3. Density class 1 is the highest density.

Following the determination of the particle flow rates in the overflow stream, the flow rates of the particles belonging to size i and density j leaving with the underflow stream can be given by the mass balance

$$f_{i,j}^{6} = (1 - p_{i,j}) f_{i,j}^{5} \tag{18}$$

Along with the flows of the ore particles, a relatively small amount of water is entrained by the particles in the underflow stream. The quantity of water entrained is proportional to the total mass flow of ore particles in the underflow stream, i.e.

$$f^6_{H_2O} = \theta \sum_{i=1}^{N} \sum_{j=1}^{M} f^6_{i,j} \tag{19}$$

where θ is a constant determined by measuring the water content and the flow rate of the underflow stream. By means of a mass balance for water, the water flow rate in the overflow stream can be determined by

$$f^7_{H_2O} = f^5_{H_2O} - f^6_{H_2O} \tag{20}$$

3. SIMULATION RESULTS

The model for the grinding mill process consists of coupled differential/algebraic equations (DAEs). This set of equations needs to be solved as a function of time. A numerical algorithm using explicit Euler's method with an integral step size of 1 second was used to solve the DAEs. A FORTRAN program was coded to carry out the simulations for the process shown in Figure 1.

3.1 Model parameters

To investigate steady-state conditions and dynamic responses of the grinding process, most values of the model parameters were assigned based on data in the literature (Morrell et al., 1996; Morrell, 2004) while some others were based on engineering judgment. In addition, transportation time lags of 0.5 minute and 2 minutes were assumed for the discharge stream leaving the mill and the recycle stream leaving the hydrocyclone, respectively.

Two density classes of the ore particles were used in the simulation. Particles belonging to density class 1 had a higher density. The cumulative mass fractions of particle sizes of density classes 1 and 2 in the fresh ore feed stream were assigned 0.5 each. Within each density class, the ore particles were further classified into ten size classes, i.e. N = 10. The ore particle size distributions in each density class in the fresh ore feed stream are

presented in Figure 5. The particles of size class 3 (corresponding to particle size 30 mm) had the highest mass fraction in the feed stream. The total mass flow rate of the fresh ore particles fed into the mill was 500 t/h. The mass flow rate of fresh water entering to the mill was 240 t/h.

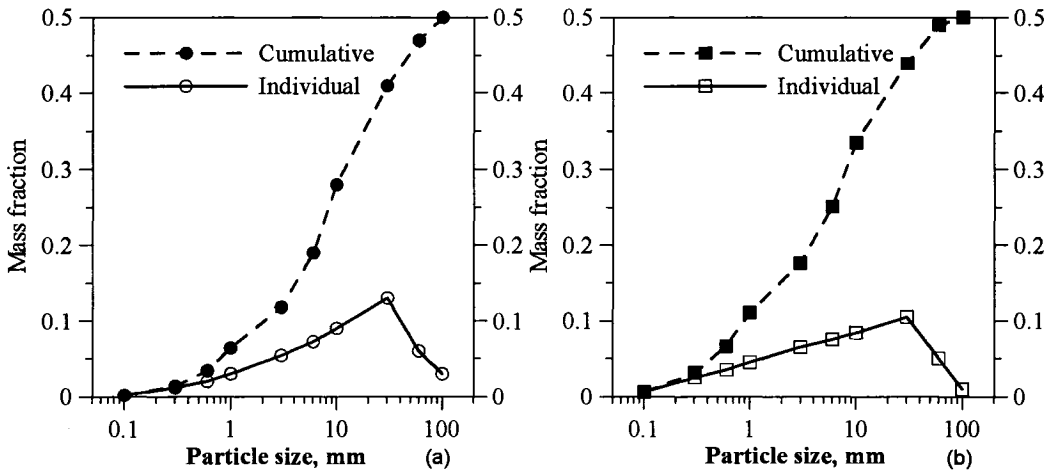


Figure 5 Particle size distributions of (a) density class 1 and (b) density class 2 in the fresh ore feed stream.

For each density class, the breakage rate constant for each size class adapted from Morrell (2004) is presented in Figure 6. This figure shows typical characteristics of the breakage rate constants as a function of particle size that is normally observed in AG/SAG mills. In general, the ore breakage rate constant is larger for lower density particles because the friability of these particles is usually higher.

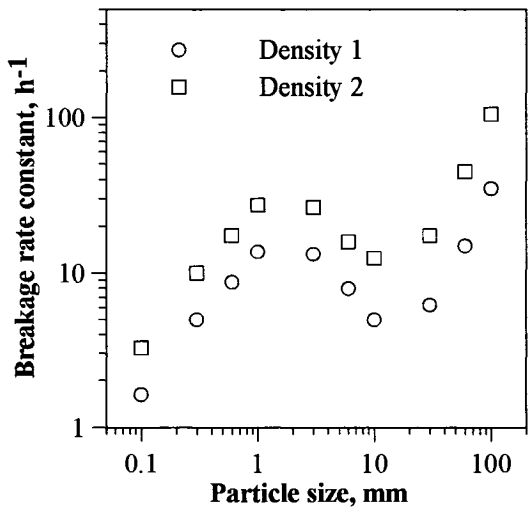


Figure 6 Breakage rate constants of ore particles.

The values, $d_{max} = 20.0\ h^{-1}$, $\phi_m = 1$ mm and $\phi_0 = 10$ mm were used for the mill discharge model (see Figure 3). The diameter of the cylindrical sump was 6.0 m. A PI controller was used to control the slurry level in order to maintain it at the setpoint of 4 m despite external disturbances that may occur. The nominal steady-state mass flow rate of the diluting water going to the sump was 120 t/h. For the hydrocyclone model, the parameters a_j and b_j in the probability function model (see Equation (17)) for each density class were obtained when the water mass fraction, $C^5_{H_2O}$, in the hydrocyclone feed stream was 0.3 and 0.4, respectively. Then, the parameters of the probability function model as a linear function of water concentration for the two density classes were determined as

$$\begin{aligned} a_1 &= 18.75C^5_{H_2O} - 2.6765 \\ b_1 &= -0.64C^5_{H_2O} + 0.0571 \end{aligned} \tag{21}$$

$$\begin{aligned} a_2 &= 18.664C^5_{H_2O} - 2.6352 \\ b_2 &= -0.9854C^5_{H_2O} + 0.008 \end{aligned} \tag{22}$$

The water entrainment by the coarser particles in the underflow stream leaving the hydrocyclone was assumed to be 5% of the total mass flow rate of the ore particles.

3.2 Steady-state conditions

From the initial values given to all process variables, the system of equations was solved until the process converged to its nominal steady state. The steady-state size distributions of the ore particles in the mill are presented in Figure 7. It shows the mass fraction in the mill was maximum for particles of size 10 mm (i.e. mill grate size) for both density classes. The particle mass fractions increased with increasing particle sizes until the maximum at 10 mm was reached and then progressively decreased to very small values at 100 mm. The particle size distributions in the mill greatly depended on the breakage rate constants associated to the size and density classes. In this investigation, the breakage rate constants for particles of density class 2 were larger than for particles of density class 1, such that the cumulative mass fraction of density class 2 was smaller than

that of density class 1. This implies that particles of density class 2 were more friable. The steady-state water mass holdup in the mill was 13.5 t and the total mass holdup was 155.5 t.

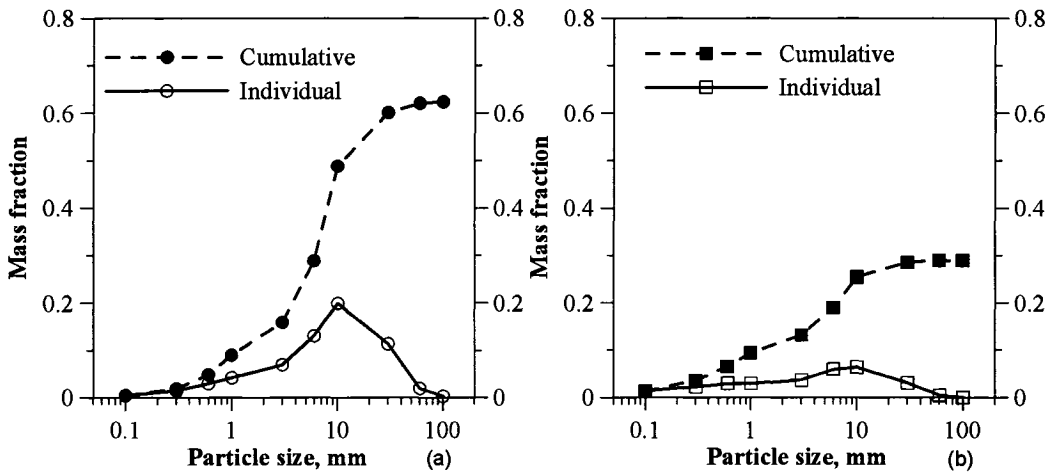


Figure 7 Particle size distributions of (a) density class 1 and (b) density class 2 in the grinding mill at process nominal steady state.

The nominal steady-state particle size distributions of the mill discharge stream are presented in Figure 8. For density class 1, the particle mass fractions increased with increasing particle size in the range 0.1 – 10 mm. For density class 2, the particle mass fractions increased with increasing particle size in the range 0.1 – 0.6 mm and remained almost constant in the range 0.6 mm – 10 mm. No particles larger than 10 mm were present due to the inherent classification performed by the grates. Akin to the mill holdup, the particles of density class 1 had a larger cumulative mass fraction than particles of density class 2. The steady-state total mass flow rate of the mill discharge stream was 1360.6 t/h, including 269.6 t/h of water.

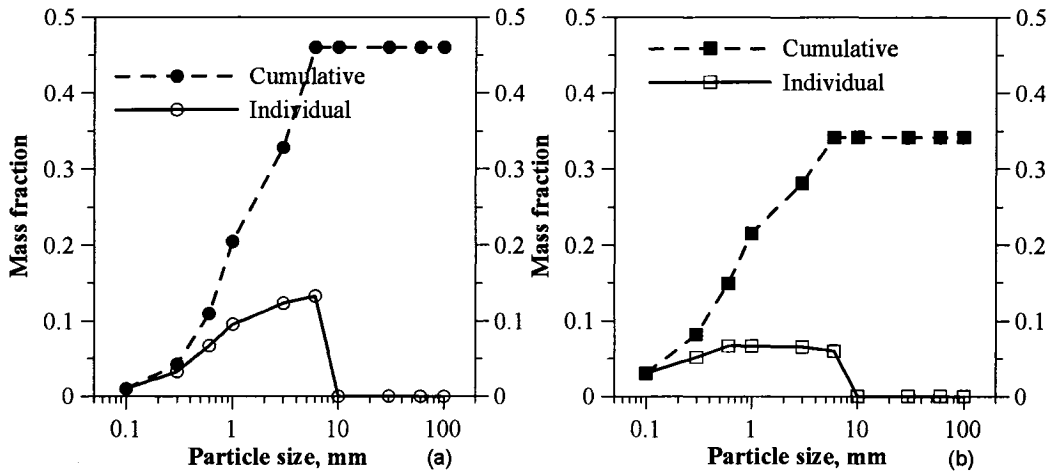


Figure 8 Particle size distributions of (a) density class 1 and (b) density class 2 in the mill discharge stream at process nominal steady state.

The steady-state size distributions of the ore particles in the sump were identical to those of the mill discharge stream, except that their mass fractions were smaller because of the addition of the diluting water into the sump. The total mass holdup in the sump was 291.1 t. The total mass flow rate of the outlet stream leaving the sump was 1480.6 t/h, of which 389.6 t/h was for water.

The steady-state ore particle size distributions in the underflow stream leaving the hydrocyclone are presented in Figure 9. It shows the particle mass fractions increased with increasing particles sizes. For particle sizes in the range 0.1 – 0.6 mm, the mass fraction for both density classes was almost identical. For particle sizes in the range 0.6 – 10 mm, the mass fractions for higher density class, i.e. density class 1, were larger than that for lower density class, i.e. density class 2. As the result, the cumulative mass fraction of particles belonging to density class 1 was higher than that of particles belonging to density class 2. The total mass flow rate of the underflow stream was 620.6 t/h, including 29.6 t/h for water. The mass flow rate of ore particles in the underflow stream brought 54 % of mass flow rate of ore particles in the feed flow stream to the hydrocyclone.

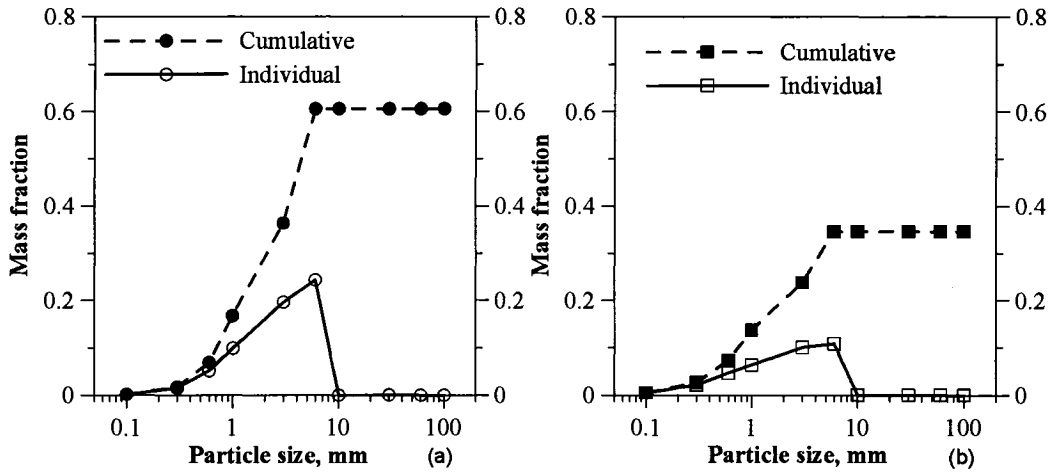


Figure 9 Particle size distributions of (a) density class 1 and (b) density class 2 in the underflow stream of the hydrocyclone at process nominal steady state.

The results of steady-state size distributions of the ore particles in the hydrocyclone overflow stream, i.e. in the product stream of the grinding process, are presented in Figure 10. It shows that the mass fraction increased with increasing particle sizes until a maximum, at 1.0 mm for density class 1 and at 0.6 mm for density class 2, was reached. Then, the mass fraction progressively decreased to zero at 10 mm. For density class 1, the maximum particle mass fraction at 1.0 mm accounted for 27% of its cumulative mass fraction. For density class 2, the maximum particle mass fraction at 0.6 mm accounted for 25 % of its cumulative mass fraction. The populations of finer particles in the range 0.1 – 1.0 mm from both density classes dominated the product stream, representing 77 % of total mass flow rate. The total mass flow rate of the product stream was 860 t/h, of which the water flow rate accounted for 360 t/h.

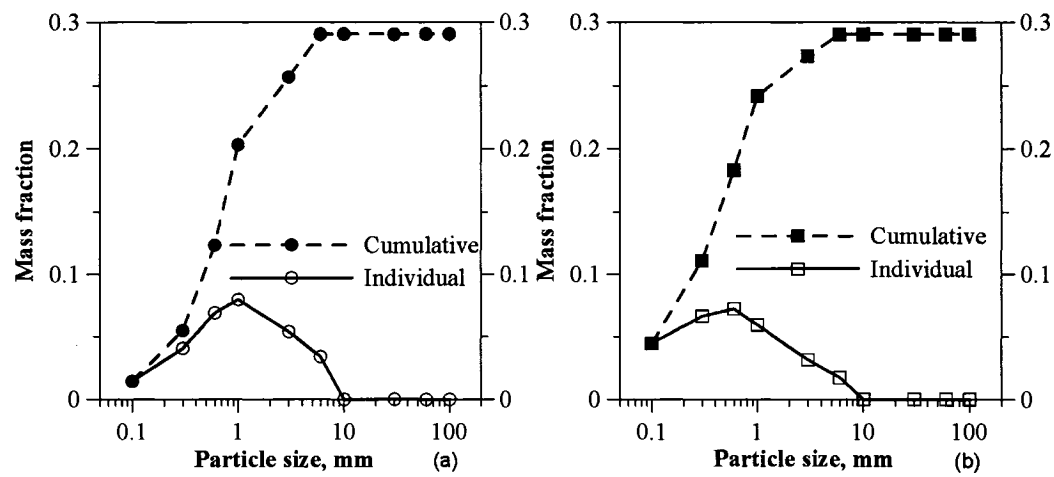


Figure 10 Particle size distributions of (a) density class 1 and (b) density class 2 in the product stream of the grinding mill process at process nominal steady state.

3.3 Dynamic simulation

3.3.1 Responses to step changes in fresh ore feed flow rate

The dynamic behavior of the grinding process was studied via simulation. Starting with the process nominal steady state, the grinding circuit was subjected to a series of step changes in the total mass flow rate of the fresh ore feed stream. The fresh ore feed rate was first increased from 500 t/h to 600 t/h, then decreased to 400 t/h before returning to the nominal fresh ore flow rate of 500 t/h. The dynamic responses of the mill holdups for particle classes from both density classes are presented in Figure 11. It shows that, as expected, the variation of the mass holdup associated to each class of particle size followed the variation of the fresh ore feed rate. The mill holdups for larger particles were more rapidly affected than for smaller particles following the disturbances. The mill holdup for the finest particles (size 0.1 mm) displayed the most sluggish responses. The dynamics of the mill holdup for each particle size appeared to behave as first order systems. Consequently, static gain and time constant for the dynamics of the mill holdup for each class of particle size were calculated and the results are presented in Table 1. It is shown that the dynamic response of the mill holdup for the 10 mm particle size had the largest static gain, while small gains were observed for the dynamic response of the smallest and largest particles. On the other hand, the time constant of the dynamic response increased from a very low value for the largest particles to a much higher value for the finest particles.

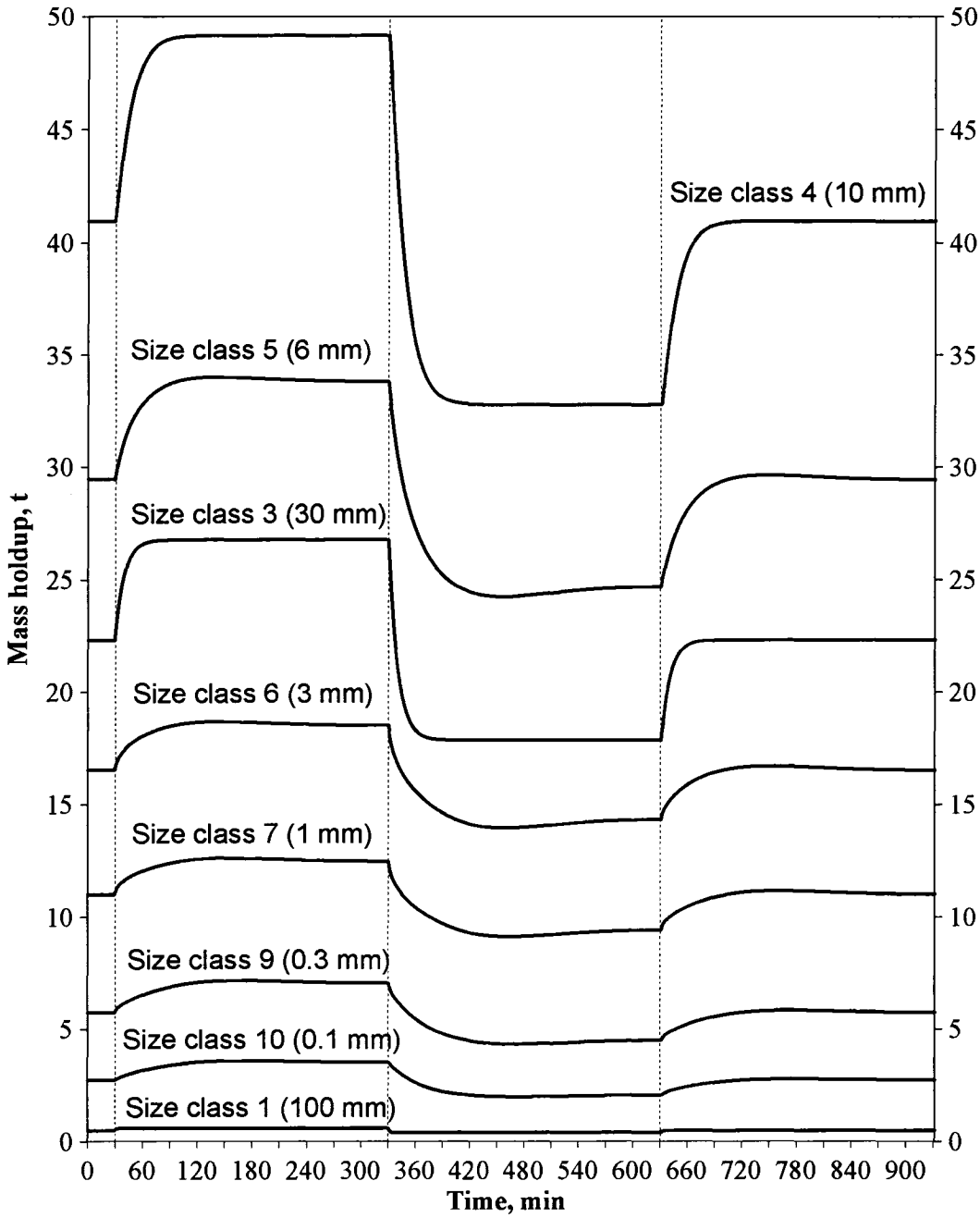


Figure 11 Dynamic responses of mass holdups in the mill for a series of step changes in the fresh ore feed flow rate.

Table 1 Static gain and time constant for the dynamic responses of mill holdups

Particle Class	Size, mm	Static gain, h	Time constant, min
1	100	0.001	1.57
2	60	0.014	3.99
3	30	0.045	9.84
4	10	0.083	17.2
5	6	0.046	22.9
6	3	0.020	27.2
7	1	0.015	33.6
8	0.6	0.016	39.9
9	0.3	0.014	39.1
10	0.1	0.009	41.8

The dynamic responses of mass flow rate for each size class in the mill discharge stream displayed similar characteristics as their corresponding holdups in the mill, except that the mass flow rates for particles larger than 10 mm were obviously absent due to the classification function performed by the grates. The dynamic responses of the sump level and mass flow rates of particles in the outlet stream of the sump are presented in Figure 12. Despite the step changes in the fresh ore feed flow rate, the PI controller was able to maintain the sump level at its setpoint by adjusting the outlet flow leaving the sump tank. Compared to the dynamics of mill discharge stream, the dynamic responses of the mass flow rates of the particles in the sump outlet stream exhibited slower dynamics due to the volume of the sump tank.

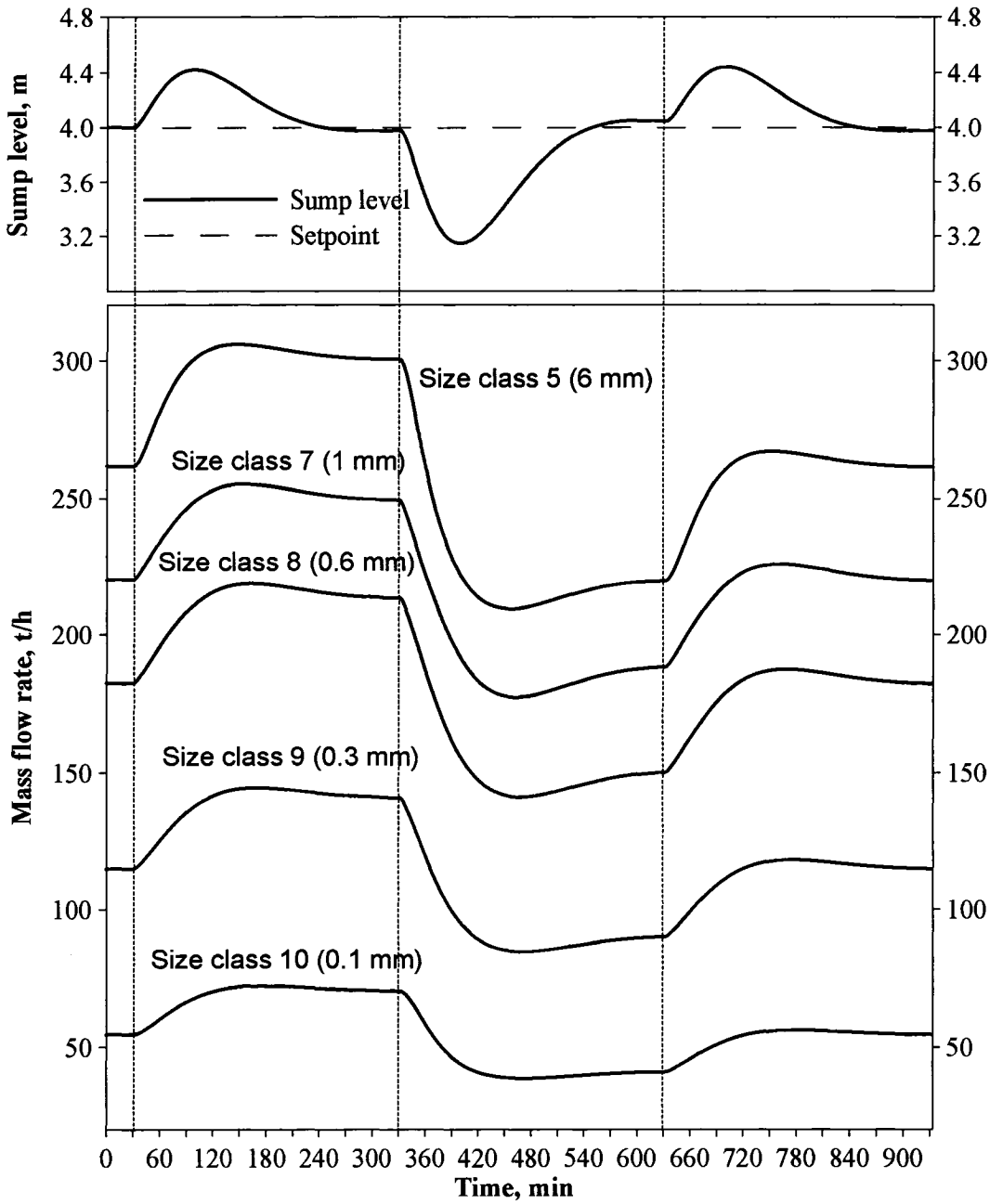


Figure 12 Dynamic responses of sump level and outlet stream for a series of step changes in the fresh ore feed flow rate.

The dynamic responses of the mass flow rates of particles in the underflow and overflow streams leaving the hydrocyclone are presented in Figures 13 and 14, respectively. The dynamic responses for larger particles were faster than for smaller particles. The dynamic responses of the mass flow rates in both streams appeared to behave like slightly

underdamped second order systems with small overshoots. It is shown that the dynamic responses of the mass flow rates for coarser particles had relatively larger gains. The mass flow rates for the coarser particles were affected more significantly by the changes in the fresh ore feed flow rate.

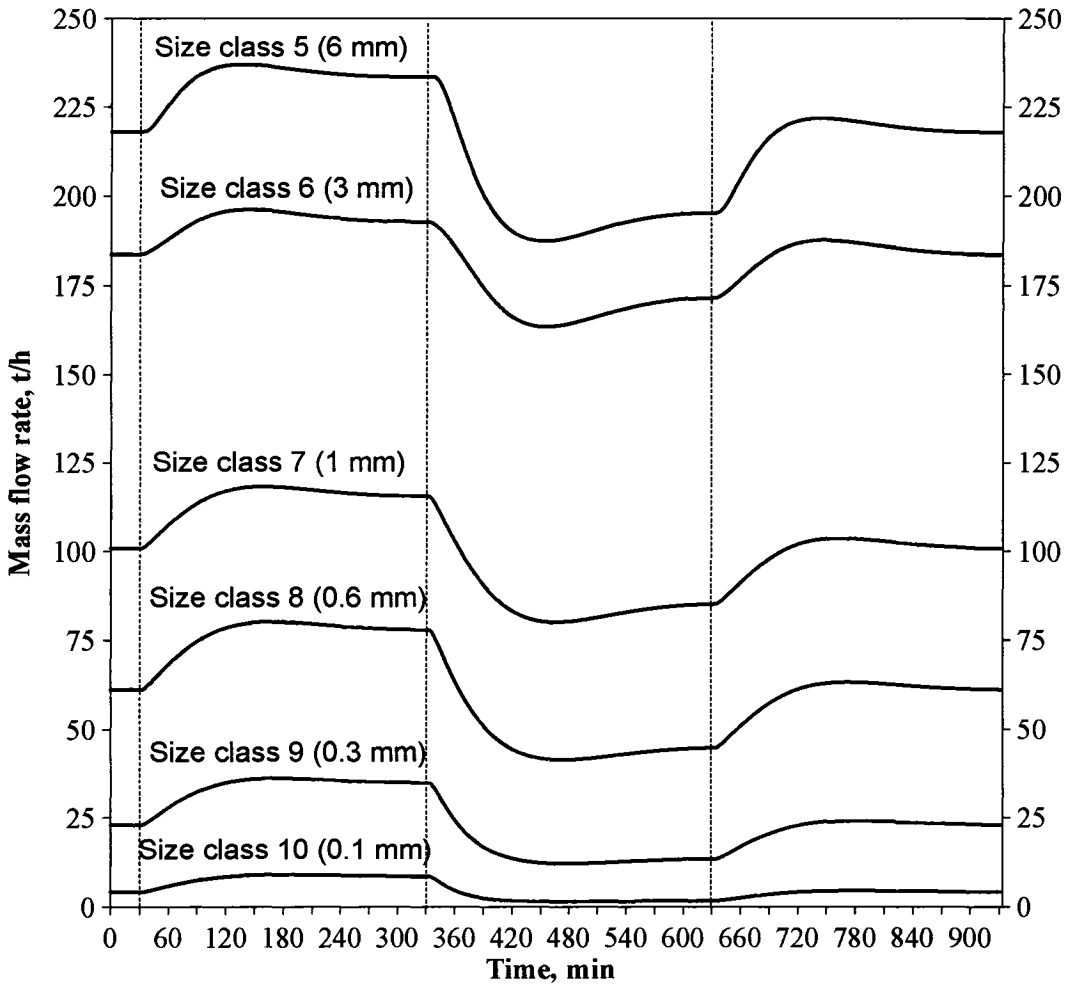


Figure 13 Dynamic responses of underflow stream leaving the hydrocyclone for a series of step changes in the fresh ore feed flow rate.

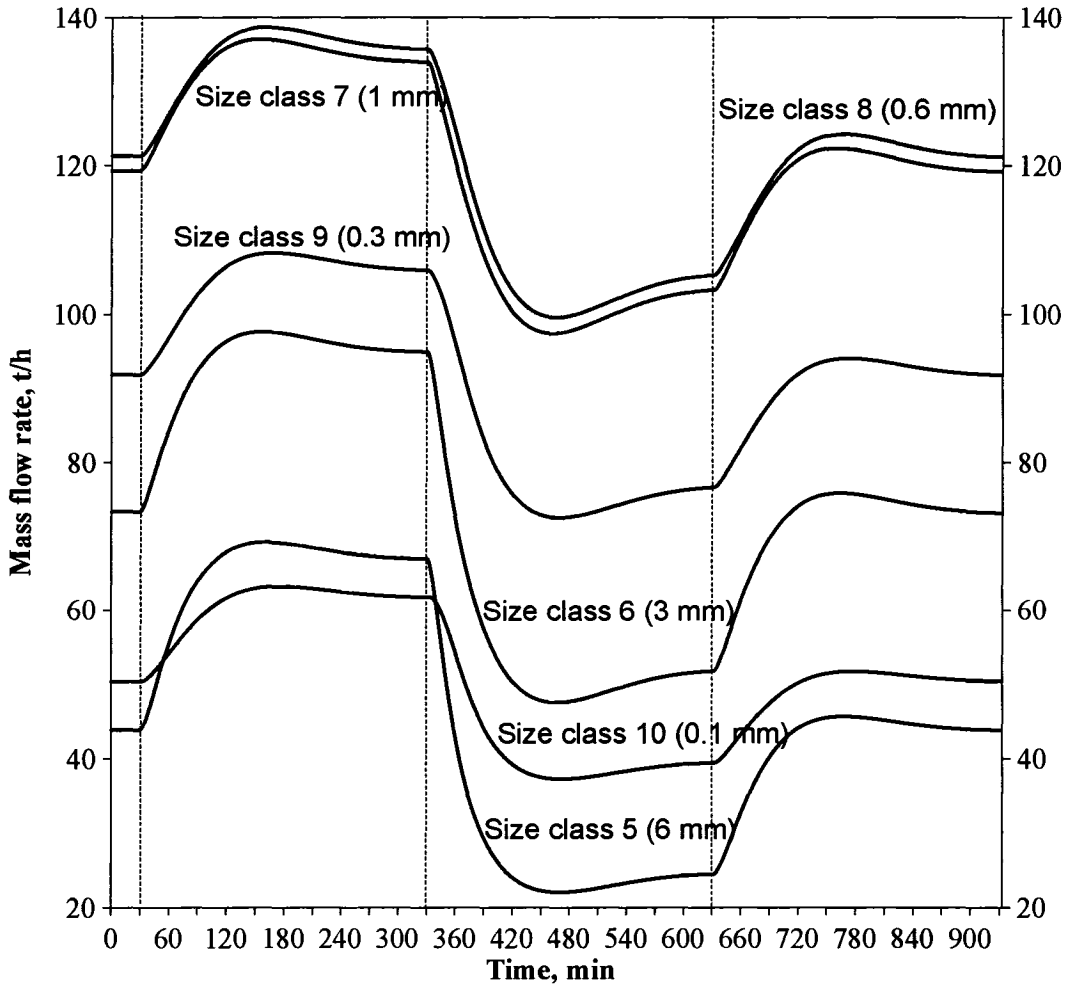


Figure 14 Dynamic responses of product stream leaving the hydrocyclone for a series of step changes in the fresh ore feed flow rate.

3.3.2 Responses to step changes in diluting water flow rate

The dynamics of the grinding process to the changes of the diluting water flow rate was also investigated. Simulation results showed that changes of diluting water flow rate had no considerable impact on the mill holdups. For the underflow stream of the hydrocyclone, the mass flow rates of coarser particles (e.g., size classes 5 and 6) increased with an increase in the diluting water flow rate, whereas the mass flow rates for finer particles (e.g., size classes 8, 9 and 10) decreased. For the overflow stream of the hydrocyclone, the mass flow rates of coarser particles decreased with increasing of the diluting water flow rate, whereas the mass flow rates of finer particles increased. An increment in the diluting water flow resulted in a larger amount of finer particles leaving

with the product stream, while a larger amount of coarser particles left with the underflow stream. The diluting water flow rate had a significant impact on the particle size distribution in the product stream. The dilution water flow rate should be considered as an important manipulated variable to control the size distribution required by floatation operations.

4. CONCLUSION

This work proposed a parsimoniously parameterized model for dynamic simulation of AG/SAG grinding processes. Compared to conventional AG/SAG models, this model has fewer parameters that can be more easily calibrated using plant data. For the hydrocyclone separator, a nonlinear model based on a sigmoid function was used. With these models, steady-state operating conditions of the grinding process were obtained. Simulation results showed particles at mill grate size dominate the mill holdup. Mass fractions of largest and smallest particles in the mill are very small. In the mill product stream, particles in the range 0.6 – 1.0 mm dominate the mass flow rate. The dynamics of the grinding process was evaluated when the process was subjected to external disturbances in fresh ore feed flow rate and diluting water flow rate. Simulation results showed that the fresh ore feed flow rate has more significant impact on mass holdups for particle sizes close to the mill grate classification size, while it has negligible impact on the largest particles. The dynamic responses of the mill holdups for finer particles are significantly more sluggish than for coarser particles. The flow rates of coarser particles in the mill product stream are more seriously affected than the flow rates of finer particles following the disturbance of fresh ore feed flow rate. With an increase in fresh ore feed flow rate, the mass fractions of coarser particles increase while the mass fractions of finer particles decrease. Disturbances in the diluting water flow rate have no considerable impact on mill holdups, but affect the ore particle size distribution in the product stream. Future work is suggested to focus on model parameterizations using plant data. Then, these dynamic models could be used for online data reconciliation as well as for process control purposes.

REFERENCES

Morrell, S., W.M. Finch, T. Kojovic and H. Delboni Jr. (1996). Modeling and simulation of large diameter autogenous and semi-autogenous mills. *Int. J. Miner. Process.* **44-45**, 289-300.

Morrell, S. and Y.T. Man (1997). Using modeling and simulation for the design of full scale mill circuit. *Minerals Engineering*, **10**, 1311-1327.

Morrell, S. (2004). A new autogenous and semi-autogenous mill model for scale-up, design and optimization. *Minerals Engineering*, **17**, 437-445.

Nikkhah, K. and C. Anderson (2001). Role of simulation software in design and operation of metallurgical plants: a case study. SME Annual meeting, Feb. 26-28, 2001, Denver, Colorado.

Radhakrishnan, V.R. (1999). Model based supervisory control of a ball mill grinding circuit. *Journal of Process Control*, **9**, 195-211.

Rajamani, K. and J. Herbst (1991). Optimal control of a ball mill grinding circuit – I. Grinding circuit modeling and dynamic simulation. *Chem. Eng. Sci.*, **46**, 861-870.

Valery, W. and S. Morrell (1995). The development of a dynamic model for autogenous and semi-autogenous grinding. *Minerals Engineering*, **8**, 1285-1297.

ANNEX C

LIST OF SIMULATION SOFTWARE

Process simulations performed in this work were carried out using coded FORTRAN program. Source code for each simulator can be found at the location in the CD room:

FOLDER 1: DISTILLATION COLUMN SIMULATOR IN CHAPTER II
FOLDER 2: CYLINDRICAL STORAGE TANK SIMULATOR IN CHAPTER III
FOLDER 3: DISTILLATION COLUMN SIMULATOR IN CHAPTER III
FOLDER 4: DISTILLATION COLUMN SIMULATOR IN CHAPTER IV
FOLDER 5: DISTILLATION COLUMN SIMULATOR IN CHAPTER V
FOLDER 6: CYLINDRICAL STORAGE TANK SIMULATOR IN CHAPTER VI
FOLDER 7: SPHERICAL STORAGE TANK SIMULATOR IN CHAPTER VI
FOLDER 8: DISTILLATION COLUMN SIMULATOR IN CHAPTER VII
FOLDER 9: AUTOASSOCIATIVE NEURAL NETWORK TRAINING PROGRAM
FOLDER 10: AG/SAG PROCESS SIMULATOR

A copy of the CD is available from either Professor David D. McLean or Professor Jules Thibault in the Department of Chemical Engineering at the University of Ottawa.