

ADAPTIVE NONLINEAR DYNAMIC DATA RECONCILIATION AND GROSS ERROR DETECTION

by

Mazyar B. Laylabadi

**B.Sc. in Electrical Engineering, Iran University of Science and
Technology, 2002**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF**

Master of Science in Engineering

In the Graduate Academic Unit of Department of Electrical and Computer
Engineering

Supervisor(s): Dr. James Taylor, Electrical and Computer Engineering Department
Examining Board: Dr. Ricky Dubay, Mechanical Engineering Department
Dr. Julian Meng, Electrical and Computer Engineering Department
Dr. Chris Diduch, Electrical and Computer Engineering Department

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

May, 2006

©Mazyar B. Laylabadi, 2006

Dedication

Dedicated to my lovely parents, my dear brother and my gorgeous fiancée. Without their patience, support, understanding and most of all love, this work would have not been possible.

Abstract

Data reconciliation (DR) is a well-known method in on-line process control engineering aimed at estimating the true values of corrupted measurements under constraints [11]. It is crucial to detect and identify gross errors first or in some cases simultaneously with DR. There have been some new gross error detection (GED) and statistical model identification approaches developed recently and combined with the original nonlinear dynamic DR (NDDR) method in order to remove the negative effects of gross errors. Among these methods there are very few approaches which address the situation where a statistical model is not available. However, they cannot handle either nonlinearity or dynamic behavior of the processes [1].

In the first step, one of the most applicable NDDR methods introduced by Liebman et al. [7], is studied in this thesis. This technique was designed and tested for inputs that undergo step changes and are otherwise constant. Next, an adaptive NDDR (ANDDR) method is proposed that includes the application to processes with an unknown statistical model. A novel GED method is developed as well and combined with the ANDDR algorithm. A new smart tracking system is also combined, to ameliorate the problem of delay seen in both the original and later NDDR methods. Finally, an extension is made to include applications with slowly and smoothly varying inputs. The proposed package has been successfully applied to the simulated continuously stirred tank reactor (CSTR) model cited commonly in the literature. As a more complex case study, this package is also tested and implemented on a simulated jacketed CSTR (JCSTR) model. The proposed package with its smart tracking features is suggested for use in distributed control systems (DCSs) or chemical process control to improve process monitoring.

Acknowledgements

I would like to first and most thank Prof. James H. Taylor, my research supervisor, whose endless assistance and patient encouragement helped me make progress in this thesis. In many challenges without his advice and insightful criticisms I would have never been able to succeed. I owe him an immense amount of appreciation.

This project is supported by Atlantic Canada Opportunities Agency (ACOA) under the Atlantic Innovation Fund (AIF) program. I gratefully acknowledge that support and the collaboration of the Cape Breton University (CBU), the National Research Council (NRC) of Canada, and the College of the North Atlantic (CNA).

I would also like to express my sincere appreciation to the faculty and staff of the Department of Electrical and Computer Engineering at University of New Brunswick for their patience and continuous encouragement. Special thanks go to Mrs. Denise Burke whose help started while before my masters program and consistently continued till the very end of my graduation.

I extend my appreciation to my colleagues, especially Mr. Atalla Sayda and Ms. Maira Omana for their helpful comments and advice. I also thank Mr. John Creighton and Mr. Khaled Akida for their suggestions and support during my masters program.

Table of Contents

Dedication	ii
Abstract	iii
Acknowledgments	iv
Table of Contents	viii
List of Tables	ix
List of Figures	xiii
Abbreviations	xiv
1 Introduction	1
1.1 Data reconciliation background	2
1.2 Gross error detection background	2
1.3 Objectives and thesis outline	3
1.3.1 Objectives	4
1.3.2 Thesis outline	4

2	Nonlinear Dynamic Data Reconciliation (NDDR)	6
2.1	Introduction	6
2.2	General NDDR problem formulation	6
2.3	Solution strategy	8
2.3.1	Moving horizon window	8
2.3.2	Discretization	8
2.4	Case study	11
3	Smart Tracking System and NDDR	15
3.1	Proposed smart tracking method	15
3.2	Case study	17
4	Novel Gross Error Detection	25
4.1	Introduction	25
4.2	Proposed GED concepts	26
4.3	Case study	27
4.3.1	Observation	28
5	Adaptive NDDR + GED and Smart Tracking System	34
5.1	Introduction	34
5.2	ANDDR + GED and covariance matrix estimation	35
5.3	Case study	38
5.3.1	Observation	38
6	Adaptive NDDR + GED for Slowly Varying Inputs	46

6.1	Introduction	46
6.2	ANDDR + GED for slowly varying inputs	47
6.3	Case study - ramp input tracking	48
6.3.1	Gross-error-free ramp input tracking	48
6.3.2	Gross errors and ramp input tracking	49
6.4	Case study - sinusoidal input tracking	52
6.4.1	Gross-error-free sinusoidal input tracking	54
6.4.2	Gross errors and sinusoidal input tracking	55
6.5	Conclusion	57
7	Implementation on JCSTR Model	60
7.1	JCSTR physical model	60
7.2	Application results	62
7.2.1	Gross-error-free case	62
7.2.2	Presence of gross errors	66
8	Thesis Observations	70
8.1	Conclusions	70
8.2	Future work	71
	Bibliography	74
	Appendix A	75
	Appendix B	76

List of Tables

3.1	Original NDDR noise reduction statistics when one step exists	24
3.2	Smart NDDR noise reduction statistics when one step exists	24
3.3	Original NDDR noise reduction statistics when two steps exist	24
3.4	Smart NDDR noise reduction statistics when two steps exist	24
6.1	ANDDR-R noise reduction statistics for ramp case	49
6.2	ANDDR-R noise reduction statistics for sinusoidal case	57
7.1	ANDDR noise reduction statistics for JCSTR model for gross-error-free case	63
8.1	CSTR model constants	75
8.2	JCSTR model constants	76

List of Figures

2.1	History horizon for NDDR	9
2.2	NDDR flowchart	10
2.3	NDDR estimation results for the first output, A	13
2.4	NDDR estimation results for the second output, T	13
2.5	NDDR estimation results for the first input, A_0	14
2.6	NDDR estimation results for the second input, T_0	14
3.1	Comparison of original NDDR and proposed smart tracking system estimation for the first input, A_0	18
3.2	Comparison of original NDDR and proposed smart tracking system estimation for the second input, T_0	18
3.3	Comparison of original NDDR and proposed smart tracking system estimation for the first output, A	19
3.4	Comparison of original NDDR and proposed smart tracking system estimation for the second output, T	19
3.5	Comparison of original NDDR and proposed smart tracking system estimation for the first input, A_0 , when two steps occur	20
3.6	Comparison of original NDDR and proposed smart tracking system estimation for the first output, A , when two steps occur	21

3.7	Comparison of original NDDR and proposed smart tracking system estimation for the second output, T , when two steps occur	22
3.8	Comparison of original NDDR and proposed smart tracking system estimation for the second input, T_0 , when two steps occur	23
4.1	GED + smart NDDR flowchart	28
4.2	Comparison of original NDDR and proposed GED combined with smart NDDR estimation for the first output, A	29
4.3	Comparison of original NDDR and proposed GED combined with smart NDDR estimation for the second output, T	30
4.4	Comparison of original NDDR and proposed GED combined with smart NDDR estimation for the first input, A_0	31
4.5	Comparison of original NDDR and proposed GED combined with smart NDDR estimation for the second input, T_0	32
5.1	ANDDR + GED flowchart	37
5.2	Comparison of original NDDR estimation with proposed ANDDR and GED combined with smart tracking system for the first output, A	39
5.3	Comparison of original NDDR estimation with proposed ANDDR and GED combined with smart tracking system for the second output, T	40
5.4	Comparison of original NDDR estimation with proposed ANDDR and GED combined with smart tracking system for the first input, A_0	41
5.5	Comparison of original NDDR estimation with proposed ANDDR and GED combined with smart tracking system for the second input, T_0	42
5.6	σ estimates obtained from ANDDR estimation for the first output, A	44
5.7	σ estimates obtained from ANDDR estimation for the second output, T	44

5.8	σ estimates obtained from ANDDR estimation for the first input, A_0	45
5.9	σ estimates obtained from ANDDR estimation for the second input, A	45
6.1	First input, A_0 , estimation. (gross-error-free)	50
6.2	Second input, T_0 , estimation. (gross-error-free)	50
6.3	First output, A , estimation. (gross-error-free)	51
6.4	Second output, T , estimation. (gross-error-free)	51
6.5	First input, A_0 , estimation. (with gross errors)	52
6.6	Second input, T_0 , estimation. (with gross errors)	52
6.7	First output, A , estimation. (with gross errors)	53
6.8	Second output, T , estimation. (with gross errors)	53
6.9	First input, A_0 , estimation. (gross-error-free)	55
6.10	Second input, T_0 , estimation. (gross-error-free)	55
6.11	First output, A , estimation. (gross-error-free)	56
6.12	Second output, T , estimation. (gross-error-free)	56
6.13	First input, A_0 , estimation. (with gross errors)	58
6.14	Second input, T_0 , estimation. (with gross errors)	58
6.15	First output, A , estimation. (with gross errors)	59
6.16	Second output, T , estimation. (with gross errors)	59
7.1	Jacketed continuous stirred tank reactor	61
7.2	First output, V , estimation. (gross-error-free)	63
7.3	Second output, T , estimation. (gross-error-free)	64

7.4	Third output, T_j , estimation. (gross-error-free)	64
7.5	First input, F_{in} , estimation. (gross-error-free)	65
7.6	Second input, F_{out} , estimation. (gross-error-free)	65
7.7	Third input, T_{in} , estimation. (gross-error-free)	66
7.8	First output, V , estimation. (with gross errors)	67
7.9	Second output, T , estimation. (with gross errors)	67
7.10	Third output, T_j , estimation. (with gross errors)	68
7.11	First input, F_{in} , estimation. (with gross errors)	68
7.12	Second input, F_{out} , estimation. (with gross errors)	69
7.13	Third input, T_{in} , estimation. (with gross errors)	69

List of Symbols and Abbreviations

Symbols

y	True values
\tilde{y}	Corrupted measurements
\hat{y}	Reconciled estimates
t_c	Current time
y_j	True values of y at discrete time t_j
\tilde{y}_j	Measured (corrupted) values at discrete time t_j
\hat{y}_j	Reconciled estimates at discrete time t_j
V	Variance-covariance matrix
A_0	CSTR feed concentration
T_0	CSTR feed temperature
A	CSTR output concentration
T	CSTR output temperature
σ	Standard deviation
$\hat{\sigma}$	Estimated standard deviation
\tilde{m}	Measurements mean
\hat{m}	Estimated mean
d	Distance
H	Window width
T	Threshold
U	Overall heat transfer coefficient
V	JCSTR tank volume
T	JCSTR temperature inside the tank
T_j	JCSTR temperature inside the jacket
F_{in}	JCSTR mixture inflow
F_{out}	JCSTR mixture outflow
T_{in}	JCSTR temperature of the mixture feed

Abbreviations

ANDDR	Adaptive nonlinear dynamic data reconciliation
CSTR	Continuous stirred tank reactor
DCS	Distributed control system
DMI	Dynamic model identification
DR	Data reconciliation
EKF	Extended Kalman filter
GED	Gross error detection
JCSTR	Jacketed continuous stirred tank reactor
MHW	Moving horizon window
NDDR	Nonlinear dynamic data reconciliation
NLP	Nonlinear programming
OCFE	Orthogonal collocation on finite elements
WSE	Weighted squared error

Chapter 1

Introduction

Measurement data are always subject to inaccuracy due to imprecision of the measurement equipment, malfunction of instruments, poor sampling and noise. Flawed information cannot be used effectively for state estimation and process control; therefore, it is crucial to have an algorithm capable of improving the accuracy of measured data with respect to actual data together with estimating of unmeasured parameters. This process is called “Data Reconciliation” (DR) [7].

Measurements might contain three possible types of error: (I) small random errors which are typically zero-mean and normally distributed (Gaussian); (II) systematic biases which occur when measurement devices provide consistently flawed data due to inaccurate calibration or installation; or (III) gross errors which are usually non-random errors causing measurements to have almost no relation with the true values. Gross errors can be due to malfunctioning sensors (e.g., data drop-out) or can be totally process-related such as process leaks [7]; here we only consider isolated outliers such as one-time data drop-outs.

Normally, it is presumed that only type (I) errors exist when DR algorithms are considered. If error types (II) or (III) are present, then a separate gross error detection

(GED) method must be developed.

1.1 Data reconciliation background

DR is a well-known method in process control aimed at estimating the true values of the corrupted measurements taking into account constraints on dynamic behavior, material and energy balance, etc.

The DR problem was first introduced by Kuehn and Davidson [5] for linear steady state models. There has been a great deal of research conducted in the area of steady state and linear processes, while nonlinear dynamic DR (NDDR) has received less attention [3]. On the other hand, as far as engineering processes are concerned they often operate dynamically in highly nonlinear regions where traditional methods such as the Kalman filter or extended Kalman filter (EKF) may be ineffective [7].

The necessity of developing NDDR methods was proposed by Liebman and Edgar [6], and the advantages of using nonlinear programming (NLP) over traditional steady state DR methods were demonstrated. In the next step Liebman et al. [7] developed their main NDDR algorithm. Their approach was based on simultaneous optimization and solution techniques where efficient state estimation was performed. In chapter 2 their approach and methodology are described.

1.2 Gross error detection background

As discussed earlier, usually it is presumed that the corrupted measurements are free of gross errors or, more specifically, that they only contain zero-mean random noise. Gross errors, however, exist normally in processes and it is crucial to detect and

identify them first or in some cases simultaneously with DR.

There was no GED and identification methodology included in the NDDR approach of Liebman et al. [7], but, since then, to complete their work, there have been some approaches which are capable of detection of gross errors, or, more specifically, identification of gross errors as well. For instance, Soderstrom et al. [9] proposed an approach to simultaneously tackle the problem of GED and identification together with DR. Abu-el-zeet et al. [4] proposed a combined method of bias and outlier identification in dynamic DR where a history of all previous bias detection and identification methods is briefly presented.

There have been studies to address GED and the estimation of the measurement error covariance matrix but either applied only to linear processes or limited to stationary processes [1]. Although several authors have stated the need for covariance estimation for DR, none of them has proved the effectiveness of using this matrix in DR except for Alici [1] who demonstrated its necessity and briefly discussed the effects that covariance matrix estimation has on DR. She also addressed the combination of dynamic model identification (DMI) with NDDR.

1.3 Objectives and thesis outline

The method presented in this thesis is suitable for most engineering processes such as found in the oil and gas industry. The two significant features of handling nonlinearities and dynamic systems in this algorithm help to be more applicable in reality. The addition of methods for adaptation also make it more practical. This algorithm will be incorporated, in future, as part of a system in the PAWS¹ project where the data collected from wireless sensors will be reconciled.

¹Petroleum Applications of Wireless Systems (PAWS) project funded by ACOA under the Atlantic Innovation Fund (AIF).

1.3.1 Objectives

The objectives of this thesis are as follows:

1. Studying and implementing the original NDDR algorithm presented by Liebman et al. [7].
2. Developing a smart tracking system to tackle the problem of delay seen in both the original and later versions of NDDR method.
3. Developing a novel GED algorithm to combine with the NDDR algorithm.
4. Combining a novel statistical model identification method with NDDR to make it an adaptive NDDR (ANDDR) algorithm that handles the applications where the statistical model is not available.
5. Introducing a novel package including a combination of proposed ANDDR, GED and smart tracking system algorithms.
6. Extending the proposed package to applications with slowly and smoothly varying inputs.
7. Demonstration of each proposed method on a simulated continuous stirred tank reactor (CSTR) model.
8. Demonstration of the whole package on a simulated model of a jacketed continuous stirred tank reactor (JCSTR) with more dynamics complexity.

1.3.2 Thesis outline

In chapter 2 the original NDDR problem formulation and the solution strategy is presented. In chapter 3 the smart tracking system is proposed which removes the delay

seen in the results of original and existing NDDR methods. Next, in chapter 4 a novel GED approach is proposed and combined with the original NDDR. In chapter 5 the basic theory of ANDDR, which is proposed as an enhancement, is presented. Then, this ANDDR is combined with the proposed GED and identification algorithm. In chapter 6 the proposed package is extended to include the applications with slowly and smoothly varying inputs. In chapter 7 the proposed package is tested and implemented on a JCSTR model. Finally, in chapter 8 conclusions and future work are discussed.

Chapter 2

Nonlinear Dynamic Data Reconciliation (NDDR)

2.1 Introduction

As mentioned in section 1.1, a general NDDR problem was formulated by Liebman et al. [7]. This problem formulation is presented in the first section of this chapter. Next, the solution strategy adopted by Liebman et al. [7] is outlined.

2.2 General NDDR problem formulation

A general NDDR formulation may be outlined as follows [7]:

$$\min_{\hat{y}} \phi(\tilde{y}, \hat{y}; \sigma) \tag{2.1}$$

subject to:

$$f\left(\frac{d\hat{y}(t)}{dt}, \hat{y}(t)\right) = 0 \quad (2.2)$$

$$h(\hat{y}(t)) = 0 \quad (2.3)$$

$$g(\hat{y}(t)) \geq 0 \quad (2.4)$$

Here the corrupted measurements \tilde{y} and reconciled estimates \hat{y} include both state variables and input variables. The first constraint (equation (2.2)) represents the process dynamics (often formulated as $\frac{d\hat{y}}{dt}=f(\hat{y})$), the second constraint (equation (2.3)) may describe energy and/or material balance, and the third (equation (2.4)) may impose process variable limits. For more details one can refer to the paper by Liebman et al. [7].

For most applications the objective function is weighted squared error (WSE):

$$\phi(\tilde{y}, \hat{y}(t); \sigma) = \sum_{j=0}^c \frac{1}{2} (\hat{y}(t_j) - \tilde{y}_j)^T V^{-1} (\hat{y}(t_j) - \tilde{y}_j), \quad (2.5)$$

where t_c is the current time, \tilde{y}_j measured (corrupted) values, \hat{y}_j the reconciled estimates at discrete time t_j , and V is the variance-covariance matrix where each diagonal element V_{ii} is σ_i^2 .

“As in the classical steady-state DR, the optimal estimates are those as close as possible (in the least-squares sense) to the measurements [true values] such that the model equations are satisfied exactly ” [7].

2.3 Solution strategy

In this section the solution strategy to solve the NLP problem of equations (2.1) to (2.4) is first demonstrated. Next, the discretization method used in this thesis is discussed.

2.3.1 Moving horizon window

The solution adopted here is a moving horizon approach which enables the user to utilize all the information at hand (process measurements) from start up to the current time [7].

The moving horizon window (MHW) approach presented here has the advantage of reduced optimization problem size together with giving the user the benefit of having only one tuning parameter, the window horizon H . This second point is obvious, in comparison with other nonlinear approaches such as the EKF where more tuning parameters need to be adjusted. Another advantage is its capability of handling constraints such as equalities and inequalities, whereas other approaches such as the EKF cannot handle the constraints. Figure 2.1 shows the basic idea of the MHW approach for NDDR. In this method after collecting the process measurements up to t_c , ϕ is optimized over the horizon from $t_c - H$ to t_c , the current time. Then one \hat{y} is saved and the procedure is repeated at the next time step [7].

2.3.2 Discretization

In order to solve the NLP problem of equations (2.1) to (2.4), we need to discretize the nonlinear model presented as the first constraint (equation (2.2)). The method adopted by Liebman et al. [7] is orthogonal collocation on finite elements (OCFE). In

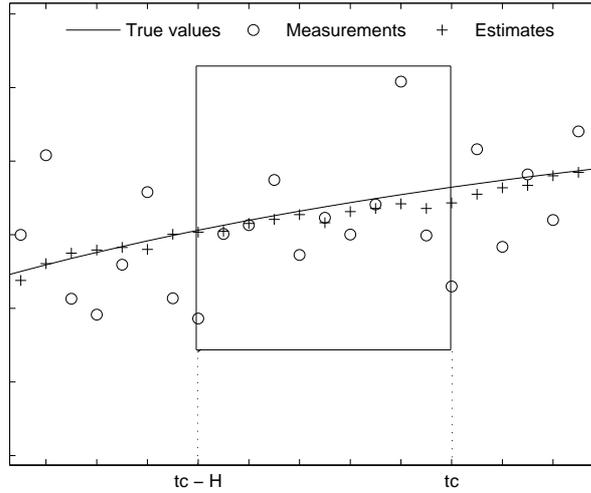


Figure 2.1: History horizon for NDDR

this thesis, however, the fourth order Runge-Kutta method has been chosen to simulate and discretize the model, as this approach has less complexity than the OCFE method with possibly better accuracy. In other words, $f(\frac{dy}{dt}, y)$ is solved numerically over the window horizon and y_j obtained by sampling this solution. Once the discretization is implemented, equations (2.1) to (2.5) can be rewritten as the following NLP problem:

$$\min_{\hat{y}} \sum_{i=0}^{n_i+n_s} \eta_i \sum_{j=c-H}^c \left(\frac{\hat{y}_{ij} - \tilde{y}_{ij}}{\sigma_i} \right)^2, \quad (2.6)$$

subject to:

$$f\left(\frac{d}{dt}\hat{y}, \hat{y}\right) = 0 \quad (2.7)$$

$$h(\hat{y}) = 0 \quad (2.8)$$

$$g(\hat{y}) \geq 0 \quad (2.9)$$

where $f(\hat{y})$, $h(\hat{y})$ and $g(\hat{y})$ now represent the constraints obtained through discretization, η is a vector of weights and n_i and n_s are the numbers of inputs and states (outputs), respectively.

Figure 2.2 is a flowchart that describes step by step the procedure of NDDR application [1].

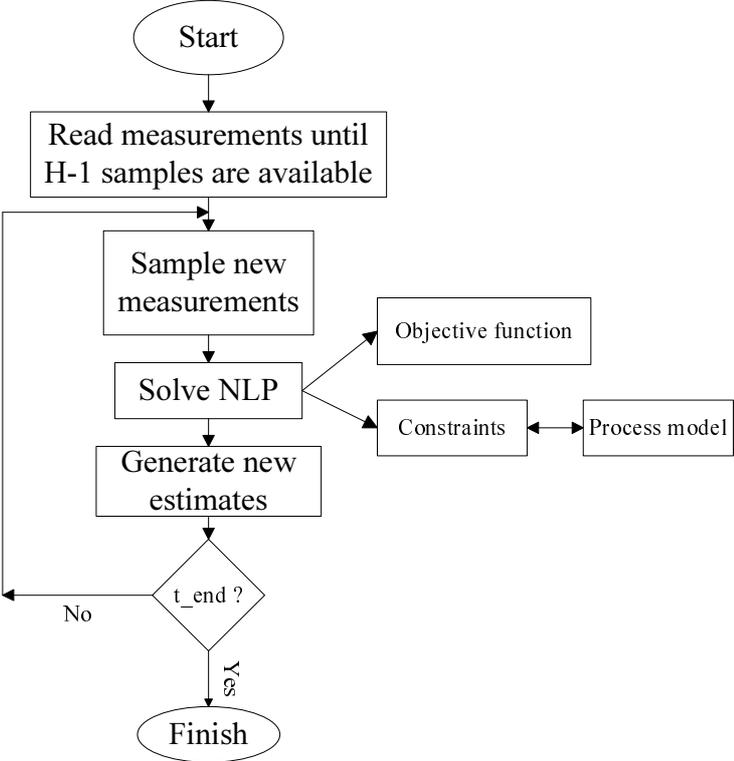


Figure 2.2: NDDR flowchart

2.4 Case study

In this thesis a simulated CSTR model studied commonly in the literature is chosen. In order to produce the comparable results with those of Liebman et al. [7], the same assumptions and parameters values for the model are used here. The normalized model can be presented as follows:

$$\frac{dA}{dt} = \frac{q}{v}(A_0 - A) - \alpha_d k A \quad (2.10)$$

$$\frac{dT}{dt} = \frac{q}{v}(T_0 - T) + \alpha_d \frac{\Delta H_r A_r}{\rho C_p T_p} k A - \frac{UA}{\rho C_p V}(T - T_c) \quad (2.11)$$

$$k = k_0 \exp\left(\frac{-E_A}{T T_r}\right) \quad (2.12)$$

where the input stream feed concentration A_0 and feed temperature T_0 are input variables and concentration A and temperature T are output variables. There are two simple constraints on both input and output variables as follows:

$$0 \leq A, A_0 \leq 20.0 \quad (2.13)$$

$$0 \leq T, T_0 \leq 10.0 \quad (2.14)$$

The values of other constants in this model are presented in Appendix A, table 8.1. One can refer to the paper by Liebman et al. [7] for more details on this CSTR model.

In this case study the two inputs and two states (outputs) are being estimated, assuming that no gross error exists. Measurements were simulated by creating the measurement noise which is assumed to be Gaussian with σ equal to 0.05 and zero mean. The time step is assumed to be 2 seconds and the simulation is run for 100 samples with window width of $H = 10$. Obviously, the first estimation is achieved at time step 10 where the first window of measurements is available. The results of the

NDDR application are demonstrated in figures 2.3 to 2.6, for the case where a step change in the feed concentration, A_0 , occurs at time step 30 from 6.5 to 7.5. The simulation was initialized at a steady-state operating point of $A_0 = 6.5$, $T_0 = 3.5$, $A = 0.1531$ and $T = 4.6091$.

Using the NDDR algorithm, both input and output variables are being estimated through solving the NLP problem introduced in equations 2.6 to 2.9. All four variable estimations (figures 2.3 to 2.6) show satisfactory results of the NDDR application to the simulated CSTR model. As these figures show, the transient behavior of the states due to the step change has been successfully observed, and there has been a significant noise reduction in all four reconciled variables. Note that at time step 30, when the step occurred, there is a significant delay in the estimation of the first input and consequently, slight delays in estimating the two states. This may be the main drawback of the original and existing NDDR methods. In the next chapter the smart tracking system is proposed to ameliorate the problem of delay seen in both input and output estimations.

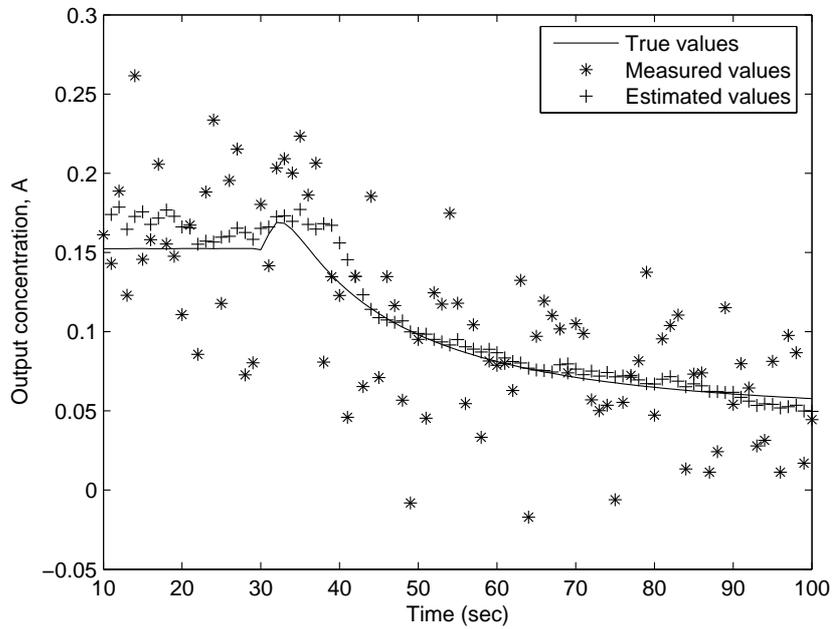


Figure 2.3: NDDR estimation results for the first output, A

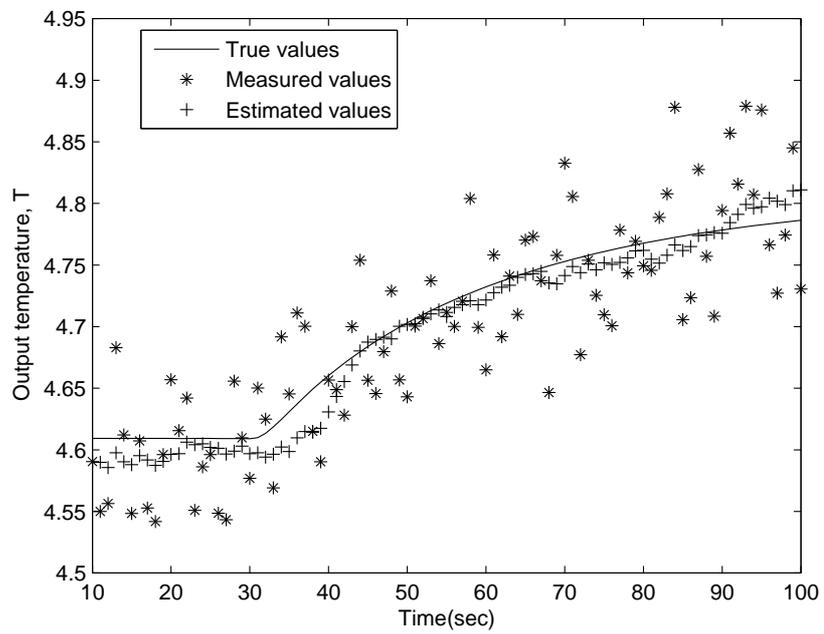


Figure 2.4: NDDR estimation results for the second output, T

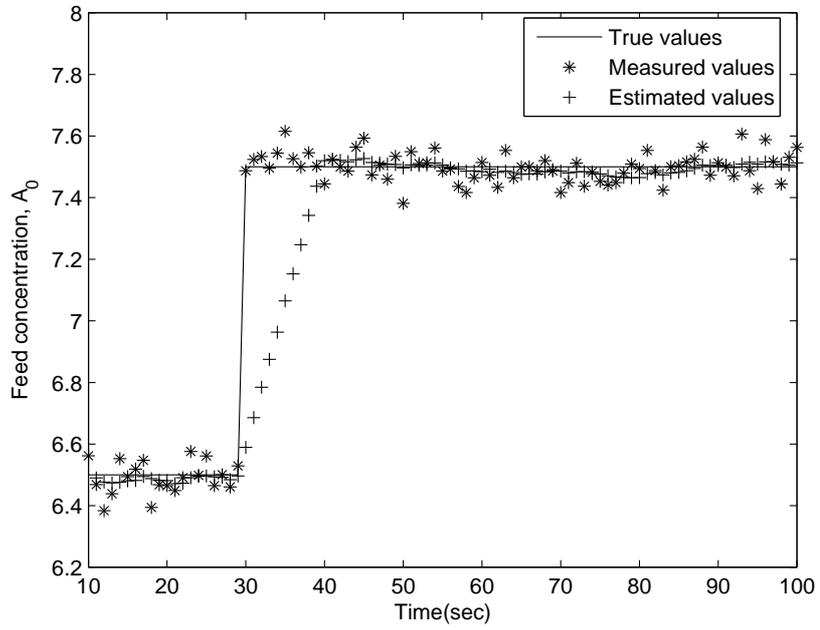


Figure 2.5: NDDR estimation results for the first input, A_0

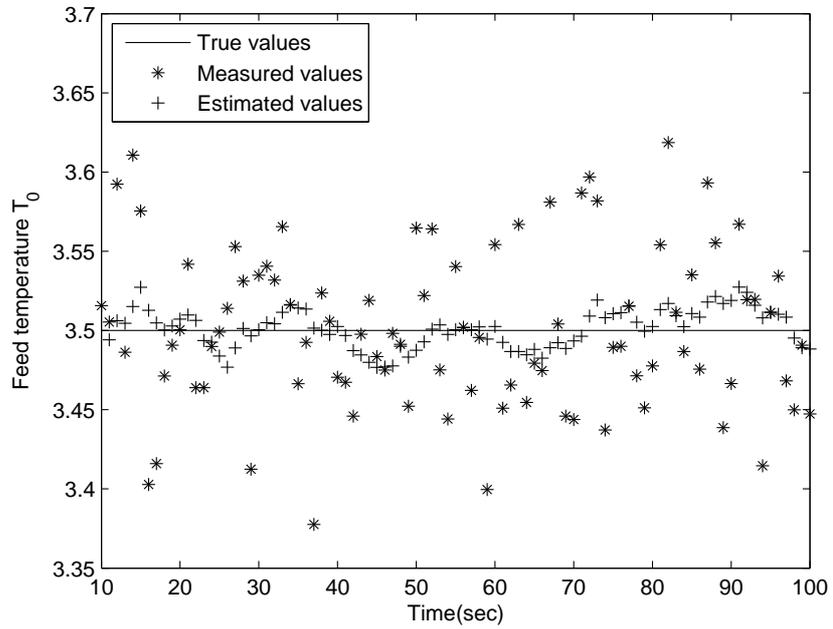


Figure 2.6: NDDR estimation results for the second input, T_0

Chapter 3

Smart Tracking System and NDDR

Studying the results of the original NDDR application in section 2.4 on the simulated CSTR model clearly shows a significant delay for the input estimation when a step occurs, which also causes some delay in the estimation of output variables. These delays, especially for the input estimation, are caused by the original assumption that the input is constant over each data window; this is the main drawback of the original NDDR algorithm and later enhanced approaches. In this chapter the concepts of the proposed new methods are first discussed, then the enhanced algorithm is applied to the same CSTR model to show the significant improvements of the estimation results.

3.1 Proposed smart tracking method

In this chapter we assume that the statistical model (i.e., σ) is available and also the measurements are gross error free. In chapters 4 and 5 we will show that these assumptions are not necessary and the proposed algorithm can be easily extended to the cases where a statistical model is not available and/or gross errors exist.

As it was discussed in section 2.4, estimation delay exists when the step change occurs in one of the input variables; as mentioned this delay is caused by the assumption that the input is constant over the entire moving window. Therefore, a complete window length is required to reach steady state at the new set-point. Obviously, the delay exists only for the first input where the step change occurs. This delay can be easily removed if we can devise a smart tracking system that enables tracking the new level of the input, when the step occurs.

Thus we need to modify and extend the original NDDR algorithm in two ways. Firstly, to enable the algorithm to detect the set-point change instantly, and secondly, to define a new input level for the time when this set-point change is sensed and let the second level track the new set-point.

To serve the first purpose, the difference $d_{c,i}$ is derived for each element of \tilde{y}_c (equation (3.1)), using the previous value of the mean, (equation (3.3)), for each time step, and it is compared with the previous σ , $\sigma_{c-1,i}$. If $|d_{c,i}|$ exceeds the threshold, as defined in equation (3.2), then the algorithm detects the set-point change and divides its input to two levels; the second level estimates the new set-point value. In this way the whole delay is removed, producing estimates that are significantly more accurate.

$$d_{c,i} = \tilde{y}_{c,i} - \hat{m}_{c-1,i} \quad (3.1)$$

$$\text{If } |d_{c,i}| > 3\sigma_{c-1,i} \text{ then } \tilde{y}_{c,i} \text{ is a set-point change} \quad (3.2)$$

$$\hat{m}_{c,i} = \sum_{j=c-H}^c \left(\frac{\tilde{y}_{i,j}}{H+1} \right) \quad (3.3)$$

In the next section the results for the input estimation show satisfactory implementation of this smart tracking system.

3.2 Case study

The same CSTR model used in the previous chapter is studied here and the new smart tracking system proposed in section 3.1 is applied. The estimation results are compared with those of the original NDDR application demonstrated in section 2.4.

Figures 3.1 to 3.4 show the comparison studies conducted to show the successful implementation of the proposed smart tracking system. As figure 3.1 shows, when the step occurs at time step 30, the reconciled values follow the set-point change instantly. The original NDDR estimation, however, needs a full window length ($H = 10$) to reach to the new set-point. Output estimations also show a slightly better estimation after the time step 30, since the inputs are more accurately estimated. Studying figures 3.3 and 3.4 show this improvement in the estimation of output concentration and temperature.

To demonstrate the smart tracking feature of the proposed algorithm, the case with two set-point steps happening at time steps 30 and 70, is also studied. First at time step 30 the feed concentration, A_0 , is stepped up from 6.5 to 7.5 and stepped down at the time step 70 from 7.5 to 4.5. Figures 3.5 to 3.8 show the comparison study of the application of the original NDDR and the proposed algorithm for this case. In each figure the complete time-history is shown, plus a zoomed section for better viewing.

Figure 3.5 clearly shows the successful implementation of the smart tracking system at both set-point changes, 30 and 70. Also figures 3.6 and 3.7 demonstrate more accurate tracking of the peak for the output estimation at time steps 30 and 70, while the original NDDR method fails to track. Figure 3.8 proves that adding this smart tracking feature does not affect the accuracy of the estimation for the second input, where we do not have any set-point changes. As this figure shows both the original NDDR and proposed algorithm have virtually the same satisfactory results. Therefore,

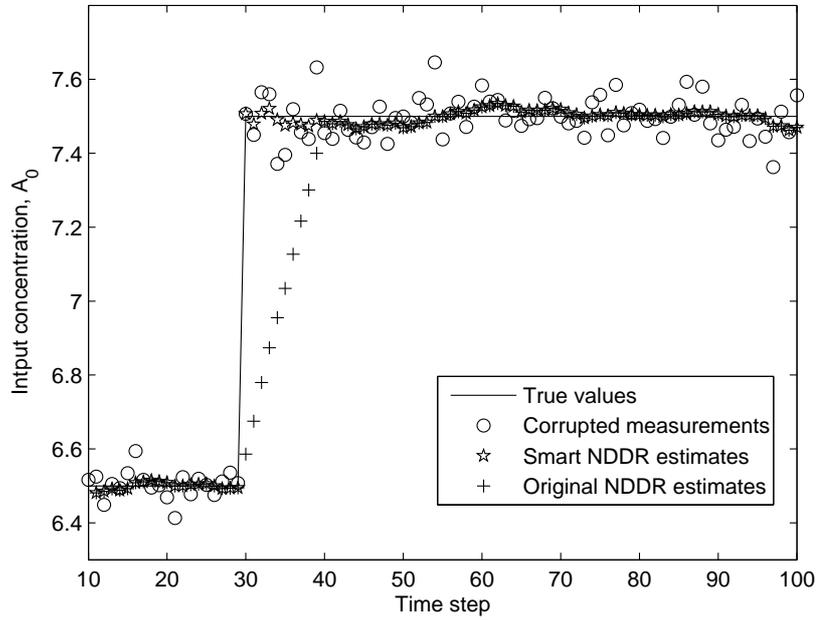


Figure 3.1: Comparison of original NDDR and proposed smart tracking system estimation for the first input, A_0

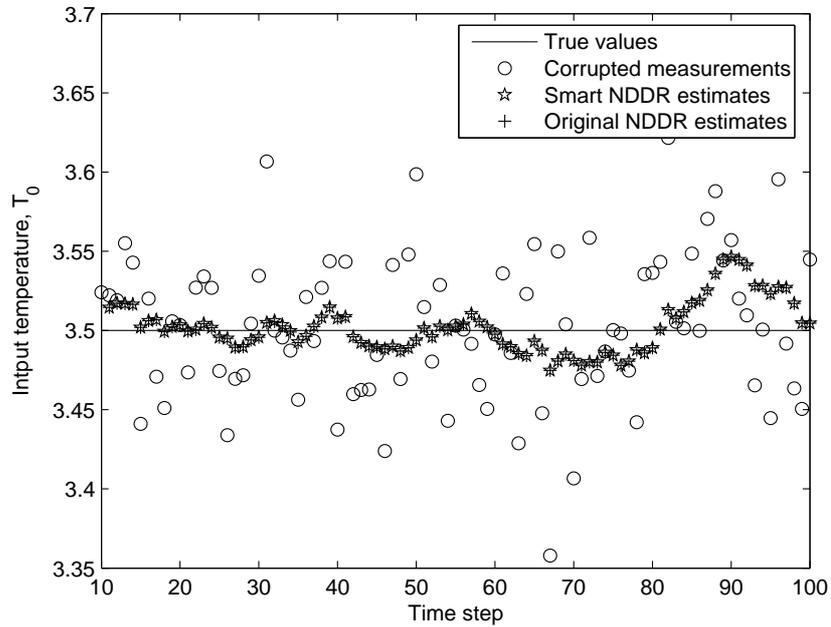


Figure 3.2: Comparison of original NDDR and proposed smart tracking system estimation for the second input, T_0

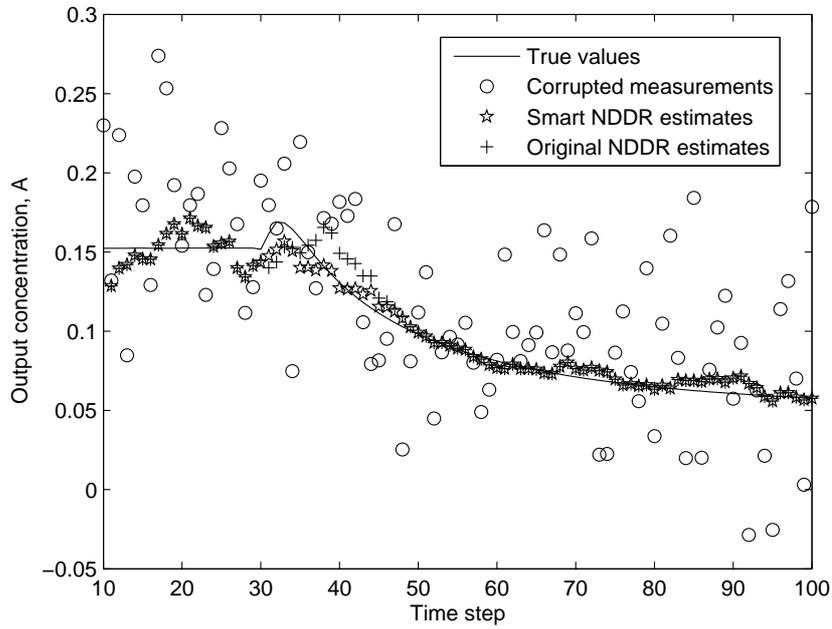


Figure 3.3: Comparison of original NDDR and proposed smart tracking system estimation for the first output, A

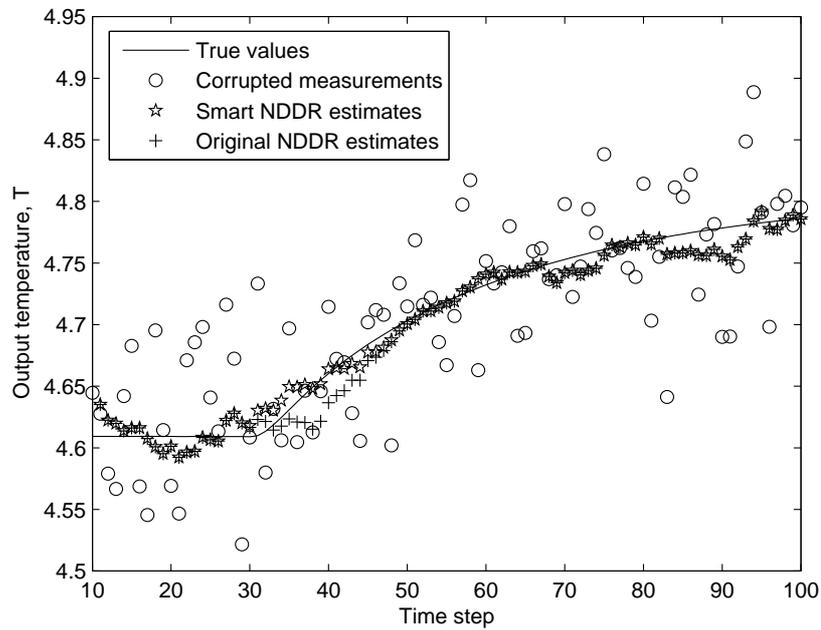


Figure 3.4: Comparison of original NDDR and proposed smart tracking system estimation for the second output, T

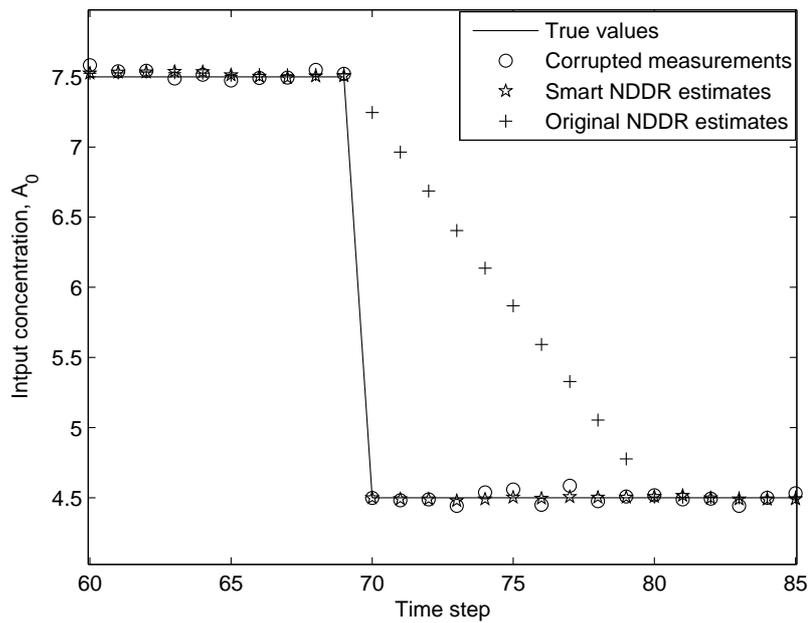
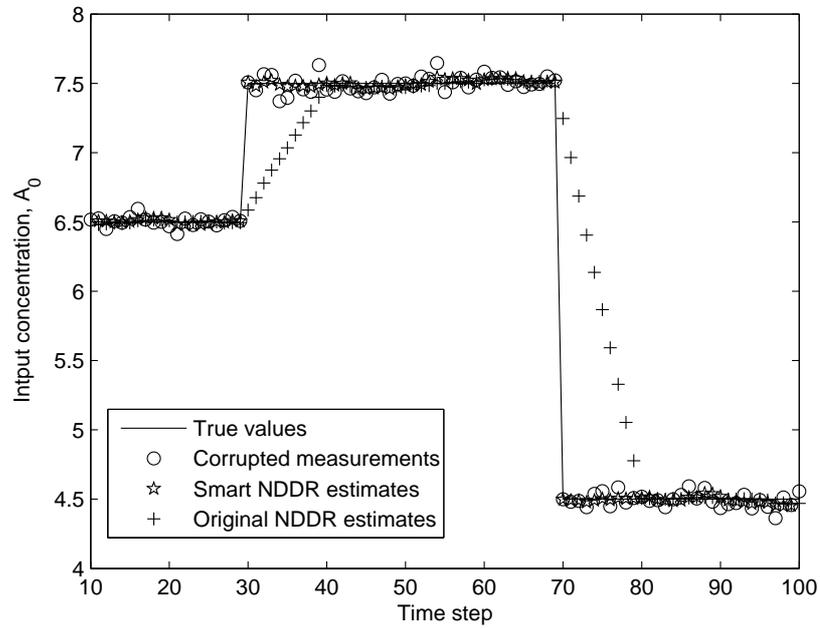


Figure 3.5: Comparison of original NDDR and proposed smart tracking system estimation for the first input, A_0 , when two steps occur

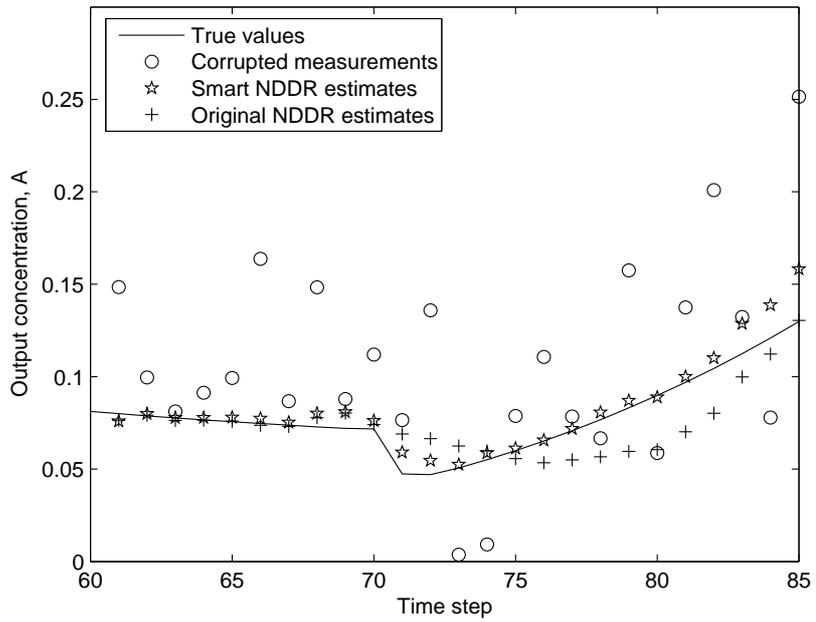
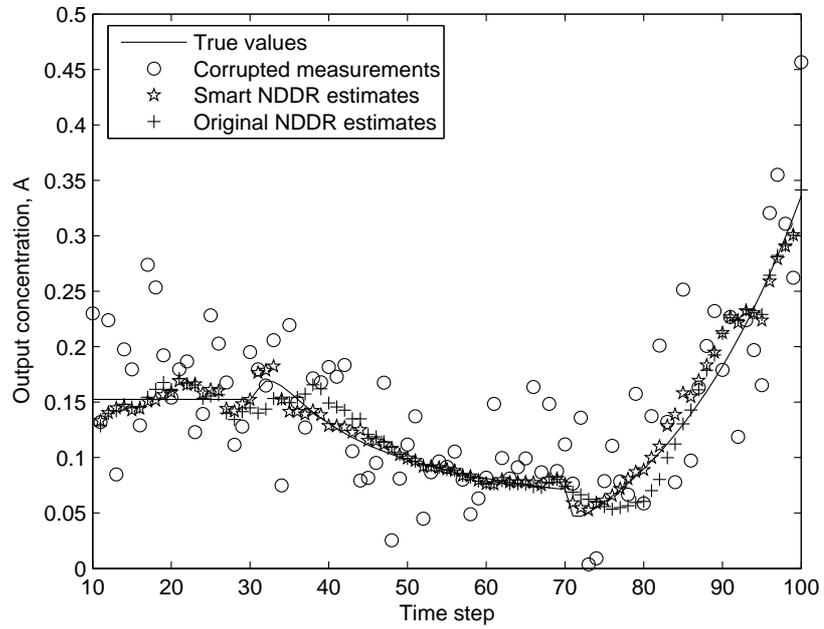


Figure 3.6: Comparison of original NDDR and proposed smart tracking system estimation for the first output, A , when two steps occur

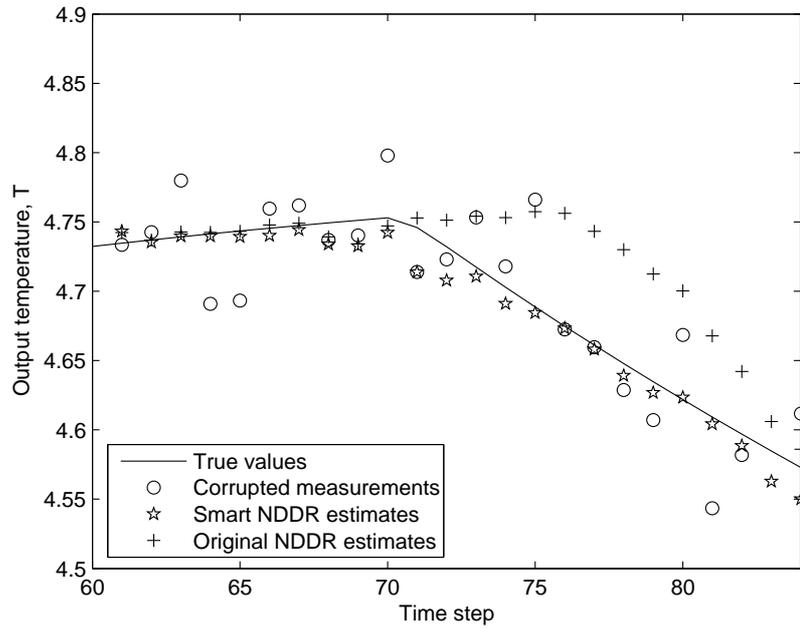
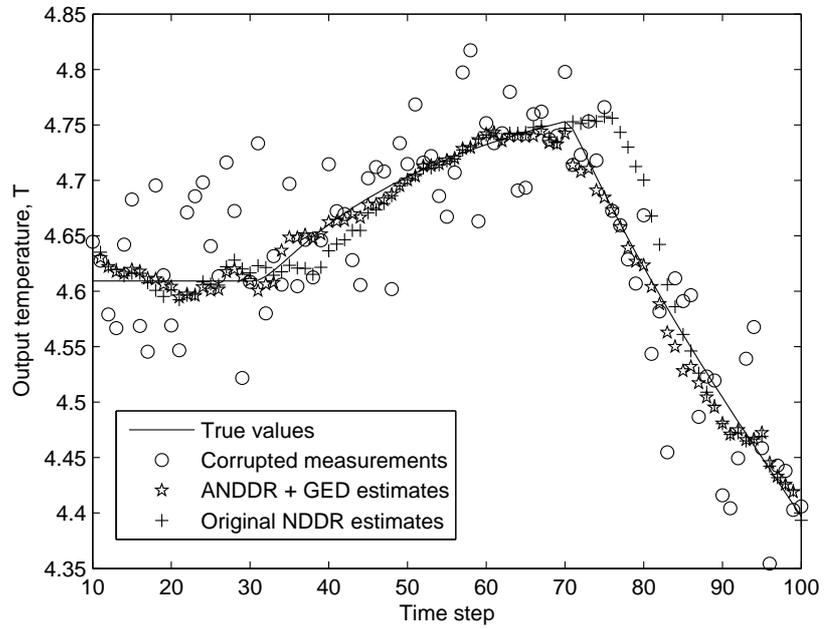


Figure 3.7: Comparison of original NDDR and proposed smart tracking system estimation for the second output, T , when two steps occur

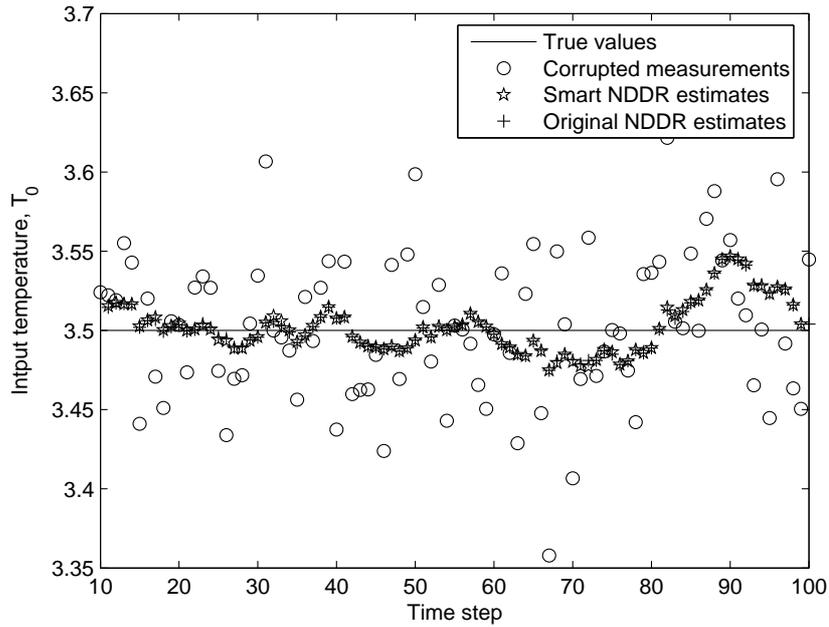


Figure 3.8: Comparison of original NDDR and proposed smart tracking system estimation for the second input, T_0 , when two steps occur

the NDDR method is successfully extended with the proposed smart tracking system and in the following chapters the addition of a GED and ANDDR methods will be discussed.

To present a better understanding of both the original and smart NDDR successful estimation, the noise reduction statistics are provided in tables 3.1 to 3.4. As these tables show, the algorithm performance has been improved by the proposed smart tracking system.

Table 3.1: Original NDDR noise reduction statistics when one step exists

Variable	Measurements σ	Estimates σ	% σ reduction
A	0.0465	0.0098	78.98
T	0.0513	0.0117	77.24
A_0	0.0513	0.1828	-
T_0	0.0466	0.0162	65.28

Table 3.2: Smart NDDR noise reduction statistics when one step exists

Variable	Measurements σ	Estimates σ	% σ reduction
A	0.0465	0.0079	83.03
T	0.0513	0.0106	79.27
A_0	0.0513	0.0160	68.79
T_0	0.0466	0.0162	65.29

Table 3.3: Original NDDR noise reduction statistics when two steps exist

Variable	Measurements σ	Estimates σ	% σ reduction
A	0.0465	0.0138	70.25
T	0.0513	0.0257	49.83
A_0	0.0513	0.5931	-
T_0	0.0466	0.0161	65.40

Table 3.4: Smart NDDR noise reduction statistics when two steps exist

Variable	Measurements σ	Estimates σ	% σ reduction
A	0.0465	0.0105	77.55
T	0.0513	0.0110	78.49
A_0	0.0513	0.0222	56.73
T_0	0.0466	0.0163	64.95

Chapter 4

Novel Gross Error Detection

4.1 Introduction

Gross errors are usually non-random errors causing measurements to have almost no relation with the true values. The gross errors considered here may be due to malfunctioning sensors (e.g., data drop-out). The main purpose of DR algorithms is to adjust the data according to the constraints. In the presence of gross errors, however, all of the adjustments may be adversely affected. Therefore, it is crucial to detect and identify the gross errors first or in some cases simultaneously with DR, and suppress or eliminate them.

As stated earlier in section 1.2 there was no GED and identification included in the NDDR approach of Liebman et al. [7], but, since then, to complete their work, there have been some approaches which are capable of detection of gross errors. Among these methods there are very few approaches which address the situation where a statistical model is not available. However, they cannot handle processes with nonlinear or dynamic behavior, which makes them unsuitable for many applications [1].

In this chapter a novel GED method is presented, assuming that a statistical model is available. Then, in the next chapter we will see how the extension to this novel GED approach will be made to handle the cases where the statistical model is not known.

4.2 Proposed GED concepts

In chapter 3 the methodology to track the step change was presented. Using the same idea, a novel GED approach is proposed. To perform GED, the difference $d_{c,i}$ is derived for each element of \tilde{y}_c (equation (4.1)), using the previous value of the mean (equation (4.3)), for each time step, and it is compared with the previous σ , $\sigma_{c-1,i}$, which is assumed to be known. If $|d_{c,i}|$ exceeds the threshold, as defined in equation (4.2), then the algorithm detects the existence of a gross error and removes it by replacing it with the previous estimate, \hat{y}_{c-1} .

$$d_{c,i} = \tilde{y}_{c,i} - \hat{m}_{c-1,i} \quad (4.1)$$

$$\text{If } |d_{c,i}| > 3\sigma_{c-1,i} \text{ then } \tilde{y}_{c,i} \text{ is an outlier} \quad (4.2)$$

$$\hat{m}_{c,i} = \sum_{j=c-H}^c \left(\frac{\tilde{y}_{i,j}}{H+1} \right) \quad (4.3)$$

Attention must be focused on points in time near set-point changes. If the algorithm is not smart, then such a change can simply be taken as a gross error. Here, a sample point where an error threshold is exceeded is designated as a possible outlier and the next point is processed to decide if a set-point change occurred or if the previous point contained an outlier (in which case it is edited out, e.g., by interpolation). This is effective under the assumption that outliers are isolated (do not happen in successive samples); if this cannot be assumed then the algorithm would have to be modified to wait several samples before the outlier/set-point-change decision can be made. This

logic causes a one or several time-step delay in a real-time setting; thus there is a trade-off between fast or robust detection. The former needs a GED algorithm based on an assumption of isolated gross errors, and the latter, which enables us to handle a number of successive gross errors, requires more detailed logic.

The proposed GED algorithm is combined with smart tracking NDDR presented in the previous chapter. Figure 4.1 is a flowchart that describes step by step the procedure of application of the proposed GED method combined with the smart NDDR [2].

4.3 Case study

The same CSTR model used in the previous chapters is studied here and the new GED algorithm is applied and combined with the smart NDDR proposed in chapter 3, and the estimation results are compared with original NDDR estimations. Here it is assumed that isolated outliers exist for each variable. The number of gross errors over the entire simulation is 5 for each variable. They have been added to the input and output measurements randomly.

In figures 4.2 to 4.5 the estimation results considering a step change at time step 30 for the first input, A_0 , from 6.5 to 7.5, and another step change from 7.5 to 4.5 at time step 70, are presented. In each figure the complete time-history is shown, plus a zoomed section for better viewing. The solid lines in these figures show the true values, circles show the corrupted measurements, stars present the proposed GED + smart NDDR estimation results and plus signs mark the original NDDR data.

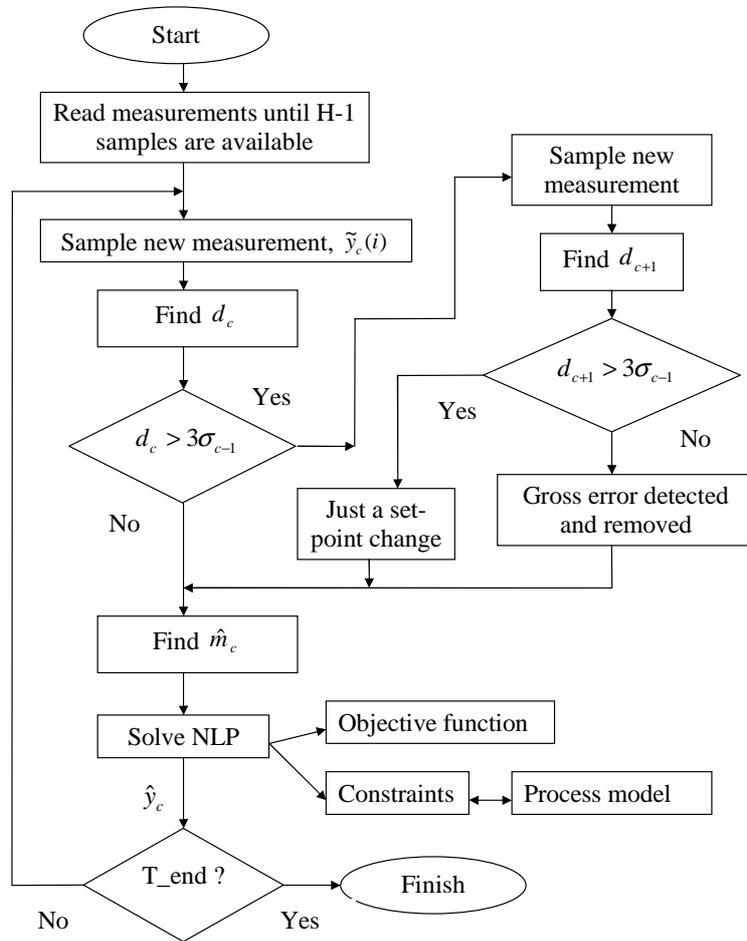


Figure 4.1: GED + smart NDDR flowchart

4.3.1 Observation

As the figures 4.2 to 4.5 show, the gross errors have been detected and successfully removed, and the estimation has not been corrupted. Observe that the outliers cause significant corruption of the NDDR data. For example, figure 4.5 shows the corrupted estimates of the original NDDR data between time steps 24 and 34. This corruption

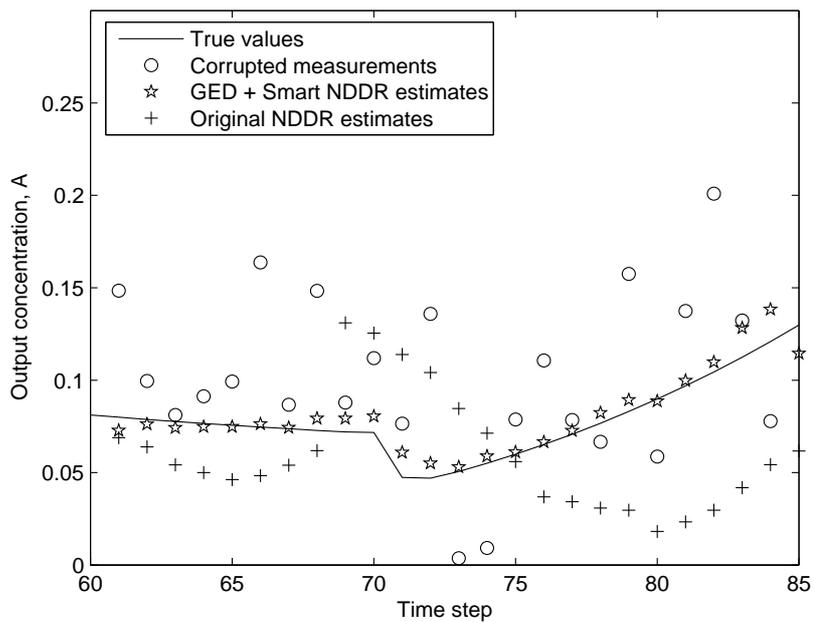
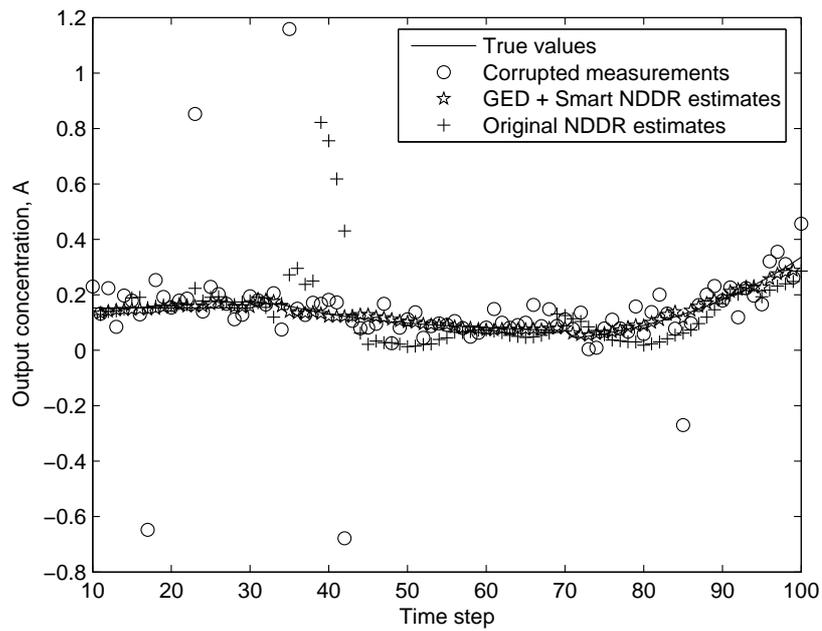


Figure 4.2: Comparison of original NDDR and proposed GED combined with smart NDDR estimation for the first output, A

is due to the existence of an outlier on T_0 at time step 24 and lasts for one window length (H samples). This is a direct effect that a gross error has caused on a variable estimation, but there might be some interaction between a gross error in one variable

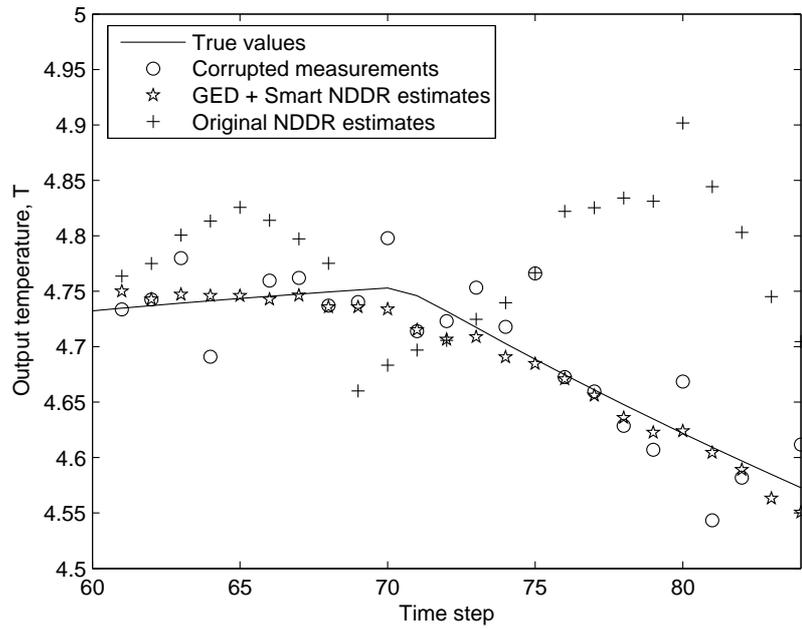
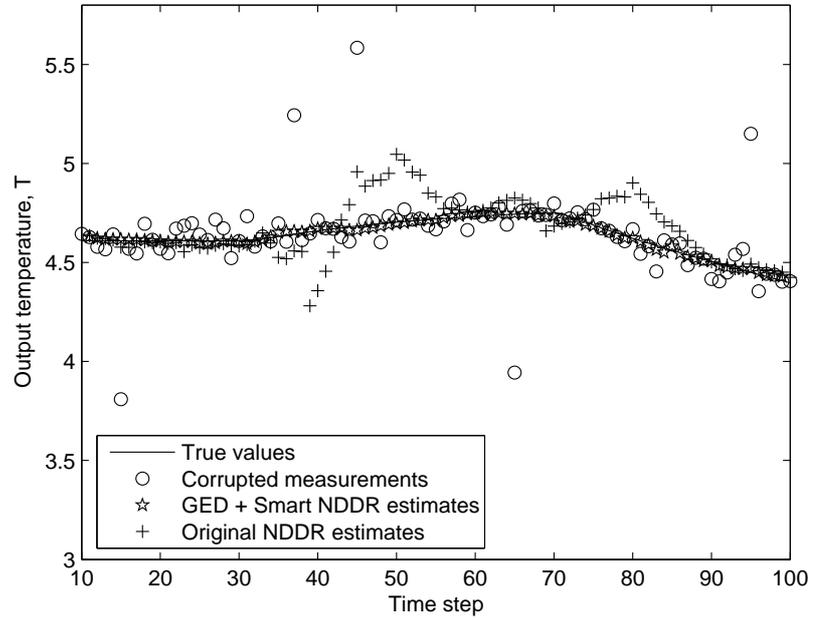


Figure 4.3: Comparison of original NDDR and proposed GED combined with smart NDDR estimation for the second output, T

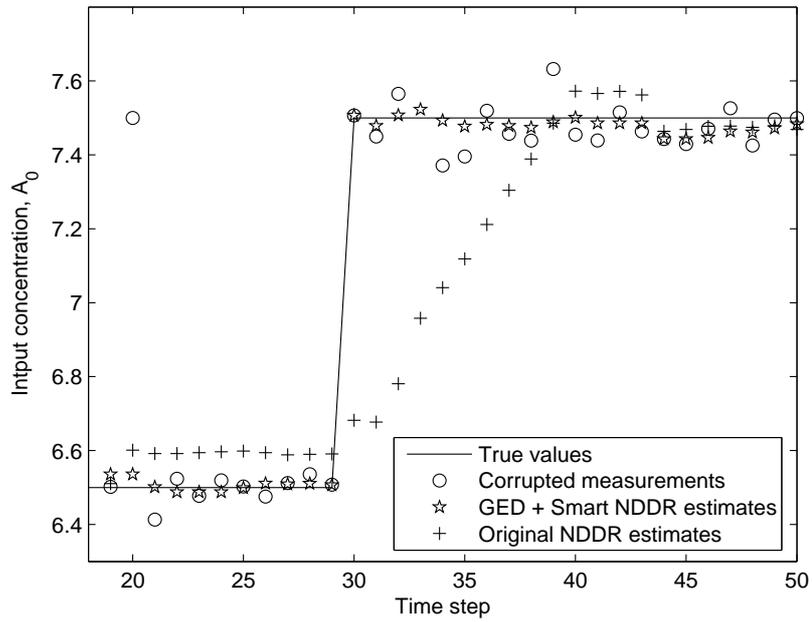
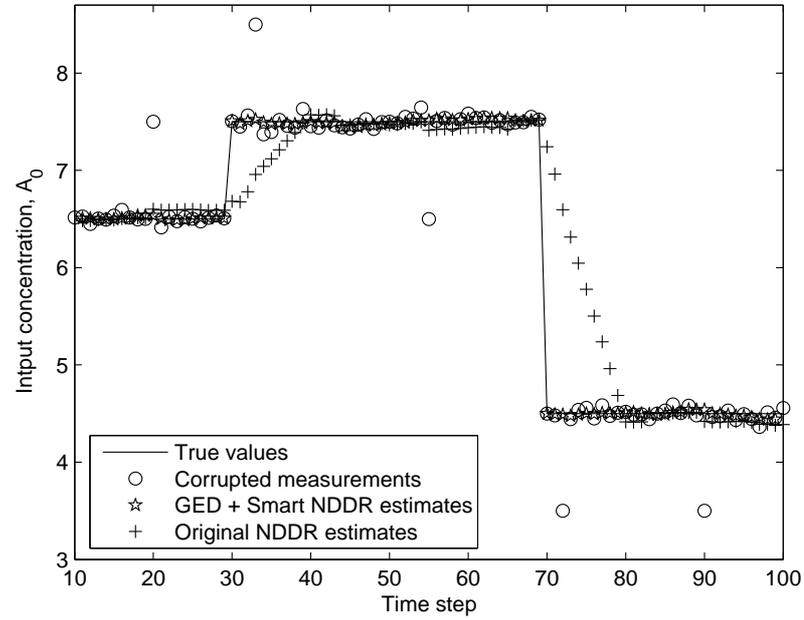


Figure 4.4: Comparison of original NDDR and proposed GED combined with smart NDDR estimation for the first input, A_0

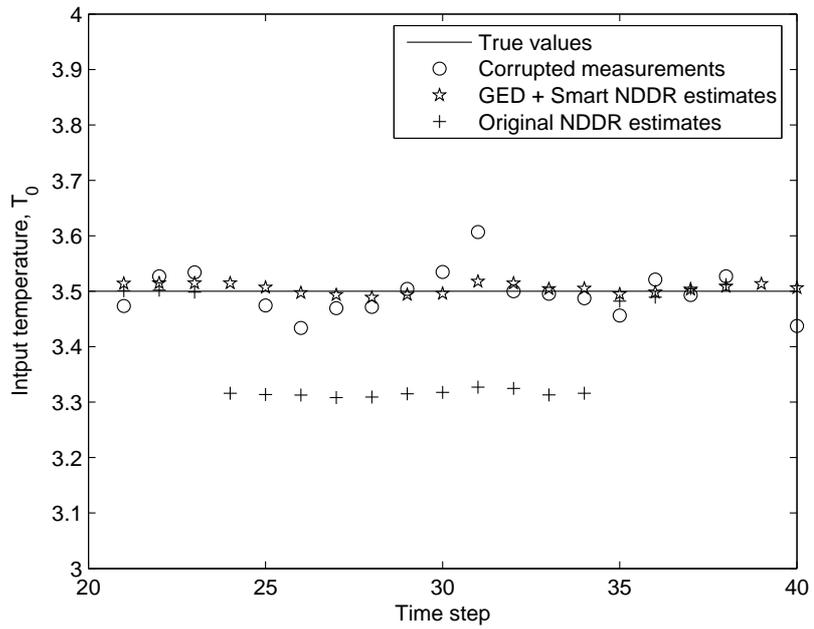
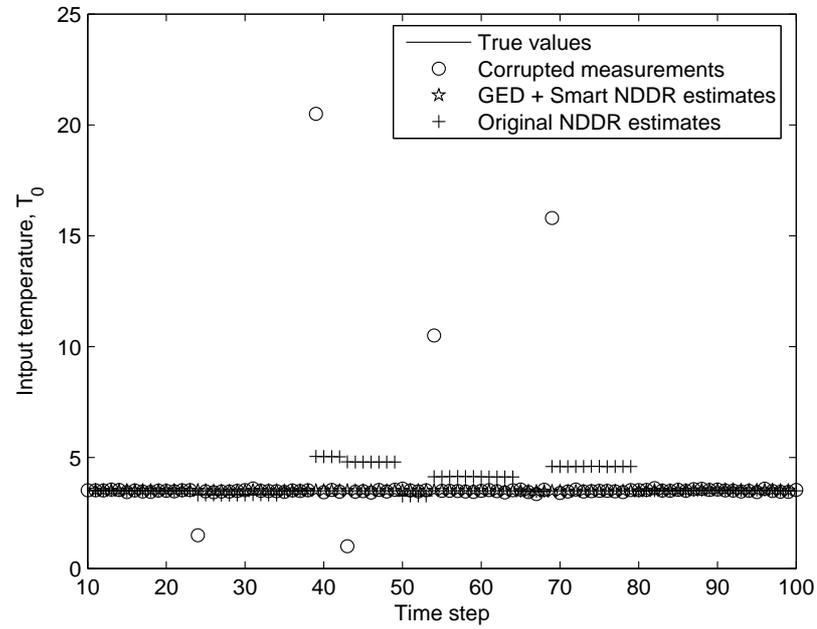


Figure 4.5: Comparison of original NDDR and proposed GED combined with smart NDDR estimation for the second input, T_0

and the estimates of the other variables as well (indirect). For instance, in figure 4.2 there is corruption of original NDDR data at time steps 38 to 41 which is due to the existence of an outlier on T_0 at time step 38. Such interactions occur when large gross errors exist on the input variables. Note that these effects, whether direct or indirect, are totally application dependent. One can see different behavior if the amplitude of gross errors and/or the place that they occur change. Using the proposed GED algorithm, however, the negative effects of gross errors are completely removed.

Here a study of noise reduction statistics cannot fairly be conducted, due to the existence of gross errors.

Chapter 5

Adaptive NDDR + GED and Smart Tracking System

5.1 Introduction

As mentioned earlier, most NDDR techniques today are based on two major assumptions: 1) having known dynamic and statistical models, and 2) having gross-error-free measurements. The novel ANDDR + GED approach presented in this chapter is suitable for cases where one does not have a statistical model for noise, or, in other words, standard deviation σ or covariance matrix V is not known, and where isolated outliers may occur.

“In DR studies almost without exception it is assumed that the statistical distribution of the measurement errors (V matrix) is known”[1]. It was Alici [1] who addressed the effect of covariance matrix estimation on the solution of NDDR.

When measurements contain gross errors, the typical covariance matrix estimation methods cannot be employed. The methodology here avoids such problems: when

measurements are received at each time step the novel GED approach developed in chapter 4 is first applied, and then a new standard deviation estimation approach (which will be presented in the next section) is employed, using measurements from which gross errors have been eliminated.

5.2 ANDDR + GED and covariance matrix estimation

In this methodology the same moving window approach proposed by Liebman et al. [7] is used and σ is estimated as each measurement variable is processed. The method can briefly be described as follows:

The moving window provides us with H measurements at each time step. Assume that $H \geq 10$ in this discussion; if this is not true, then a longer window may be used for estimating σ . Since the measurement error is Gaussian and white (uncorrelated from sample-to-sample), and the true process variables change slowly over a data window, the sample variance can be used to estimate σ for each variable. It is known that the σ of the sample variance for a Gaussian process is [10]:

$$\sigma_{\hat{V}} = \sqrt{2\frac{V}{H}} \quad (5.1)$$

For $H \geq 10$ the estimate is adequate for a threshold test that is usually conservatively chosen, e.g., $T = 3\hat{\sigma}$. This means, for example, that for $H = 10$, \hat{V} has a standard deviation of $0.447\sqrt{V}$ which is crude for statistical purposes but reasonable for a threshold test. This is the basis of the σ estimation method, which justifies using $\hat{\sigma} = \sqrt{\hat{V}}$ in solving the ANDDR problem.

The methodology here is the same as the GED approach presented in chapter 4.2,

with a major extension being that the estimated standard deviation, $\hat{\sigma}$, is up-dated at each time step, so it will vary depending on the statistical behavior of recent measurements. To perform GED, the difference $d_{c,i}$ is derived for each element of \tilde{y}_c (equation (5.2)), using the previous value of the mean (equation (5.4)), for each time step, and it is compared with the previous $\hat{\sigma}$, $\hat{\sigma}_{c-1,i}$, (equation (5.5)). If $|d_{c,i}|$ exceeds the threshold, as defined in equation (5.3), then the algorithm detects the existence of a gross error and removes it by replacing it with the previous estimate, \hat{y}_{c-1} .

$$d_{c,i} = \tilde{y}_{c,i} - \hat{m}_{c-1,i} \quad (5.2)$$

$$\text{If } |d_{c,i}| > 3\hat{\sigma}_{c-1,i} \text{ then } \tilde{y}_{c,i} \text{ is an outlier} \quad (5.3)$$

$$\hat{m}_{c,i} = \sum_{j=c-H}^c \left(\frac{\tilde{y}_{i,j}}{H+1} \right) \quad (5.4)$$

$$\hat{\sigma}_{c,i} = \sqrt{\sum_{j=c-H}^c \left(\frac{(\tilde{y}_{i,j} - \hat{m}_{c,i})^2}{H} \right)} \quad (5.5)$$

Again, as in the previous discussion in section 4.2, attention must be focused on the points in time near set-point changes. A sample point where an error threshold is exceeded should be designated as a possible outlier and the next point is processed to decide if a set-point change occurred or the previous point contained an outlier (in which case it is edited out, e.g., by interpolation). Using the same argument as in section 4.2, a one time-step delay occurs for isolated outliers and more delay is needed if multiple outliers can occur successively. Again, there is a trade-off between fast or robust detection. The former needs a GED algorithm based on an assumption of isolated gross errors, and the latter, which enables us to handle a number of successive gross errors, requires more detailed logic.

Figure 5.1 is a flowchart which describes step by step the procedure of application

5.3 Case study

The same CSTR model used in the previous chapters is studied here and the new ANDDR combined with the GED algorithm is implemented with the smart tracking system proposed in chapter 3. The implementation results are compared with the original NDDR estimates. Here it is assumed that isolated outliers exist for each variable. The number of gross errors over the entire simulation is 5 for each variable.

In figures 5.2 to 5.5 the estimation results considering a step change at time step 30 for the first input, A_0 , from 6.5 to 7.5, and another step change from 7.5 to 4.5 at time step 70, are presented. In each figure the complete time-history is shown, plus a zoomed section for better viewing. The solid lines in these figures show the true values, circles show the corrupted measurements, stars present the proposed ANDDR + GED + smart tracking system estimation results, and plus signs mark the original NDDR data.

5.3.1 Observation

As the figures 5.2 to 5.5 show, the gross errors have been detected and successfully removed, and the estimation has not been corrupted. Observe that the outliers cause significant corruption of the NDDR data. Similar to the case study in the previous chapter, both direct and indirect effects of gross errors are seen here. For instance, at time step 42, the gross error on output A has caused the next 11 estimates of the original NDDR data to be adversely affected (figure 5.2). Also, as an indirect effect example, figure 5.2 shows the corrupted estimation of NDDR data for output A from time step 69 to 79, which is caused by the gross error on T_0 at time step 69. Using the proposed GED algorithm, however, the negative effects of gross errors have been completely removed.

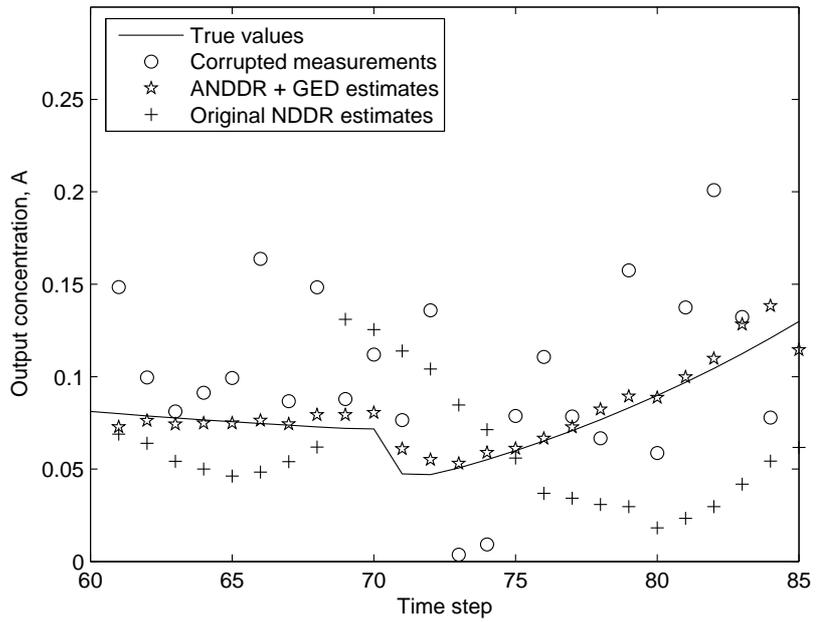
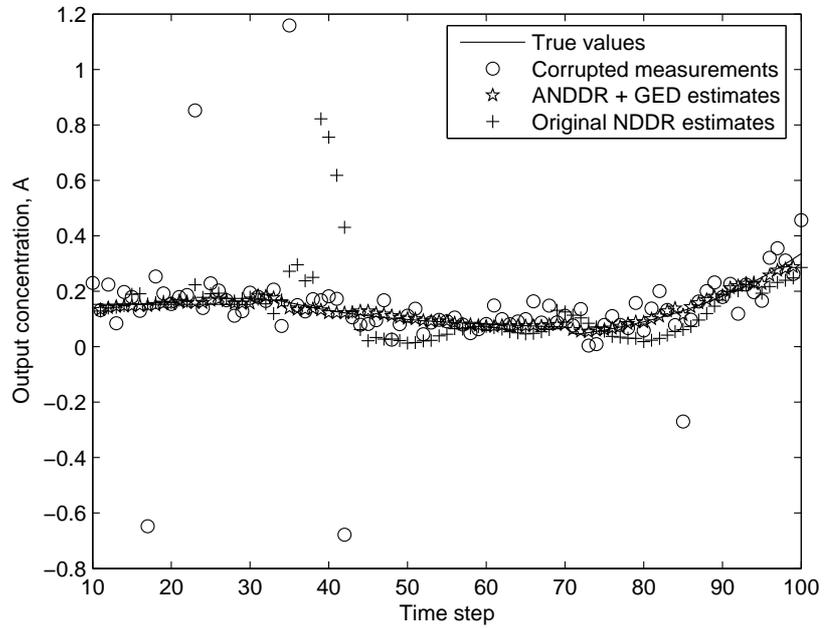


Figure 5.2: Comparison of original NDDR estimation with proposed ANDDR and GED combined with smart tracking system for the first output, A

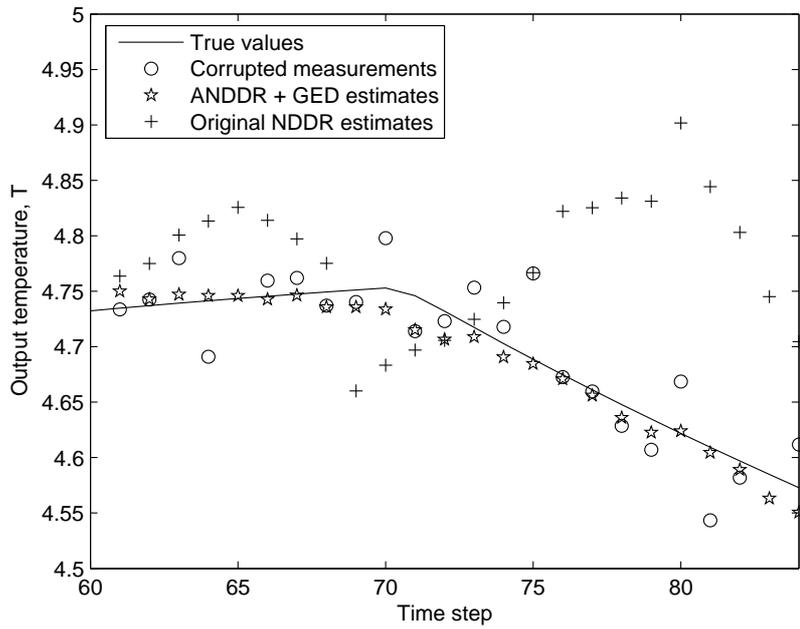
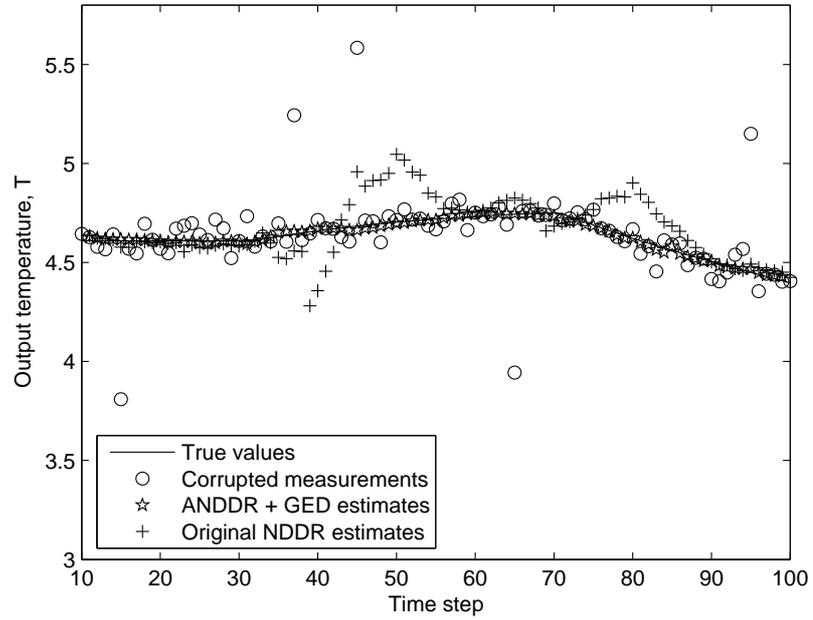


Figure 5.3: Comparison of original NDDR estimation with proposed ANDDR and GED combined with smart tracking system for the second output, T

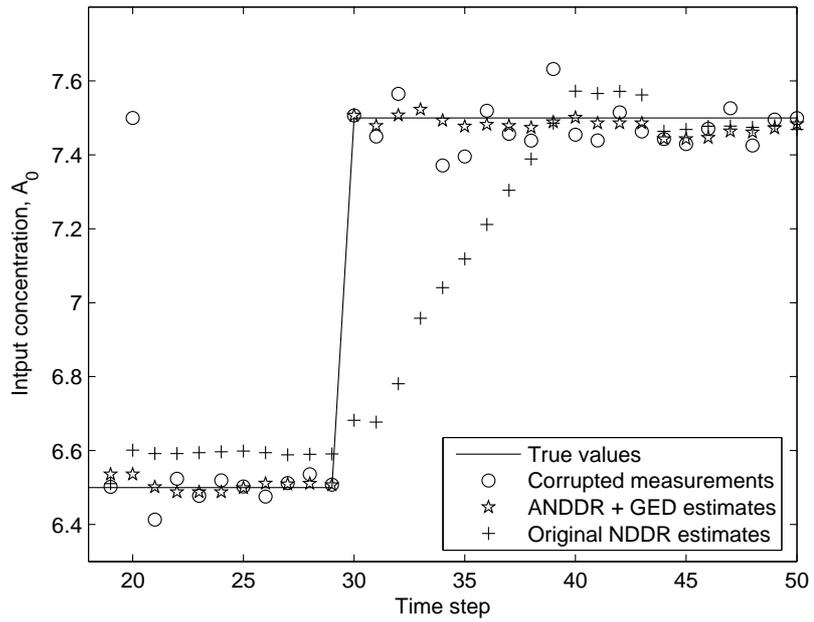
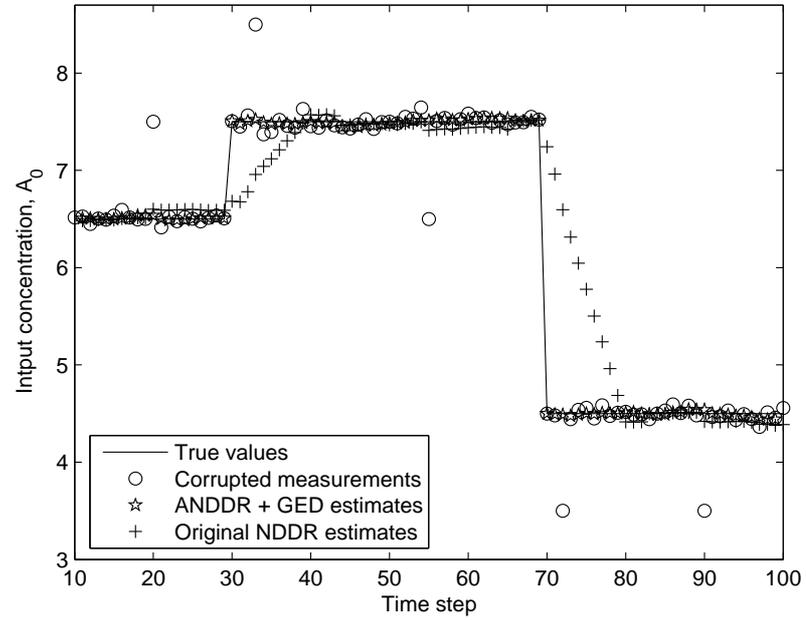


Figure 5.4: Comparison of original NDDR estimation with proposed ANDDR and GED combined with smart tracking system for the first input, A_0

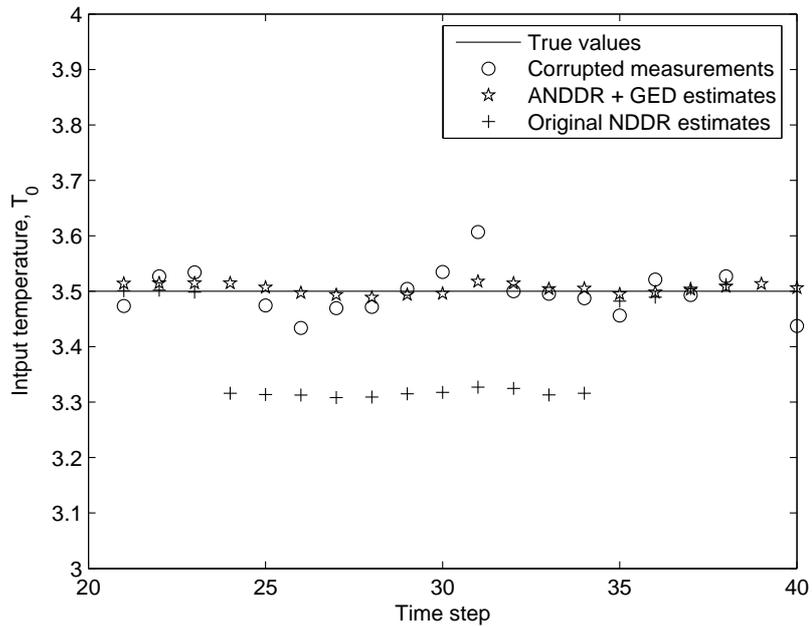
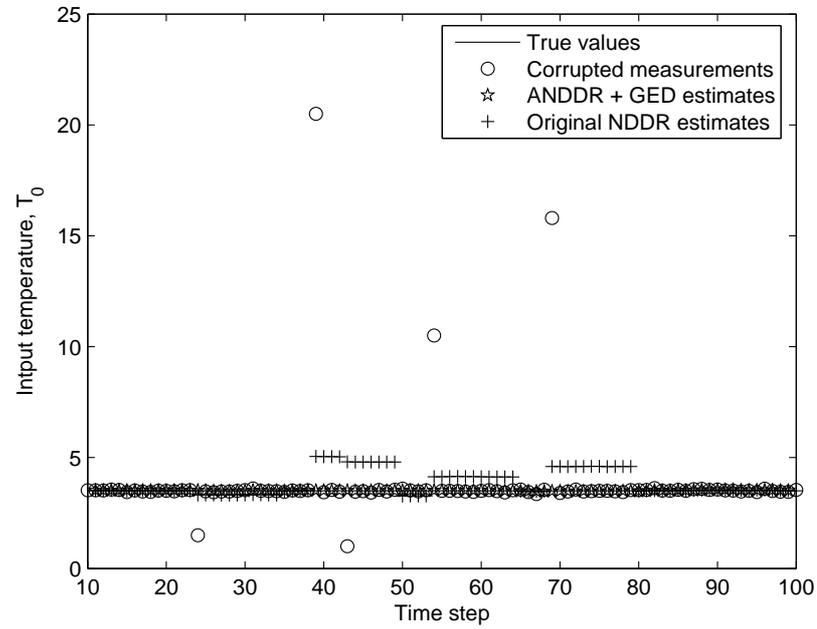


Figure 5.5: Comparison of original NDDR estimation with proposed ANDDR and GED combined with smart tracking system for the second input, T_0

Here, as in section 4.3, a study of noise reduction statistics cannot fairly be conducted, due to the existence of gross errors. Also, comparing these reconciled values with those of previous chapter show no degradation of the estimation for both input and output variables, even though in this chapter the assumption is made that the statistical noise model, σ , is not known. To demonstrate the successful σ estimation feature of the ANDDR algorithm, the values of $\hat{\sigma}_i$ for each variable are depicted in figures 5.6 to 5.9 knowing that the true σ is 0.05. Observe that the variation in $\hat{\sigma}$ for each variable is in the order of $\pm 40\%$ in each case, which is much less than the threshold defined in this methodology ($3\sigma_{true}$).

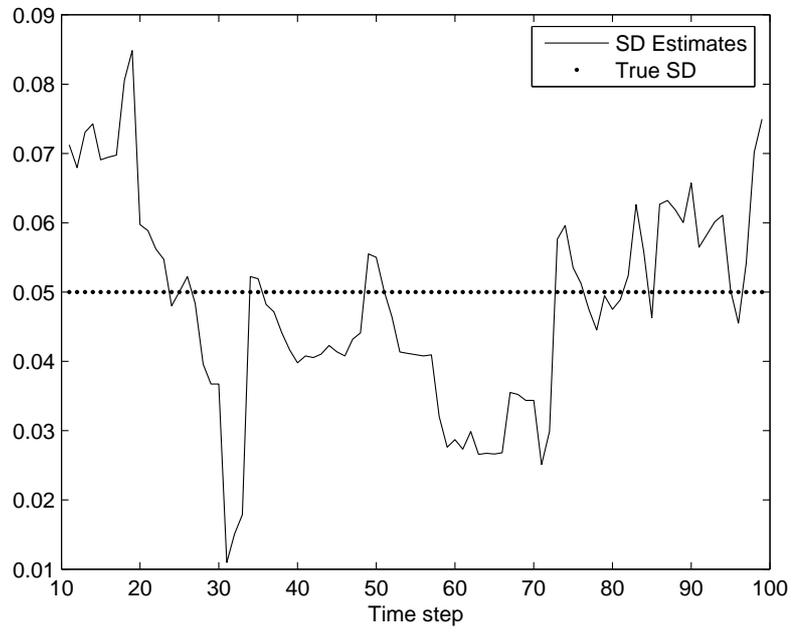


Figure 5.6: σ estimates obtained from ANDDR estimation for the first output, A

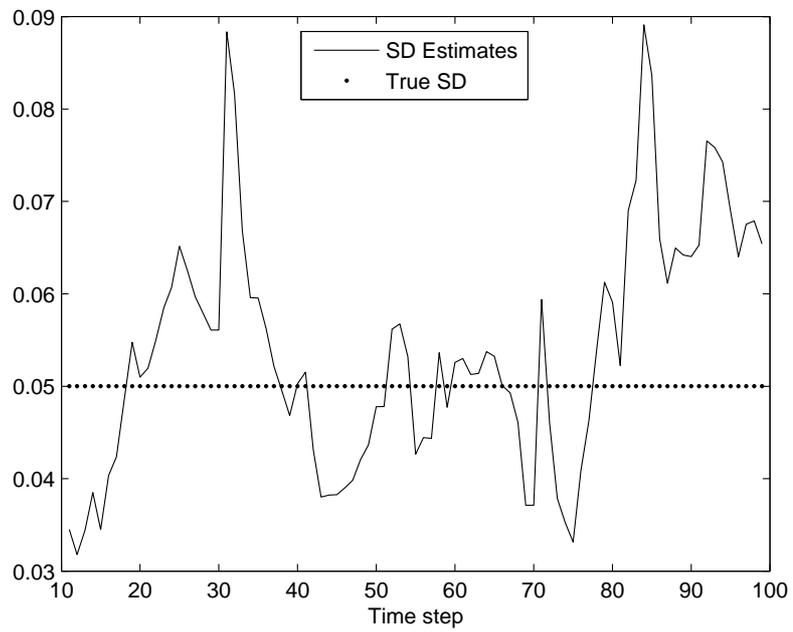


Figure 5.7: σ estimates obtained from ANDDR estimation for the second output, T

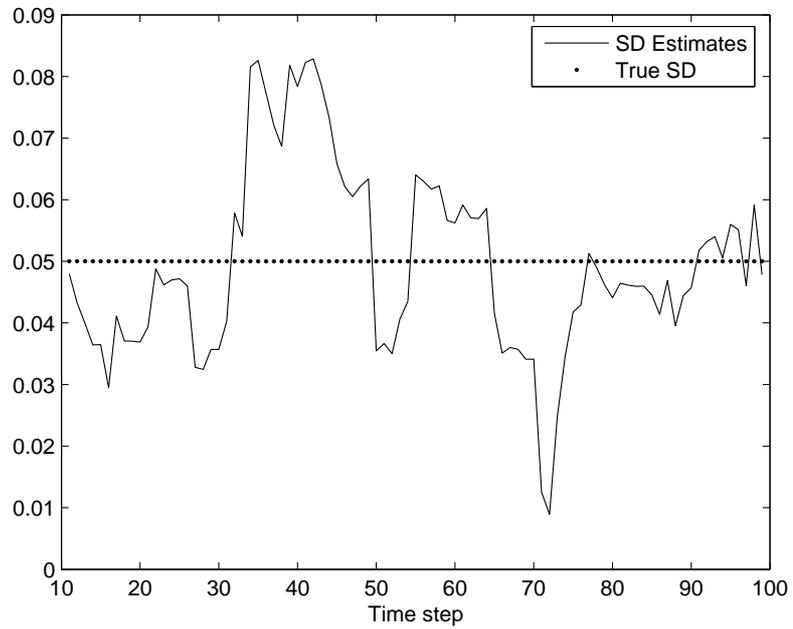


Figure 5.8: σ estimates obtained from ANDDR estimation for the first input, A_0

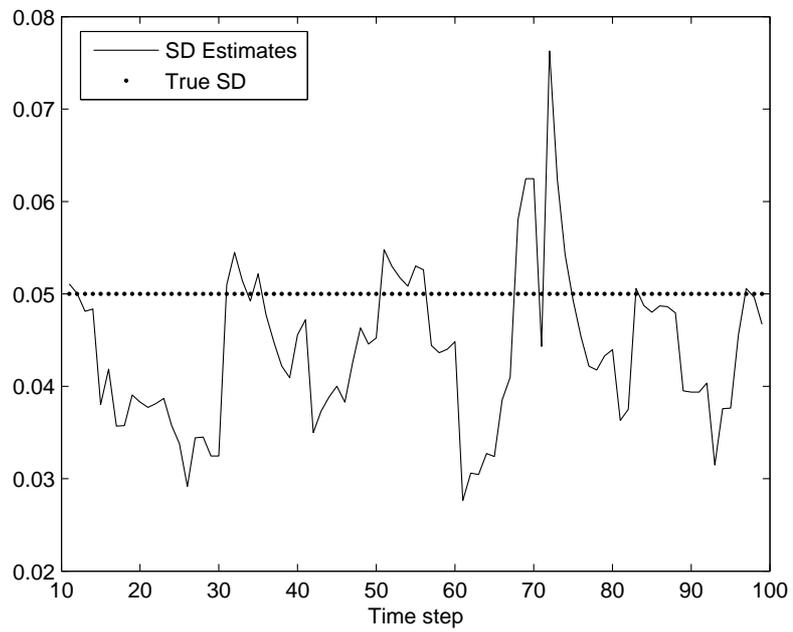


Figure 5.9: σ estimates obtained from ANDDR estimation for the second input, A

Chapter 6

Adaptive NDDR + GED for Slowly Varying Inputs

6.1 Introduction

Most nonlinear dynamic data reconciliation methods have studied cases where the input variables are constant over relatively long periods of time separated by simple step changes (e.g., set-point changes). While this scenario is not uncommon in process control, it imposes strong limitations on a method's applicability.

In this chapter a novel ANDDR algorithm is presented that extends the method presented in chapter 5 to the cases where the input variables are ramps or slow sinusoidal functions or, for that matter, any slow, smooth variation. The proposed package has been successfully applied to the same CSTR model used in the previous chapters.

The combined ANDDR and GED method together with the enhanced dynamic tracking feature presented before, plus the extended capability to track slowly-varying inputs discussed in this chapter, produce a package that is suitable to most control process

applications. This latest extension we refer to as ANDDR-R where -R denotes its design for ramp inputs.

6.2 ANDDR + GED for slowly varying inputs

Here the method presented in chapter 5 is extended in order to include applications with ramp and sinusoidal inputs. As before, the ANDDR-R + GED approach is applicable to cases where the statistical model for noise is not given, or, in other words, the standard deviation σ or covariance matrix V is unknown, and where outliers may occur.

Since in this study the input to the model is assumed to be a ramp, sinusoid or other smooth function, the difficulty of set-point change detection does not exist. One can refer to chapters 4 and 5 where the problem of the set-point change is tackled while GED is performed in order to appreciate this simplification.

The key modification that we made here for the ANDDR-R approach is the way that input variables are estimated for each moving window. Traditionally, in the original NDDR algorithm, inputs over a moving window were assumed to be constant, which caused a significant delay seen in the estimation of the step input [7]. This problem was eliminated for step input functions in chapter 3 where the smart tracking system with two input levels over the window instead of one, was introduced. In this chapter the inputs are assumed to be ramps or slow sinusoids. Therefore, the extended ANDDR-R solution is obtained by assuming that the input over each moving window is a ramp. In this way, the algorithm can be significantly faster and smoother in tracking the dynamic behavior of a system with slowly-varying continuous inputs.

6.3 Case study - ramp input tracking

In this section the performance of the proposed algorithm is demonstrated. The case studied in this chapter is the same simulated CSTR model cited in the previous chapters. First, the situation where the inputs to the model are ramps is considered. Then, in the next study the response of the system to slow sinusoidal inputs is considered and the tracking capability for general slow and smoothly varying inputs is illustrated.

6.3.1 Gross-error-free ramp input tracking

We consider the CSTR model used in the previous chapters with the same assumptions and parameters values, except the inputs may be ramp functions. In this case study both the two inputs and two states (outputs) are estimated, assuming that no gross errors exist. Measurements were simulated by creating the measurement noise which is assumed to be Gaussian with σ equal to 0.05 and zero mean. The time step is assumed to be 2 seconds and the simulation is run for 100 samples with window width of $H = 10$. Obviously, the first estimates are obtained at time step 10 where the first window of measurements is available. The first input, A_0 , is assumed to be a ramp starting from 6.5 and increasing to 8.5, as shown in figure 6.1, and the second input, T_0 , is assumed to be constant at 3.5.

Figures 6.1 to 6.4 demonstrate the successful application of the ANDDR-R approach for the ramp function. The solid lines in these figures show the true values, circles show the corrupted measurements, stars represent the proposed ANDDR-R estimates, and plus signs mark the original NDDR estimates. There is a significant noise reduction in both input and output estimation for ANDDR-R application. A large delay, however, is seen for the estimation of the first input, A_0 , when the original NDDR algorithm is

used. Obviously, the purpose of ANDDR-R development was to track slowly varying inputs, which is successfully achieved in this study. However, as the figure 6.1 shows, original NDDR fails to track A_0 due to its constant input assumption for each moving window. Table 6.1 shows the noise reduction statistics for ANDDR-R estimates.

Table 6.1: ANDDR-R noise reduction statistics for ramp case

Variable	Measurements σ	Estimates σ	% σ reduction
A	0.0437	0.0030	93.21
T	0.0437	0.0045	89.75
A_0	0.0437	0.0197	54.90
T_0	0.0437	0.0092	65.28

6.3.2 Gross errors and ramp input tracking

With the same assumptions made for the previous case, random gross errors are added to the measurements of each input and output variable in the second study. There are four random gross errors for each variable with different amplitudes. Figures 6.5 to 6.8 demonstrate the successful GED algorithm implementation. These figures show the comparison of the situation where GED method was implemented along with ANDDR-R, and where ANDDR-R is individually implemented. The solid lines in these figures show the true values, circles show the corrupted measurements, stars present the proposed ANDDR-R + GED estimation results, and plus signs mark the ANDDR-R data without GED. As the figures show, the gross errors have been detected and successfully removed, and the estimation has not been corrupted. Observe that the outliers cause significant corruption of the estimates when the ANDDR-R is implemented solely; the use of the GED algorithm eliminates this problem. Note that the method is applicable where many gross errors may exist in the measurements, whether they are successive or isolated, since the algorithm can detect and identify them without the need to

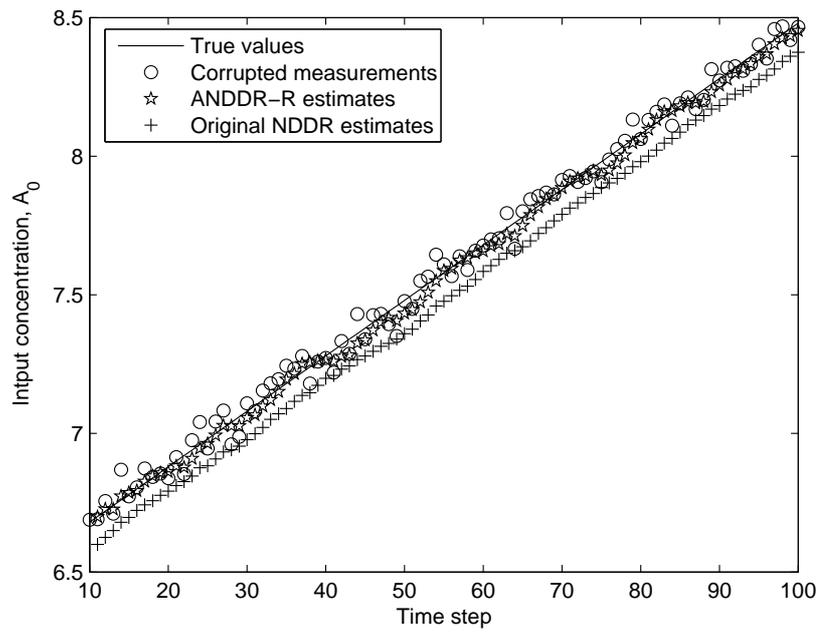


Figure 6.1: First input, A_0 , estimation. (gross-error-free)

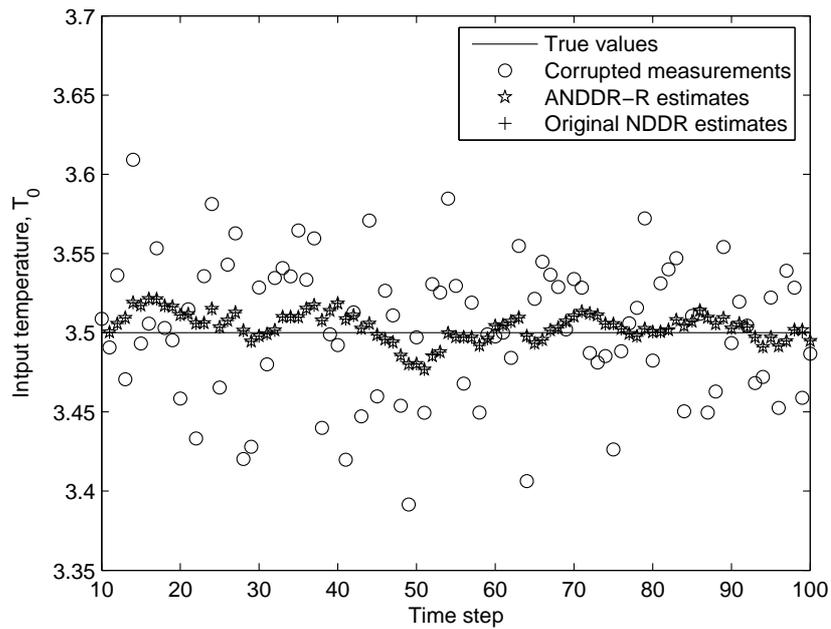


Figure 6.2: Second input, T_0 , estimation. (gross-error-free)

distinguish outliers from step changes.

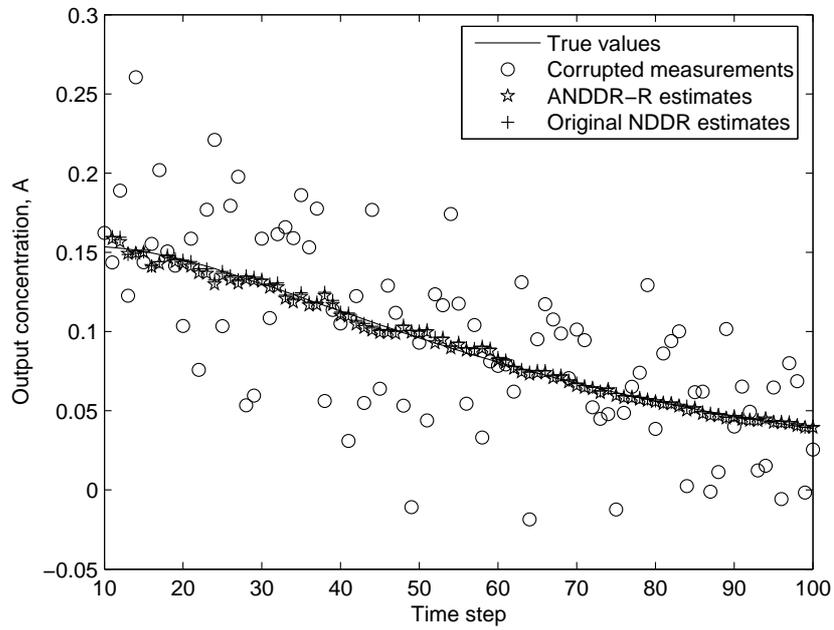


Figure 6.3: First output, A , estimation. (gross-error-free)

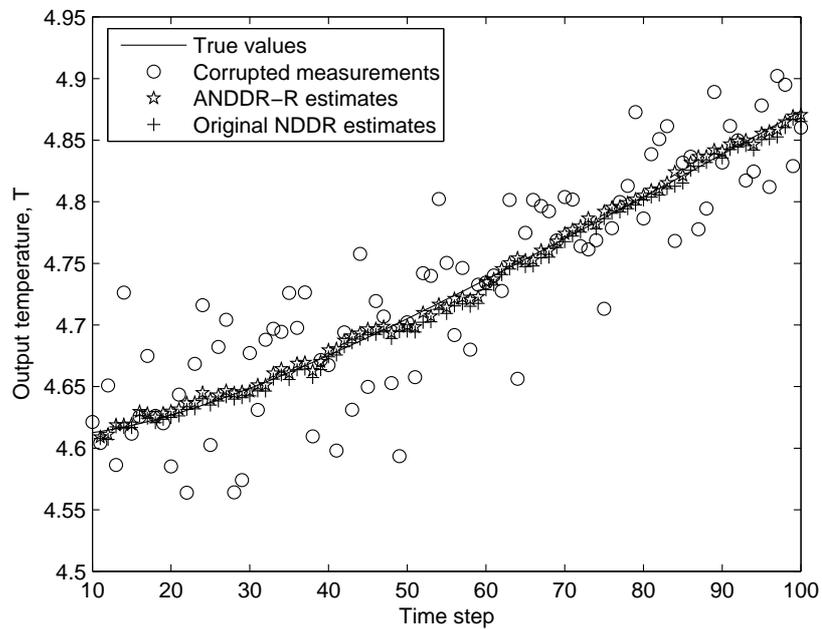


Figure 6.4: Second output, T , estimation. (gross-error-free)

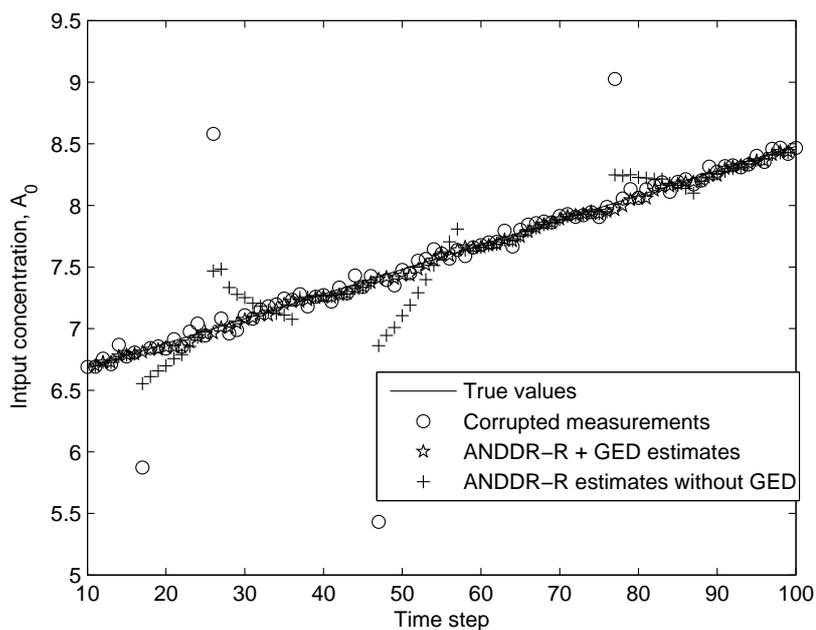


Figure 6.5: First input, A_0 , estimation. (with gross errors)

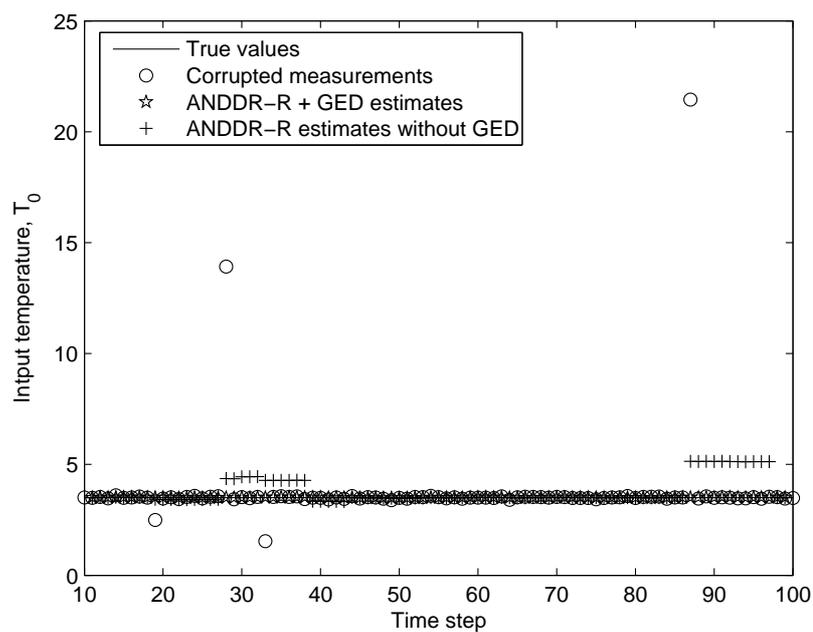


Figure 6.6: Second input, T_0 , estimation. (with gross errors)

6.4 Case study - sinusoidal input tracking

In this section the performance of the algorithm for sinusoidal inputs is studied. The previous CSTR model with the same parameters and assumptions is used. The feed

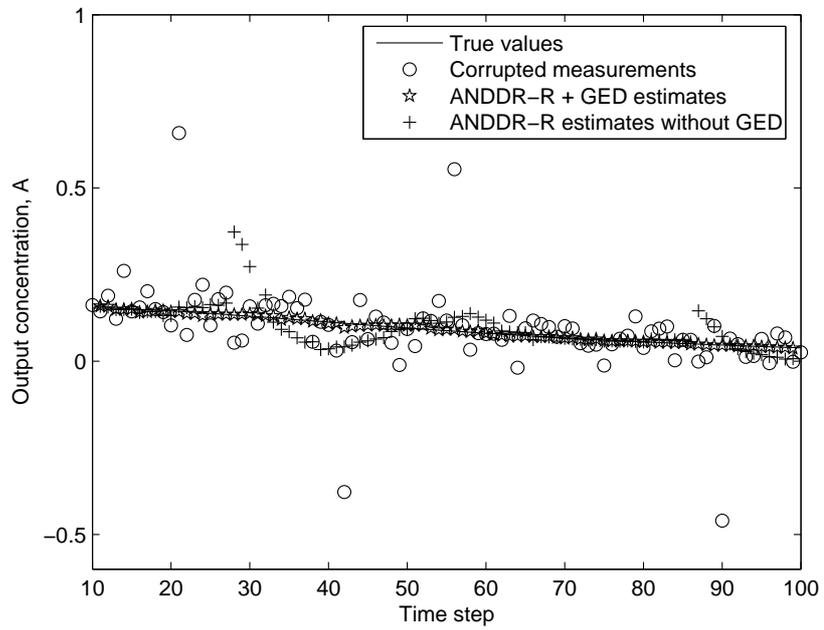


Figure 6.7: First output, A , estimation. (with gross errors)

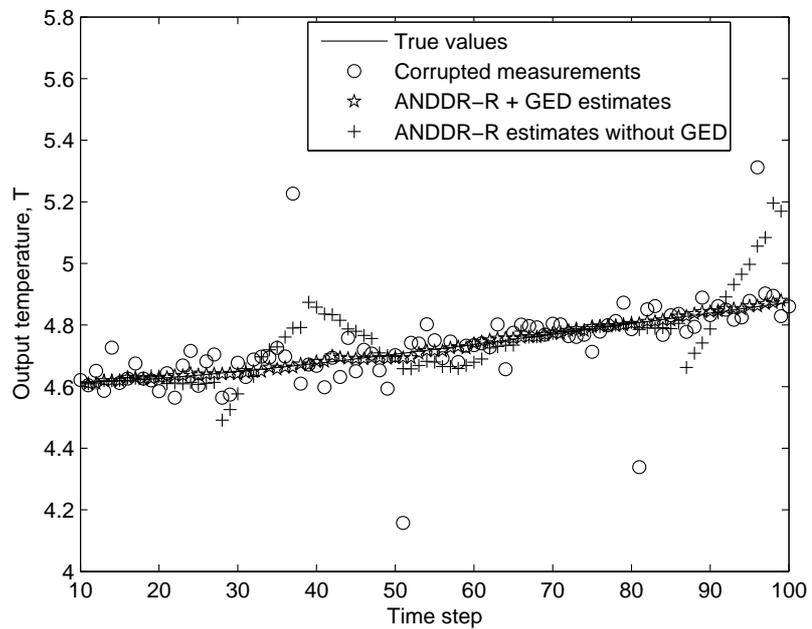


Figure 6.8: Second output, T , estimation. (with gross errors)

concentration, A_0 , is a slow sinusoidal function as follows:

$$A_0 = \sin\left(\frac{\pi}{100}t\right) + 6.5, \quad (6.1)$$

and the feed temperature, T_0 , is constant with the same value of 3.5.

6.4.1 Gross-error-free sinusoidal input tracking

First we assume that the measurements are not corrupted by gross errors and only contain the same zero-mean Gaussian noise defined in section 6.3.1. As figures 6.9 to 6.12 demonstrate, the ANDDR-R algorithm has successfully tracked the sinusoidal behavior of the model, for both input and output estimates. The solid lines in these figures show the true values, circles show the corrupted measurements, stars represent the proposed ANDDR-R estimates, and plus signs mark the original NDDR estimates. There is a significant noise reduction in both input and output estimation for ANDDR-R application. A large delay, however, is seen for the estimation of the first input, A_0 , when the original NDDR algorithm is used (figure 6.9). This delay is significantly reduced using the ANDDR-R algorithm, but still a slight lag exists. One suggestion to reduce this delay would be to assume two different slopes for the input estimation in each moving window. This modification would significantly reduce the delay seen in the estimates of feed concentration.

Table 6.2 shows the noise reduction statistics for both the inputs and outputs. Obviously, due to the slight delay seen in the sinusoid input estimation, A_0 , no noise reduction can be achieved for this variable. The reason is that systematic error due to the delay mentioned above is greater than the very low level of random measurement error.

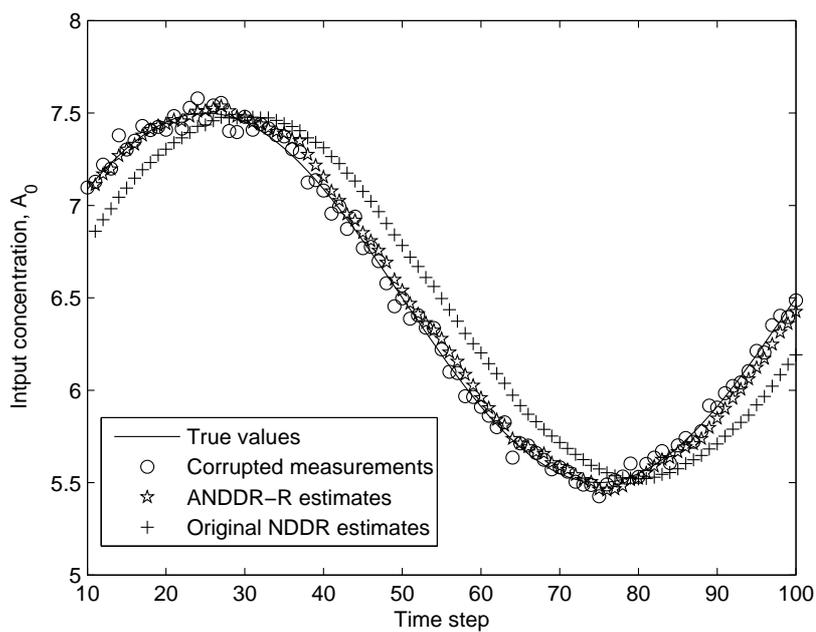


Figure 6.9: First input, A_0 , estimation. (gross-error-free)

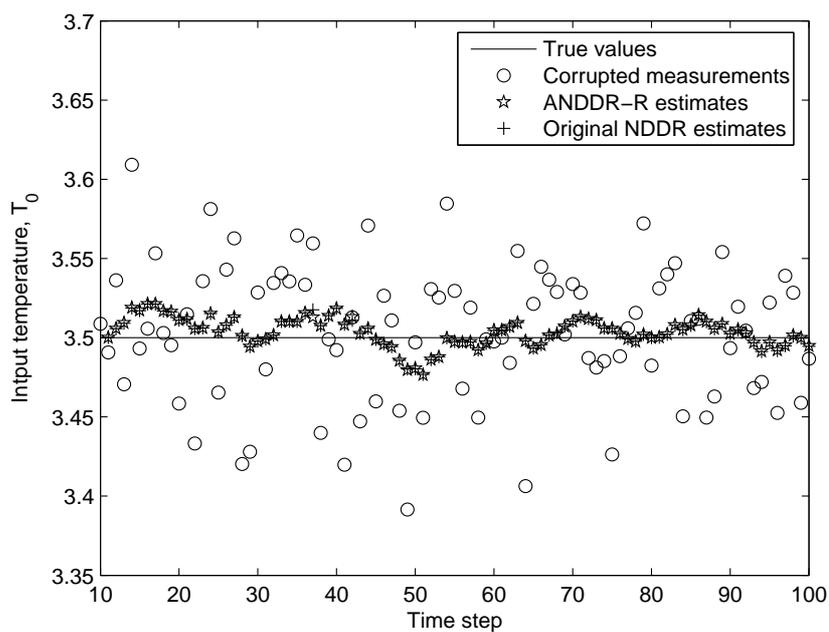


Figure 6.10: Second input, T_0 , estimation. (gross-error-free)

6.4.2 Gross errors and sinusoidal input tracking

To prove the performance of the proposed GED algorithm, we consider the existence of gross errors. There are four random outliers added to each input and output variable

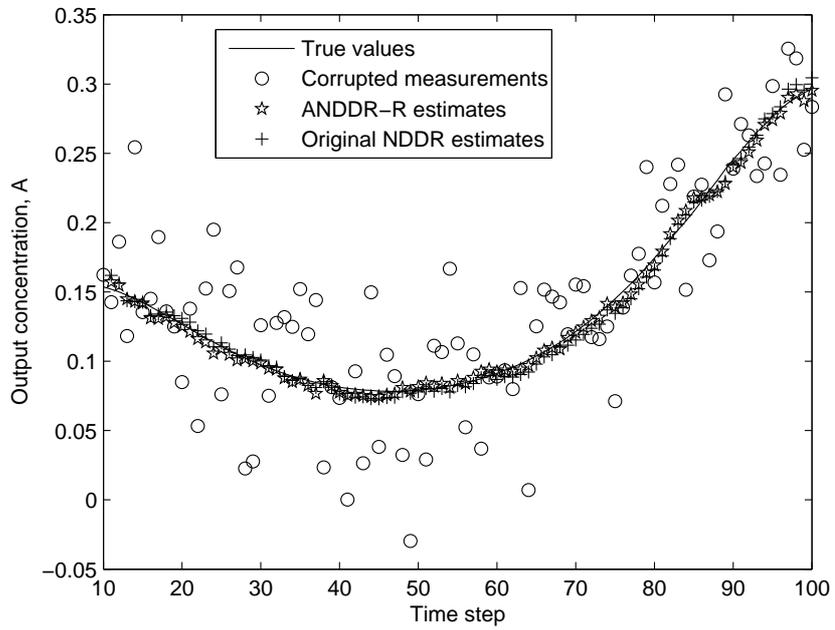


Figure 6.11: First output, A , estimation. (gross-error-free)

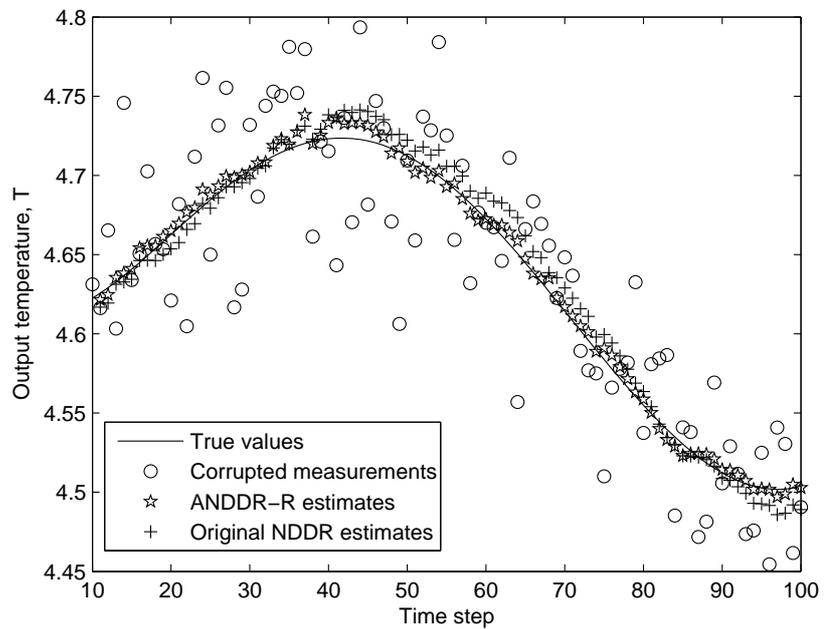


Figure 6.12: Second output, T , estimation. (gross-error-free)

Table 6.2: ANDDR-R noise reduction statistics for sinusoidal case

Variable	Measurements σ	Estimates σ	% σ reduction
A	0.0437	0.0040	90.85
T	0.0437	0.0050	88.57
A_0	0.0437	0.0480	-
T_0	0.0437	0.0091	79.16

measurement, and the proposed ANDDR-R + GED method is implemented. Figures 6.13 to 6.16 show the successful detection of outliers and show the comparison studies of the two cases of ANDDR-R and ANDDR-R + GED implementations. It is clear that without GED the estimates become seriously corrupted.

6.5 Conclusion

Studying the results achieved in the case studies presented in sections 6.3 and 6.4, the applicability of the proposed ANDDR-R + GED package for cases with slowly varying inputs is demonstrated. As the estimates show, when there are no gross errors the ANDDR-R algorithm performed well for both ramp and slow sinusoidal cases, except for a slight delay seen in estimation of the sinusoidal input. However, it is believed that this delay can be reduced by a further simple extension (using two input slopes over the window, as mentioned in section 6.4.1) which is suggested for future work. Also, in presence of the gross errors, the GED algorithm combined with ANDDR-R, detected and removed outliers successfully for the both case studies.

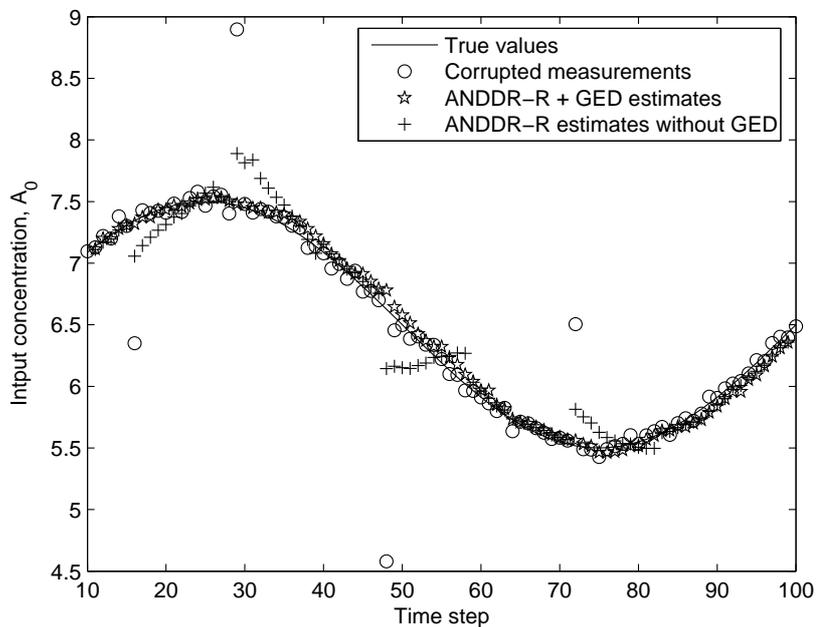


Figure 6.13: First input, A_0 , estimation. (with gross errors)

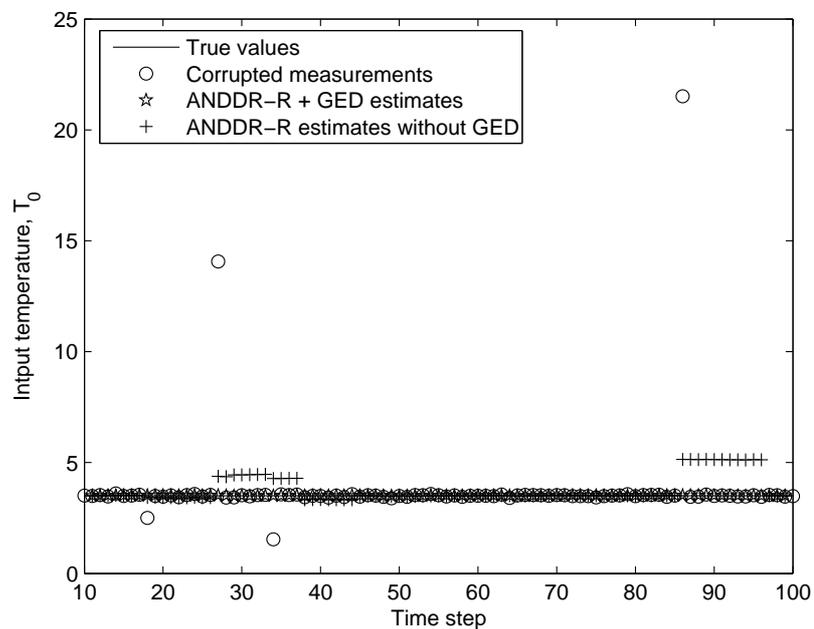


Figure 6.14: Second input, T_0 , estimation. (with gross errors)

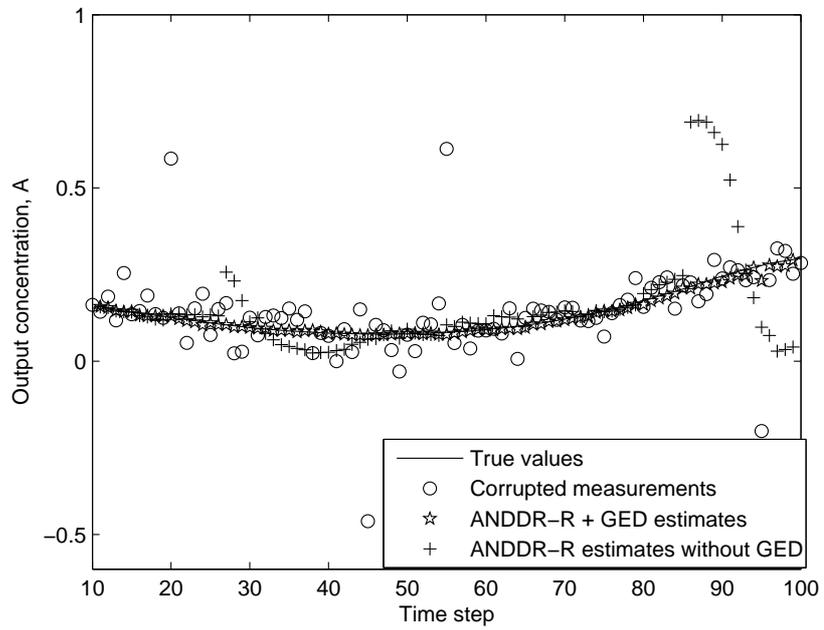


Figure 6.15: First output, A , estimation. (with gross errors)

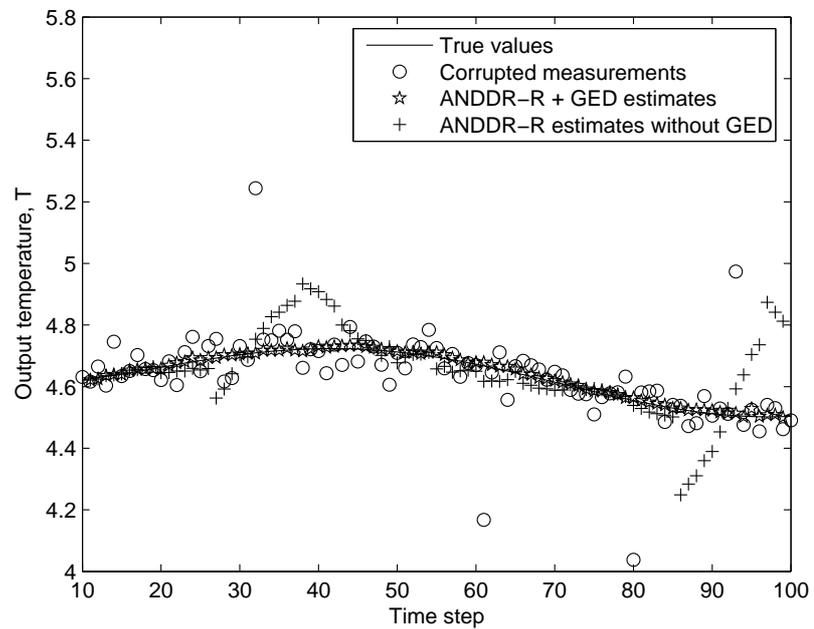


Figure 6.16: Second output, T , estimation. (with gross errors)

Chapter 7

Implementation on JCSTR Model

In this chapter a novel application of the proposed ANDDR + GED package is studied. In order to study a situation where we have both set-point changes and slowly varying inputs happening at the same time, the proposed ANDDR + GED algorithm is applied to a more sophisticated model. A jacketed continuous stirred tank reactor (JCSTR) is chosen with the physical characteristics discussed in the following section.

7.1 JCSTR physical model

The JCSTR model studied in this thesis is portrayed in figure 7.1. In this JCSTR model the tank inlet stream is received from another process unit and there is a heat transfer fluid circulating through the jacket to heat the fluid in the tank. The objective is to control the temperature and the volume inside the tank by varying the jacket inlet valve flow rate and the tank outlet valve flow rate. In order to derive the dynamic modeling equations of the tank and jacket temperatures, the following assumptions were made:

- Liquids have constant density and heat capacity.
- Mixing in both the tank and jacket are perfect.
- The rate of heat transfer from the jacket to the tank is governed by the equation $Q = U A(T_j - T)$, where U is the overall heat transfer coefficient and A is the area for heat transfer.

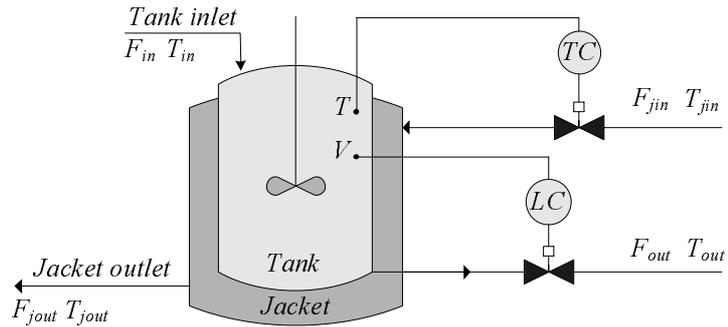


Figure 7.1: Jacketed continuous stirred tank reactor

The following equations describe the model for the JCSTR reactor [8]:

$$\dot{V} = F_{in} - F_{out} \quad (7.1)$$

$$\dot{T} = \frac{F_{in}(T_{in} - T)}{V} + \frac{UA(T_j - T)}{V\rho C_p} \quad (7.2)$$

$$\dot{T}_j = \frac{F_{jin}(T_{jin} - T_j)}{V_j} + \frac{UA(T_j - T)}{V_j\rho C_p} \quad (7.3)$$

where subscripts *in*, *out* and *j* refer to inlet, outlet and jacket respectively. The parameters values are given in table 8.2 in Appendix B.

This is a third order model with three input variables (F_{in}, F_{out}, T_{in}) and three states or outputs (V, T, T_j). Output variables are tank volume, V , temperature inside the tank, T , and temperature inside the jacket, T_j . The input variables are mixture inflow, F_{in} , mixture outflow, F_{out} , and the temperature of the mixture feed, T_{in} . The

operating point of the model considered for this study is:

$$V_o = 100 \text{ m}^3, T_o = 290^\circ K, T_{j_o} = 300^\circ K \quad (7.4)$$

The time step in this simulation is 10 sec, and the simulation is run for 1500 seconds. The inputs are assumed to include only step changes. The mixture inflow, F_{in} , is equal to $0.1 \text{ m}^3/s$ initially until the step is applied at time step 50 (500 sec) from 0.1 to 0.15. The other two input variables, mixture outflow, F_{out} , and the temperature of the mixture feed, T_{in} , are constant with $0.1 \text{ m}^3/s$ and $283^\circ K$, respectively.

7.2 Application results

In this section the extended ANDDR + GED algorithm is applied to the JCSTR model. Two cases are considered: 1) gross-error-free measurements, to evaluate the performance of the ANDDR + GED approach in estimating the statistical model; and 2) existence of gross errors, to evaluate the whole package's estimation results.

7.2.1 Gross-error-free case

The measurement noise is assumed to be Gaussian with σ equal to 5% of the nominal values ($\sigma = 5, 0.85, 1.35, 0.005, 0.005, 0.5$) and zero mean. Measurements are created by adding this noise to the actual values of the model achieved through simulation. Window width of $H = 10$ is selected and obviously, the first estimation is achieved at time step 10 where the first window of measurements is available. Figures 7.2 to 7.7 demonstrate the results of the ANDDR + GED implementation on the simulated JCSTR model. Solid lines represent the true values (from the simulation), circles the corrupted measurements, and plus signs mark the ANDDR + GED estimates. As these figures show, both input and output variables are estimated successfully with

Table 7.1: ANDDR noise reduction statistics for JCSTR model for gross-error-free case

Variable	Measurements σ	Estimates σ	% σ reduction
V	4.4352	1.2630	71.52
T	0.7540	0.1949	74.15
T_j	1.1975	0.1179	90.15
F_{in}	0.0044	0.0019	56.48
F_{out}	0.0044	0.0012	72.20
T_{in}	0.4435	0.1229	72.30

a significant noise reduction in each case. Observe that in the estimation of the first input, mixture inflow, there is no delay when the step occurs at time step 50 (figure 7.5). Another prominent feature is how fast the proposed algorithm has tracked the output variables. All three output variables show fast transient responses, and the estimation results demonstrate the successful ANDDR + GED tracking feature. There is a significant noise reduction in both input and output estimation. Table 7.1 shows the noise reduction statistics.

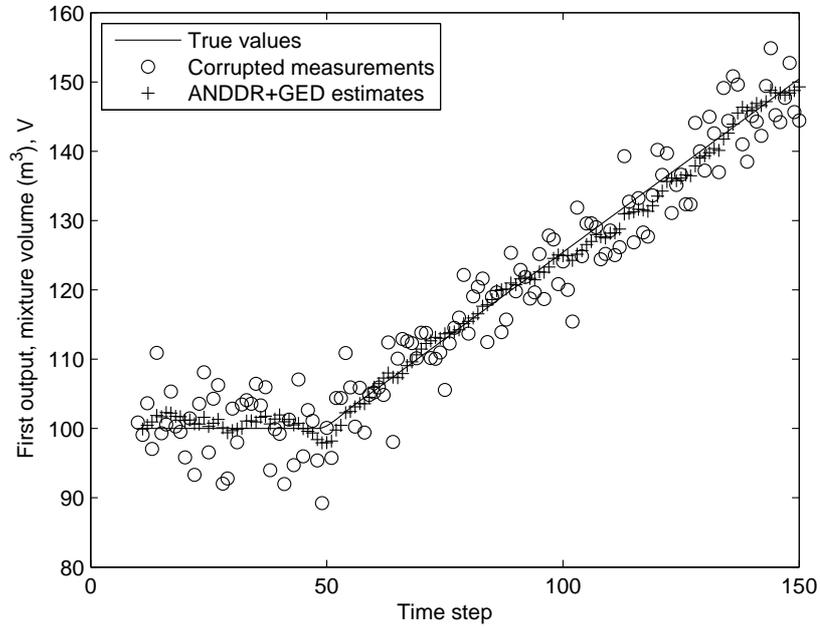


Figure 7.2: First output, V , estimation. (gross-error-free)

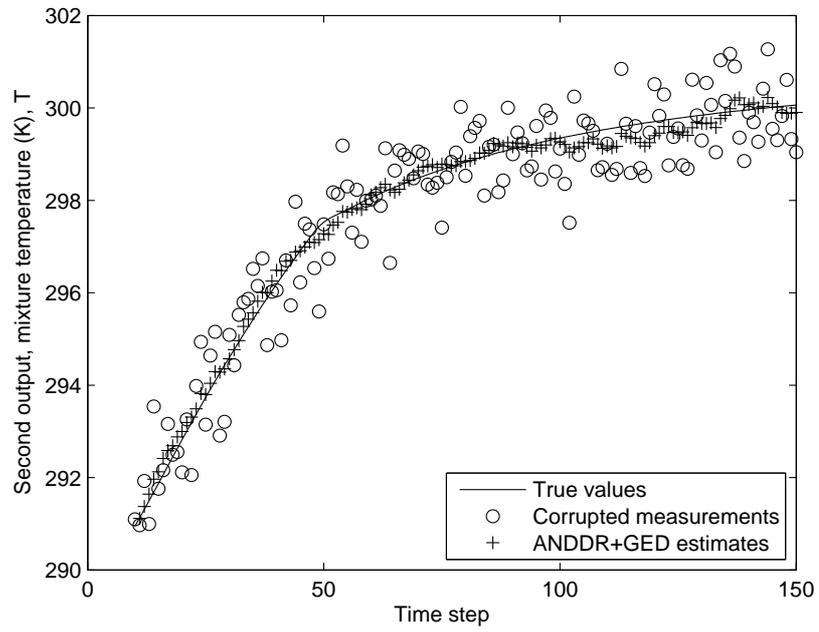


Figure 7.3: Second output, T , estimation. (gross-error-free)

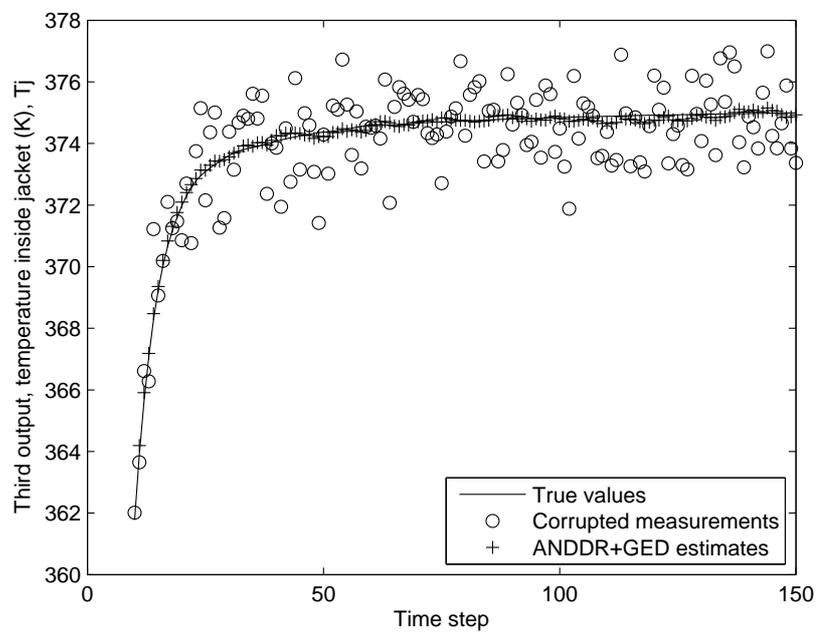


Figure 7.4: Third output, T_j , estimation. (gross-error-free)

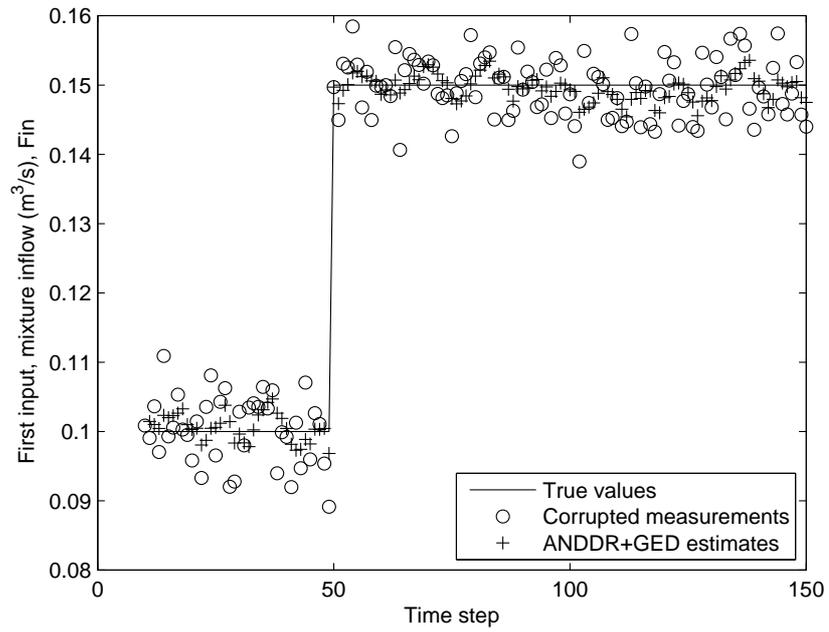


Figure 7.5: First input, F_{in} , estimation. (gross-error-free)

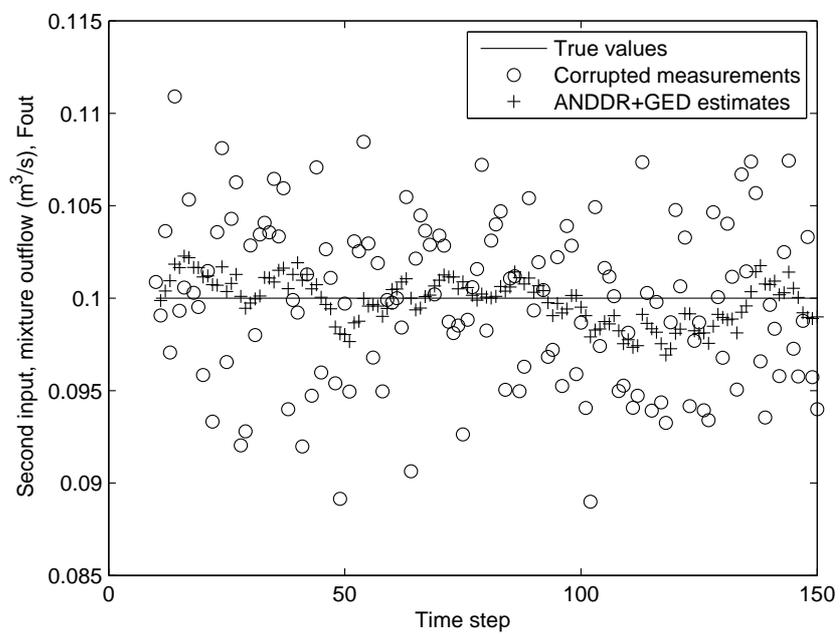


Figure 7.6: Second input, F_{out} , estimation. (gross-error-free)

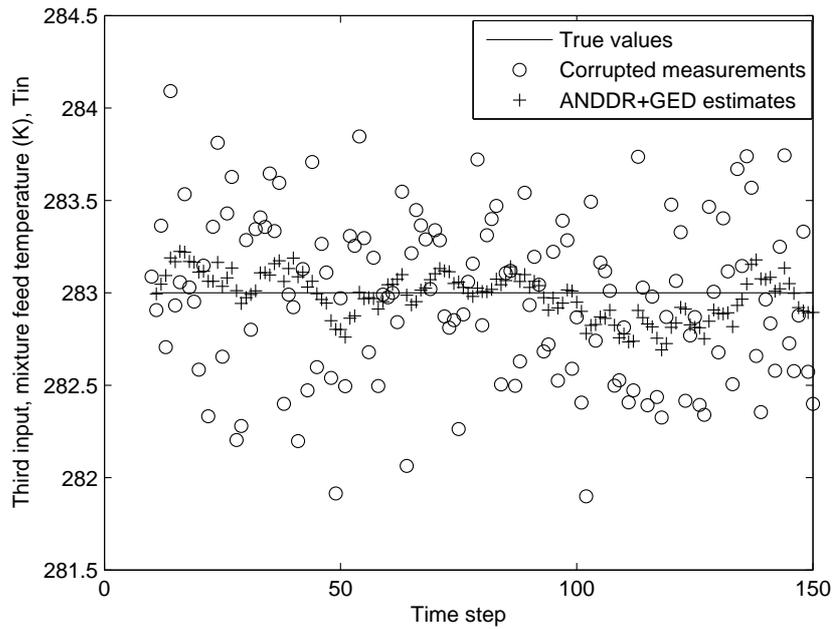


Figure 7.7: Third input, T_{in} , estimation. (gross-error-free)

7.2.2 Presence of gross errors

In this section the results of the application of the extended ANDDR + GED to the JCSTR model are presented in a case when gross errors exist. There are three gross errors assumed for each input and output variable. Figures 7.8 to 7.13 show successful estimates with plus signs, corrupted measurements with circles, and true values with solid lines. In order to see the effect of gross errors, the results of the ANDDR estimation without GED are also depicted with stars. Observe that outliers cause significant corruption of the ANDDR data.

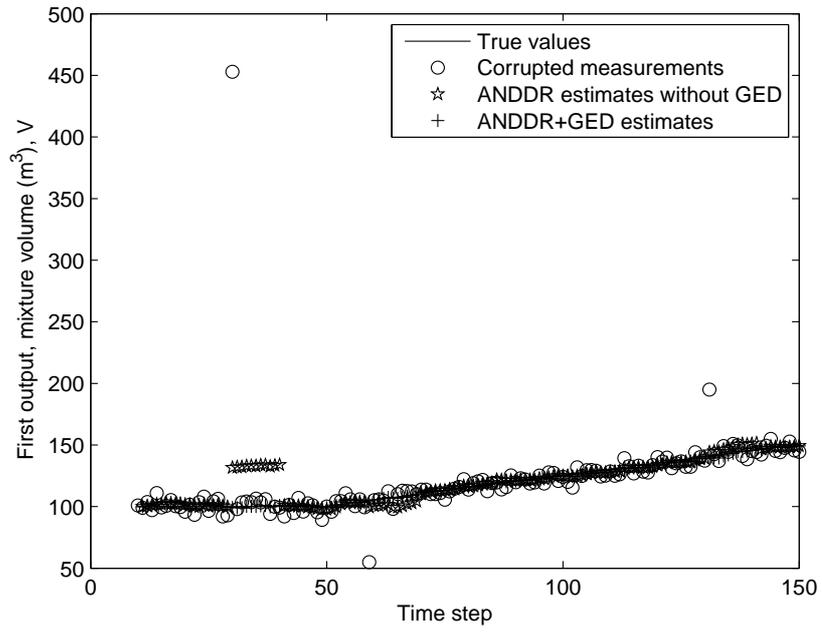


Figure 7.8: First output, V , estimation. (with gross errors)

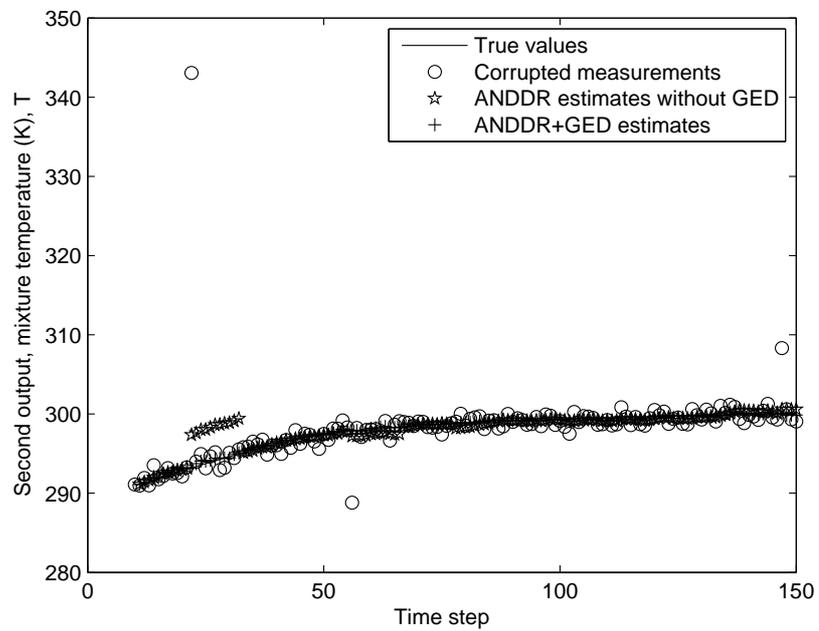


Figure 7.9: Second output, T , estimation. (with gross errors)

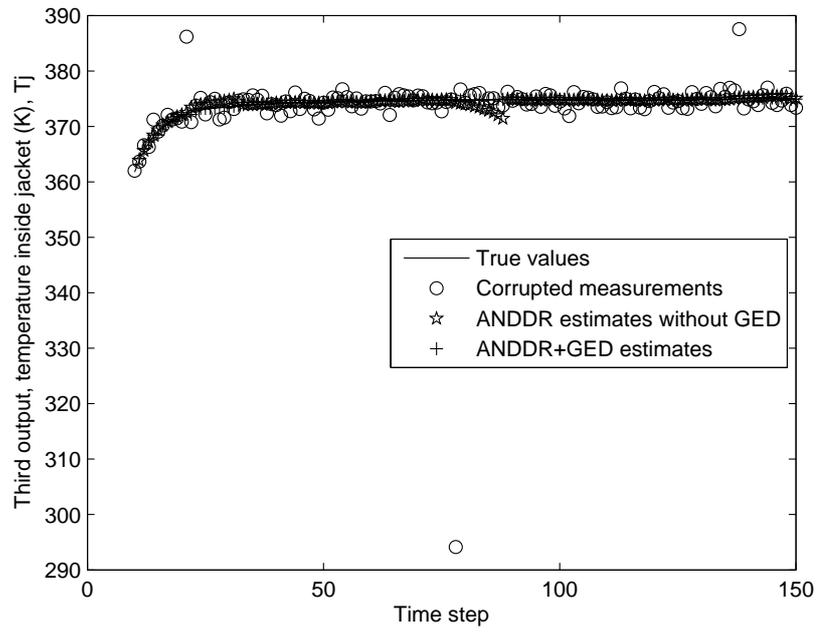


Figure 7.10: Third output, T_j , estimation. (with gross errors)

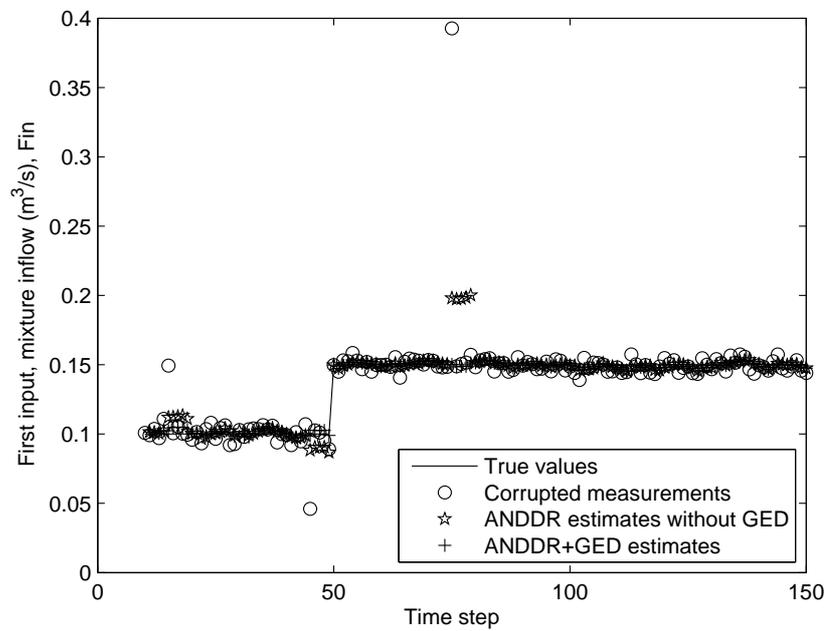


Figure 7.11: First input, F_{in} , estimation. (with gross errors)

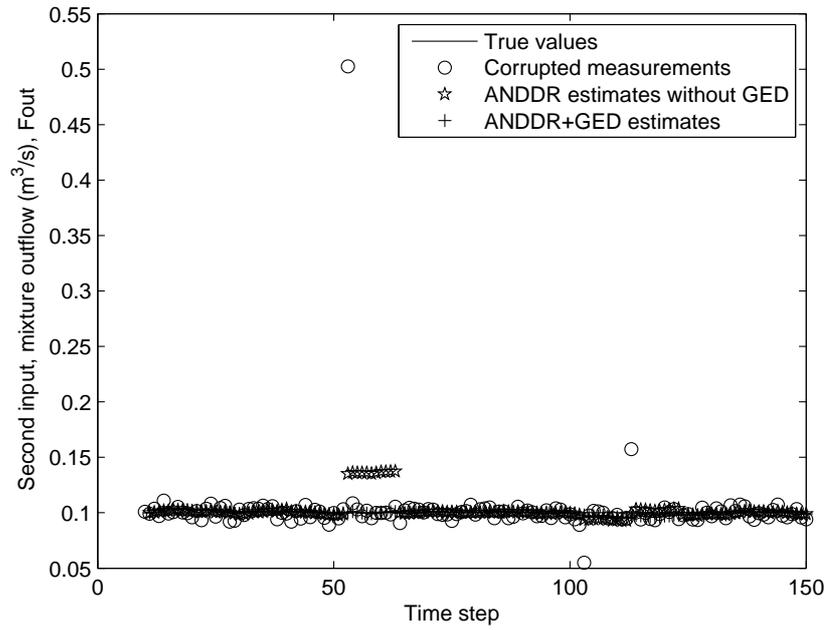


Figure 7.12: Second input, F_{out} , estimation. (with gross errors)

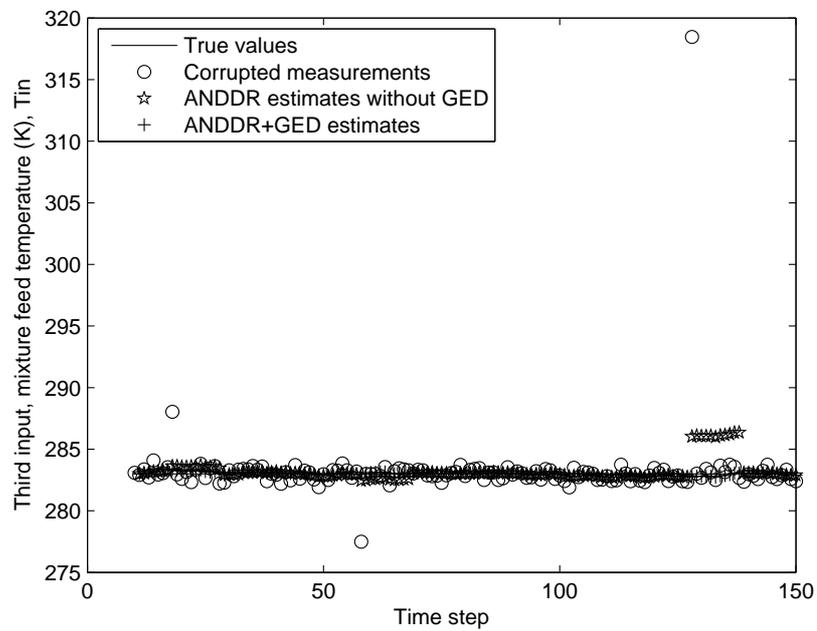


Figure 7.13: Third input, T_{in} , estimation. (with gross errors)

Chapter 8

Thesis Observations

8.1 Conclusions

1. The original NDDR approach developed by Liebman et al. [7] has been studied and extensions have been made to improve the tracking system and adaptability of the algorithm.
2. A smart tracking system has been developed which ameliorates the problem of delay seen in original and later versions of NDDR.
3. A novel ANDDR method has been developed. As the results in chapters 5, 6 and 7 show, the ANDDR algorithm can accurately estimate both input and output variables even if the statistical model is not known.
4. A novel GED algorithm is proposed. The proposed GED method successfully detected and removed the gross errors occurring in the measurements.
5. The proposed ANDDR and GED methods have been combined in order to detect and remove gross errors before the ANDDR estimation. This combination did not degrade the performance of the ANDDR algorithm.

6. The proposed package in this thesis has been successfully implemented and applied to the CSTR model of Liebman et al. [7]. Simulation results in chapter 5 demonstrated the performance improvements in applications where set-point step changes occur and/or the covariance matrix is not known.
7. The ANDDR-R + GED algorithm was also successfully implemented and tested on the simulated CSTR model of Liebman et al. [7] when the inputs are ramps or sinusoid functions. In both cases the estimates are satisfactory (chapter 6). A very small delay exists in sinusoidal input tracking; however, it is believed that this delay can be reduced further by another simple extension (using two input slopes over the window, as mentioned).
8. To study another application of the proposed ANDDR + GED algorithm, the whole package has been successfully tested on the simulated JCSTR model with more complex dynamic characteristics. The reconciled estimates depicted in chapter 7 prove the applicability of the proposed method to cases where both set-point changes and ramps exist.
9. This package with its smart tracking features is suggested for use in distributed control systems (DCSs) or chemical process control to improve process monitoring and lessen operator work load.

8.2 Future work

The next step will be an evaluation on a more realistic model of a pilot plant facility at the College of North Atlantic as part of the Petroleum Application of Wireless Systems (PAWS) project, followed by studies with actual data acquired from the plant.

Adding a dynamic model identification (DMI) feature to the proposed approach may

be a useful extension for this study. As stated in section 1.2, DMI has been addressed in Alici et al. [1], but the selection of a suitable DMI approach requires serious attention, since it can be application dependent. Therefore, the combination of a suitable DMI method and the new ANDDR and GED package is suggested as future work.

Bibliography

- [1] S. Alici, *Dynamic data reconciliation using process simulation software and model identification tools*, Ph.D. thesis, University of Texas at Austin, 2001.
- [2] S. Alici and T. F. Edgar, *Nonlinear dynamic data reconciliation via process simulation software and model identification tools*, *Industrial and Engineering Chemistry Research* **41** (2002), no. 16, 3984–3992.
- [3] C. M. Crowe, *Data reconciliation progress and challenges*, *Process Control* **6** (1996), no. 2-3, 89–98.
- [4] Z. H. Abu el zeet, V. M. Becerra, and P. D. Roberts, *Combined bias and outlier identification in dynamic data reconciliation*, *Computers and Chemical Engineering* **26** (2002), 921–935.
- [5] D. R. Kuehn and H. Davidson, *Computer control*, *Chemical Engineering Progress* **57** (1961), 44–47.
- [6] M. J. Liebman and T. F. Edgar, *Data reconciliation for nonlinear process*, *Proceedings of the AIChE Annual Meeting*, Washington, DC (1998).
- [7] M. J. Liebman, T. F. Edgar, and L. S. Lasdon, *Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques*, *Computers and Chemical Engineering* **16** (1992), no. 11, 963–986.

- [8] Maira Omana, *Robust fault detection and isolation using a parity equation implementation of directional residuals*, Master's thesis, University of New Brunswick, 2005.
- [9] T. A. Soderstrom, D. M. Himmelblau, and T. F. Edgar, *A mixed integer optimization approach for simultaneous data reconciliation and identification of measurement bias*, *Control Engineering Practice* **9** (2001), 869–876.
- [10] J. H. Taylor, *Statistical performance analysis of nonlinear stochastic systems by the monte carlo method*, *Mathematics and Computers in Simulation* **XXIII** (1981).
- [11] H. Tong and C.M. Crowe, *Detecting persistent gross errors by sequential analysis of principal components*, *AIChE* **43** (1997), no. 5, 1242–1249.

Appendix A

Table 8.1: CSTR model constants

Parameter	Value	Units
q	10.0	$cm^3 s^{-1}$
V	1000.0	cm^3
ΔH_r	-27,000.0	$cal\ gmol^{-1}$
ρ	0.001	$g\ cm^3$
C_p	1.0	$cal(gK)^{-1}$
U	5.0×10^{-4}	$cal(cm^2 sK)^{-1}$
A_R	10.0	cm^2
T_c	340.0	K
k_0	7.86×10^{12}	s^{-1}
E_A	14,090.0	K
α_d	1.0	-

Appendix B

Table 8.2: JCSTR model constants

Parameter	Value	Units
D_r	5.0	m
C_p	4.1868×1000	$J/kg.K$
ρ	997.95	kg/m^3
U	851.74	$W/m^2.K$
F_{Jout}	0.15	m^3/S
V_j	9.0	m^3
T_{jin}	419.0	K

Curriculum Vita

- Candidate's full name: Mazyar B. Laylabadi
- University attended: Iran University of Science & Technology, Bachelor in Electrical Engineering, 2002
- Publications:
 1. ANDDR with Novel Gross Error Detection and Smart Tracking System (Accepted to IFAC, INCOM Conf., May 2006, France)
 2. A Novel Adaptive Nonlinear Dynamic Data Reconciliation and Gross Error Detection Method (Submitted to IEEE International Conference on Control Applications, CCA 2006, Munich, Germany)