

City Research Online

City St George's, University of London

Citation: Abu-el-zeet, Z.H. (2000). Optimisation techniques for advanced process supervision and control. (Unpublished Doctoral thesis, City University London)

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://city-ref-test.eprints-hosting.org/id/eprint/8162/

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

OPTIMISATION TECHNIQUES FOR ADVANCED PROCESS SUPERVISION AND CONTROL

By

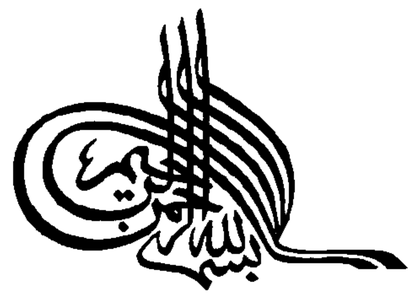
Ziad Hasan Abu-el-zeet

A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

CITY UNIVERSITY, LONDON

CONTROL ENGINEERING RESEARCH CENTRE
DEPARTMENT OF ELECTRICAL, ELECTRONIC
AND INFORMATION ENGINEERING

JULY, 2000.



To

My dear mother,

The beautiful memory of my father.

My wife, Najah.

My brothers

Imad, Jafar, Mahmood.

My sisters

Nadia, Fatima, Nawal, Firyal.

My children

Tariq, Mohammed, Muna.

The memory of

Mr. Lackson K. Chishimba BEng, MSc, MPhil.

TABLE OF CONTENTS

TABLE OF CONTENTS **3**

LIST OF TABLES **8**

LIST OF FIGURES **9**

ACKNOWLEDGEMENTS **15**

DECLARATION **16**

ABSTRACT **17**

LIST OF SYMBOLS **18**

LIST OF ABBREVIATIONS **21**

CHAPTER 1. INTRODUCTION **23**

1.1 OPTIMISATION OF INDUSTRIAL PROCESSES 23

1.2 DATA RECONCILIATION 24

1.2.1 Identification of steady-state 25

1.2.2 Gross error detection and identification 25

1.2.3 Bias detection and identification 26

1.3 MODEL PREDICTIVE CONTROL 26

1.4 SCOPE AND AIMS OF THE THESIS 27

1.4.1 Contributions of the thesis 28

1.5 THESIS OUTLINE 28

1.6 SUMMARY 31

CHAPTER 2. THE MODIFIED TWO STEP METHOD 32

2.1 INTRODUCTION 32

2.2 ISOPE ALGORITHM - A BRIEF REVIEW 33

2.3 FORMULATION OF THE MODIFIED TWO STEP METHOD 34

2.4 CASE WITH QUADRATIC OBJECTIVE, LINEAR ADAPTIVE MODEL AND LINEAR CONSTRAINTS 37

2.5 PRACTICAL VERSION OF THE MODIFIED TWO STEP ALGORITHM.. 40

2.6 MODEL IDENTIFICATION 41

2.7 DYNAMIC INTEGRATED SYSTEM OPTIMISATION AND PARAMETER ESTIMATION (DISOPE) 42

2.7.1 CASE WITH LINEAR QUADRATIC MODEL-BASED PROBLEM 46

2.8 SUMMARY 47

CHAPTER 3. DEVELOPMENTS IN PREDICTIVE OPTIMISATION 49

3.1 INTRODUCTION 49

3.2 MODEL PREDICTIVE CONTROL 49

3.2.1 Model predictive control strategy 51

3.2.2 Models used for MPC 53

3.2.3 Model predictive control algorithms 54

3.3 RECEDING HORIZON OPTIMAL CONTROL ALGORITHM 57

3.4 LINEAR MODEL IDENTIFICATION SCHEME 60

3.5 SIMULATION CASE STUDY 62

3.5.1 The ALSTOM gasifier benchmark challenge 63

3.5.1.1 Control aims 64

3.5.2 Gasifier model implementation 65

3.5.3 Model identification 65

3.5.4 Controller implementation 66

3.5.5 Controller tuning and simulations 67

3.5.6 Simulation results 67

3.6 CONCLUSIONS 69
3.7 SUMMARY 70

CHAPTER 4. STEADY-STATE DATA RECONCILIATION AND ESTIMATION OF SYSTEMATIC BIASES 82

4.1 DATA RECONCILIATION BACKGROUND 82
4.2 STEADY-STATE DATA RECONCILIATION 84
4.2.1 Benefits of data reconciliation 85
4.2.2 Sources and types of error 86
4.2.3 Variable classification 87
4.2.4 Review of previous work 89
4.2.5 Applications of data reconciliation 92
4.2.6 Formulation of the data reconciliation problem 92
4.2.6.1 Linear solution 93
4.2.6.2 Successive linearisation methods 94
4.2.6.3 Non-linear methods 94
4.2.7 Formulation of the bias estimation problem 95
4.3 THE STATIC DATA RECONCILAITION MODULE 95
4.3.1 Simulation case study 95
4.3.2 Implementation issues 97
4.3.3 Simulation results 98
4.3.3.1 Reconciliation of biased data and estimation of systematic bias 99
4.3.3.2 Estimation of physical parameters 100
4.3.3.3 Behaviour in the presence of transients 100
4.3.3.4 Behaviour in the presence of parametric differences 102
4.4 CONCLUSIONS 112
4.5 SUMMARY 113

CHAPTER 5. AUTOMATIC IDENTIFICATION OF STEADY-STATE 115

5.1 STEADY-STATE DETECTION BACKGROUND 115

5.2 REVIEW OF EXISTING METHODS 116
5.3 STEADY-STATE IDENTIFICATION METHOD - CAO AND RHINEHART 118
5.3.1 Algorithm for the identification of steady-state 122
5.3.2 Selection of λ values 123
5.4 FILTERING OF THE STEADY-STATE IDENTIFIER OUTPUT 123
5.5 IMPLEMENTATION OF THE STEADY-STATE IDENTIFICATION MODULE 124
5.5.1 Simulation results 124
5.6 USE OF THE STEADY-STATE DETECTION MODULE IN STATIC DATA RECONCILIATION 129
5.6.1 Simulation results 130
5.7 SUMMARY 134

CHAPTER 6. STEADY-STATE OPTIMISATION USING DATA RECONCILIATION AND BIAS ESTIMATION 135
6.1 STATIC DATA RECONCILIATION 135
6.2 STEADY-STATE OPTIMISATION 136
6.3 SIMULATION CASE STUDY 137
6.3.1 Implementation issues 138
6.3.2 Discussion of results 141
6.4 SUMMARY 146

CHAPTER 7. BIAS AND GROSS ERROR DETECTION IN DYNAMIC DATA RECONCILIATION 147
7.1 INTRODUCTION 147
7.2 DYNAMIC DATA RECONCILIATION 148
7.3 MOVING HORIZON ESTIMATION 149
7.3.1 Formulation of the moving horizon estimator 150
7.4 GROSS ERROR DETECTION AND IDENTIFICATION 153

7.4.1 Gross error detection and identification algorithm 157
7.5 DETECTION AND IDENTIFICATION OF SYSTEMATIC BIAS 159
7.5.1 Bias detection and identification algorithm 161
7.6 SIMULATION CASE STUDY 167
7.6.1 Implementation issues 167
7.6.2 Results 168
7.6.2.1 Gross error detection and identification 168
7.6.2.2 Bias detection and identification 170
7.6.2.3 Combined gross error and bias detection and identification 171
7.7 CONCLUSIONS 172
7.8 SUMMARY 175

CHAPTER 8. MODEL PREDICTIVE CONTROL USING DYNAMIC DATA RECONCILIATION TECHNIQUES 192

8.1 INTRODUCTION 192
8.2 SIMULATION CASE STUDIES 193
8.2.1 Results 195
8.3 CONCLUSIONS 197
8.4 SUMMARY 198

CHAPTER 9. CONCLUSIONS AND FURTHER WORK 210

9.1 CONCLUSIONS 210
9.2 SUGGESTIONS FOR FURTHER RESEARCH 215

REFERENCES 217

APPENDIX 229

A. SEQUENTIAL QUADRATIC PROGRAMMING 229

LIST OF PUBLICATIONS 231

LIST OF TABLES

Table No.		Page
3.1	Predictive control algorithm	59
3.2	Measured variable parameters	71
3.3	Manipulated variable parameters	71
3.4	Scaling factors	71
3.5	Controller tuning parameters	71
3.6	Case with 100% load and step disturbance	72
3.7	Case with 100% load and sine disturbance	72
3.8	Case with 50% load and step disturbance	72
3.9	Case with 50% load and sine disturbance	73
3.10	Case with 0% load and step disturbance	73
3.11	Case with 0% load and sine disturbance	73
3.12	Comparison of adopted MPC method with other strategies	74
4.1	Various simulations carried out on the SDR module	102

LIST OF FIGURES

Figure No.		Page
3-1	Model predictive control strategy	52
3-2	Basic structure of MPC	53
3-3	Gasifier plant functional diagram	64
3-4	Model validation plot	74
3-5	Response of process outputs to a step disturbance at 100% load	75
3-6	Response of manipulated variables to a step disturbance at 100% load	75
3-7	Response of process outputs to a sine disturbance at 100% load	76
3-8	Response of manipulated variables to a sine disturbance at 100% load	76
3-9	Response of process outputs to a step disturbance at 50% load	77
3-10	Response of manipulated variables to a step disturbance at 50% load	77
3-11	Response of process outputs to a sine disturbance at 50% load	78
3-12	Response of manipulated variables to a sine disturbance at 50% load	78
3-13	Response of process outputs to a step disturbance at 0% load	79
3-14	Response of manipulated variables to a step disturbance at 0% load	79
3-15	Response of process outputs to a sine disturbance at 0% load	80
3-16	Response of manipulated variables to a sine disturbance at 0% load	80
3-17	Response of process outputs to a sine disturbance at 0% load - extended simulation time	81
4-1	Three steps in the processing of measurement data	89
4-2	Two Continuous Stirred Tank Reactors (CSTR) in series	97
4-3	Reconciliation of measurement data when first measurement is biased	103

4-4 Reconciliation of measurement data when second measurement is biased 103
4-5 Reconciliation of measurement data when both measurements are biased 104
4-6 Reconciliation of measurement data while estimating a physical parameter when first measurement is biased 105
4-7 Estimate of the physical parameter when the first measurement is biased 105
4-8 Reconciliation of measurement data while estimating a physical parameter when both measurements are biased 106
4-9 Estimate of the physical parameter when both measurements are biased 106
4-10 Reconciliation of measurement data while estimating a physical parameter when no measurements are biased 107
4-11 Estimate of the physical parameter when no measurements are biased 107
4-12 Behaviour in the presence of a transient when no measurements are biased 108
4-13 Behaviour in the presence of a transient when the first measurement is biased 108
4-14 Behaviour in the presence of a transient when both measurements are biased 109
4-15 Behaviour in the presence of a transient when estimating a physical parameter and the first measurement is biased 110
4-16 Estimate of the physical parameter in the presence of a transient and when the first measurement is biased 110
4-17 Behaviour when large parametric differences exist between the static and dynamic models 111
4-18 Behaviour when small parametric differences exist between the static and dynamic models 111

5-1 Measured process variables and output of SSD module, filtering not applied, test based on first measurement 126

5-2 Ratio of the two variances, critical value and output of SSD module 127

5-3 Filtering technique applied to SSD algorithm, test based on first measurement 127

5-4 Filtering technique applied to SSD algorithm, test based on second measurement 128

5-5 Filtering technique applied to SSD algorithm, test based on both measurements 128

5-6 Schematic of the SDR-SSD simulation case study 129

5-7 Biased measurements, estimates and SSD module output, test based on both measurements 132

5-8 Biased measurements, estimates and SSD module output, test based on first measurement 133

5-9 Biased measurements, estimates and SSD module output, test based on both measurements using new critical values 133

5-10 Biased measurements, estimates and SSD module output, test based on both measurements using different critical values 134

6-1 Detailed schematic of module interconnections in OTISS (REACTORS-ISOPE case) 139

6-2 Schematic of case study (1), steady-state optimisation without the use of data reconciliation techniques 140

6-3 Schematic of case study (2), steady-state optimisation using data reconciliation techniques 140

6-4 Measurements and real output of the plant - case with steady state optimisation and no SDR 142

6-5 Measurements and real output of the plant - case with steady state optimisation using SDR 142

6-6 Manipulated variables - case with steady-state optimisation and no SDR 143

6-7 Manipulated variables - case with steady-state optimisation using SDR 143
6-8 Measurements and real output of the plant - case with steady-state optimisation and no SDR, second measurement contains bias 144
6-9 Measurements and real output of the plant - case with steady-state optimisation using SDR, second measurement contains bias 144
6-10 Manipulated variables - case with steady-state optimisation and no SDR, second measurement contains bias 145
6-11 Manipulated variables - case with steady-state optimisation using SDR, second measurement contains bias 145
7-1 Flow chart - bias detection method 1 164
7-2 Flow chart - bias detection method 2 165
7-3 Flow chart - A new bias detection algorithm 166
7-4 Effects of an outlier on the first measurement - gross error detection algorithm disabled 176
7-5 Effects of an outlier on the first measurement - gross error detection algorithm enabled 177
7-6 Effects of an outlier on the second measurement - gross error detection algorithm disabled 178
7-7 Effects of an outlier on the second measurement - gross error detection algorithm enabled 179
7-8 Effects of outliers on the both measurements - gross error detection algorithm disabled 180
7-9 Effects of outliers on the both measurements - gross error detection algorithm enabled 181
7-10 Effects of outliers in the dynamic case - gross error detection algorithm disabled 182
7-11 Effects of outliers in the dynamic case - gross error detection algorithm enabled 183
7-12 Performance of the bias detection algorithm in the case of static data 184

7-13 Performance of the bias detection algorithm in the case of dynamic data 185
7-14 Effects on bias detection algorithm when the bias changes 186
7-15 Performance of the bias detection algorithm with changing bias and transients 187
7-16 Effects of bias and outliers when both bias and gross error detection algorithms are disabled 188
7-17 Effects of bias and outliers when both bias and gross error detection algorithms are enabled 189
7-18 Effects of bias and outliers when both bias and gross error detection algorithms are disabled 190
7-19 Effects of bias and outliers when both bias and gross error detection algorithms are enabled 191
8-1 Schematic of case study (1) - MPC without DDR 194
8-2 Schematic of case study (2) - MPC using DDR 195
8-3 MPC scheme, DDR disabled, regulatory objective, measurements are bias free. 199
8-4 MPC scheme, DDR disabled, regulatory objective, one measurement is biased. 200
8-5 MPC scheme, DDR enabled, regulatory objective, one measurement is biased. 201
8-6 MPC scheme, DDR disabled, economic objective, one measurement is biased. 202
8-7 MPC scheme, DDR enabled, economic objective, one measurement is biased. 203
8-8 MPC scheme, DDR disabled, Model adaptation disabled. 204
8-9 MPC scheme, DDR enabled, Model adaptation disabled. 205
8-10 MPC scheme, DDR disabled, economic objective, measurements are bias free. 206

8-11 MPC scheme, DDR enabled, economic objective, measurements are bias free. 207
8-12 MPC scheme, DDR disabled, combined objective, one measurement is biased. 208
8-13 MPC scheme, DDR enabled, combined objective, one measurement is biased. 209

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and gratitude to my supervisor, Professor Peter D. Roberts for his guidance, encouragement and support throughout the course of this research. Thank you Professor for the superb supervision, the quality of which is absolutely first class.

I can never sufficiently thank Dr. Victor M. Becerra who has also actively participated in the excellent supervision of this research. Thank you Victor for all the time you spent in helping me, for the guidance and encouragement and thank you for all the various pointers along the way.

The financial support of the following organisations is gratefully acknowledged:

- The United Kingdom Engineering and Physical Sciences Research Council (EPSRC) grant no. GR/L64478.
- Bin Hamoodah Company, Abu Dhabi, UAE.

The support and provision of software licenses for OTISS™ by Aspentech (U.K.) Ltd. are gratefully acknowledged.

I would like to thank my mother for her patience and prayers throughout and my wife Najah for her loving support, encouragement and hard work looking after the children throughout the past few years.

My thanks are due to Jalal and Mo Omar for their help and encouragement.

Last, but by no means least, I would like to express my thanks to my fellow research colleagues in the Control Engineering Research Centre for their help and encouragement. Thanks to Daniel, Moufid, Stavros, Costas, Ali, Ermina, Tanyia and Gabriel. My thanks also to Joan Rivellini and Linda Carr in the Electrical Engineering office for their tireless and caring attitude to help at all times.

DECLARATION

The author grants powers of discretion to the University Librarian to allow this thesis to be copied in whole or part without further reference to him. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

ABSTRACT

This thesis is concerned with the use and development of optimisation techniques for process supervision and control. Two major areas related to optimisation are combined namely model predictive control and dynamic data reconciliation. A model predictive control scheme is implemented and used to simulate the control of a coal gasification plant. Static as well as dynamic data reconciliation techniques are developed and used in conjunction with steady-state optimisation and model predictive control schemes. The inaccuracy of process data due to measurement errors can be considerably reduced by data reconciliation techniques. This in turn improves process knowledge and control system performance. The static and dynamic data reconciliation techniques developed in this thesis are tested using dynamic models of process plants.

In the steady-state case, a static data reconciliation algorithm that uses a static model of the process is implemented. This algorithm has capabilities of estimating measured variables, unmeasured variables, systematic bias and unknown physical parameters. The technique is applied to static optimisation to show the improvements in performance of the optimiser when using reconciled data. In order for static data reconciliation to be applied, it is necessary to employ a steady-state detection scheme since the underlying assumption is that the process is at steady-state. An algorithm for steady-state detection is implemented and tested in conjunction with the static data reconciliation technique.

In the dynamic case, a moving horizon estimator that employs a dynamic model of the process is used to reconcile dynamic process data. An algorithm for the detection, identification and elimination of gross errors is implemented and tested. Furthermore, an algorithm for the detection and identification of systematic bias is developed and implemented. These techniques are then applied in combination to the dynamic model of a process. The effect of dynamic data reconciliation on the performance of model predictive control is observed by means of applying the above techniques to such a scheme.

The various algorithms outlined above are implemented in software and tested using appropriate simulations. It is shown that it is possible to implement a steady-state detection algorithm and to successfully use it in conjunction with static data reconciliation. The application of static data reconciliation to steady-state optimisation shows a marked improvement in the performance of the optimiser. It is further shown that it is possible to combine bias and gross error detection and identification algorithms and to successfully apply them to dynamic data reconciliation procedures. The application of dynamic data reconciliation techniques to model predictive control shows improvement in the performance in cases where the objective is not purely economic.

LIST OF SYMBOLS

Symbol	Description
$A^*(q^{-1})$	Polynomial matrix
$B^*(q^{-1})$	Polynomial matrix
b	Vector of bias parameters
$\hat{b}$	Bias estimate
$\bar{\hat{b}}$	Average bias over the data window
$C^*(q^{-1})$	Polynomial matrix
e_i	Residual of ith process measurement
F	Steady-state objective
f^*	Set of discrete time state equations
f	Approximate dynamic model of f^*
f_x	Mapping of state equations
G	Static matrix gain
g	Mapping of output dependent inequality constraints
h	Integration step
J	Objective function
J^*	Performance index
J_e	EOP performance index
J_m	Model-based (MOP) performance index
J_M	Modified model-based (MMOP) performance index
K^*	Process static relationships between manipulated and measured variables
K	Approximate model of K^*
k	Discrete sampling time index
L^*	Discrete performance (or weighting) function
L	Simplification of L^*
M	Control horizon
N	Prediction horizon

nh	Number of measurements in the data window
P	Penalty term
p	Costate vector
p_u	Vector of price coefficients for manipulated variables
p_y	Vector of price coefficients for measured variables
Q	Intermediate state weighting matrix
Q_y	Output target weighting matrix
q^{-1}	Backward shift operator
R	Control weighting matrix
R_i	Ratio of two variance measures (variability measure)
r	Convexification factor
S	Bias distribution matrix
$S_{1,i}, S_{2,i}$	Estimates of noise variance
u	Vector of decision variables
$u_{\min}, u_{\max}$	Control bounds
u_r	Reference value for the manipulated variable
V	Covariance matrix
v	Vector of manipulated variables
W_i	Trust degree of measurement
w	Set of relaxation variables
x	State vector
y^*	Vector of measured variables
y	Vector of model outputs
$\hat{y}$	Predicted future outputs
$\overline{y}$	Estimates of process variables
$y_{\min}, y_{\max}$	Bounds on measured variables
y_m	Vector of measured process variables
y_r	Reference value for the measured variable
y_{true}	Vector of true process variables
Δ	Increment operator
Φ	Terminal state weighting matrix

Ψ	Vector of inequality constraints
α	Vector of model parameters
β	Lagrange multiplier vector
ε	Vector of random measurement errors
φ	Scalar valued terminal weighting function
γ	Discrete parameter
λ	Lagrange multiplier vector
$\lambda_1, \lambda_2, \lambda_3$	Filter factors
μ	Penalty factor
ρ	Penalty factor
σ	Measurement noise standard deviation
ψ	Mapping of inequality constraints

LIST OF ABBREVIATIONS

Abbreviation	Description
ARMAX	Auto Regressive Moving Average with Exogeneous input
ARX	Auto Regression with Extra inputs
CSTR	Continuous Stirred Tank Reactors
DDR	Dynamic Data Reconciliation
DIMT	Dynamic Integrated Measurement Test
DISOPE	Dynamic Integrated System Optimisation and Parameter Estimation
DMC	Dynamic Matrix Control
DMT	Dynamic Measurement Test
EKF	Extended Kalman Filter
EMT	Extended Measurement Test
EOP	Expanded Optimal control Problem
GLR	Generalized Likelihood Ratio
GPC	Generalised Predictive Control
IAE	Integrated Absolute Errors
IGCC	Integrated Gasification Combined Cycle
IMT	Iterative Measurement Test
ISOPE	Integrated System Optimisation and Parameter Estimation
LQ	Linear Quadratic
MAC	Model Algorithmic Control
MIMT	Modified Iterative Measurement Test
MMD	Mean Minimum Distance
MMOP	Modified Model based Optimal control Problem
MOP	Model based Optimal control Problem
MPC	Model Predictive Control
MT	Measurement Test
NDDR	Nonlinear Dynamic Data Reconciliation

NLMPC	Non-Linear Model Predictive Control
NLP	Non-Linear Programming
NLQDMC	Non-Linear Quadratic Dynamic Matrix Control
NT	Nodal Test
ODE	Ordinary Differential Equation
PCT	Principal Component Test
pdf	probability density function
PI	Proportional plus Integral
PRBS	Pseudo Random Binary Sequence
QDMC	Quadratic Dynamic Matrix Control
QP	Quadratic Programme
RHOP	Receding Horizon Optimisation Problem
SC	Screened Combinatorial method
SDR	Static Data Reconciliation
SLHE	Successively Linearised Horizon based Estimator
ROP	Real Optimal control Problem
SQP	Sequential Quadratic Programming
SSD	Steady-State Detection
SSPC	Steady-State Predictive Control
UBET	Unbiased Estimation Technique

CHAPTER 1

INTRODUCTION

"Engineering is concerned with understanding and controlling the materials and forces of nature for the benefit of humankind. Control system engineers are concerned with understanding and controlling segments of their environment, often called *systems*, in order to provide useful economic products for society. Perhaps the most characteristic quality of control engineering is the opportunity to control machines, and industrial and economic processes for the benefit of society", Dorf (1992).

1.1 OPTIMISATION OF INDUSTRIAL PROCESSES

The principle of optimality was first mentioned by Johann Bernoulli in 1696 in connection with the Brachistochrone problem. Various optimality principles were investigated in the 1600s by P. de Fermat and in the 1700s by L. Euler and Hamilton. In 1958 a Russian group headed by L.S. Pontryagin developed the *maximum principle* which solved optimal control problems relying on the calculus of variations (see Pontryagin et al., 1962). Kalman (1960a, 1960b) published some major work concerning optimal control of systems and discussed optimal filtering and estimation theory.

In optimisation the optimum operating conditions of a system are predicted such that some performance criterion is satisfied. In an industrial process, for example, the criterion for optimum operation is often in the form of minimum cost, where

the product cost can depend on a large number of interrelated controlled parameters.

The fundamental elements of steady-state process optimisation, sometimes referred to as *optimising control* are a performance criterion (index) and a mathematical model of the plant along with relevant process constraints. The application of steady-state optimisation produces a set of optimal controller set-points. These set-points define the optimum operating point at which the process should be regulated until a change in economic objectives is desired.

Dynamic opimisation or *optimal control* requires a dynamic model of the process since the aim is to manipulate certain process inputs so as to optimise a dynamic criterion during transient conditions.

1.2 DATA RECONCILIATION

Process data is the foundation upon which all control and evaluation of process performance is based. Inaccurate process data can easily lead to poor decisions which will adversely affect many parts of the process. Many process control and optimisation activities are also based on small improvements in process performance; errors in process data can easily exceed the actual changes in process performance. Moreover, because of the immense scale of operation, the impact of any error is greatly magnified in absolute terms (Mah et al., 1976). When flawed information is used for state estimation and process control, the state of the system is misrepresented and the resulting control performance may be poor and can lead to suboptimal and even unsafe process operation (Liebman et al., 1992).

Data reconciliation is the adjustment of a set of data so the quantities derived from the data obey natural laws, such as material and energy balances. The adjustments are made using redundancies in the measurements. After adjustment, the material

and, if considered, the energy balances are satisfied exactly (Bodington, 1995). Data reconciliation may be performed on a set of steady state data, using a steady-state model of the process or it may be applied to dynamic data, using a dynamic model of the process.

1.2.1 Identification of steady-state

Some control and estimation techniques that use steady-state models assume that process measurements correspond to steady-state conditions. Steady-state models are widely used in model identification, optimisation and data reconciliation. For the purpose of data reconciliation, it is important to know when the system is at steady-state in order to be able to apply static data reconciliation techniques. The identification of steady state is also applicable to the compression of process data (Mo et al., 1998) and fault diagnosis.

Although there are a few existing methods for steady-state identification, work in this field has been limited. A survey of methods for detecting changes in signals which are applicable to data reconciliation was published by Basseville (1988). A brief review of some of the methods available for detecting changes in steady-state was presented by Crowe (1996).

1.2.2 Gross error detection and identification

It is quite natural to assume the presence of random, normally distributed measurement errors, with zero mean and known covariance to be present in the process data. This is treated using straightforward data reconciliation procedures. However, a different type of error known as a gross error is sometimes present in the data. Gross errors can be subdivided into two categories: *measurement related* such as malfunctioning sensors and *process related* such as process leaks.

The presence of gross errors invalidates the statistical basis of data reconciliation procedures and thus their treatment is essential. The treatment of gross errors can be divided into three stages (Madron, 1992). In the first stage *gross error detection* is performed to ascertain whether gross errors are present in the measurements. If the presence of gross errors is detected the next stage is the *identification* of the sources of those errors. The final stage is the *elimination* of the gross errors.

Considerable effort by a number of researchers has been expended on developing methods for gross error identification.

1.2.3 Bias detection and identification

A further type of error that is sometimes classified as a special type of gross error by some authors is systematic bias. This type of error usually occurs when measurement devices provide consistently erroneous values and may be caused by incorrect calibration of measurement devices. Again it is important that data containing such errors is identified and either treated or removed before the process of data reconciliation takes place.

Only a handful of researchers have addressed the problem of identifying systematic bias and most of the previous work has concentrated on steady-state processes.

1.3 MODEL PREDICTIVE CONTROL

Model-based predictive control has been the subject of intensive research for about 20 years. The technique is based on the receding horizon concept where the current control action is obtained by solving a finite horizon open-loop optimal control problem at each sampling instant using the current state of the plant as the

initial state. By minimising an objective function, the optimisation yields an optimal control sequence from which only the first control is applied to the plant.

Model predictive control (MPC) has enjoyed wide acceptance in industrial applications such as the petro-chemical industry. This success has been mainly due to the fact that MPC algorithms handle process constraints and multivariable processes. Furthermore, MPC algorithms are intuitive and relatively easy to tune.

The literature is rich with hundreds of important contributions in the field. A good survey, among a number of others, is the one recently published by Mayne et al. (2000).

1.4 SCOPE AND AIMS OF THE THESIS

The aims of this thesis can be set out as follows:

- To develop a static data reconciliation module which should have capabilities to estimate measured and unmeasured process variables, systematic bias and unknown physical parameters.
- To find and implement a practical algorithm for the identification of steady-state.
- To apply static data reconciliation techniques to static optimisation in order to investigate their potential.
- To use a model predictive control technique in the control of a practical process.
- To develop a bias detection and identification algorithm.
- To implement a gross error detection algorithm specifically for the identification and elimination of outliers on the process variables.
- To investigate the potential of a dynamic data reconciliation technique based on the moving horizon concept which also uses the bias and gross error detection and identification algorithms outlined above.

- To apply the above dynamic data reconciliation techniques to a model predictive control scheme.
- To implement all the above algorithms in software and to test their performance using simulation case studies.

1.4.1 Contributions of the thesis

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

- Implementation of a static data reconciliation module.
- Implementation of an algorithm for the identification of steady-state and its refinement such that errors in the identification at the transition points between steady and non-steady state conditions are considerably reduced.
- The application of static data reconciliation to static optimisation.
- Development and implementation of a bias detection and identification algorithm.
- Implementation of a gross error detection and identification algorithm.
- Implementation of combined bias and gross error detection and identification algorithms within a dynamic data reconciliation framework.
- The application of dynamic data reconciliation techniques to model predictive control.
- The implementation of all the above algorithms in C/C++ code and interfacing them with the industrial process simulation software Aspen-OTISS.

1.5 THESIS OUTLINE

This thesis is structured in the following manner:

Chapter 2 introduces static optimisation and specifically the modified two step algorithm otherwise known as Integrated System Optimisation and Parameter Estimation (ISOPE) developed by Roberts (1979). A short review of the ISOPE

family of algorithms is presented and a practical formulation developed by Becerra and Roberts (2000) is reproduced. Finally, the dynamic version of ISOPE known as DISOPE is introduced. DISOPE is used in chapter 3 while ISOPE is used later in chapter 6.

Chapter 3 introduces the area of model predictive control and reviews some of the main algorithms available and major research activities in the field. A state-space model predictive control algorithm employing the receding horizon concept and developed at City University is used to control a gasifier plant as part of a benchmark challenge set by ALSTOM Mechanical Engineering Centre. The gasifier plant is used for the generation of power from coal. The control scheme is implemented and simulation results are presented.

Chapter 4 introduces the area of static data reconciliation and presents a historical review of the work carried out in the field. A static data reconciliation module using sequential quadratic programming having capabilities of estimating measured and unmeasured process variables, estimating bias and physical parameters is implemented. The underlying assumptions here are that the process is at steady-state, that there are no gross errors (outliers) in the data and that it is known a priori which measurements (if any) are corrupted by systematic bias. Simulations are carried out, the results from which are then presented.

Chapter 5 presents a workable solution to the problem of steady-state identification. Following a review of the work published on the subject, an algorithm suggested by Cao and Rhinehart (1995) is implemented. This is first tested separately on a model of a chemical reactor system and then it is tested in conjunction with the static data reconciliation module developed in chapter 4. The aim of this set up is to enable or disable the static data reconciliation module based on the information obtained from the steady-state detection algorithm regarding the actual state of the system. Simulation results from both exercises are presented.

Chapter 6 applies the static data reconciliation techniques developed in chapter 4 to static optimisation introduced in chapter 2. For the purpose of comparison. two case studies are investigated. In the first case study static optimisation is performed on the untreated biased data. In the second case study, however, the process data is first reconciled before the static optimisation is performed. Simulation results from both case studies are presented to show how reconciling the data can improve the performance of static optimisation techniques.

Chapter 7 introduces the areas of dynamic data reconciliation and bias and gross error detection and identification. Historical reviews of published research relating to these fields are presented. A dynamic data reconciliation algorithm based on the moving horizon concept is used to reconcile dynamic process data. An algorithm for the detection, identification and elimination of gross errors is implemented. Algorithms for the detection and identification of systematic bias are developed and an intuitive method is implemented for that purpose. Simulation studies are carried out to test the various algorithms in isolation as well as in combination.

Chapter 8 applies the dynamic data reconciliation techniques developed and implemented in chapter 7 to a model predictive control scheme. The advantages and disadvantages of using dynamic data reconciliation in model predictive control are highlighted through a comparison between a scheme that uses dynamic data reconciliation and one that does not. Results from simulation studies are then presented.

Chapter 9 draws some conclusions from the results and makes a number of suggestions for further research related to the work in this thesis.

1.6 SUMMARY

In this chapter, a general introduction to the area of optimisation has been presented. The areas of data reconciliation and model predictive control which are related in their common use of optimisation techniques have also been introduced. The scope and aims of the thesis have been clearly defined and a list of the major contributions of the thesis has been presented. An outline of the thesis chapters has also been given.

The following chapter introduces static and dynamic optimisation techniques used later in this thesis.

CHAPTER 2

THE MODIFIED TWO-STEP METHOD

2.1 INTRODUCTION

In steady-state optimisation, the calculation of the optimal set points is usually based on a mathematical model of the plant. Since the mathematical model is not an exact representation of the real plant and the process will generally operate in a changing environment, the calculated set-points will only be optimal for the model. It is necessary, therefore, to make sure that the mathematical model is adaptable.

In order to overcome the problem of model-reality differences, the well known *two-step* method was proposed. In this method, parameters which are estimated by comparing model based and measured outputs are contained in the model. This defines the parameter estimation problem. Further, the steady-state model is used to determine the optimum controller set point values to satisfy a given performance index (e.g. to maximise a particular product). This forms the system optimisation problem. Since, in general, the model will not be an exact representation of the real process, the two problems interact. The solution of the optimisation problem is dependent upon the values of the model parameters and the parameter estimates will change according to the controller settings.

The two-step technique treats the system optimisation and parameter estimation problems separately and solves them repeatedly until convergence is obtained. However, Durbeck (1965) showed that this simple procedure will not converge to the correct optimum when the model is inaccurate unless the derivatives of the

real process outputs with respect to the controller set points are matched exactly with the corresponding derivatives in the model.

Roberts (1979) proposed the *modified two-step* method which allows interaction between the parameter estimation and system optimisation problems. This method came to be known as Integrated System Optimisation and Parameter Estimation (ISOPE).

2.2 ISOPE ALGORITHM - A BRIEF REVIEW

Like the standard two-step method, ISOPE is iterative in nature using repeated solutions of optimisation and estimation of parameters within the model used for calculating the optimum. Since it was first proposed, a number of researchers have worked on the algorithm. Roberts and Williams (1981) investigated the performance and studied the stability and convergence properties of the algorithm. Conditions which ensure that the algorithm converges were developed by Brdys and Roberts (1987). Ellis et al. (1988) compared the modified two-step method to two other techniques one using a direct approach while the other used an approximate linear model. Four forms of the ISOPE family of algorithms were applied to a mixing process and compared (Ellis et al., 1993). A considerable disadvantage of the modified two-step method is that real process derivatives have to be measured using perturbation techniques. To overcome this considerable disadvantage, Zhang and Roberts (1990) employed the dynamic model identification method of Bamberger and Isermann (1978) and applied it successfully to ISOPE. Augustin and Roberts (1993) presented a hierarchical version of ISOPE. Becerra and Roberts (2000) developed a practical version of ISOPE. An extensive review of the ISOPE algorithms was published by Roberts (1995).

Further work extending the capabilities of the modified two-step method to dynamic optimal control was done. This came to be known as Dynamic ISOPE or DISOPE and is introduced in Section 2.7.

2.3 FORMULATION OF THE MODIFIED TWO-STEP METHOD

The following formulation of the modified two-step method (ISOPE) is a revised practical version presented by Becerra and Roberts (2000). The objective used is a function with quadratic and linear terms and the model used for the computations as well as the inequality constraints are linear. The derivatives of the real process outputs are approximated by an identified model. This version of the modified two-step method requires the solution of a quadratic program at every iteration.

The steady-state optimisation problem consists of finding the values of manipulated variables that minimise an objective function subject to steady-state relationships of the process and constraints on manipulated and measured variables, if appropriate. The problem can be stated as follows:

Problem 1:

$$\min_{v} \; J(y^*, v) \tag{2.1}$$

subject to:

$$y^* = K^*(v) \tag{2.2}$$

$$g(y^*) \leq 0 \tag{2.3}$$

$$u_{\min} \leq v \leq u_{\max} \tag{2.4}$$

where $y^* \in \Re^{n_y}$ is a vector of measured variables, $v \in \Re^{n_u}$ is a vector of manipulated variables, $J: \Re^{n_y} \times \Re^{n_u} \rightarrow \Re$ is the objective function, $K^*: \Re^{n_u} \rightarrow \Re^{n_y}$ represents the process static relationships between manipulated

variables v and measured variables y^*, $g: \Re^{n_y} \rightarrow \Re^{n_g}$ is a mapping of output dependent inequality constraints.

In reality, the true process mapping $K^*(\cdot)$ is not known exactly and therefore a model is introduced to represent the real system. Consider the following model of the real system:

$$y = K(u,\alpha) \tag{2.5}$$

where $y \in \Re^{n_y}$ is a vector of model outputs, $u \in \Re^{n_u}$ is a vector of decision variables, $K: \Re^{n_u} \times \Re^{n_\alpha} \rightarrow \Re^{n_y}$ is an approximate model of K^*, and $\alpha \in \Re^{n_\alpha}$ is a vector of model parameters.

Consider now the following problem, which is based on a mathematical model known to be an approximation of the real process. The problem is made equivalent to problem 1 by the introduction of suitable equality constraints:

Problem 2:

$$\min_u \; J(y,u) \tag{2.6}$$

subject to:

$$y = K(v,\alpha) \tag{2.7}$$

$$K^*(v) = K(v,\alpha) \tag{2.8}$$

$$v = u \tag{2.9}$$

$$g(y) \leq 0 \tag{2.10}$$

$$u_{\min} \leq u \leq u_{\max} \tag{2.11}$$

Note that in equation (2.8) it is assumed that there exists a value α such that the equality holds. The variable u is a separation variable. By introducing the model parameters α and introducing u and equating it to v (equation (2.9)), the problems of optimisation and parameter estimation are separated into two independent sub-problems.

Analysis of the optimality conditions of problem 2 shows that the optimality conditions of problem 1 are satisfied after convergence by iteratively solving the following problem (based on a model of the process), given the values of $\alpha, \lambda, v, y^*, \rho \; and \; r$:

Problem 3:

$$\min_{u,w} \quad J(K(u,\alpha),u) - \lambda u + \frac{1}{2}\rho w^T w + \frac{1}{2} r \|u - v\|^2 \tag{2.12}$$

subject to:

$$g(K^*(v)) + M(u - v) + w \leq 0 \tag{2.13}$$

$$\overline{u}_{\min} \quad \leq \quad u \quad \leq \quad \overline{u}_{\max} \tag{2.14}$$

where

$$\overline{u}_{\min} \quad = \quad \max(u_{\min}, v - \delta) \tag{2.15}$$

$$\overline{u}_{\max} \quad = \quad \min(u_{\max}, v + \delta) \tag{2.16}$$

where δ_j is the maximum allowed value of $|u_j - v_j|$, $j = 1, \cdots, n_u$, M is given by:

$$M \quad = \quad \left[\frac{\partial g(y)}{\partial y}\right]_{y=y^*} \times \left[\frac{\partial K^*(\mu)}{\partial \mu}\right]_{\mu=v} \tag{2.17}$$

λ is computed from:

$$\lambda \quad = \quad \left[\frac{\partial J(K(\mu,\alpha),\mu)}{\partial \mu}\right]^T_{\mu=v} - \left[\frac{\partial K^*(\mu)}{\partial \mu}\right]^T_{\mu=v} \times \left[\frac{\partial J(y,v)}{\partial y}\right]^T_{y=y^*} \tag{2.18}$$

and α is obtained from:

$$y^* - K(v,\alpha) = 0 \tag{2.19}$$

Equation (2.13) is a linear approximation to (2.10), $w \in \Re^{n_g}$ is a set of relaxation variables and ρ is a penalty factor. This penalty relaxation technique is used to allow the treatment of output dependent constraints as *soft* constraints. In this way output constraints are enforced when they are feasible, and any violations are minimised when the constraints are infeasible. Also the factor $\frac{1}{2} r \|u - v\|^2$ is added to convexify the objective function, which is used in non-convex and other difficult cases to improve the convergence of the algorithm when problems are encountered ($r \geq 0$ is called the convexification factor and is treated as a tuning parameter).

As can be seen from equations (2.17) and (2.18), process derivative measurements are theoretically required by the modified two-step method. The application of perturbations on the manipulated variables to estimate the process derivatives by finite differences has been regarded as inefficient in the case of slow, multivariable processes. Alternative methods have been devised to replace the real derivatives by suitable approximations. The earlier mentioned dynamic model identification method by Bamberger and Isermann (1978) is one such method. This was introduced into the Modified two-step method by Zhang and Roberts (1990) and is used here.

2.4 CASE WITH QUADRATIC OBJECTIVE, LINEAR ADAPTIVE MODEL AND LINEAR CONSTRAINTS

Assume that the objective J is a quadratic function:

$$J(y,u) = \frac{1}{2}(y - y_r)^T P_y (y - y_r) + \frac{1}{2}(u - u_r)^T P_u (u - u_r) + p_y{}^T y + p_u{}^T u \tag{2.20}$$

where P_y and P_u are matrices of the appropriate dimensions, $p_y \in \Re^{n_y}$ and $p_u \in \Re^{n_u}$ are vectors of price coefficients for the measured and manipulated variables, respectively, y_r and u_r are reference values for the measured and manipulated variables, respectively.

Assume that the model (2.5) is linear:

$$y = K(u,\alpha) = Gu+\alpha \tag{2.21}$$

where the static matrix gain G is updated periodically by a system identification method (see Section 2.6).

Assume also that the output dependent constraints (2.10) are linear:

$$g(y) = Ay-b \leq 0 \tag{2.22}$$

A common form of (2.22) is $y_{\min} \leq y \leq y_{\max}$ which may be written as:

$$\underbrace{\begin{bmatrix} -I_{n_y} \\ I_{n_y} \end{bmatrix}}_{A} y - \underbrace{\begin{bmatrix} -y_{\min} \\ y_{\max} \end{bmatrix}}_{b} \leq 0 \tag{2.23}$$

Then, the calculation of α is reduced to

$$\alpha = y^* - Gv \tag{2.24}$$

and the calculation of λ simplifies to

$$\lambda = P_u(v-u_r)+p_u-G^T P_y(y^*-y_r)-G^T p_y \tag{2.25}$$

where the derivative of the real process function $\partial K^*(\mu)/\partial\mu\big|_{\mu=v}$ has been replaced by its approximation, the identified static gain G.

Equation (2.13) can now be written as follows:

$$Ay^* - b + AG(u-v) + w \leq 0 \tag{2.26}$$

where $\partial K^*(\mu)/\partial\mu\big|_{\mu=v}$ has also been approximated by G.

It follows by derivation that problem 3 reduces to the following Quadratic Programming problem:

Problem 4:

$$\min_x \ \frac{1}{2}x^T Hx + f^T x \tag{2.27}$$

subject to:

$$\overline{A}x \ \leq \ \overline{b} \tag{2.28}$$

$$x_{\min} \leq x \leq x_{\max} \tag{2.29}$$

where:

$$x \ = \ \begin{bmatrix} u \\ w \end{bmatrix} \tag{2.30}$$

$$H \ = \ \left[\begin{array}{c|c} G^T P_y G + P_u + rI_{n_y} & 0_{n_u \times n_g} \\ \hline 0_{n_g \times n_u} & \rho I_{n_g} \end{array}\right] \tag{2.31}$$

$$f \ = \ \begin{bmatrix} G^T P_y(\alpha - y_r) - P_u u_r + p_u - \lambda + G^T p_y - rv \\ \hline 0_{n_g \times 1} \end{bmatrix} \tag{2.32}$$

$$\overline{A} \ = \ \left[AG \,\middle|\, I_{n_g}\right] \tag{2.33}$$

$$\overline{b} \ = \ \left[b - Ay^* - AGv\right] \tag{2.34}$$

The above formulation gives rise to the following ISOPE algorithm.

2.5 PRACTICAL VERSION OF THE MODIFIED TWO-STEP ALGORITHM

The following is a practical version of the modified two-step algorithm as set out by Becerra and Roberts (2000). The algorithm is designed to drive the real process to its true optimum as model-plant mismatch is addressed in its formulation. Achieving the true optimum is subject to the convergence of the procedure and the accuracy of the estimates of G.

Algorithm 2.1: The modified two-step (ISOPE) Algorithm

Data: $P_y, P_u, p_y, p_u, y_r, u_r, A, b, r, \rho, k, i = 0, v^{(0)}$ and means for measuring $y^{*(i)}$ and computing $G^{(i)}$.

Step 1: Apply the current input $v^{(i)}$ to the plant, wait for a steady-state to be reached and measure the process output $y^{*(i)}$.

Step 2: Update the static gain $G^{(i)}$ by using an identification method.

Step 3: Compute $\alpha^{(i)}$ using (2.24) and $\lambda^{(i)}$ using (2.25).

Step 4: Solve Problem 4 using QP to obtain the next input candidate $u^{(i)}$.

Step 5: Compute the next process input by means of the following relaxation formula, which is introduced to regulate convergence:

$$v^{(i+1)} = v^{(i)} + k(u^{(i)} - v^{(i)}) \tag{2.35}$$

where $k \in [0, 1]$ is a relaxation gain.

Step 6: Set $i = i + 1$ and go to step 1.

2.6 MODEL IDENTIFICATION

As mentioned earlier, a way of estimating the derivatives of the real process mapping $K^*(\cdot)$ with respect to the manipulated variables, is to identify a dynamic model on-line and to reduce it to a steady-state model. For the purpose of this work, a multivariable ARMAX model with the following structure is identified:

$$\begin{aligned} y(k) = &- A_1 y(k-1) - \cdots - A_{n_a} y(k-n_a) \\ &+ B_1 u(k-1) + \cdots + B_{n_b} u(k-n_b) \\ &+ \varepsilon(k) + C_1 \varepsilon(k-1) + \cdots + C_{n_c} \varepsilon(k-n_c) + c \end{aligned} \tag{2.36}$$

where $y \in \Re^{n_y}$ is the vector of measured outputs, $u \in \Re^{n_u}$ is the vector of process inputs, $\varepsilon \in \Re^{n_y}$ is assumed to be zero mean white noise, k is a discrete time index, A_i, B_i, C_i are matrix coefficients of the appropriate dimensions, and $c \in \Re^{n_y}$ is an offset vector.

The identification algorithm used for this work is a multivariable moving horizon least squares based method, which is described in detail in Becerra et al. (1998a). Note that it is often necessary to add small perturbation signals to the manipulated variables, such that the inputs are sufficiently exciting and a model can be estimated from the measured data.

A static model is obtained by assuming that outputs y and inputs u are at steady-state , and that the noise ε is zero. This gives the following input-output relationship:

$$y = \left[I_{n_y} + A_1 + \cdots + A_{n_a} \right]^{-1} \left[B_1 + \cdots + B_{n_b} \right] u + c = Gu + c \tag{2.37}$$

As mentioned previously, further work extending the capabilities of the modified two-step method to dynamic optimal control was done. This came to be known as

Dynamic ISOPE or DISOPE. The following is an introduction to the formulation of the DISOPE algorithm.

2.7 DYNAMIC INTEGRATED SYSTEM OPTIMISATION AND PARAMETER ESTIMATION (DISOPE)

Dynamic Integrated System Optimisation and Parameter Estimation (DISOPE) was developed by Roberts (1992). Extensive further work on the algorithm was done by Becerra (1994) and Becerra and Roberts (1996) to include, among other things, constraints handling and application to batch processes and nonlinear predictive control. DISOPE has been successfully applied within a number of model predictive control schemes, see for instance, Becerra et al. (1996, 1997, 1998a, 1998b).

The formulation of the DISOPE algorithm is as follows. Suppose that the real plant dynamics are described by the following nonlinear time-varying difference equation

$$x(k+1) = f^*(x(k), u(k), k) \tag{2.38}$$

where k is a discrete sampling time index, $f^*: \Re^n \times \Re^m \times \Re \to \Re^n$ represents a set of discrete-time state equations which describe the process with state $x(k) \in \Re^n$ and control input $u(k) \in \Re^m$. Further assume that the following performance index has been chosen:

$$J^* = \varphi(x(N_f)) + \sum_{k=N_0}^{N_f-1} L^*(x(k), u(k), k) \tag{2.39}$$

where $[N_0, N_f]$ is the fixed interval of sampling indices of interest, $\varphi: \Re^n \to \Re$ is a scalar valued terminal weighting function and $L^*: \Re^n \times \Re^m \times \Re \to \Re$ is a discrete performance (or weighting) function.

If the state of the system at the initial sampling time index N_0 is assumed known, with value $x(N_0) = x_0$ and if no constraints on the values of control and state variables are taken into consideration, apart from the dynamic constraint (2.36), the discrete time real optimal control problem (ROP) can be formulated as follows:

ROP

$$\min_{\substack{u(k) \\ k\in[N_0, N_f-1]}} J^* = \varphi(x(N_f)) + \sum_{k=N_0}^{N_f-1} L^*(x(k), u(k), k) \tag{2.40}$$

subject to

$$x(k+1) = f^*(x(k), u(k), k) \tag{2.41}$$

$$x(N_0) = x_0 \tag{2.42}$$

Instead of solving ROP, the following, possibly simplified, discrete-time model-based optimal control problem (MOP) is considered:

MOP

$$\min_{\substack{u(k) \\ k\in[N_0, N_f-1]}} J_m = \varphi(x(N_f)) + \sum_{k=N_0}^{N_f-1} L(x(k), u(k), \gamma(k)) \tag{2.43}$$

subject to

$$x(k+1) = f(x(k), u(k), \alpha(k)) \tag{2.44}$$

$$x(N_0) = x_0 \tag{2.45}$$

where state and control vectors have the same dimensions as in ROP, J_m is a model-based performance index, $L: \Re^n \times \Re^m \times \Re \to \Re$ is a discrete weighting function and perhaps a simplification of a known L^*, $f: \Re^n \times \Re^m \times \Re^r \to \Re^n$, is an approximate dynamic model of f^*; $\gamma(k) \in \Re$ and $\alpha(k) \in \Re^r$ are discrete parameters. The role of $\alpha(k)$ will be to take into account differences in value between f^* and f, while $\gamma(k)$ takes into account differences in value between L^* and L.

By adding appropriate equality constraints, an equivalence is made between ROP and MOP, giving rise to the following expanded optimal control problem (EOP):

EOP

$$\min_{\substack{u(k)\\ k\in[N_0,N_f-1]}} J_e = \varphi(x(N_f)) + \sum_{k=N_0}^{N_f-1} L(x(k),u(k),\gamma(k)) \tag{2.46}$$

subject to

$$x(k+1) = f(x(k),u(k),\alpha(k)) \tag{2.47}$$

$$x(N_0) = x_0 \tag{2.48}$$

$$f(z(k),v(k),\alpha(k)) = f^*(z(k),v(k),k) \tag{2.49}$$

$$L(z(k),v(k),\gamma(k)) = L^*(z(k),v(k),k) \tag{2.50}$$

$$u(k) = v(k) \tag{2.51}$$

$$x(k) = z(k) \tag{2.52}$$

By using Lagrange multiplier theory and differential calculus, it is possible to find that the values of $\alpha(k),\gamma(k),\lambda(k)$ and $\beta(k)$, $k\in[N_0N_f-1]$ are given by (see Becerra, 1994)

$$f(z(k),v(k),\alpha(k)) = f^*(z(k),v(k),k) \tag{2.53}$$

$$L(z(k),v(k),\gamma(k)) = L^*(z(k),v(k),k) \tag{2.54}$$

$$\begin{aligned}\lambda(k) = &\left[\frac{\partial f}{\partial v(k)} - \frac{\partial f^*}{\partial v(k)}\right]\hat{p}(k+1)\\ &+\left[\nabla_{v(k)}L(z(k),v(k),\gamma(k)) - \nabla_{v(k)}L^*(z(k),v(k),k)\right]\end{aligned} \tag{2.55}$$

$$\begin{aligned}\beta(k) = &\left[\frac{\partial f}{\partial z(k)} - \frac{\partial f^*}{\partial z(k)}\right]\hat{p}(k+1)\\ &+\left[\nabla_{z(k)}L(z(k),v(k),\gamma(k)) - \nabla_{z(k)}L^*(z(k),v(k),k)\right]\end{aligned} \tag{2.56}$$

where

$$v(k) = u(k), k\in[N_0, N_f-1] \tag{2.57}$$

$$z(k) = x(k), k\in[N_0, N_f] \tag{2.58}$$

$$\hat{p}(k) = p(k), k \in \left[N_0, N_f\right] \tag{2.59}$$

If the values of $\alpha(k), \gamma(k), \lambda(k), \beta(k), v(k), z(k)$ and $\hat{p}(k)$ satisfy the equations (2.53) to (2.59) above, then the solution of the following problem satisfies the necessary optimality conditions of EOP. This problem is called the discrete-time modified model-based problem (MMOP), defined as follows:

MMOP

$$\min_{\substack{u(k) \\ k \in [N_0, N_f - 1]}} J_M = \varphi(x(N_f)) + \sum_{k=N_0}^{N_f - 1} \left[L(x(k), u(k), \gamma(k)) - \lambda(k)^T u(k) - \beta(k)^T x(k) \right] \tag{2.60}$$

subject to

$$x(k+1) = f(x(k), u(k), \alpha(k)) \tag{2.61}$$

$$x(N_0) = x_0 \tag{2.62}$$

Assuming convergence, the iterative solution of MMOP by means of the following algorithm satisfies the necessary optimality conditions of ROP.

Algorithm 2.2: Discrete-time DISOPE algorithm

Data	$f, L, \varphi, x_0, N_0, N_f$ and means of calculating f^* and L^*
Step 0	Compute or choose a nominal solution $u^0(k), x^0(k)$ and $p^0(k)$. Set $i = 0, v^0(k) = u^0(k), z^0(k) = x^0(k), \hat{p}^0(k) = p^0(k)$.
Step 1	Compute the parameters $\alpha(k)$ and $\gamma(k)$ to satisfy: $f(z(k), v(k), \alpha(k)) = f^*(z(k), v(k), k)$ $L(z(k), v(k), \gamma(k)) = L^*(z(k), v(k), k)$ This is called the parameter estimation step.
Step 2	Compute the multipliers $\lambda^i(k)$ and $\beta^i(k)$ $\lambda(k) = \left[\frac{\partial f}{\partial v(k)} - \frac{\partial f^*}{\partial v(k)} \right] \hat{p}(k+1) + \left[\nabla_{v(k)} L(z(k), v(k), \gamma(k)) - \nabla_{v(k)} L^*(z(k), v(k), k) \right]$

$$\beta(k)=\left[\frac{\partial f}{\partial z(k)}-\frac{\partial f^{*}}{\partial z(k)}\right]\hat{p}(k+1)+\left[\nabla_{z(k)}L(z(k),v(k),\gamma(k))-\nabla_{z(k)}L^{*}(z(k),v(k),k)\right]$$

Step 3 With specified $\alpha(k),\gamma(k),\lambda(k)$ and $\beta(k),k\in\left[N_0,N_f-1\right]$ solve the discrete-time modified model-based optimal control problem MMOP to obtain $u^{i+1}(k),x^{i+1}(k)$ and $p^{i+1}(k)$. This is called the system optimisation step.

Step 4 This step tests convergence and updates the estimate for the optimal solution of ROP. In order to provide a mechanism for regulating convergence, a simple relaxation method is employed. This is:

$$v^{i+1}(k)=v^{i}(k)+k_v(u^{i+1}(k)-v^{i}(k))$$
$$z^{i+1}(k)=z^{i}(k)+k_z(x^{i+1}(k)-z^{i}(k))$$
$$\hat{p}^{i+1}(k)=\hat{p}^{i}(k)+k_p(p^{i+1}(k)-\hat{p}^{i}(k))$$

where k_v,k_z and k_p are scalar gains.
If $v^{i+1}(k)=v^{i}(k),k\in\left[N_0,N_f-1\right]$ within a given tolerance stop,
else set $i=i+1$ and continue from step 1.

2.7.1 Case with Linear Quadratic model-based problem

If the MOP is chosen to be Linear Quadratic such that

$$L(x(k),u(k),\gamma(k))=\frac{1}{2}x(k)^{T}Qx(k)+\frac{1}{2}u(k)^{T}R\,u(k)+\gamma(k) \qquad (2.63)$$

$$\varphi(x(N_f))=\frac{1}{2}x(N_f)^{T}\Phi x(N_f)$$

and

$$f(x(k),u(k),\alpha(k))=Ax(k)+Bu(k)+\alpha(k)$$

where Q is the intermediate state weighting matrix and Φ is the terminal state weighting matrix, then the solution of MMOP can be achieved by using standard,

non-iterative methods, such as the following solution procedure (see Becerra, 1994).

Procedure 2.1: Solution of MMOP

Data $A, B, Q, R, \Phi, N_0, N_f, \alpha(k), \lambda(k), \beta(k), k \in \left[N_0 N_f - 1\right]$

Step1 Solve backwards from $k = N_f - 1$ to N_0 the following difference equation, with terminal conditions $S(N_f) = \Phi$, and $h(N_f) = 0$

$$S(k) = Q + A^T S(k+1)(A - BG(k))$$
$$G(k) = \left[R + B^T S(k+1) B\right]^{-1} B^T S(k+1) A$$
$$h(k) = (A - BG(k))^T h(k+1) + (A - BG(k))^T S(k+1)\alpha(k)$$
$$-\beta(k) + G(k)^T \lambda(k)$$

Step 2 Compute the driving input $g(k)$ from

$$g(k) = \left[R + B^T S(k+1) B\right]^{-1} \left[-B^T S(k+1)\alpha(k) - B^T h(k+1) + \lambda(k)\right]$$

Step 3 Compute the state sequence $x(k)$, by solving from the initial condition x_0 the following difference equation:

$$x(k+1) = (A - BG(k))x(k) + Bg(k) + \alpha(k)$$

Step 4 Compute the costate $p(k)$ from:

$$p(k) = S(k)x(k) + h(k)$$

Step 5 Compute the control sequence $u(k)$ from:

$$u(k) = -G(k)x(k) + g(k)$$

2.8 SUMMARY

In this chapter, the modified two-step method otherwise known as ISOPE has been reviewed. A brief review of some of the work done on the ISOPE family of algorithms has been presented. A new practical formulation of the algorithm

devised by Becerra and Roberts (2000) has also been presented. Finally, the basic formulation of the dynamic version of ISOPE (DISOPE) has been presented. The reason for presenting these algorithms here is because they are used in subsequent chapters. ISOPE is used in chapter 6 while DISOPE is used in chapter 3.

CHAPTER 3

DEVELOPMENTS IN PREDICTIVE OPTIMISATION

3.1 INTRODUCTION

Model Predictive Control (MPC) methods are very much related to data reconciliation problems in that optimisation techniques used to solve both problems are essentially the same. In this chapter Model Predictive Control is introduced. Following a short historical review of the methods available, the formulation of a predictive optimiser developed and used at City University is presented. The MPC algorithm is applied to a benchmark challenge set by ALSTOM involving the control of a gasifier used for the generation of environmentally clean and efficient power from coal. This chapter paves the way for chapter 8 in which dynamic data reconciliation techniques developed in this thesis are applied to MPC.

3.2 MODEL PREDICTIVE CONTROL

Model-based predictive control has been the subject of intensive research for about 20 years. Although the theoretical solutions have been available for some time, industrial application only took place relatively recently due mainly to the lack of availability, at an acceptable price, of computing capacity (Balchen et al., 1991).

Model Predictive Control (MPC) has become a powerful tool for dynamic optimisation and control. There are a number of different MPC schemes

available, however the basic idea behind them all is essentially the same and can be summarised as follows (Lee and Ricker, 1994):

- A prediction of future output behaviour expressed in terms of current and future manipulated input moves is built using a dynamic model and on line measurements.
- Optimisation is then performed, based on the prediction; to find a sequence of input moves that minimises a chosen measure of the output deviation from their respective reference values while satisfying all the given constraints.
- Since the quality of prediction may improve as more measurements are collected, only the first of the calculated input sequences is implemented and the whole optimisation is repeated at the next sampling time. This *receding horizon* implementation makes MPC a feedback control algorithm.

A strong attribute of MPC is that various process constraints can be incorporated directly into the on-line optimisation performed at each time step. As a result MPC has been steadily gaining acceptance by the process industry since most control problems faced by the industry involve multivariable systems with constraints for which no other effective control technique exists. Other advantages of MPC include (Camacho and Bordons, 1999; Roberts, 1999):

- The concepts behind MPC are intuitive which makes the schemes attractive to industry.
- MPC can be used to control a great variety of processes including systems with long delay times, or of non-minimum phase or unstable ones.
- MPC compensates for measurable disturbances in a natural way by introducing feed-forward control.
- The resulting controller is an easy to implement linear control law.
- MPC is an open methodology based on basic principles which allow for future extensions.
- MPC is readily applicable to batch processes.

There are, of course, disadvantages associated with MPC: the most significant disadvantage is the requirement for an appropriate model of the process. The benefits obtained from MPC can be seriously affected by the discrepancies existing between the real process and the model. The second disadvantage is that although the resulting control law is easy to implement and requires little computation, its derivation is quite complex. Since the system dynamics will change and therefore the scheme must be adaptive, the amount of computation required at each sampling instant becomes substantial. The computations increase further if constraints are considered. However, with the advancement in computer power, this problem has become less of an obstacle.

Nonlinear model predictive control (NLMPC) schemes use nonlinear models for predictions. One of the advantages of NLMPC is that nonlinear models can be more accurate for long term prediction beyond the local operating point. A further advantage is that manipulated and state variable constraints are explicitly handled (Sistu and Bequette, 1991).

3.2.1 Model Predictive Control Strategy

The strategy usually shared by the MPC family of algorithms is illustrated in Figure (3-1) and can be described as follows:

1. The predicted future outputs $\hat{y}(t+k \mid t), \quad k=1 \ldots N$ are calculated at each instant t over the prediction horizon N using the process model. These depend on the known values up to instant t (past inputs and outputs), including the current output $y(t)$ and on the future control signals $u(t+k \mid t), \quad k=1 \ldots N-1$, to be calculated. (Note that $u(t+k \mid t)$ indicates the value of u at time instant $t+k$ calculated at time t).
2. The sequence of future control signals is computed to optimise a performance criterion, often a quadratic function to minimise the error between the

predicted process output signal and a reference trajectory. The control effort is usually included in the performance criterion.

3. Only the current control signal $u(t \mid t)$ is transmitted to the process. At the next sampling instant $y(t+1)$ is measured and step 1 is repeated and all sequences updated. Thus $u(t+1 \mid t+1)$ is calculated using the receding horizon concept.

The basic structure shown in Figure (3-2) is used to implement the above strategy. Future plant outputs are predicted using a model and based on past and current values and on the optimal control actions calculated by the optimiser.

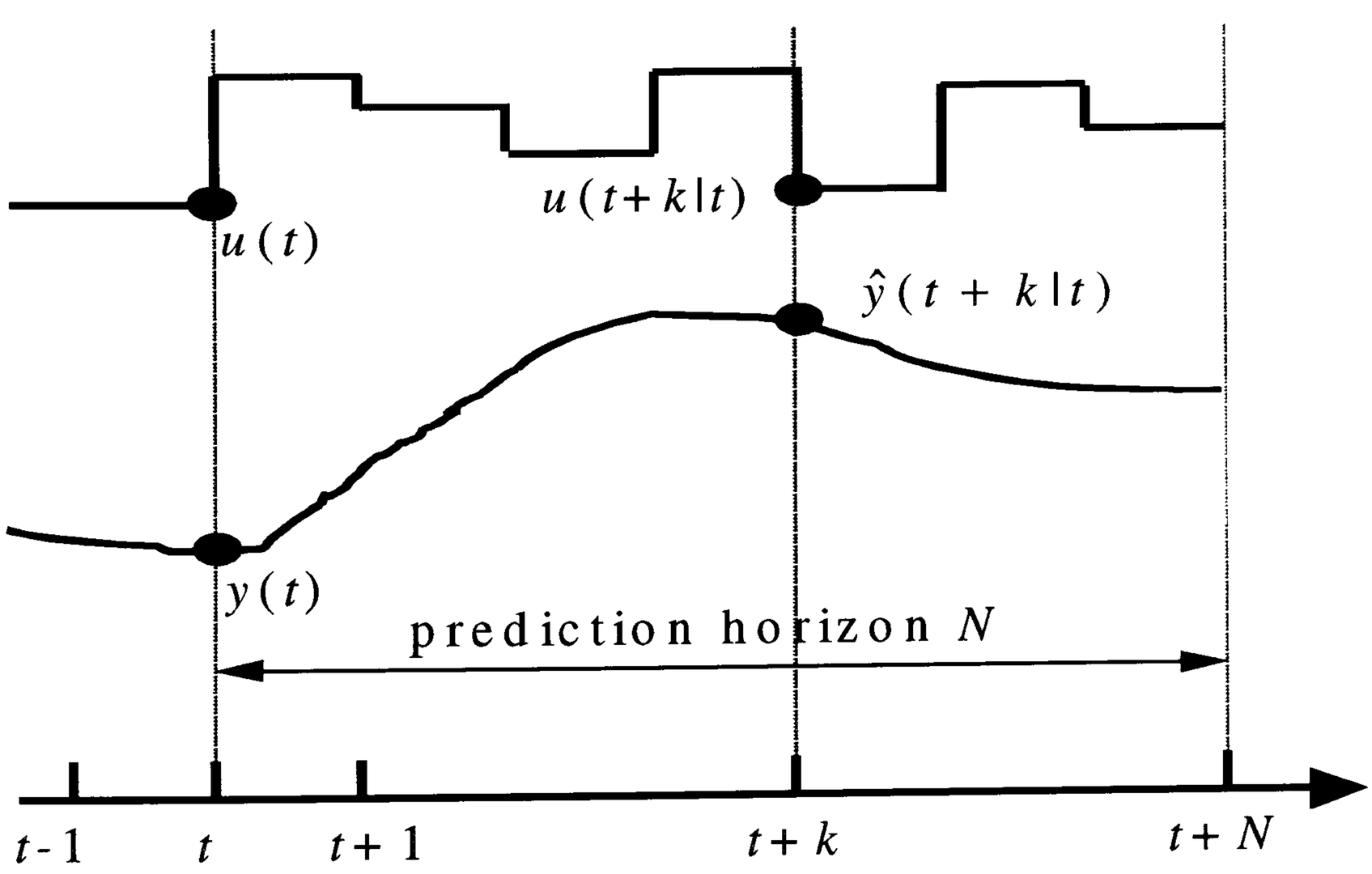


Figure (3-1): Model Predictive Control Strategy.

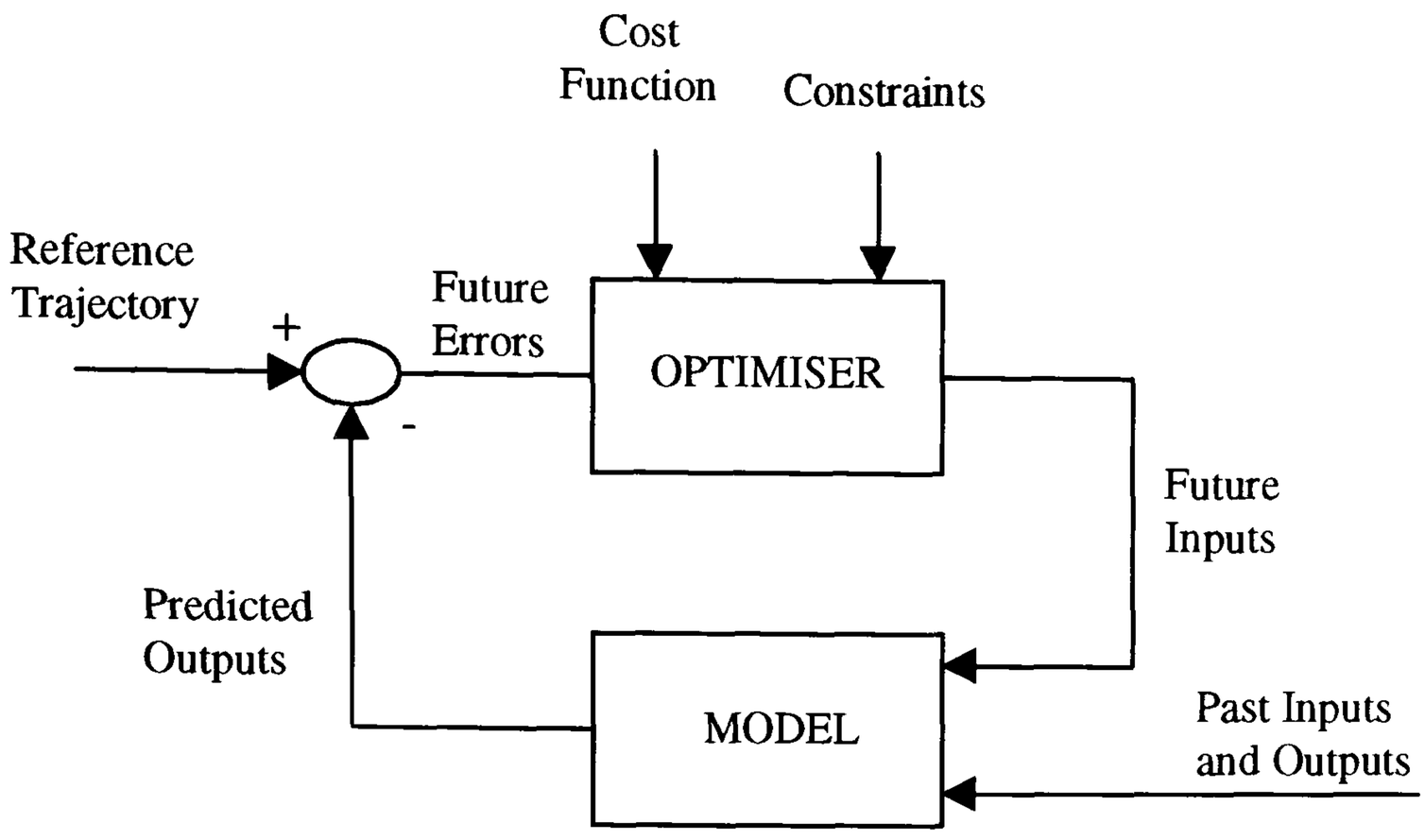


Figure (3-2): Basic structure of MPC

3.2.2 Models used for MPC

The various MPC algorithms available differ only in the model used to represent the process and the cost function to be minimised. The choice of model and model accuracy is of paramount importance to a successful MPC scheme. The chosen model must be capable of capturing the process dynamics so as to precisely predict the future outputs as well as being simple to implement. There are a number of types of model used with MPC, they are:

- Impulse response model,
- Step response model,
- Transfer function model,
- Linear state-space model,
- Nonlinear state-space model.

The Impulse response model, also known as the truncated impulse response model, is one of the most popular in industry. This type of model is easy to obtain requiring only the measurement of the output when the process is excited with an

impulse input. The impulse which is physically unrealisable is of course approximated by a pulse. As well as being intuitive, the impulse response model can be used for multivariable processes. The main drawbacks of this method. however, are the large number of parameters needed and the fact that for a process to be represented in this way, it must be open-loop stable. The step response model is closely related to this model and it is obtained when the input is a step.

Most widespread in the academic community is probably the transfer function model because it is valid for a wide range of processes and requires only a few parameters. State variable models are also used in some formulations as they can easily describe multivariable processes.

3.2.3 Model Predictive Control Algorithms

Various versions of MPC exist and, as mentioned previously, they are differentiated by the type of model they use and the cost function. The most well-known MPC algorithms are Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC) and Genralised Predictive Control (GPC).

Dynamic Matrix Control was developed by Cutler and Ramaker in 1980. DMC uses a step response model and has the following advantages and disadvantages (Pike et al., 1996):

DMC Advantages

- Implementation of the model is straight forward - simple calculations,
- Attractive for use by industrial personnel without extensive training,
- No assumption about the order of the process is required.

Disadvantage

- Open-loop unstable processes cannot be modeled or controlled.

Further work on DMC was carried out by Lundstrom et al. (1995) who highlighted its limitations and suggested the use of an alternative algorithm which includes an observer. Garcia and Morshedi (1986) published an extension of DMC called Quadratic DMC (QDMC) which uses Quadratic Programming to solve the constrained optimisation problem. However Cutler et al. (1983) had outlined the technique a few years previously. Gattu and Zafiriou (1992) extended QDMC for use with nonlinear process models and refer to the algorithm as Nonlinear QDMC (NLQDMC).

Model Algorithmic Control (MAC), developed by Richalet et al. (1978), and originally known as Model Predictive Heuristic Control, is similar to the DMC approach but possesses fewer tuning parameters and uses an impulse response model. The mathematical framework for MAC was formalised by Rouhani and Mehra (1982).

Generalised Predictive Control (GPC), developed by Clarke et al. (1987), uses a transfer function model and thus can easily be implemented in an adaptive mode by using an online estimation algorithm such as recursive least squares. The advantages and disadvantages of this approach are:

GPC Advantages

- GPC is normally able to stabilise and control open-loop unstable processes through the choice of tuning parameters.
- GPC is related to the properties of LQ control.

Disadvantage

- There are no guaranteed stability properties for GPC except under special conditions.

A number of researchers have worked on various extensions and applications of GPC. For instance, Gawthrop et al. (1998) developed a state-space version which unlike the transfer function version extends readily to the nonlinear case. Rossiter et al. (1996) presented an approach which guarantees the retention of feasibility

and stability for any setpoint change. Mahfouf et al. (1997) looked at the application of GPC to the on-line administration of muscle relaxant drugs in the operating theatre.

While the majority of the aforementioned techniques use input/ output models, there are a number of researchers who use state-space models and the receding horizon concept. Balchen et al. (1992) developed the State-Space Predictive Controller (SSPC) which accounts for severe process nonlinearities and general constraints on process variables. Becerra et al. (1998a) published a technique which integrates predictive control with on-line optimisation of economic objectives removing the need for a separate steady-state optimiser. Becerra et al. (1999a, 1999b) applied this technique to the Tennessee Eastman process model and a coal gasification plant model respectively. Some of the other researchers who have contributed to this field are Kwon and Byun (1989) and Mayne and Michalska (1990).

There are a number of review articles on MPC such as Qin and Badgwell (1996), Garcia et al. (1987) and Eaton and Rawlings (1992) to name a few. A number of textbooks are also available on the subject, see for instance Camacho and Bordons (1999).

Model predictive control has enjoyed wide industrial application. The key features contributing to its success are that multivariable systems and constraints can be accommodated effectively in the control problem, and the use of empirical models which can be measured from input/ output data. Qin and Badgwell (1996) reported the number of MPC applications at around 2200 and noted that the majority of applications were in refining and petrochemicals but significant growth was being noticed in the areas such as chemicals, pulp and paper, food processing, aerospace and automotive industries. From the perspective of Predictive Control Limited, U.K., Sandoz (1998) also noted that while in the petrochemical industry MPC was well exploited, the technology was slow in gaining ground in wider industry. A number of vendors have developed and

marketed predictive control software under different names such as DMC, IDCOM and CONNOISSEUR.

For the purpose of this work, an MPC algorithm developed by Becerra et al. (1998a) has been used. As well as for this chapter, the algorithm is utilised in chapter 8 where data reconciliation techniques are applied to MPC. This algorithm is based on state-space models, the derivation of which is presented in detail in Becerra et al. (1998a). The following is a brief outline of the algorithm.

3.3 RECEDING HORIZON OPTIMAL CONTROL ALGORITHM.

In this model predictive control framework, a *Receding Horizon Optimisation Problem (RHOP)* is solved at every sampling instant. The problem is formulated as follows:

RHOP

$$\min_{\Delta u(k)} \quad J(i) = \frac{1}{2}\Delta x(i+N)^T \Phi \Delta x(i+N) + \sum_{k=i}^{i+N-1} \Big\{ F(y(k), u_m(k)) + \frac{1}{2}\Delta x(k)^T Q \Delta x(k) + \frac{1}{2}\Delta u(k)^T R \Delta u(k) + P(y(k)) \Big\} \tag{3.1}$$

subject to

$$\Delta x(k+1) = A\Delta x(k) + B_m \Delta u_m(k) + \delta(k-i) B_d \Delta u_d(i) \tag{3.2}$$

$$\Delta y(k) = C\Delta x(k) \tag{3.3}$$

$$u_l \leq u_m \leq u_h \tag{3.4}$$

$$\left|\Delta u_m\right| \leq u_r \tag{3.5}$$

$$\Delta u_m(k) = 0, \quad M \leq k \leq N \tag{3.6}$$

where $F(y)$ is a steady-state objective, Φ, Q, R are weighting matrices of the appropriate dimensions, Δ is the increment operator, i.e. $\Delta u(k+1) = u(k) - u(k-1)$, u_m is the vector of manipulated variables, u_d is the disturbance input, $\delta(k-i)$ is 1 when $k = 1, 0$ otherwise (so the disturbance is assumed to be a step), B_m, B_d are submatrices of the identified B matrix, $B = [B_m \quad B_d]$, N is the prediction horizon in samples, M is the control horizon in samples, i denotes current time.

Output constraint violations along the predictions are penalised in equation (3.1) by means of the term $P(y(k))$. This implies that output constraints are treated as soft constraints. The penalty term is calculated from:

$$P(y(k)) = \sum_j \rho(P_\varepsilon(\Psi_j(y(k))))^2 + \mu P_\varepsilon(\Psi_j(y(k))) \tag{3.7}$$

where ρ and μ are scalar penalty factors. The output constraints are given by:

$$\Psi(y(k)) \le 0 \tag{3.8}$$

and

$$P_\varepsilon(\chi) = \begin{cases} \chi & if \ \chi > \varepsilon \\ -(\chi - \varepsilon)^2 / 4\varepsilon & if \ -\varepsilon \le \chi \le \varepsilon \\ 0 & if \ \chi < -\varepsilon \end{cases} \tag{3.9}$$

This type of penalty term, which combines quadratic and exact (linear) penalties, helps avoid problems when the output constraints are infeasible but attempts to enforce active but feasible output constraints.

The steady-state objective function used in this work was a quadratic regulatory objective:

$$F(y(k)) = \frac{1}{2}(y(k) - y_r)^T Q_y (y(k) - y_r) \tag{3.10}$$

where Q_y is a weighting matrix of the appropriate dimensions and y_r is a vector of reference values for the outputs. Note that $F(y(k))$ may in general include other terms such as economic objectives, which are often linear, and other quadratic terms implementing targets for the manipulated variables.

The optimisation algorithm used to solve the receding horizon problem formulated above was Dynamic Integrated System Optimisation and Parameter Estimation (DISOPE) which was briefly introduced in chapter 2. Further details of the DISOPE algorithm can be found in Becerra and Roberts (1996).

Table (3.1): Predictive control algorithm

Data: $Q, R, \Phi, N, u_l, u_h, u_r$

Step 1 *Parameter estimation*: Obtain values of parameter matrices A, B and C

Step 2 *Optimal prediction*: Solve *RHOP* to obtain the predicted control sequence $\{u_m(i), \ldots u_m(i+N-1)\}$

Step 3 Apply the first element $u_m(i)$ of the predicted input sequence to the plant.

Step 4 Wait until next sampling time, set $i = i+1$ and go to step 1.

3.4 LINEAR MODEL IDENTIFICATION SCHEME

Consider a state space model of the type:

$$\Delta x(k+1) = A\Delta x(k) + B\Delta u(k)$$
$$\Delta y(k) = C\Delta x(k) \qquad (3.11)$$

where k is an integer sampling index, $x \in \Re^n$ is a state vector, $u \in \Re^{n_u}$ is a set of independent inputs, $y \in \Re^{n_y}$ is a vector of measured outputs, A, B, C are matrices of the appropriate dimensions. The estimation of linear state space models of this type has been tackled using different system identification methods. Becerra et al. (1997) developed a non-iterative technique based on the least squares method and the multivariable ARMAX model structure. The resulting ARMAX model is then transformed into a non-minimal state-space realization. In this way the need for a separate linear state estimator is avoided since the state vector is formed from delayed values of the output and input vectors. The technique is based on the moving-horizon concept, but it exploits the displacement structure of the data window to considerably reduce the computational load. The following is a brief outline of the technique (Becerra et al., 1998a). For further details regarding the formulation and various advantages of this technique, the reader is referred to this particular paper as well as to Becerra et al. (1997).

Assume that the output of the system at discrete time k is denoted as $y(k) \in \Re^{n_y}$, and the input variable at time k is given by $u(k) \in \Re^{n_u}$. An ARMAX model of the system can be written as:

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k-d) + C^*(q^{-1})\varepsilon(k) + b \qquad (3.12)$$

where

$$A^*(q^{-1}) = I + A_1 q^{-1} + \cdots + A_{n_a} q^{-n_a}$$
$$B^*(q^{-1}) = B_1 q^{-1} + B_2 q^{-2} + \cdots + B_{n_b} q^{-n_b}$$

$$C^*(q^{-1}) = I + C_1 q^{-1} + \cdots + C_{n_c} q^{-n_c}$$

are matrix polynomials of degrees n_a, n_b, n_c respectively, in the backward shift operator q^{-1}, d is the minimum pure time delay in samples from inputs to outputs, the sequence $\varepsilon(k) \in \Re^{n_y}$ is assumed to be zero mean discrete white noise, and $b \in \Re^{n_y}$ is an off-set parameter vector introduced to take into account non-zero levels in the signals involved.

An equivalent non-minimal state-space realization of the deterministic part of the ARMAX model (3.12) is as follows:

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + c \\ y(k) &= Cx(k) \end{aligned} \tag{3.13}$$

where

$$\begin{aligned} x(k) = \big[& y(k)^T \, y(k-1)^T \cdots y(k-n_a+1)^T \, u(k-1)^T \\ & \cdots u(k-d)^T \cdots u(k-d-n_b+2)^T \big]^T \end{aligned}$$

is a state vector which contains present and past values of the output at time k, and past values of the input variables, dim $x = n = n_y n_a + n_u (n_b + d - 2)$, A and B_u are matrices of the appropriate dimensions which are formed in terms of the ARMAX model polynomial coefficients, $c \in \Re^n$ is an off-set vector. Notice that because the state vector is formed from past input and output variables, a state observer is not required when using these models for control purposes. For instance, for the case when $d = 1$, matrices A, B_u and C are given below:

$$A=\begin{bmatrix} -A_1 & -A_2 & \cdots & -A_{n_a} & B_2 & \cdots & B_{n_b} \\ I_{n_y} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots I_{n_y} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots I_{n_u} & 0 \end{bmatrix}_{(n_y n_a + n_u(n_b-1))\times(n_y n_a + n_u(n_b-1))}$$

$$B_u=\begin{bmatrix} B_1 \\ 0 \\ \vdots \\ 0 \\ I_{n_u} \\ \vdots \\ 0 \end{bmatrix}_{(n_y n_a + n_u(n_b-1))\times n_u}$$

$$C=\begin{bmatrix} I_{n_y} & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}_{n_y\times(n_y n_a + n_u(n_b-1))} \tag{3.14}$$

Multiplying equations (3.13) by the difference operator $\Delta = 1 - q^{-1}$, the following incremental state space model is obtained:

$$\begin{aligned} \Delta x(k+1) &= A\Delta x(k) + B_u \Delta u(k) \\ \Delta y(k) &= C\Delta x(k) \end{aligned} \tag{3.15}$$

This model is a locally valid linear state space model in the form used in the definition of the receding horizon optimisation problem above.

3.5 SIMULATION CASE STUDY

The predictive control technique outlined above was applied to a benchmark challenge to control a gasifier plant used in the generation of power from coal.

The following subsection presents an introduction to the challenge and the control aims. This is followed by a discussion of some implementation issues and a presentation of the results obtained.

3.5.1 The ALSTOM Gasifier Benchmark Challenge

In September 1997 GEC ALSTHOM Mechanical Engineering Centre (now ALSTOM Energy Technology Centre), presented the U.K. control community with a control design challenge to control a gasifier plant. The challenge drew considerable interest form both academia and industry and led to a seminar (Dixon et al., 1998) in which a number of advanced control schemes were presented as solutions to the problem. Dixon (1999) published a report on the seminar and the solutions presented.

Integrated Gasification Combined Cycle (IGCC) power plants are being developed around the world in order to provide environmentally clean and efficient power generation from coal. The gasification plant or gasifier (Figure 3-3) can be considered as a reactor where coal reacts with air and steam. The products of the gasification process are a low calorific value fuel gas, which can be burnt in a suitably adapted gas turbine, and char (ash from coal, limestone and unreacted carbon). Limestone (sorbent) is also added to the vessel to capture the majority of sulphur present in the coal.

The key inputs to the gasifier are air flow, steam flow, coal flow, sorbent flow and char extraction flow. The main output variables to be regulated are gas temperature, gas pressure, gas calorific value and bed mass. The gasifier system is difficult to control since it is multivariable and highly nonlinear, with significant cross-coupling between the input and output variables.

3.5.1.1 Control aims

As part of the challenge 'information pack', ALSTOM provided three linear models obtained from a rigorous nonlinear model of the system which was not made available to the participants. The challenge brief was to design a controller based on a linear model of the gasifier running at 100% load. The specification included constraints on the control signals and bounds on the output variables (see Tables 3.2 and 3.3) as well as a set of tests and performance criteria to facilitate the easy comparison of the different schemes proposed. The tests included system response to both step and sinusoidal disturbances. Furthermore, the controller was to be evaluated without redesign on a further two linear models representing the gasifier at 50% and 0% load.

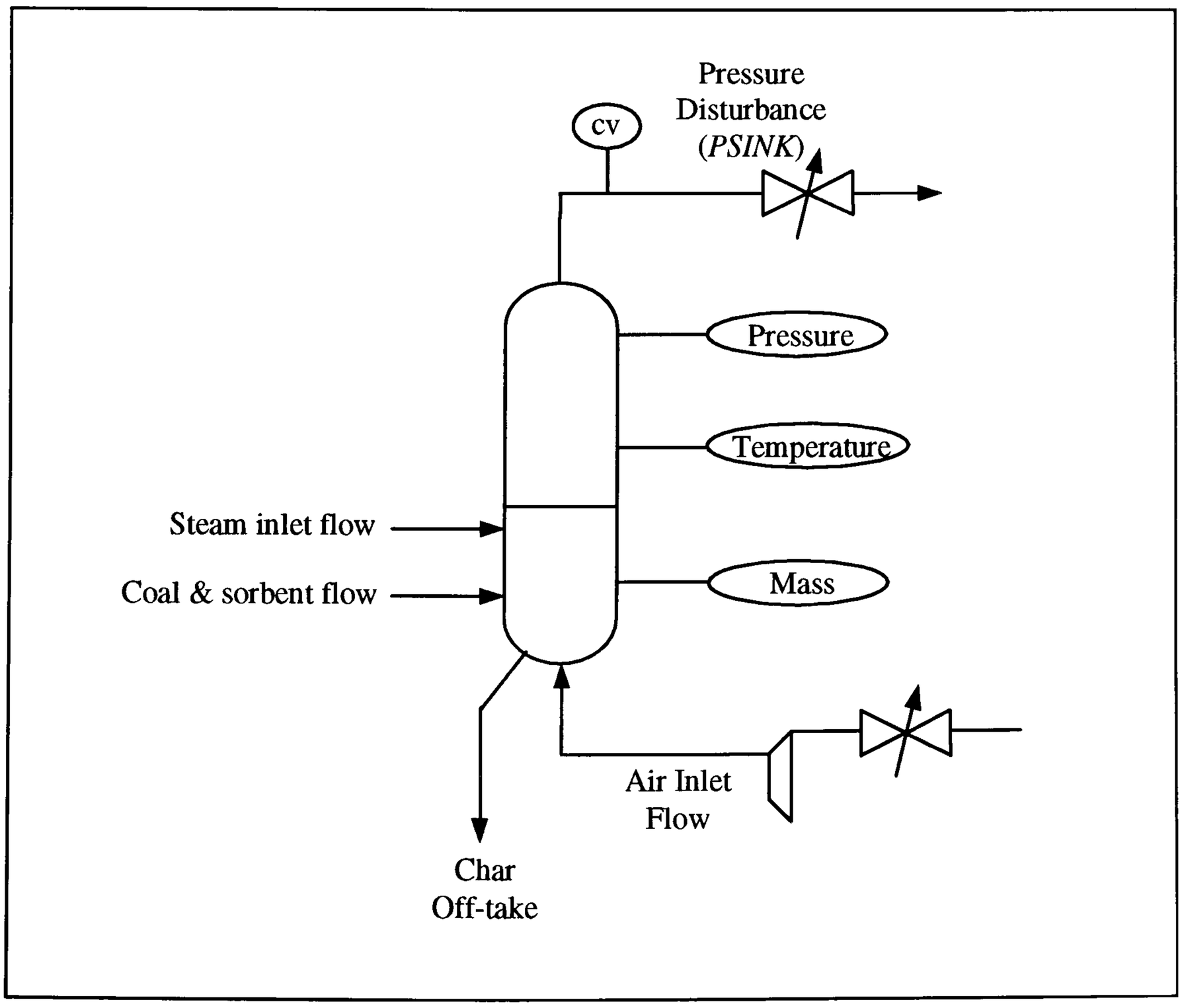


Figure (3-3): Gasifier plant functional diagram.

3.5.2 Gasifier Model Implementation

The supplied gasifier models were implemented as a module in the OTISS^{TM} process simulator, (SAST, 1993). This software package is used for simulation in the process industries, particularly in the oil and petrochemical industries. OTISS^{TM} allows the implementation of rigorous process models by using a library of components. It also allows its modular extension by the user by adding C/C++ routines. The integration algorithm used was the 4th order Runge-Kutta method, with a fixed step size of 0.005 s . The model matrices were read directly from the supplied Matlab data files. The gasifier load value could be specified as a parameter from the user interface, so that the corresponding model matrices could be used. Also, the corresponding steady-state values were subtracted from the inputs and added to the outputs.

3.5.3 Model Identification

In order to follow an approach that was independent from the form of the supplied plant models, an independent identification experiment was carried out. A similar experiment could have been carried out if the plant model was a nonlinear model. The purpose of the identification experiment was to obtain input-output data which was later used for identifying a discrete time model. The identified model was then used for control design purposes.

A pseudo random binary sequence (PRBS) test was carried out on the model for 100% load in order to identify a possibly lower order model. Independent PRBS signals were applied to the manipulated inputs and the disturbance input *psink*. The application of a PRBS to the disturbance variable is of course not very realistic. In a real application the PRBS would only be applied to the manipulated variables and it would be necessary to use the normal changes in the disturbance variable as an excitation. The fixed relationship between *wls* and *wcol* was taken into account (*wls* = 0.1 *wcol*). The data collected from the PRBS test contained

information for a period of one hour. The sampling time used was 0.25 seconds. A multivariable ARX model was identified with the following structure:

$$y(k) = -A_1 y(k-1) - A_2 y(k-2) + B_1 u(k-1) + B_2 u(k-2) \qquad (3.16)$$

where y is the vector of measured outputs, $y = [cvgas, mass, pgas, tgas]^T$, u is the vector of process inputs, $u = [wchr, wair, wcol, wstm, psink]^T$, k is a discrete time index, A_i, B_i are matrix coefficients. The resulting ARX model was transformed into a state-space model using a non-minimal realisation (equations 3.13) where the state vector $x(k) = [y(k), y(k-1), u(k-1)]^T$.

For the ARX structure given above, the corresponding dimension of the state vector $x(k)$ is 13. A validation plot is shown in Figure (3-4), which compares the response of the supplied and identified models to a 10% increase in the air flow, *wair*, while at 100% load. In this simulation, a PI controller with proportional gain -0.003 and integral gain -0.00001 manipulates the char offtake in order to maintain the bedmass.

3.5.4 Controller Implementation

The predictive controller introduced in Section 3.3 was implemented as an OTISS module by Becerra et al. (1998a). A moving horizon identification algorithm is integrated with this module, to allow the identification of a suitable process model. Periodic adaptation of the model is also possible, although this feature could not be used in this case because of the rules of the challenge. A graphical interface allows the specification of the different parameters associated with the predictive controller, such as prediction horizon, weights, model structure, etc.

3.5.5 Controller Tuning and Simulations

The scaling factors used internally by the predictive controller are given in Table (3.4). The tuning parameters of the predictive controller are given in Table (3.5). A different set of results employing a different set of tuning parameters was presented by Becerra et al. (1999b).

The disturbance variable was the sink pressure *psink*. The steady-state values of *psink* were $18.5\times10^5\ (N/m^2)$ at 100% load, $14.8\times10^5\ (N/m^2)$ at 50% load and $11.1\times10^5\ (N/m^2)$ at 0% load. The step disturbance consists of a change of $-2\times10^4\ (N/m^2)$ at 30s. The sinusoidal disturbance had a peak value of $2\times10^4\ (N/m^2)$ and a period of 25 s.

In order to tune the controller, several simulations were carried out at 100% load for the step and sinusoidal disturbances. Finally, to satisfy the rules of the challenge, simulations were carried out using both types of disturbance at 50% and 0% load using the same identified model and controller tuning.

3.5.6 Simulation Results

The simulation results including peak rate, maximum and minimum absolute values for each variable, the maximum constraint violation and the relevant integrated absolute errors (IAE) are presented in Tables (3.6) to (3.11). These include cases for step and sinusoidal disturbances, as well as three different values of the gasifier load: 100%, 50% and 0%. Figures (3-5) to (3-17) show the simulation trajectories for the outputs and manipulated variables for both types of disturbance and for different load values.

Figures (3-5) and (3-6) show the trajectories of the process outputs and manipulated inputs respectively for the case when using a step disturbance with 100% load. It can be observed that there are no violations of any of the

constraints except for the variable *pgas* which just violates the lower constraint. The level of violation of this variable can be seen in Table (3.6).

Figures (3-7) and (3-8) are related to the case when the disturbance is sinusoidal with a load of 100%. It is apparent that the variables *cvgas* and *pgas* periodically violate the constraints in small amounts. This is similar to the case when a sinusoidal disturbance is used with 50% load (Figures 3-11 and 3-12).

In the case of a step disturbance with 50% load, the results are good (Figures 3-9 and 3-10). There are no violations at all by any of the variables and the controller quickly rejects the disturbance.

In the cases where the load was 0%, the results were not very satisfactory for both disturbance types. In the case of a step disturbance (Figures 3-13 and 3-14), there can be seen violations by the variables *cvgas* and *pgas* and a large violation by *tgas*. There is also some noticeable oscillation in the variable *cvgas*. In the case of a sinusoidal disturbance (Figures 3-15 and 3-16), *cvgas* and *pgas* display a similar response, in shape, to the 100% and 50% load cases with larger periodic violations. Again, however, there is a large violation by the variable *tgas* which is seen to settle in the extended simulation in Figure (3-17).

A comparison of the results obtained here with a results obtained using a number of other control strategies presented as solutions to the problem at Coventry University in June 1998 was carried out. It is important to note that the results presented here are an improved set of results to those of Becerra et al. (1998). The comparison was made in terms of a sum of IAE values for all the step cases compared to the average from all the control techniques presented. A similar exercise was carried out for the sinusoidal disturbance cases. The results of this exercise are shown in Table (3.12). It can be seen that with the cases involving a step disturbance, the presented MPC algorithm performs better than the average (considerably better in the case of *pgas*). However, in the case of sinusoidal disturbances, this algorithm performs considerably worse than the average.

3.6 CONCLUSIONS

A state-space model predictive control algorithm has been used to control a gasifier plant as part of a challenge set by the Mechanical Engineering Centre at ALSTOM. The challenge involved the control of models of the coal gasifier at different operating conditions. The models are multivariable and marginally stable. The controller was required to take into account constraints on input and output variables and be able to deal with step and sinusoidal disturbances. The dynamics of the process change significantly at the different load levels. For the above reasons the satisfaction of all the control objectives and constraints was truly a difficult task.

The approch used in tackling the set problem produced an acceptable response for the cases of 100% and 50% load conditions which involved minor and transient violations of two of the output constraints. However, for the case of 0% load condition, the process response did not satisfy many of the constraints and control objectives.

The results obtained above highlight one important aspect of model predictive control and that is the dependence on the accuracy of the model. In the cases when the load was 100% and 50%, the results were good. However, when the load condition was set to 0% where the process dynamics must be very different to those captured by the original model, the results were poor. The predictive control algorithm presented in this chapter has on-line model adaptation capabilities but this feature was disabled in order to comply with the challenge rules. The on-line identifier has the potential of increasing the robustness of the controller.

Although in the cases with 100% and 50% load where a sinusoidal disturbance was used the results were somewhat satisfactory, the algorithm could be seen to be struggling with all cases involving a sinusoidal disturbance. This is probably due to the fact that the algorithm assumes a step-like disturbance and therefore

finds it difficult to cope with sinusoidal disturbances. The use of a more general disturbance model may help in improving the response for cases where a sinusoidal disturbance is present.

3.7 SUMMARY

In this chapter, model predictive control has been introduced. Following a brief introduction, a short review of the available algorithms was presented along with some of the major research activities in the field. A state-space model predictive control algorithm employing the receding horizon concept developed at City University has been presented. This algorithm was applied to a benchmark challenge set by ALSTOM Mechanical Engineering Centre involving the control of a gasifier used for the generation of power from coal. The scheme was shown to be successful for the majority of the cases set by the challenge but had difficulty in dealing with cases involving considerable departure of the process dynamics from the identified model.

This chapter represents an introduction to the field of model predictive control. Chapter 8 applies the data reconciliation methods developed in the subsequent chapters of this thesis to model predictive control techniques.

Variable	100%	50%	0%	Max. Limit	Min. Limit
cvgas (J/kg)	$4.36*10^6$	$4.49*10^6$	$4.71*10^6$	10000	-10000
mass (kg)	10000	10000	10000	500	-500
pgas(N/m^2)	$2.0*10^6$	$1.55*10^6$	$1.12*10^6$	10000	-10000
tgas (K)	1223.2	1181.1	1115.1	1	-1

Table (3.2): Measured variable parameters

Variable	100%	50%	0%	Max. Limit	Min. Limit	Rate Limit
wchr (kg/s)	0.9	0.89	0.5	3.5	0	0.2
wair (kg/s)	17.42	10.89	4.34	20	0	1
wcol (kg/s)	8.55	5.34	2.136	10	0	0.2
wstm (kg/s)	2.7	1.69	0.676	6	0	1

Table (3.3): Manipulated variable parameters

Variable	Scaling Factor
wchr	1/0.90
wair	1/17.42
wcol	1/8.55
wstm	1/2.70
cvgas	$1/4.36*10^6$
mass	1/10000.0
pgas	$1/2.0*10^6$
tgas	1/1223.2

Table (3.4): Scaling Factors.

Parameter	Description	Value
N	Prediction horizon	80
M	Control horizon	5
R	Input move weight	diag (10^{-5},2,1,0.005)
Q_y	Output target weight	diag (2.5, 0.005, 4.0, 20.0)
Q	state weight	$0.1I_{13}$
Φ	Terminal state weight	I_{13}

Table (3.5): Controller tuning parameters

Name	Max & Min abs. Value	Peak Rate	IAE	Max. Viol.
Inputs				
wchr	0.90192 0.88069 (kg/s)	0.000864 (kg/s^2)		
wair	18.5637 17.3485 (kg/s)	0.39469 (kg/s^2)		
wcol	9.5231 8.5486 (kg/s)	0.2 (kg/s^2)		
wstm	3.0839 2.6999 (kg/s)	0.10902 (kg/s^2)		
Ouputs				
cvgas	4360465.5 4350331 (J/kg)	2678 (J/kg s)	**151963.4 (J s/kg)**	**0 (J/kg)**
mass	10000.0537 9975.6143 (kg)	0.32812 (kg/s)		
pgas	2001290.6 1989721.4 (N/m^2)	7148.5 (N/m^2 s)	**117578.3 (N s/m^2)**	**279.9 (N/m^2)**
tgas	1223.5901 1222.6963 (K)	0.1499 (K/s)		

Table (3.6): Case with 100% load and step disturbance.

Name	Max & Min abs. Value	Peak Rate	IAE	Max. Viol.
Inputs				
wchr	0.9085 0.8989 (kg/s)	0.000464 (kg/s^2)		
wair	17.7682 16.7934 (kg/s)	0.31876 (kg/s^2)		
wcol	9.2641 7.828 (kg/s)	0.2 (kg/s^2)		
wstm	2.9907 2.3962 (kg/s)	0.1029 (kg/s^2)		
Ouputs				
cvgas	4373863.5 4344578.5 (J/kg)	3858 (J/kg s)	**2446746.9 (J s/kg)**	**5417.5 (J/kg)**
mass	10007.367 9998.862 (kg)	0.51562 (kg/s)		
pgas	2015234 1984210.1 (N/m^2)	4159.5 (N/m^2 s)	**2783506.1 (N s/m^2)**	**5845.1 (N/m^2)**
tgas	1223.363 1222.8 (K)	0.04541 (K/s)		

Table (3.7): Case with 100% load and sine disturbance.

Name	Max & Min abs. Value	Peak Rate	IAE	Max. Viol.
Inputs				
wchr	0.8908 0.8685 (kg/s)	0.00084 (kg/s^2)		
wair	12.0727 10.8524 (kg/s)	0.3943 (kg/s^2)		
wcol	6.3426 5.34 (kg/s)	0.2 (kg/s^2)		
wstm	2.0804 1.69 (kg/s)	0.1089 (kg/s^2)		
Ouputs				
cvgas	4490318 4480453 (J/kg)	2620 (J/kg s)	**150025.4 (J s/kg)**	**0 (J/kg)**
mass	10000.06 9976.92 (kg)	0.33203 (kg/s)		
pgas	1551826.1 1540196.4 (N/m^2)	7149.5 (N/m^2 s)	**263858.1 (N s/m^2)**	**0 (N/m^2)**
tgas	1181.45 1180.46 (K)	0.1494 (K/s)		

Table (3.8): Case with 50% load and step disturbance.

Name	Max & Min abs. Value	Peak Rate	IAE	Max. Viol.
Inputs				
wchr	0.8945 0.8867 (kg/s)	0.000464 (kg/s^2)		
wair	11.2726 10.2846 (kg/s)	0.31872 (kg/s^2)		
wcol	6.0741 4.6594 (kg/s)	0.2 (kg/s^2)		
wstm	1.9795 1.3949 (kg/s)	0.10719 (kg/s^2)		
Ouputs				
cvgas	4503755.5 4473603.5 (J/kg)	4142 (J/kg s)	**2457445.6 (J s/kg)**	**6393.5 (J/kg)**
mass	10007.657 9998.596 (kg)	0.51563 (kg/s)		
pgas	1565529.5 1534592.6 (N/m^2)	4040.5 (N/m^2 s)	**2781674.3 (N s/m^2)**	**5474.5 (N/m^2)**
tgas	1181.27 1180.65 (K)	0.04736 (K/s)		

Table (3.9): Case with 50% load and sine disturbance.

Name	Max & Min abs. Value	Peak Rate	IAE	Max. Viol.
Inputs				
wchr	0.5161 0.0348 (kg/s)	0.007412 (kg/s^2)		
wair	7.9161 3.5682 (kg/s)	1 (kg/s^2)		
wcol	3.9867 1.8086 (kg/s)	0.2 (kg/s^2)		
wstm	2.7985 0.1114 (kg/s)	1 (kg/s^2)		
Ouputs				
cvgas	4720690 4690196 (J/kg)	9322 (J/kg s)	**1297783.9 (J s/kg)**	**9798 (J/kg)**
mass	10000.001 9822.6299 (kg)	1.5078 (kg/s)		
pgas	1124692.4 1103290 (N/m^2)	14639.5 (N/m^2 s)	**688951.3 (N s/m^2)**	**6710.8 (N/m^2)**
tgas	1121.12 1114.63 (K)	0.4541 (K/s)		

Table (3.10): Case with 0% load and step disturbance.

Name	Max & Min abs. Value	Peak Rate	IAE	Max. Viol.
Inputs				
wchr	0.5034 0.1647 (kg/s)	0.007756 (kg/s^2)		
wair	5.5728 2.594 (kg/s)	1 (kg/s^2)		
wcol	2.5097 0.3974 (kg/s)	0.2 (kg/s^2)		
wstm	2.3603 0 (kg/s)	1 (kg/s^2)		
Ouputs				
cvgas	4740785 4688832 (J/kg)	12658 (J/kg s)	**2946452.4 (J s/kg)**	**20790.5 (J/kg)**
mass	10000.165 9882.545 (kg)	1.1445 (kg/s)		
pgas	1141348.8 1096198.1 (N/m^2)	5699.5 (N/m^2 s)	**3590381.9 (N s/m^2)**	**13916.4 (N/m^2)**
tgas	1120.7 1115.05 (K)	0.25098 (K/s)		

Table (3.11): Case with 0% load and sine disturbance.

Disturbance	Average IAE (Seminar)	Sum of IAE using MPC
Step - *cvgas*	$1.6326*10^6$	$1.5998*10^6$
Step - *pgas*	$1.7611*10^6$	$1.0704*10^6$
Sine - *cvgas*	$2.2605*10^6$	$7.8506*10^6$
Sine - *pgas*	$4.9729*10^o$	$9.1556*10^o$

Table (3.12): Comparison of adopted MPC method with other control stategies.

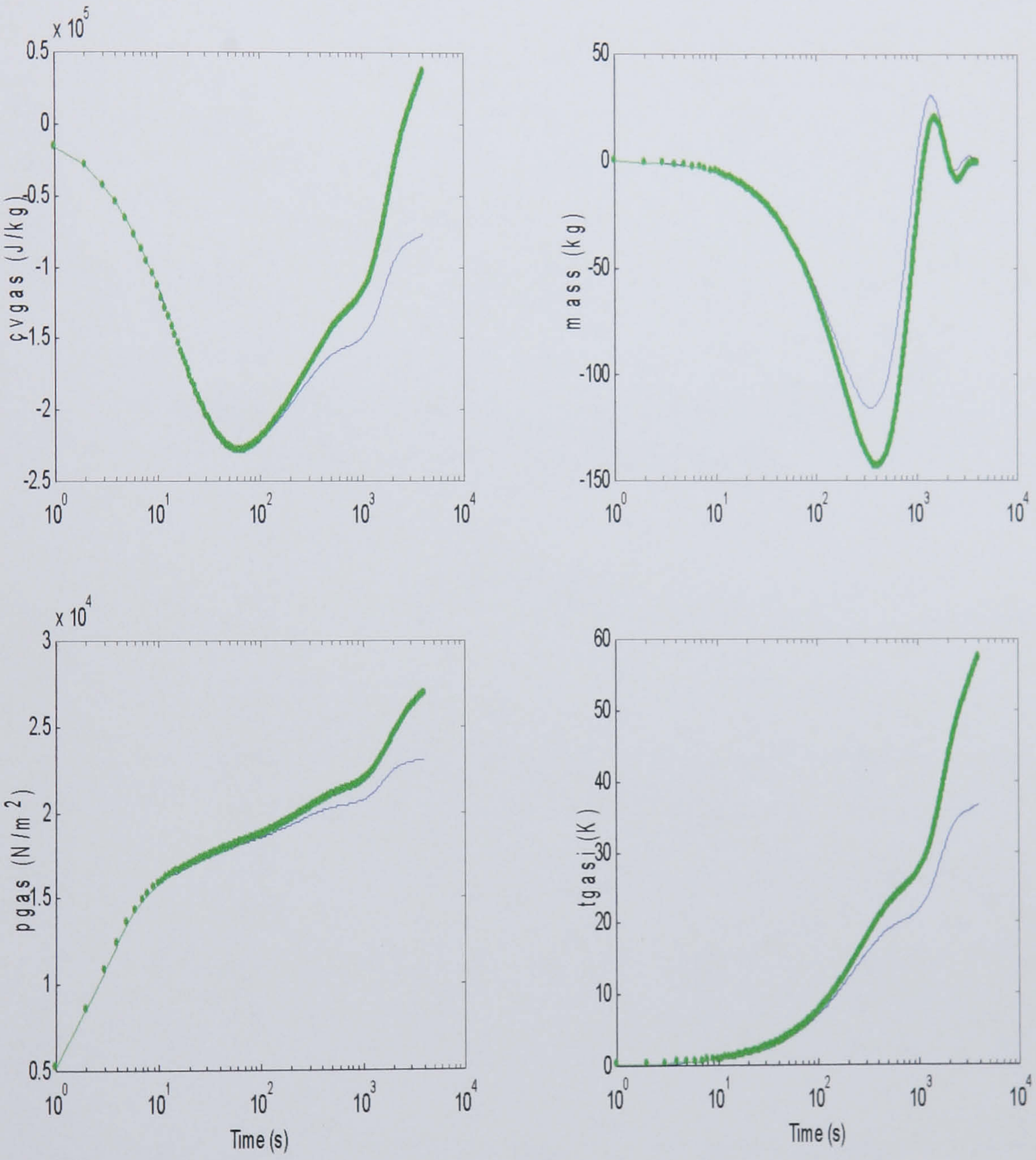


Figure (3-4): Model validation plot -
response of the identified and supplied models after a step in the air flow, *wair*.

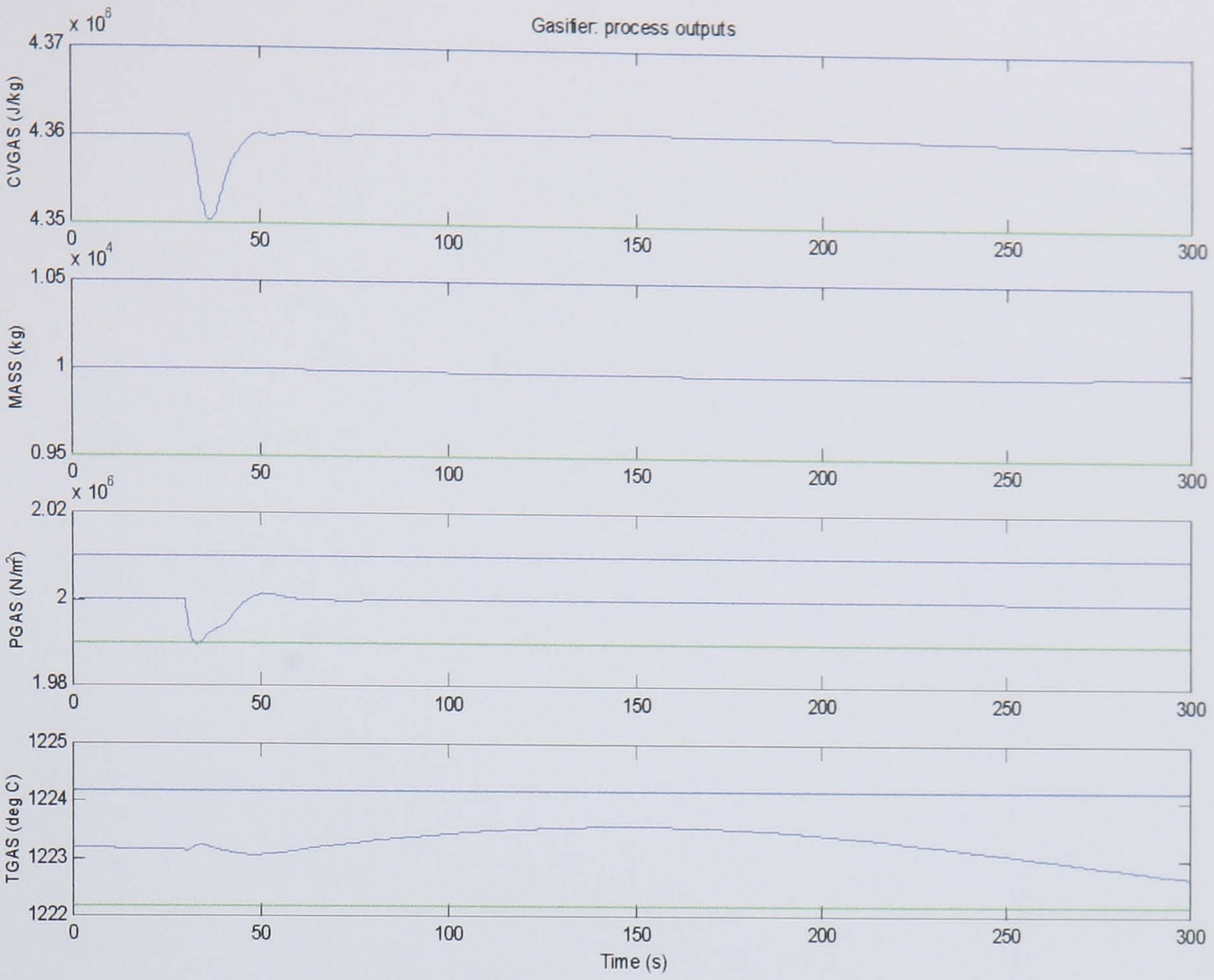


Figure (3-5): Response of the process outputs at 100% load to a step disturbance.

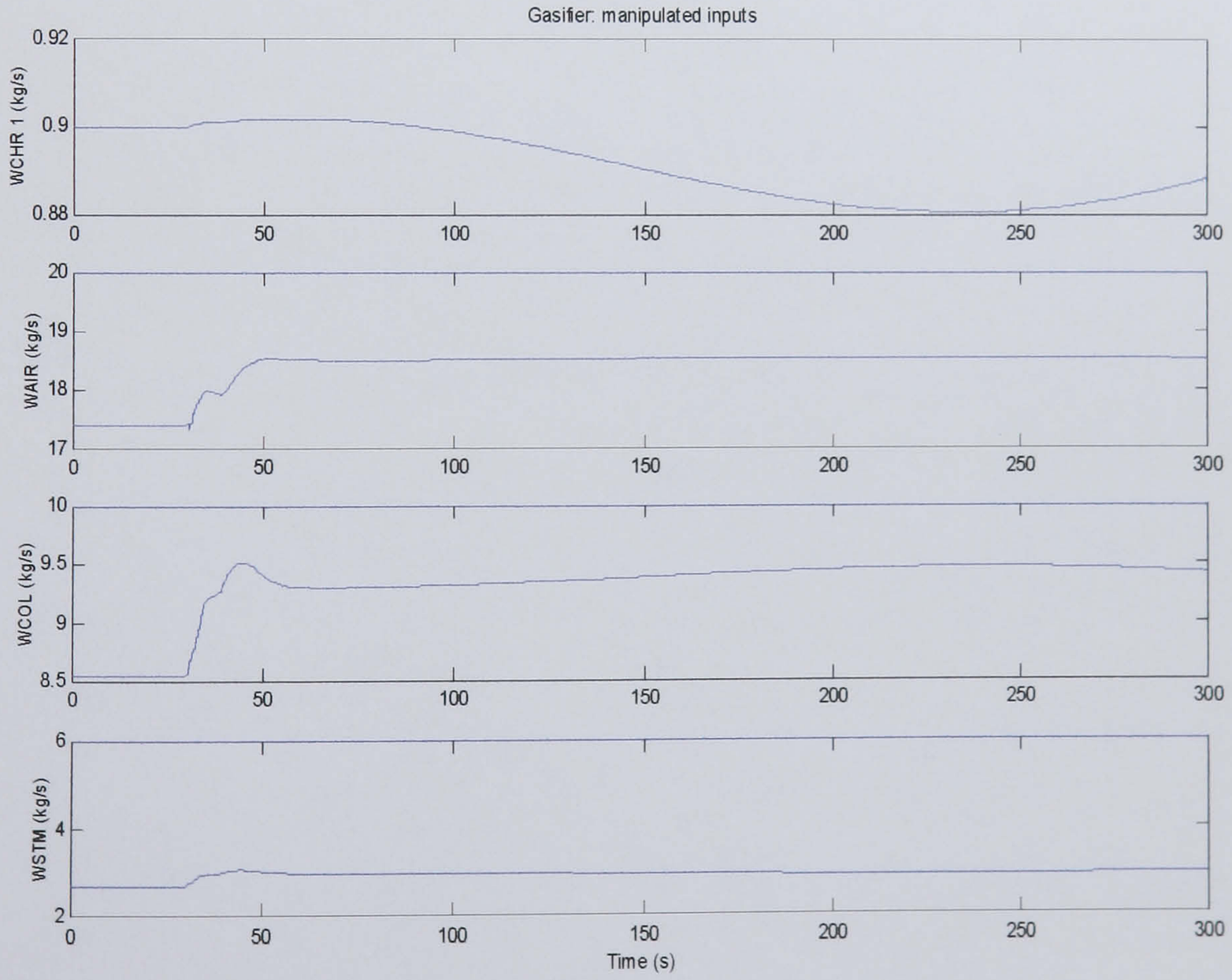


Figure (3-6): Response of the manipulated variables at 100% load to a step disturbance.

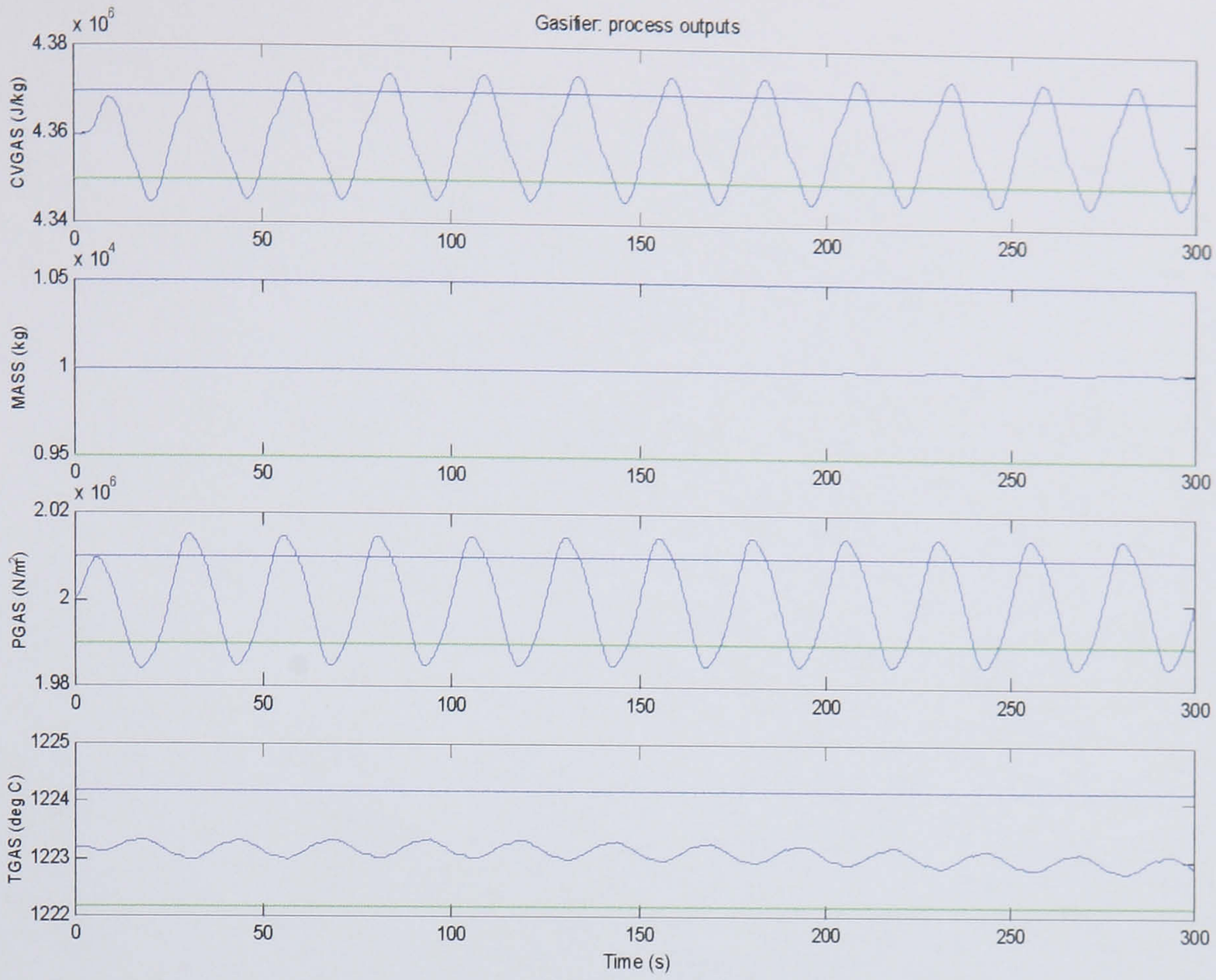


Figure (3-7): Response of the process outputs at 100% load to a sinusoidal disturbance.

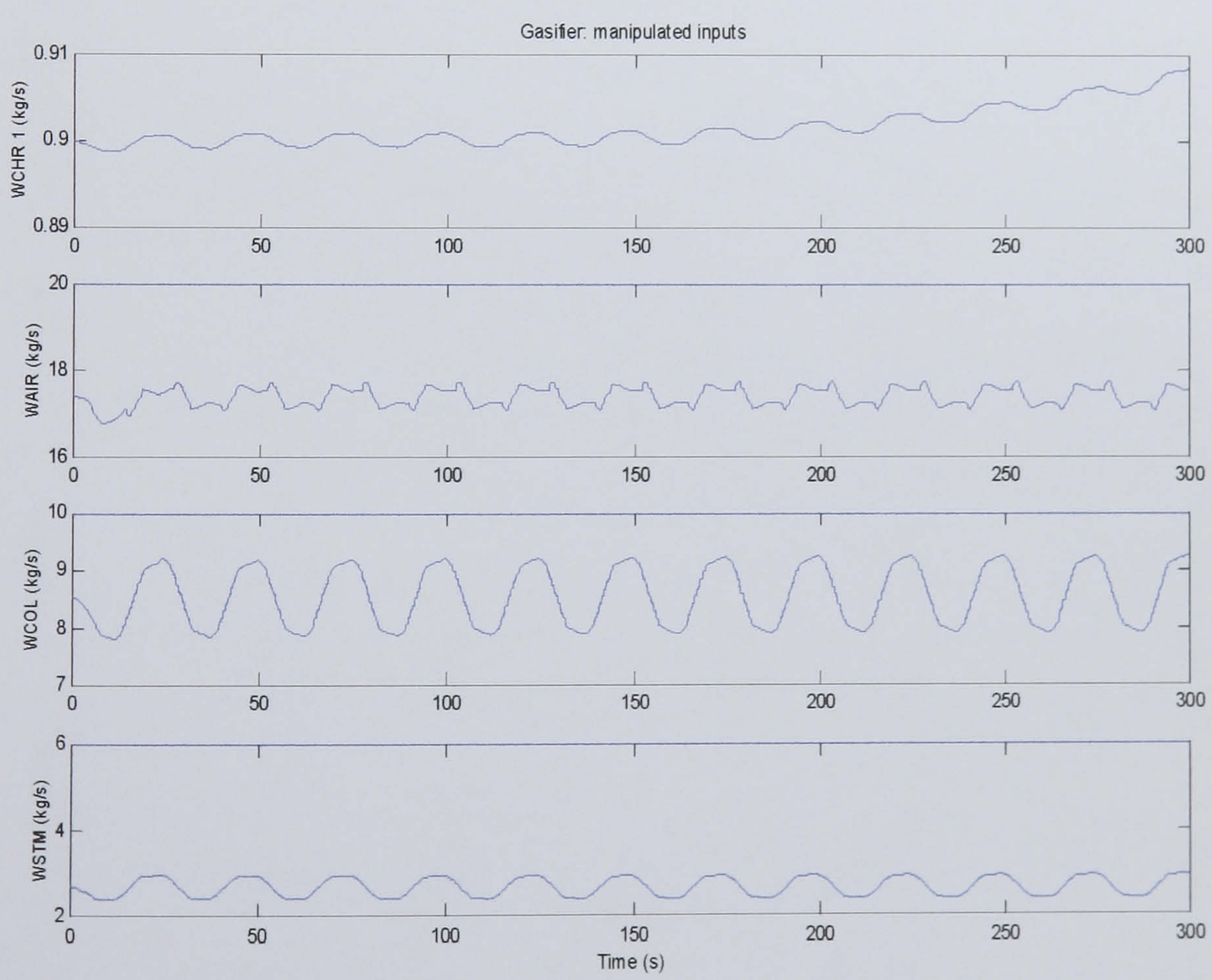


Figure (3-8): Response of the manipulated variables at 100% load to a sinusoidal disturbance.

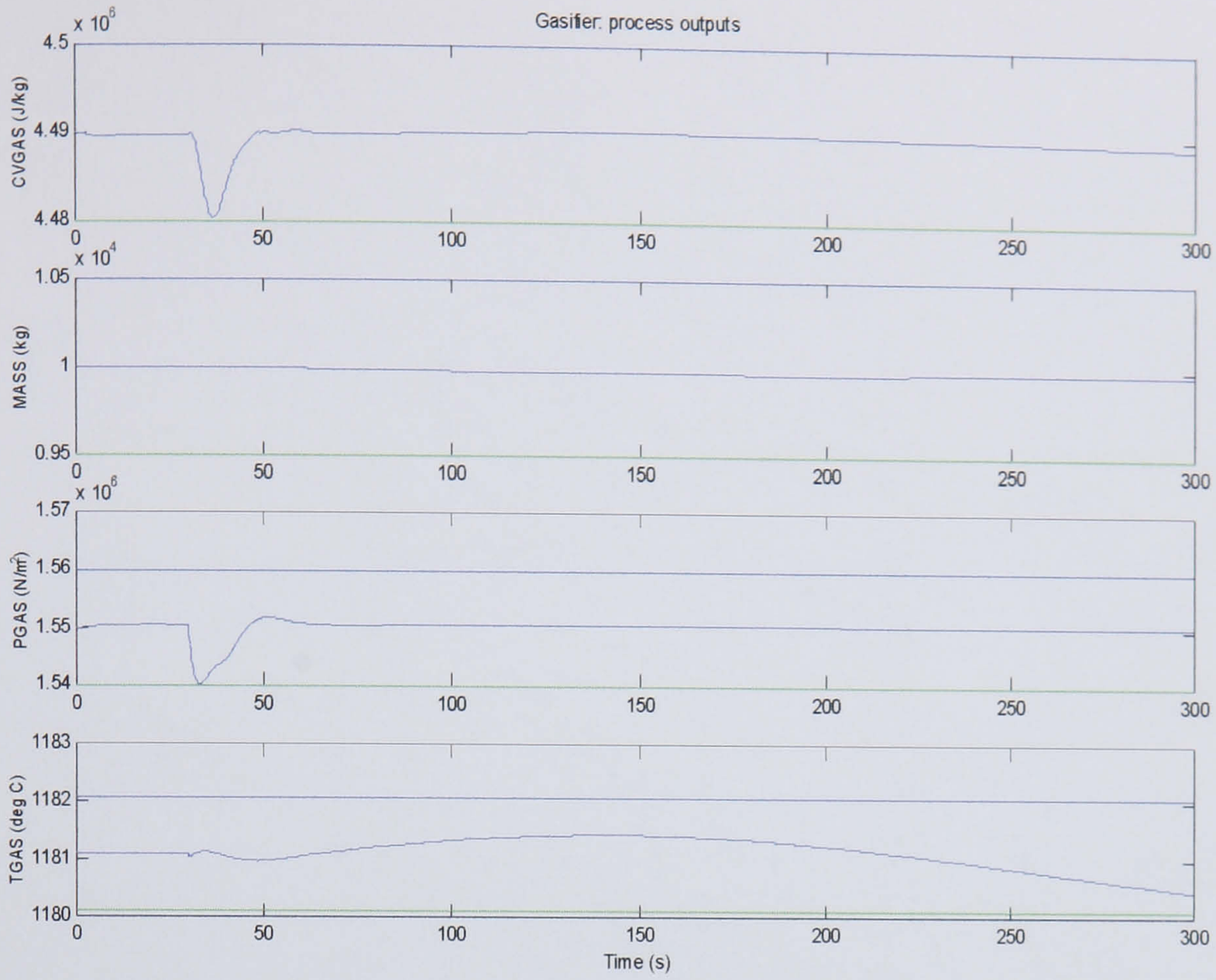


Figure (3-9): Response of the process outputs at 50% load to a step disturbance.

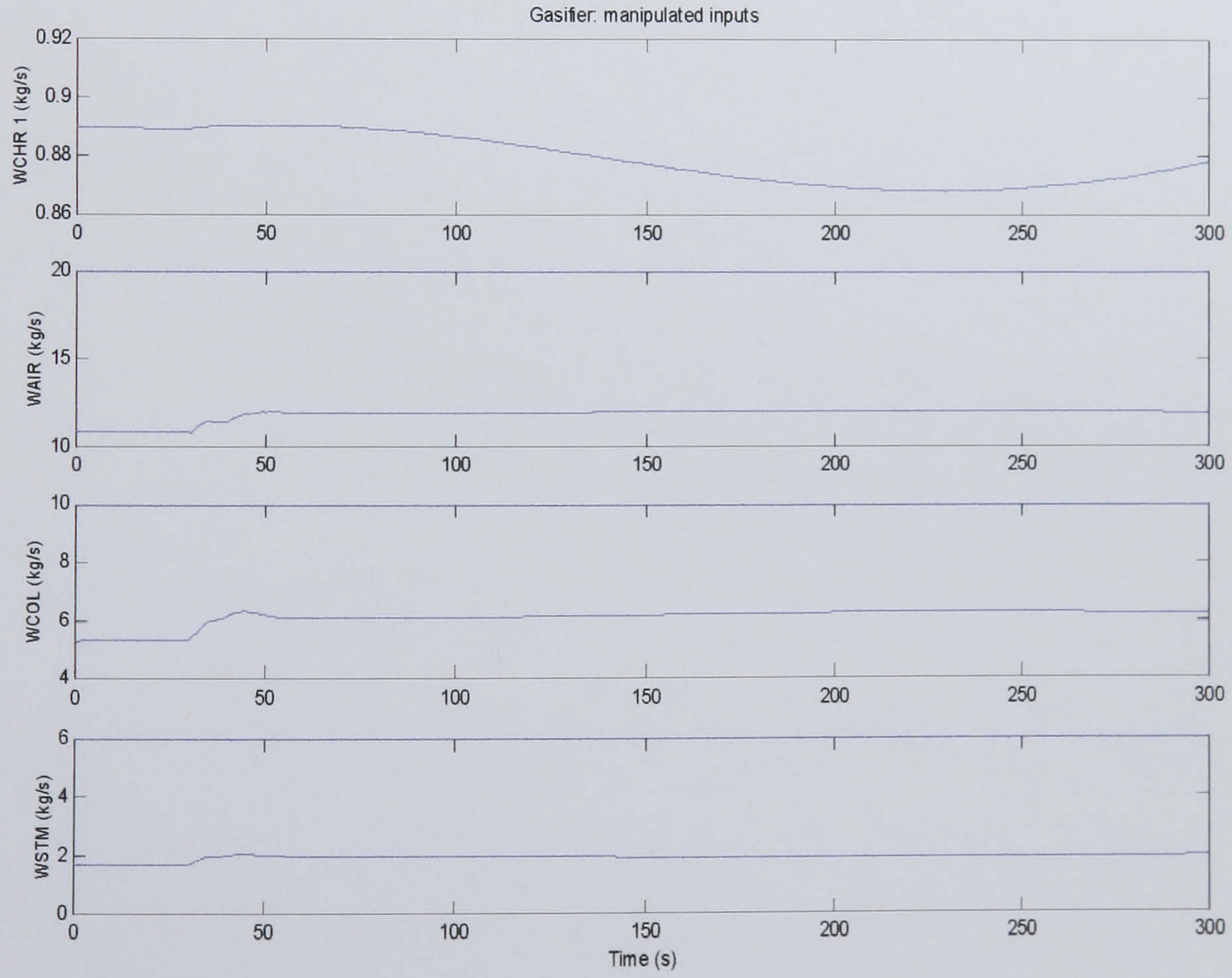


Figure (3-10): Response of the manipulated variables at 50% load to a step disturbance.

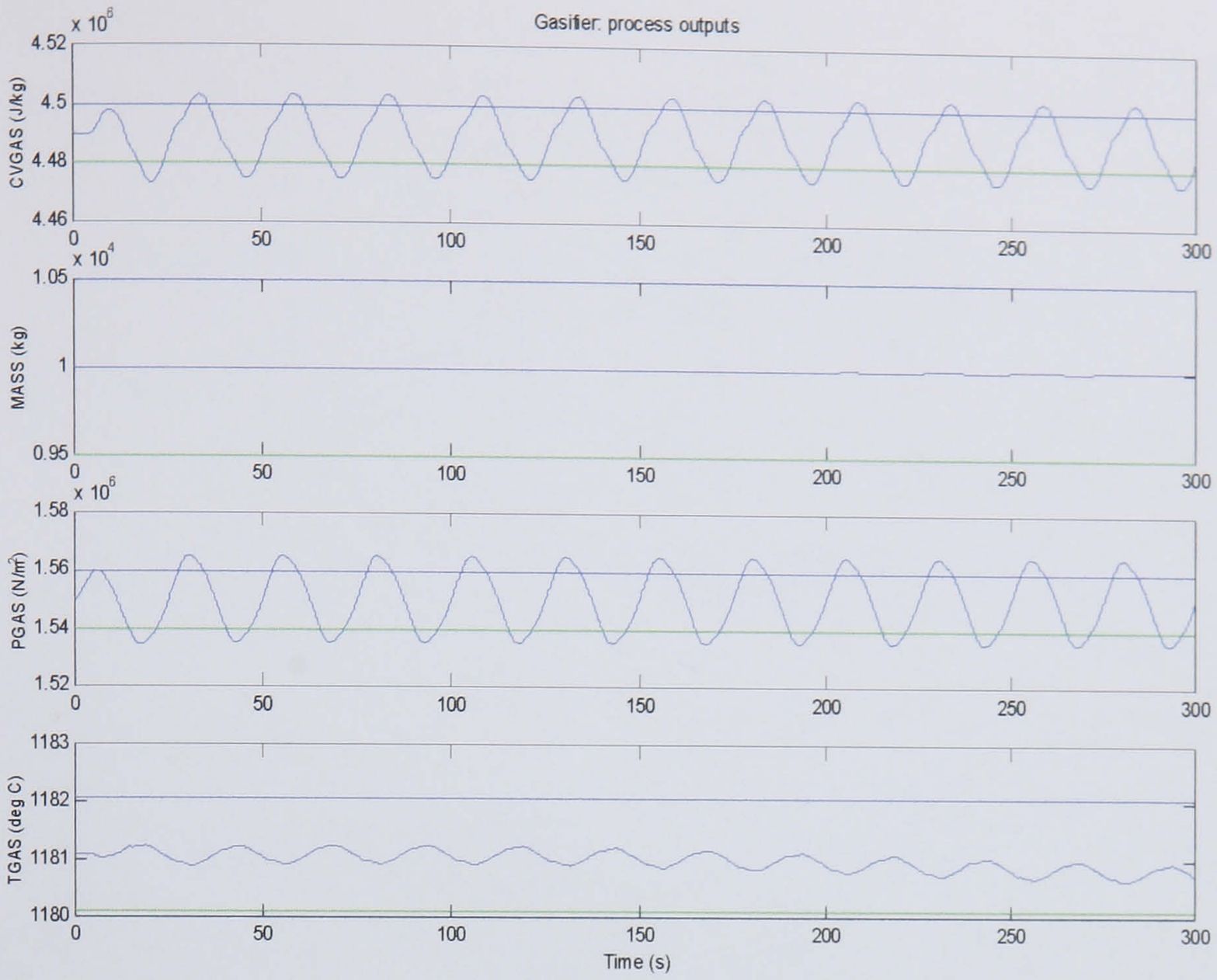


Figure (3-11): Response of the process outputs at 50% load to a sinusoidal disturbance.

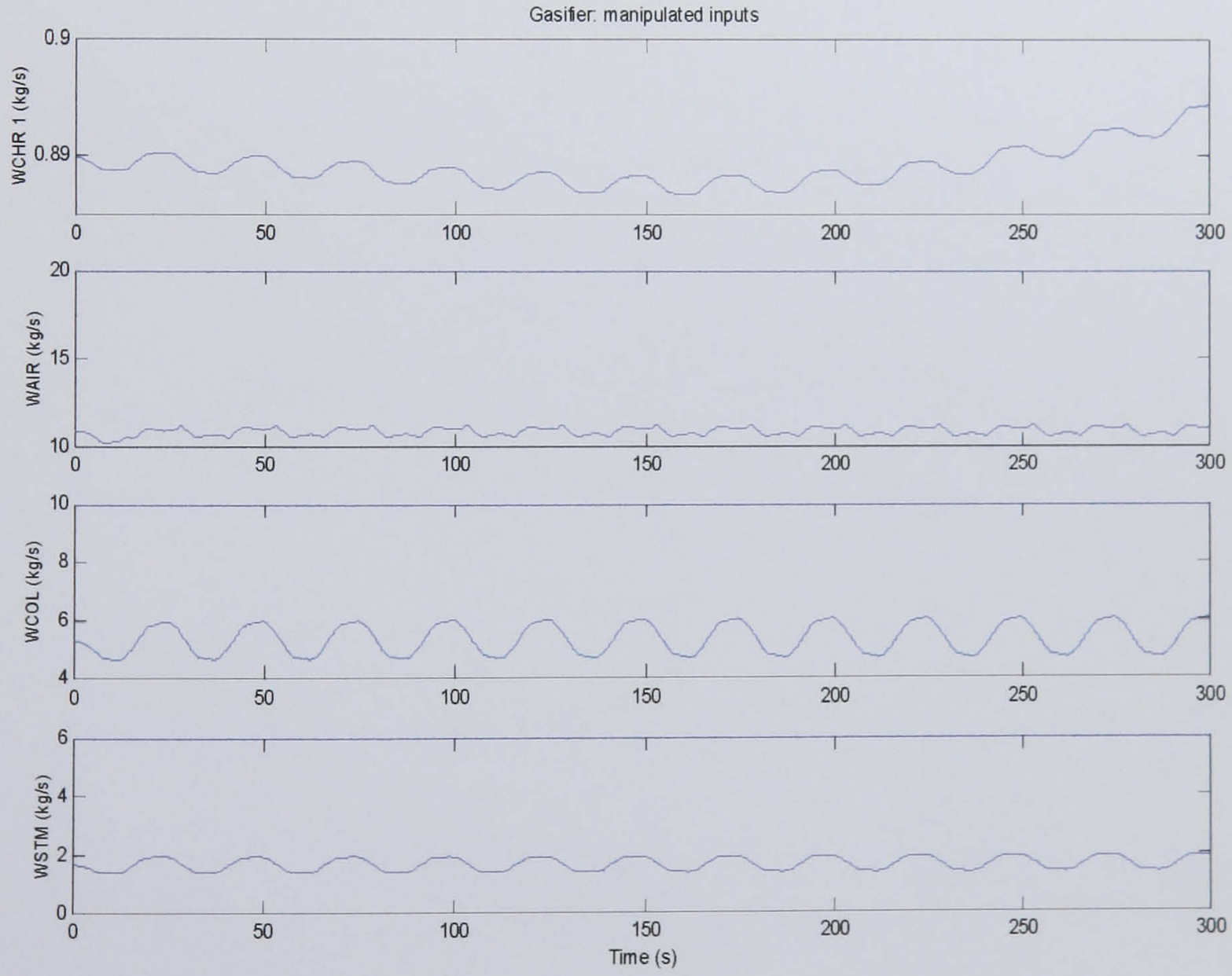


Figure (3-12): Response of the manipulated variables at 50% load to a sinusoidal disturbance.

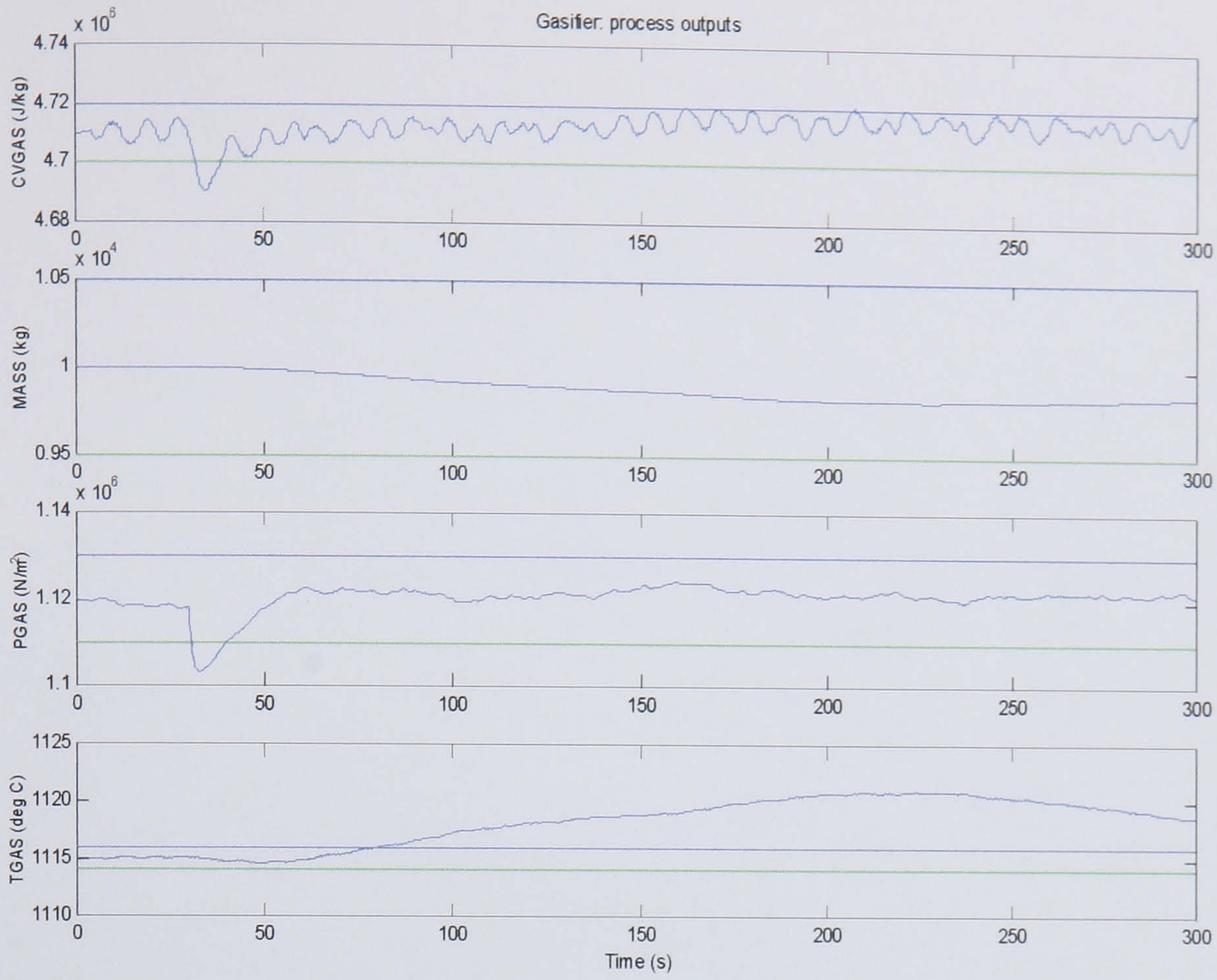


Figure (3-13): Response of the process outputs at 0% load to a step disturbance.

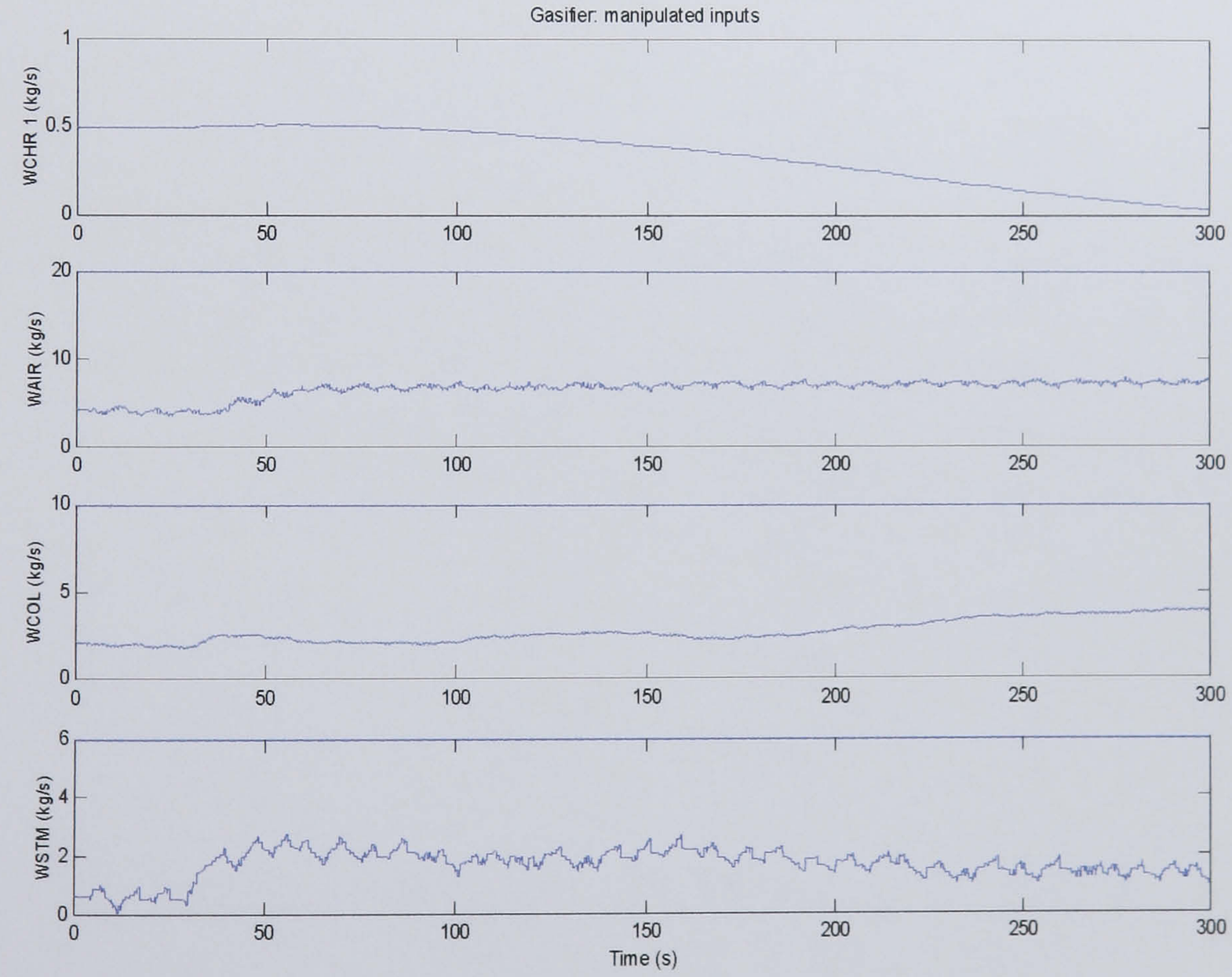


Figure (3-14): Response of the manipulated variables at 0% load to a step disturbance.

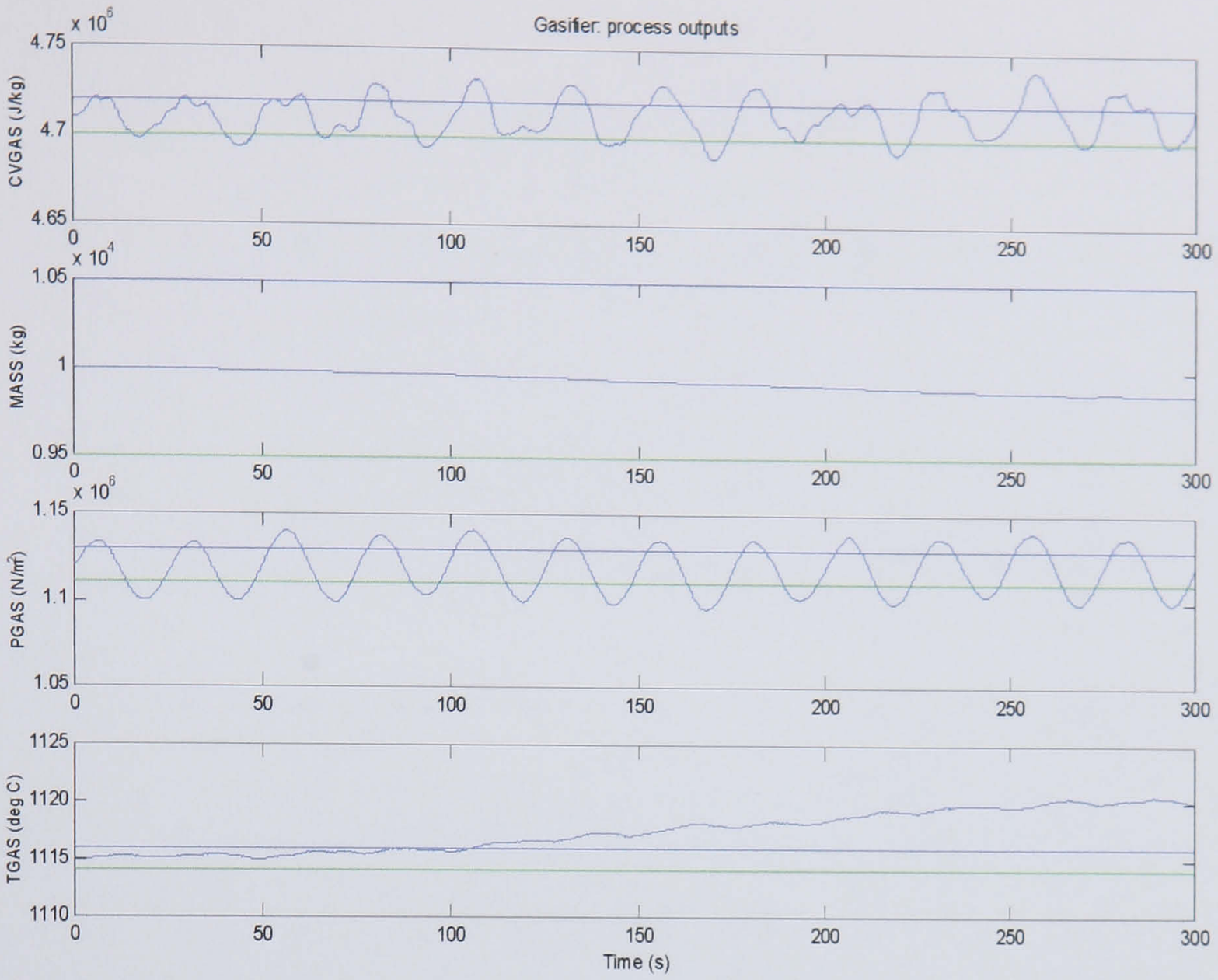


Figure (3-15): Response of the process outputs at 0% load to a sinusoidal disturbance.

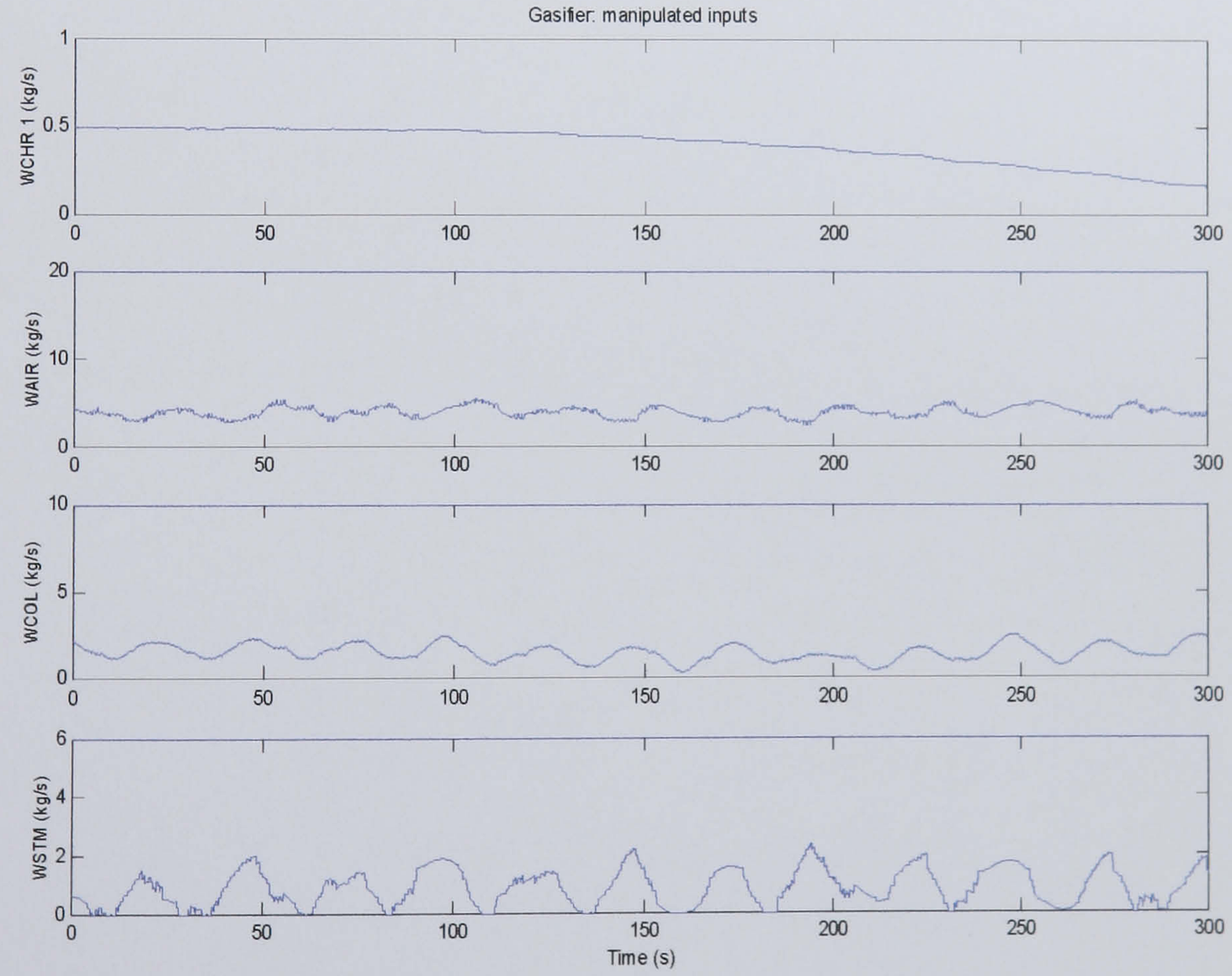


Figure (3-16): Response of the manipulated variables at 0% load to a sinusoidal disturbance.

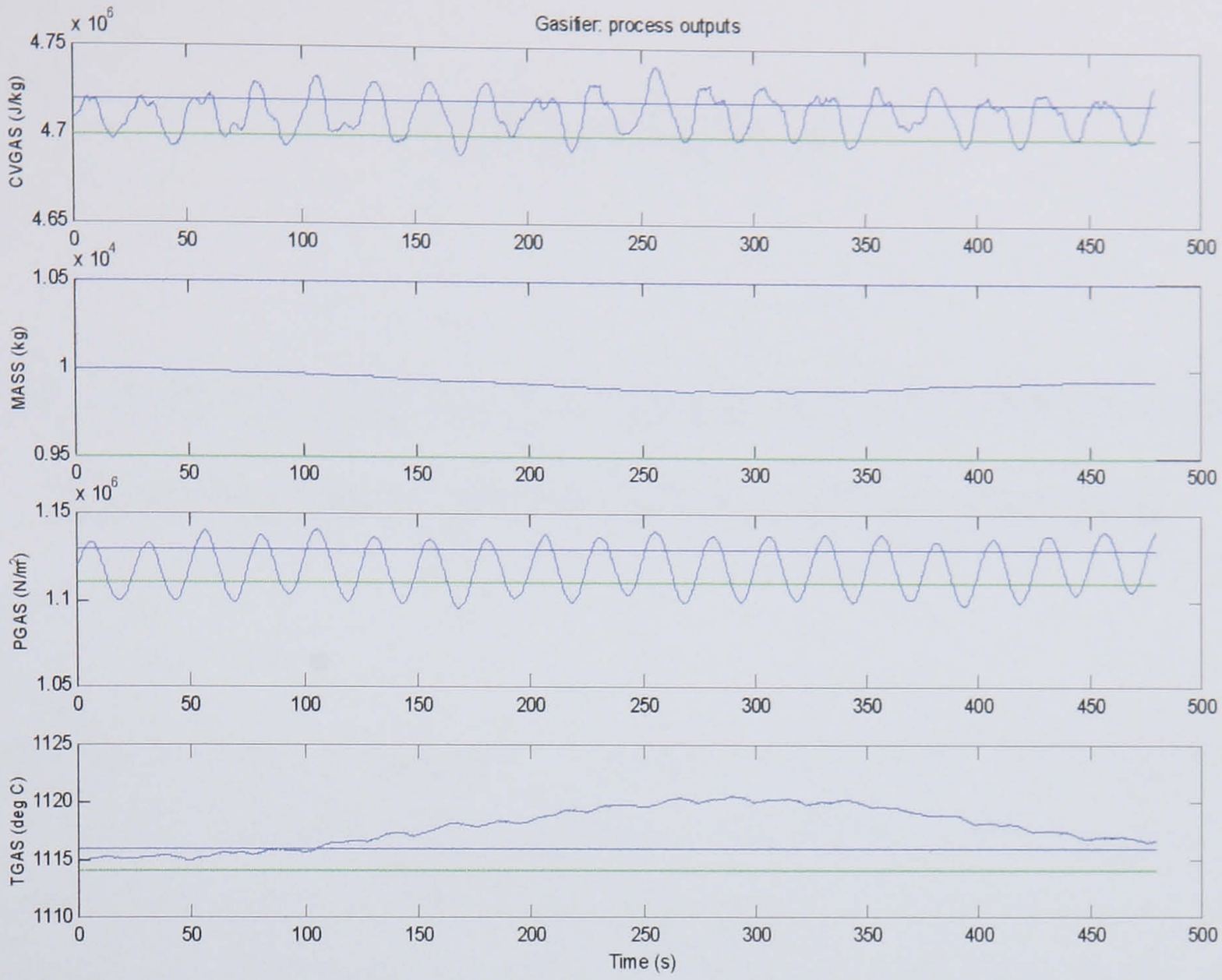


Figure (3-17): Response of the process outputs
at 0% load to a sinusoidal disturbance - extended simulation time.

CHAPTER 4

STEADY-STATE DATA RECONCILIATION AND ESTIMATION OF SYSTEMATIC BIASES.

This chapter presents a static data reconciliation module with capabilities of estimating systematic bias. The underlying assumption is that the process is at steady-state (although some simulations are carried out when this is not the case) and that there are no gross errors (outliers) present. A further assumption is that it is known which variable is corrupted by systematic bias. The fundamental principles of data reconciliation are introduced and a thorough review of previous work is presented. The algorithm is implemented as an OTISS module using C++ code and is intended to fit into a larger collection of modules which together make up an integrated dynamic data reconciliation framework.

4.1 DATA RECONCILIATION BACKGROUND

Data reconciliation which is sometimes referred to as measurement error reconciliation, is the adjustment of a set of data so the quantities derived from the data obey natural laws, such as material and energy balances. Measurements made on processes, such as flow, tank level, or temperature, are adjusted in some proportion to the standard error of the measurement. The adjustments are made using redundancies in the measurements. After adjustment, the material and, if considered, the energy balances are satisfied exactly (Bodington, 1995).

Measured process data inherently contain inaccurate information since the measurements are obtained with imperfect instruments. Therefore any set of measurements cannot be expected to obey the laws of conservation. Although we

would normally expect such errors to be random i.e. having an expected value of zero, it is often the case that biased errors are present, the expected values of which are other than zero. These could occur through malfunction of instruments, miscalibration or poor sampling and are usually broadly classified as gross errors or systematic biases (these are discussed in more detail later). Most data reconciliation techniques require that both gross errors and systematic biases be absent from the data before the reconciliation is carried out. If these error types are present, the reconciled values will exhibit *'smearing'* when compared with the true values. An additional difficulty with process data is that not all variables are measured because of cost considerations or technical infeasibility and therefore must, if possible, be estimated instead.

It is often relevant to validate and adjust the measurements taking into account the degree of precision in each measurement and key physical laws. The term *data validation* has been often used to refer to techniques for detecting gross errors and measurement bias, while the term *data reconciliation* usually refers to techniques for reconciling measured data with physical laws. Data reconciliation may be performed on a set of steady-state data, using a steady-state model of the process, or it may be applied to dynamic data, using a dynamic model of the process.

Process data is the foundation upon which all control and evaluation of process performance is based. Inaccurate process data can easily lead to poor decisions which will adversely affect many parts of the process. Many process control and optimisation activities are also based on small improvements in process performance; errors in process data can easily exceed the actual changes in process performance. Moreover, because of the immense scale of operation, the impact of any error is greatly magnified in absolute terms (Mah et al., 1976). When flawed information is used for state estimation and process control, the state of the system is misrepresented and the resulting control performance may be poor and can lead to suboptimal and even unsafe process operation (Liebman et al., 1992).

The main aim of data reconciliation is to reduce or eliminate as much as possible the effect of random measurement error on the analysis of process performance and on the predictions for future operation. Additional objectives are to improve confidence in the calculation of unmeasured variables and to identify process losses and faulty measurements.

The fundamental procedure for data reconciliation is to develop a set of natural-law balances within a process that must balance exactly. These balances are used as constraints in a mathematical minimisation. An objective function is developed that is the sum of squares of the ratios of the measurement adjustments to the measurement standard deviations. This objective is minimised subject to the balance constraints. If the balances are volume or mass balances, the constraints are linear equations. The problem is a least squares minimisation subject to linear constraints (Ham et al., 1979). Other balances involving products of variables such as flow times composition or temperature functions (enthalpy) result in nonlinear constraints. The objective function remains the same (Bodington, 1995). A quadratic objective function is desirable in reconciliation applications because the squared penalty associated with a quadratic reflects the Gaussian nature of random measurement errors (Ham et al., 1979).

The data reconciliation module described in this chapter is intended for use with systems at steady-state. Although specific to steady-state systems, these tools may also be applied in a modular fashion to dynamic systems where parts of the system have no dynamics, e.g. a valve.

4.2 STEADY-STATE DATA RECONCILIATION

Techniques for reconciling steady-state process data are well developed (Bodington, 1995). Commercial products are available from different vendors. For instance, Aspen Technology's SPEEDUP dynamic process simulator has steady-state data reconciliation and parameter estimation features. DATACON, a

product from Simulation Sciences Inc., can be interfaced to a distributed computer control system to perform gross error detection and data reconciliation. RTO+, a plant optimisation system from MDC Process Control Consultants (UK), includes steady-state data validation and reconciliation modules, which are used for periodically tuning a rigorous steady-state model of the process (Roberts, 1997). While methods for solving linear steady-state data reconciliation problems have been available for many years, similar treatment for nonlinear and dynamic systems has received much less attention (Liebman et al., 1992).

Bagajewicz and Jiang (1997) state that steady-state data reconciliation has been able to perform well in practice. However, when it comes to gross error detection, the steady-state model does not perform successfully. Due to this fact, Bagajewicz and Jiang conclude that recent work in the field is now moving towards sequential analysis, where data from different days is sequentially analysed with different statistical techniques, aiming at the identification of biases and leaks.

4.2.1 Benefits of data reconciliation

The benefits of reconciling process data are many; the following are some listed in the literature (Ham et al., 1979; Liebman, 1991):

- Monitoring of performance and higher accuracy of process yield measurements.
- More accurate operating data for technical analysis and process improvement.
- More accurate accounting and loss control.
- Production planning.
- Aid in detecting unintentional transactions involving raw material or product.
- Aid in detecting process leaks.

- Aid in detecting faulty instrumentation and in prioritisation of instrument maintenance.

In reporting on the application of Data Reconciliation at Exxon Corporation, Ham et al. (1979) refer to several specific instances where financial benefits have been realised from performing data reconciliation. Furthermore, they report significant manpower savings as a result of having a reconciliation program to analyse the data.

4.2.2 Sources and types of error

All measurements are composed of the sum of the *true* value of the measurement and an *error*. Errors can arise from (Bodington, 1995):

- Drift – a slow change in the calibration of an instrument.
- Bias – a permanent one-sided error perhaps caused by an improper installation.
- Deterioration of components, seals, etc.
- Wear of parts.
- Corrosion of sensor equipment.
- Fouling of sensors or measurement lines.
- Improper calibration of an instrument.
- Interference in an analytical procedure.
- Improper analytical procedure.

Measurement errors can be classified into three different types (Liebman et al., 1992):

(a) Small random errors;

Small random errors are typically assumed to be independent, zero-mean and normally distributed. These errors are due to the fact that measurement devices are unable to exactly reproduce measurements.

(b) Systematic biases;

Systematic biases occur when measurement devices provide consistently erroneous values, either too high or too low. In this case the expected value of the measurement error is not zero. Bias may arise from sources such as incorrect installation or calibration of the measurement device.

(c) Gross errors.

Gross errors are usually caused by non-random events. In this case, the measurement bears little or no relation to the *true* measurement value. Gross errors can be further subdivided into *measurement-related* errors such as malfunctioning sensors and *process related* such as process leaks.

It can be noticed in the literature that these classifications of error types are not well defined. A number of researchers implicitly imply that systematic biases are a type of gross error. Chen and Romagnoli (1998) explicitly suggest that there are two types of gross errors: (i) systematic biases and (ii) outliers. On the other hand, Rollins and Davis (1992) do not regard measurement bias or process leaks as gross errors. In order to avoid confusion, the classifications outlined above will be used throughout this text.

The term *outlier* is sometimes used by researchers to mean *gross error*, (Tamhane and Mah, 1985). Albuquerque and Biegler (1996) define outliers (or gross errors) as being a measurement in which the error does not follow the statistical distribution of the bulk of the data.

4.2.3 Variable classification

The processing of measurement data is usually a three step process, see Figure 4-1 (Liebman, 1991). The data reconciliation step will be tackled further in this chapter and in chapter 7 when the dynamic case is taken into account. The gross

error detection and identification step will also be addressed in chapter 7. An important aspect in the processing of measurement data is that of variable classification. It is not always convenient or desirable to measure every process variable due to cost considerations or technical feasibility (e.g. extreme process operating conditions). However, it is often possible to estimate the value of some unmeasured variables through mass, energy and component balances. This process is called *coaptation* (Liebman, 1991) and can be done simultaneously with measured data reconciliation (Liebman et al., 1992). The ability to estimate a variable depends on the placement of the measuring instruments. If it is possible to change the value of a variable without violating the conservation constraints then the change is said to be *feasible* (Mah, 1990). Further if it is possible to make a feasible change for a variable without being detected by the instruments, then the variable is *unobservable*. Thus a measured variable is certainly observable, but an unmeasured variable may or may not be observable.

Another fundamental concept to data reconciliation is that of redundancy. A measurement is said to be *redundant* if its value can be calculated based on other measurements. In other words, a given variable remains observable even if the measurement associated with it is deleted. Redundant measurements provide the conflicts with the imposed constraints which are resolved through data reconciliation. The concept of being able to calculate a value to be compared with the direct measurement is fundamental to data reconciliation (Bodington, 1995). The redundancy of measurements can be one of two types (Liebman, 1991): *spatial* or *temporal*. Spatial redundancy is as defined above, i.e. measurements are *spatially redundant* if there are more than enough data to completely define the process model at any instant in time. Measurements are *temporally redundant* if past measurement values are available and can be used for estimation purposes.

Variable classification plays an important role in the design of instrumentation schemes and for providing more insight into how the measurements relate to the physical model being used. A number of researchers have addressed the issue of variable classification, with some describing algorithms for the classification of

variables (Albuquerque and Biegler, 1996; Meyer et al., 1993; Kretsovalis and Mah, 1987). Comprehensive reviews on this topic are available (Mah, 1987, Mah, 1990; Crowe, 1996).

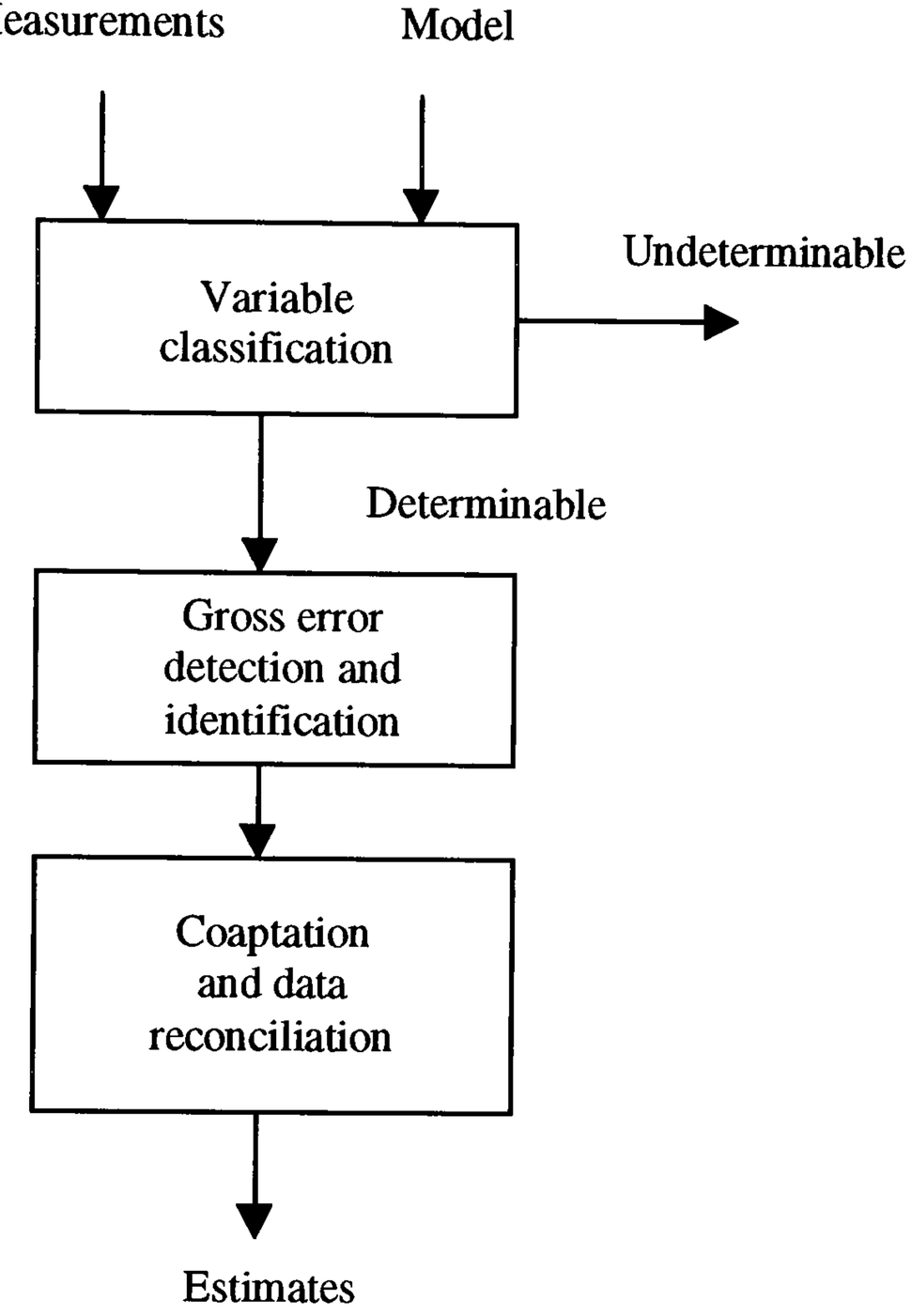


Figure (4-1): Three steps in the processing of measurement data (Liebman, 1991).

4.2.4 Review of previous work.

The issue of steady-state data reconciliation has been well addressed by a large number of researchers in the field. Kuehn and Davidson (1961) were probably the first to address the problem of data reconciliation and solved for the optimal adjustments using Lagrange multipliers for the case where all component flow rates are measured. Vaclavek (1969) published a number of articles and was a

main contributor to the field in the 1960's. In particular, he looked at the case where a stream is completely measured or is unmeasured. Ham et al. (1979) reported on the use of a data reconciliation program at the Exxon Corporation noting the benefits achieved.

Crowe et al. (1983) used a projection matrix to obtain a reduced set of balance equations from the original component balances. In particular, they examined the linear case where it is assumed that the total flow rate is measured in any stream in which a concentration is measured. Crowe (1986) omitted that assumption and looked at the nonlinear case where the balance equations contain products of unknowns. Almasy and Mah (1984) outlined an indirect method of estimating the variances of measurement errors. The direct method of obtaining the variances from serially correlated data can prove unreliable if there is a departure from the steady-state. Tamhane and Mah (1985) presented a thorough review of the data reconciliation and gross error detection problems. They outlined some problem areas and presented them in their basic essential mathematical framework in order to attract the attention of a wider circle of statisticians. They described three types of statistical tests that have been proposed for detecting gross errors.

Kretsovalis and Mah (1987) studied the effect of redundancy on estimation accuracy and concluded that redundancy never adversely affects estimation accuracy but rather it always enhances it. Pai and Fisher (1988) developed an iterative procedure for solving nonlinear data reconciliation problems. The technique makes use of Crowe's matrix projection and combines a quasi-Newton update with the Gauss-Newton scheme. MacDonald and Howat (1988) presented two methods for estimating process parameters in data reconciliation. The first method is a sequential decoupled procedure in which the data is reconciled and then the process parameters are estimated using maximum-likelihood estimation. The second method is a coupled procedure that simultaneously reconciles the data to satisfy the constraints and estimates the process parameters. The two approaches were compared.

Holly et al. (1989) applied data reconciliation and gross error detection to a chemical extraction plant. They used the composite statistical test (Narasimhan, 1984) to determine whether or not the process was at steady-state. Tjoa and Biegler (1991) presented a method to deal with data reconciliation and gross error detection simultaneously. They focused on problems with nonlinear constraints but applied to steady-state processes. The test is based on a bivariate distribution function constructed using the maximum likelihood principle. The resulting objective function which takes into account both contributions from random and gross errors is minimised. Takiyama et al. (1991) proposed a sensor-based data reconciliation method which is based on direct use of the measured variables instead of the conventional use of balanced variables. They applied their method to a pilot plant. Meyer et al. (1993) presented a data reconciliation method putting great emphasis on an algorithm for variable classification.

Narasimhan and Harikumar (1993) incorporated upper and lower bounds on process variables in the data reconciliation and gross error detection problems. Bounds on process variables are directly incorporated as constraints. Islam et al. (1994) developed a comprehensive nonlinear data reconciliation package for an industrial pyrolysis reactor. They used successive linearisation and SQP for data reconciliation and a global test along with serial elimination for detection and rectification of gross errors. Fillon et al. (1995) proposed a revised formulation of the data reconciliation problem for application to batch reactors. Bagajewicz (1996) presented a new mathematical formulation of the data reconciliation problem to take into account distortions of the probability distribution of the original signal. This distortion is caused by the propagation of measurement errors through different devices. Chen et al. (1998a) applied data reconciliation and gross error detection to a Monsanto sulfuric acid contact plant using measurement signals contaminated by Gaussian noise for gross error rectification as proposed by Tjoa and Biegler (1991).

Most recently, Romagnoli and Sanchez (2000), published a book which covers steady-state as well as dynamic data reconciliation and the treatment of gross

errors. The literature is backed up by excellent review papers, the earliest probably being that of Hlavacek (1977). Later reviews have been published by Mah (1982); Tamhane and Mah (1985); Mah (1990); Madron (1992); and Crowe (1996).

4.2.5 Applications of data reconciliation.

Applications of data reconciliation to problems of practical interest have been reviewed by Crowe (1996). Successful industrial application and testing has been reported by Ham et al. (1979) and Serth and Heenan (1986). Holly et al. (1989) applied data reconciliation and gross error detection to a chemical extraction plant. Ramamurthi et al. (1993) carried out open-loop and closed-loop simulation studies on a continuous stirred tank reactor to demonstrate the effectiveness of their successively linearized horizon based estimator (SLHE).

Bussani et al. (1995) applied a data reconciliation and optimisation package called On-line Reconciliation and Optimisation (ORO) to a hydrogen plant. The ORO package adopts a sequential modular approach. Islam et al. (1994) and Weiss et al. (1996) applied static data reconciliation to an industrial pyrolysis reactor using successive linearization. Reconciliation has also been applied to mineral processes, Hodouin et al. (1988). A few years previously, Hodouin et al. (1982) applied data reconciliation to a cement clinker grinding process and looked at sensitivity analysis of material balance calculations.

4.2.6 Formulation of the data reconciliation problem

Data reconciliation is the process of adjusting the process measurements – which are subject to error, to obtain values that are consistent with the material and energy balances. The simplest case is a process operating in steady-state where all the desired variables are measured and no gross errors are present in the

measurements. This implies that the measurement error is Gaussian with known variances and the mean of measurement errors is assumed to be zero. The measurement vector (y_m) can be written as:

$$y_m = y_{true} + \varepsilon \tag{4.1}$$

where y_{true} is the vector of the true values of the variables, ε is a vector of random measurement errors that are normally distributed with zero mean, and a covariance matrix V.

The data reconciliation problem can be stated as a constrained least squares estimation problem where the weighted sum of errors is to be minimised subject to constraints:

$$\min_{y_{true}} \ (y_m - y_{true})^T V^{-1} (y_m - y_{true})$$

subject to:

$$f(y_{true}) = 0. \tag{4.2}$$

The constraints arise because the mass balances, energy balances and any other performance equations must be satisfied, and are encapsulated in the term $f(y_{true})$. Several methods have been used to solve the optimisation problem. They are listed below.

4.2.6.1 Linear solution

If the constraints are linear, or linearised if they are almost linear, then problem (4.2) can be reduced to an unconstrained Quadratic Programming problem which can be solved analytically. The solution is obtained by the use of Lagrange multipliers and is given by:

$$y_{true} = y_m - VA^T(AVA^T)^{-1}\delta \tag{4.3}$$

where A is the Jacobian of the constraint equations and δ is the residual of the unsatisfied balances and is described by:

$$\delta = A\varepsilon = Ay_m \tag{4.4}$$

since Ay_{true} is zero.

4.2.6.2 Successive linearisation methods

A shortcoming of the linear solution is that the solution does not necessarily satisfy the non-linear constraints. In successive linearisation, the linear problem is iterated until an optimal point is obtained satisfying the non-linear constraints. As in the linear solution method, the advantage of successive linearisation is its relative simplicity and fast calculation.

4.2.6.3 Non-linear methods

These methods directly solve problem (4.2) as a general non-linear programming (NLP) problem. The non-linear programming solution makes it simple to augment equations (4.2) with upper and lower bounds on the variables, which may lead to a better formulated problem. The additional constraints are:

$$y_{true,l,i} \le y_{true,i} \le y_{true,u,i} \qquad \forall i, \tag{4.5}$$

where $y_{true,l,i}$ and $y_{true,u,i}$ refer to the lower and upper constraints on variable $y_{true,i}$.

4.2.7 Formulation of the bias estimation problem

Systematic bias can be estimated as a parameter (McBrayer and Edgar, 1995). The objective function is formulated as follows:

$$J = \left(\frac{\bar{y}_1 - \left(y_{m1} - \hat{b}_1\right)}{\sigma_1}\right)^2 + \left(\frac{\bar{y}_2 - \left(y_{m2} - \hat{b}_2\right)}{\sigma_2}\right)^2 + \cdots + \left(\frac{\bar{y}_i - \left(y_{mi} - \hat{b}_i\right)}{\sigma_i}\right)^2$$

subject to:

$$\begin{aligned} & f(\bar{y}) = 0. \\ & \bar{y}_{l,i} \le \bar{y}_i \le \bar{y}_{u,i} \qquad \forall i, \\ & \hat{b}_{l,i} \le \hat{b}_i \le \hat{b}_{u,i} \qquad \forall i, \end{aligned} \tag{4.6}$$

where y_{mi} is the ith measured variable, $\bar{y}_i$ is the ith estimate, σ_i is the measurement noise standard deviation of the ith measured variable and $\hat{b}_i$ is the estimate of bias on the ith measured variable. Note that $\hat{b}_i$ is also included in the inequality constraints. This allows for physical limits on the range of admissible biases.

4.3 THE STATIC DATA RECONCILIATION MODULE

4.3.1 Simulation case study

Simulations were carried out on a dynamic model of two Continuous Stirred Tank Reactors (CSTR) connected in series where an exothermic autocatalytic reaction takes place (Figure 4-2). The two units interact in both directions due to the recycle of a 50% fraction of the product stream into the first reactor. Regulatory controllers are used to control the temperature in both reactors, and the dynamics of these controllers are neglected. Full details of this model can be found in Garcia and Morari (1981) and are briefly outlined below.

The reaction that takes place in the reactors is:

$$A + B \underset{k^-}{\overset{k^+}{\Leftrightarrow}} 2B \tag{4.7}$$

where one molecule of species A reacts with one molecule of species B to produce two molecules of species B and this reaction is reversible.

The dynamic equations describing the model are as follows:

$$\begin{aligned}
\frac{dC_{a1}}{dt} &= \frac{0.5}{\tau_1}(C_{a0} + C_{a2}) - \frac{C_{a1}}{\tau_1} - (k_{1+}C_{a1}C_{b1} - k_{1-}C_{b1}{}^2) \\
\frac{dC_{b1}}{dt} &= \frac{0.5}{\tau_1}C_{b2} - \frac{C_{b1}}{\tau_1} + (k_{1+}C_{a1}C_{b1} - k_{1-}C_{b1}{}^2) \\
\frac{dC_{a2}}{dt} &= \frac{C_{a1}}{\tau_2} - \frac{C_{a2}}{\tau_2} - (k_{2+}C_{a2}C_{b2} - k_{2-}C_{b2}{}^2) \\
\frac{dC_{b2}}{dt} &= \frac{C_{b1}}{\tau_2} - \frac{C_{b2}}{\tau_2} + (k_{2+}C_{a2}C_{b2} - k_{2-}C_{b2}{}^2)
\end{aligned} \tag{4.8}$$

where C_{xi} is the concentration of species x in tank i, $\tau_1 = 30$ [min] is the residence time of reactor 1, $\tau_2 = 25$ [min] is the residence time of reactor 2, $k_{i\pm} = A_{\pm}\exp(-E_{\pm}/RT_i)$ are the reaction rates where $E_+/R = 17{,}786$ [K], $E_-/R = 23{,}523$ [K], $A_+ = 9.73\times10^{22}$ [m^3/Kmol s] and $A_- = 3.1\times10^{30}$ [m^3/Kmol s]. $C_{a0} = 0.1$ [Kmol/ m^3] is the feed concentration of A, T_1 is the temperature in reactor 1, T_2 is the temperature in reactor 2.

In all the simulation cases that follow, the system was started from the steady-state condition given by the set-points $T_1(0) = 307\,K$ and $T_2(0) = 302\,K$, which yield steady state values $C_{b1}(0) = 0.05165$ [Kmol/ m^3], $C_{b2}(0) = 0.05864$ [Kmol/m^3]. The sampling time for the measurements was 1 minute. Note that the overall open loop time constant of the process is approximately 40 minutes (Garcia and Morari, 1981). It is worth noting that the simulation times which appear in the results that follow and in subsequent chapters relate to the real plant. The simulations would typically run at speeds of between 10 to 100 times faster depending on the computational load on the algorithm.

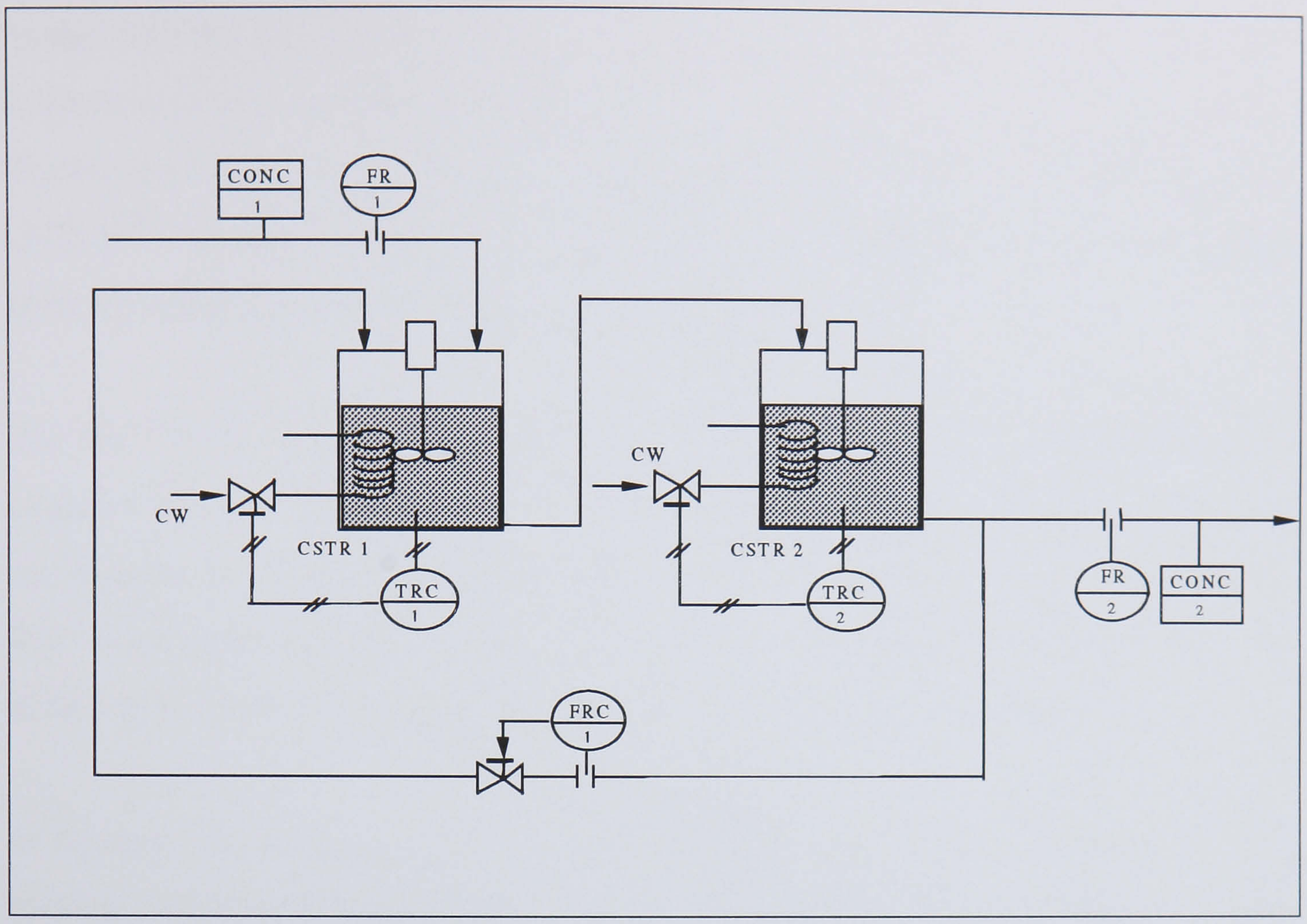


Figure (4-2): Simulation case study - two Continuous Stirred Tank Reactors in series.

4.3.2 Implementation issues

The static data reconciliation module was first implemented standalone using C++ code and utilising an SQP algorithm developed at City University (Becerra, 1998). As a test case, a static model of two continuous stirred tank reactors connected in series was used. Tests showed that the data reconciliation algorithm was working very well in estimating bias. Although some of the tests concerning the estimation of physical parameters were not good, the fact that it worked in some cases indicated that the algorithm itself was working. Where in some cases bad results were obtained may just point to the possibility that the problem itself is ill conditioned or it is just not possible to estimate the parameters from the information available.

With confidence in the algorithm achieved, implementation using a dynamic model was the next natural step. The static data reconciliation algorithm was implemented as a separate OTISS™ module (Abu-el-zeet et al., 1999b, 2000). Testing took place using the existing dynamic model of the two CSTR plant in OTISS™. A static model of the two CSTR was still used as part of the Static Data Reconciliation (SDR) module to reconcile the data.

The two CSTR plant described in section 4.3.1 has 4 outputs which are basically concentrations in the two tanks. It is assumed here that we are able to measure two outputs only and must estimate the other two. The SDR module must also be able to estimate any bias present on any or both of the measured outputs. In addition the estimation of physical parameters must also be possible.

A number of tests were carried out involving in the outset the choice of the starting positions (initialisation) of the various variables. An arbitrary set was used first, then the choice of the initial values was made as follows:

- Unmeasured variables were set at values 10% higher than the nominal values.
- Measured variables as well as their estimates were set at values 7% higher than the nominal values. Note that the measured variables were also contaminated with noise.

The values of standard deviations used in the simulations were σ_1, which was set at 5% of the nominal value of C_{b1}, and σ_2 which was set at 2% of the nominal value of C_{b2}.

4.3.3 Simulation results

Four sets of simulations were carried out to test the behaviour of the SDR module, these are:

1. Reconciliation of biased data and estimation of systematic bias.
2. Estimation of physical parameters.
3. Behaviour in the presence of transients.
4. Behaviour when there are parametric differences between the process (dynamic model) and the static model.

4.3.3.1 Reconciliation of biased data and estimation of systematic bias.

A number of tests were carried out to test how the algorithm handles bias on the measurements. Different values of bias were tested, firstly on the measured variable C_{b1} then on C_{b2} and finally on both variables together. Table (4.1) shows a summary of all the tests carried out and the number of SDR iterations required before a solution is found. Also shown in the table is a list of Figures relating to the various simulations. Simulations 1 to 12 are those related to the estimation of bias. The level of bias added to the measurement C_{b1} was in the range -50% to +50% of the nominal value, whereas the level of bias added to C_{b2} was in the range -20% to +20% of the nominal value.

The simulations show that the module is correctly carrying out data reconciliation, see Figures (4-3) and (4-4). The measured as well as the unmeasured variables are being correctly estimated. In the majority of the figures that follow Figure (4-3), the estimates of the unmeasured variables are not shown for the purpose of clarity. In terms of accuracy, the estimates of the unmeasured variables and those of measured variables are very similar. Therefore, the estimates of unmeasured variables will only be presented if there is a significant difference from those of measured variables or to highlight the algorithm's ability under varying circumstances. In all the simulations (1 to 12), the algorithm finds a solution in 1 to 2 SDR iterations, a maximum of 2 minutes, despite the presence of relatively large magnitudes of bias. Some of these simulations were carried out with bias added to both C_{b1} and C_{b2}, see for example Figure (4-5) where bias on C_{b1} is -25% of the nominal value and bias on C_{b2} is +10% of the nominal value.

4.3.3.2 Estimation of physical parameters

A number of simulations were carried out to test the algorithm's ability to estimate physical parameters. These are summarised in Table (4.1), simulations 13-16. Plots of the results related to simulations 13,15 and 16 can be seen in Figures (4-6), (4-7), (4-8), (4-9), (4-10) and (4-11). The parameter that was estimated was C_{a0}, the feed concentration.

Figures (4-10) and (4-11) show the results when estimation is being carried out with no added bias on either of the measured variables. It can be seen from Figure (4-11) that the correct value of $C_{a0} = 0.1$ is being estimated. Figures (4-6) and (4-7) show the estimation of C_{a0} with bias added to C_{b1} equivalent to +25% of the nominal value. Again Figure (4-7) shows that C_{a0} is being estimated correctly. Figures (4-8) and (4-9) show the estimation of C_{a0} with bias added to both measured variables. Bias on C_{b1} is equivalent to –25% of the nominal value and bias on C_{b2} is +10% of the nominal value. It can be observed from Figure (4-9) that the algorithm is not correctly estimating the parameter C_{a0} and in fact it is clear from Figure (4-8) that the estimator is not coping at all with the estimation of any of the variables. As mentioned previously, this may indicate that the problem itself is ill conditioned or it is not possible to correctly estimate parameters from the limited information available. It can be observed from Figures (4-6) and (4-10) that the estimates of the measured variables follow the noise contaminated measurements.

4.3.3.3 Behaviour in the presence of transients

To test the module when a transient was present, the temperature in Tank 1 (T_1) was changed from its steady-state value of $307K$ to $310K$. Several simulations were carried out, summarised in Table (4.1), simulations 17 to 20.

Figure (4-12) shows the behaviour of the static data reconciliation module in the presence of a transient but without the presence of bias. It can be observed that the correct estimates are achieved following the introduction of the transient. There is a slight delay before this happens as can be seen from the plot and in fact according to Table (4.1), this is equivalent to 25 SDR iterations. This delay is probably due to the fact that during the transient the static model does not properly represent the behaviour of the process. Despite this slight delay, it is nevertheless interesting to note that the SDR algorithm seems to predict the steady-state that will be achieved by the process variables following a step disturbance. It is important to note that in the SQP algorithm used, the initial guess is retained as the final value if SQP reaches the maximum number of iterations without converging. This is why the estimates remain constant when a solution is being found.

Figure (4-13) shows the behaviour in the presence of bias as well as the transient. The added bias is +25% of the nominal value on C_{b1}. The correct estimates of both measured and unmeasured variables (not shown in Figure (4-13)) are found although some delay is present. Figure (4-14) shows the behaviour when bias is present on both measured variables (-25% on C_{b1} and +10% on C_{b2}). The correct estimates are achieved instantly with one SDR iteration. The swiftness in achieving a solution may seem quite strange as the algorithm needed a number of iterations even when bias was not present (Figure 4-13). An explanation of this may be found by closely looking at the initialisation points of the algorithm. Some cases of bias may be offset by the initialisation point. A further explanation is that the measured variables are contaminated with random noise. So in fact even two identical simulation runs will not produce exactly the same results.

Figure (4-15) shows the behaviour with bias added to C_{b1} and also estimating C_{a0}. Figure (4-16) shows the estimate of C_{a0}. It can be observed that the measured and unmeasured variables are all being estimated correctly. Also the disturbance when the transient enters is rejected and C_{a0} is estimated correctly.

4.3.3.4 Behaviour in the presence of parametric differences

Simulations 21 to 24 in Table (4.1) were carried out to show the algorithm's behaviour when there existed some parametric differences between the process (dynamic model) and the static model used for estimation. Tests were carried out assuming a difference in the value of C_{a0} which equals 0.1 in the dynamic model of the process. Different values for C_{a0} in the static model were tried (from 0.07 to 0.0999). The algorithm did not converge when C_{a0} in the static model was set to 0.07, Figure (4-17). However for values closer to the actual value, the algorithm converged but gave a slightly incorrect solution depending on how large the difference, see for example Figure (4-18). For a reasonable estimate the difference had to be 0.0001 or less (i.e. C_{a0} in static model = 0.0999). In all cases where the algorithm converged, a solution was achieved in 1 to 2 SDR iterations.

Sim.	Fig.	Cb1 Bias	Cb2 Bias	Ca0 Estimated	No. of iterations	Comments
1	3	0.0129		No	1	
2		0.0258		No	1	
3		-0.0129		No	2	
4		-0.0258		No	2	
5	4		0.00585	No	2	
6			0.0117	No	1	
7			-0.00585	No	1	
8			-0.0117	No	1	
9		0.0129	0.00585	No	2	
10		-0.0129	-0.00585	No	1	
11	5	-0.0129	0.00585	No	1	
12		0.0129	-0.00585	No	1	
13	6,7	0.0129		Yes	1	
14			0.00585	Yes	1	
15	8,9	-0.0129	0.00585	Yes	No Convergence	Unstable like behaviour
16	10,11			Yes	1	
17	12			No	25 (after transient)	T1 step from 307 to 310
18	13	0.0129		No	20 (after trans.)	T1 step from 307 to 310
19	14	-0.0129	0.00585	No	1 (after trans.)	T1 step from 307 to 310
20	15,16	0.0129		Yes	1 (after trans.)	T1 step from 307 to 310
21				No	2	Ca0 = 0.095
22	17			No	No Convergence	Ca0 = 0.07
23	18			No	1	Ca0 = 0.099
24				No	1	Ca0 = 0.0999

Table (4.1): Various simulations carried out on the SDR module.

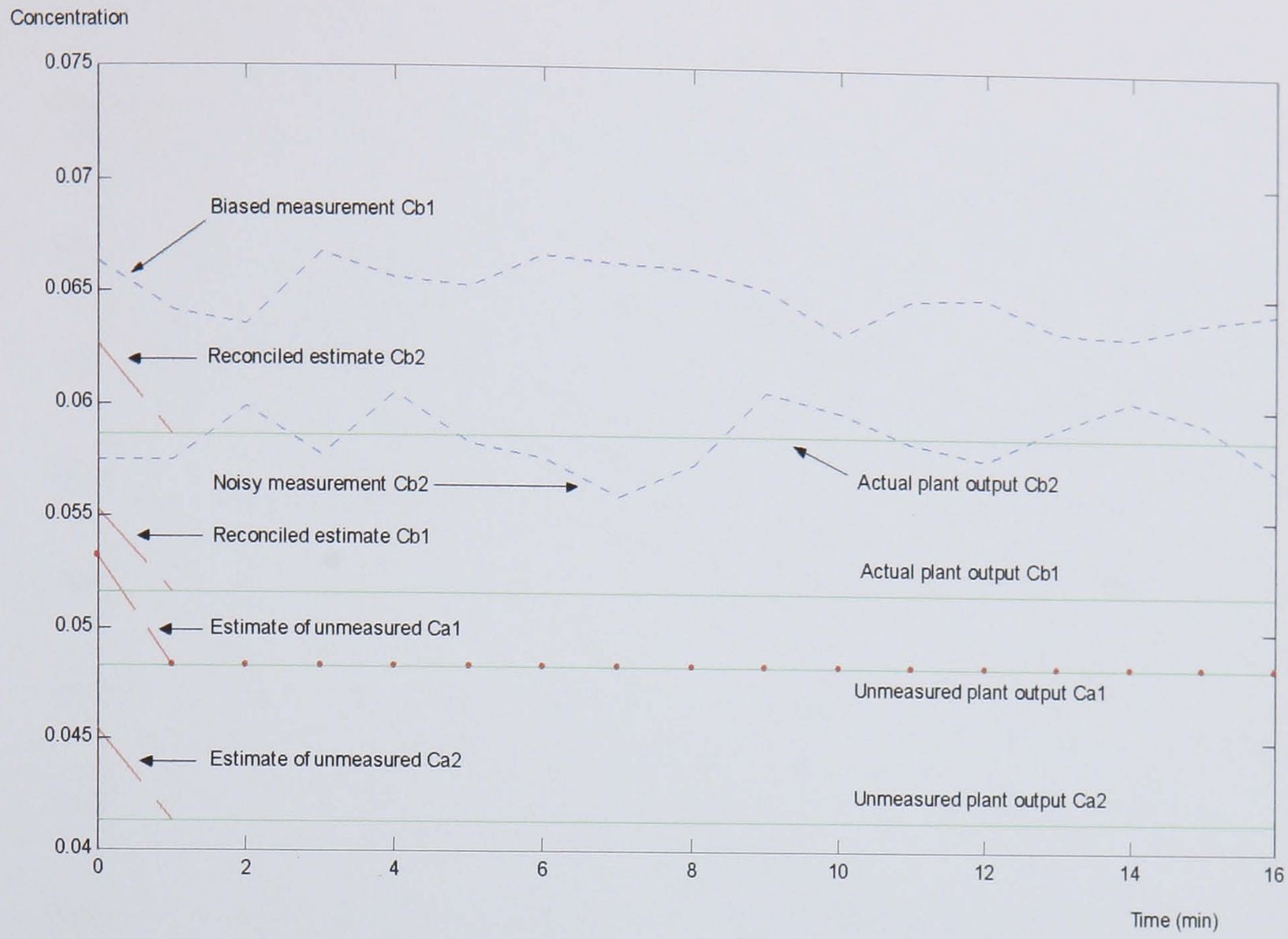


Figure (4-3): Reconciliation of measurement data when C_{b1} is biased.

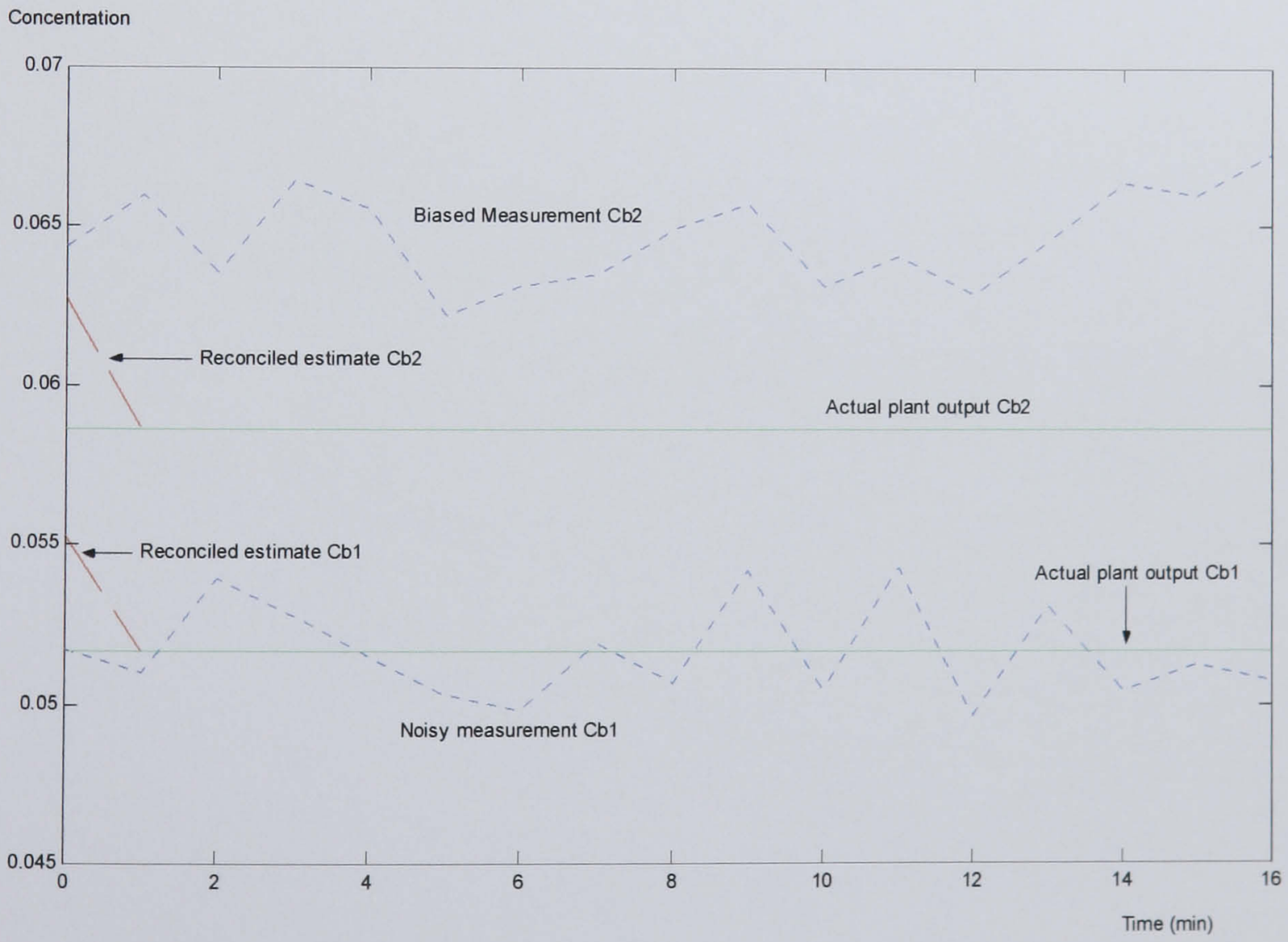


Figure (4-4): Reconciliation of measurement data when C_{b2} is biased.

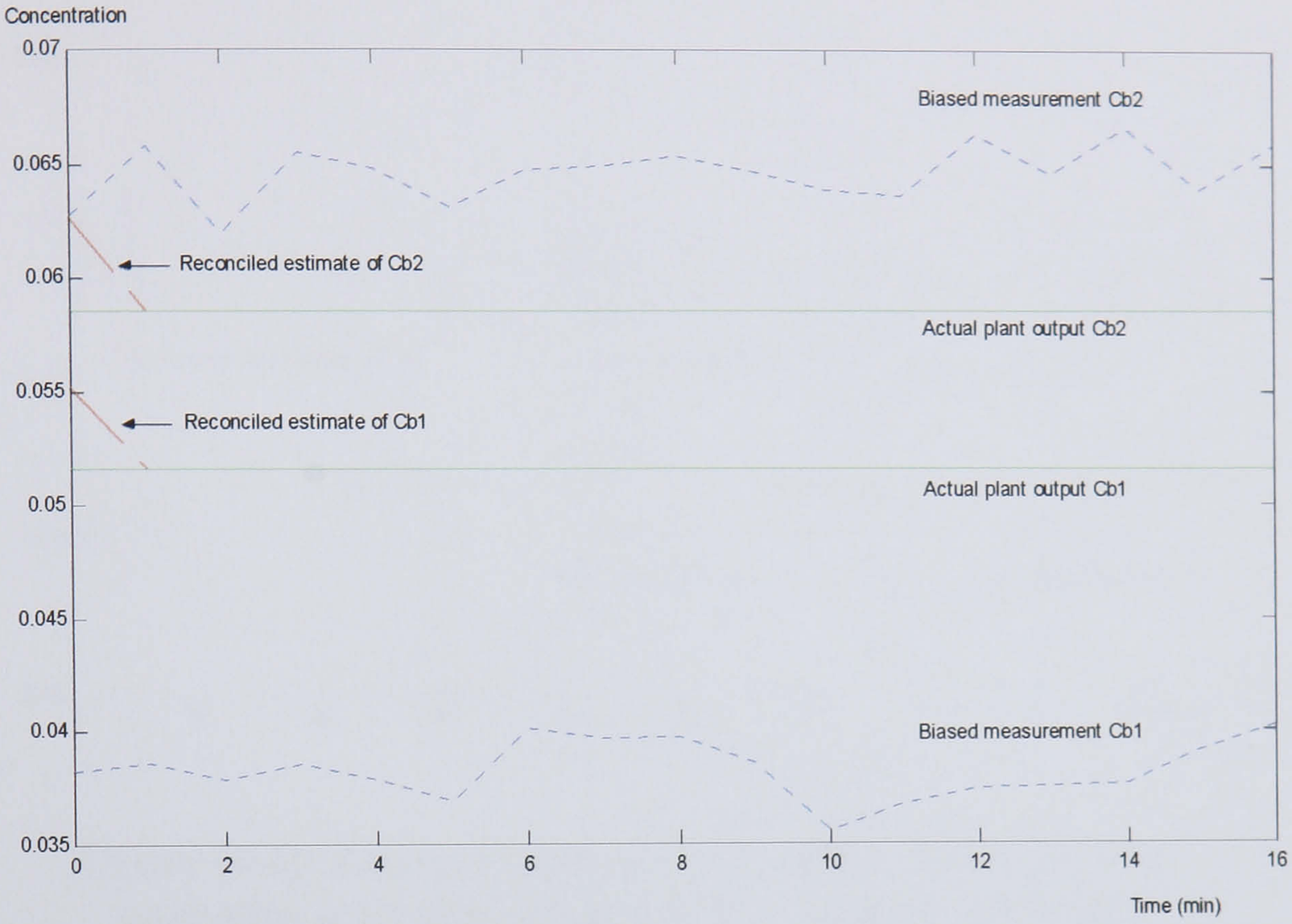


Figure (4-5): Reconciliation of measurement data when both measurements are biased.

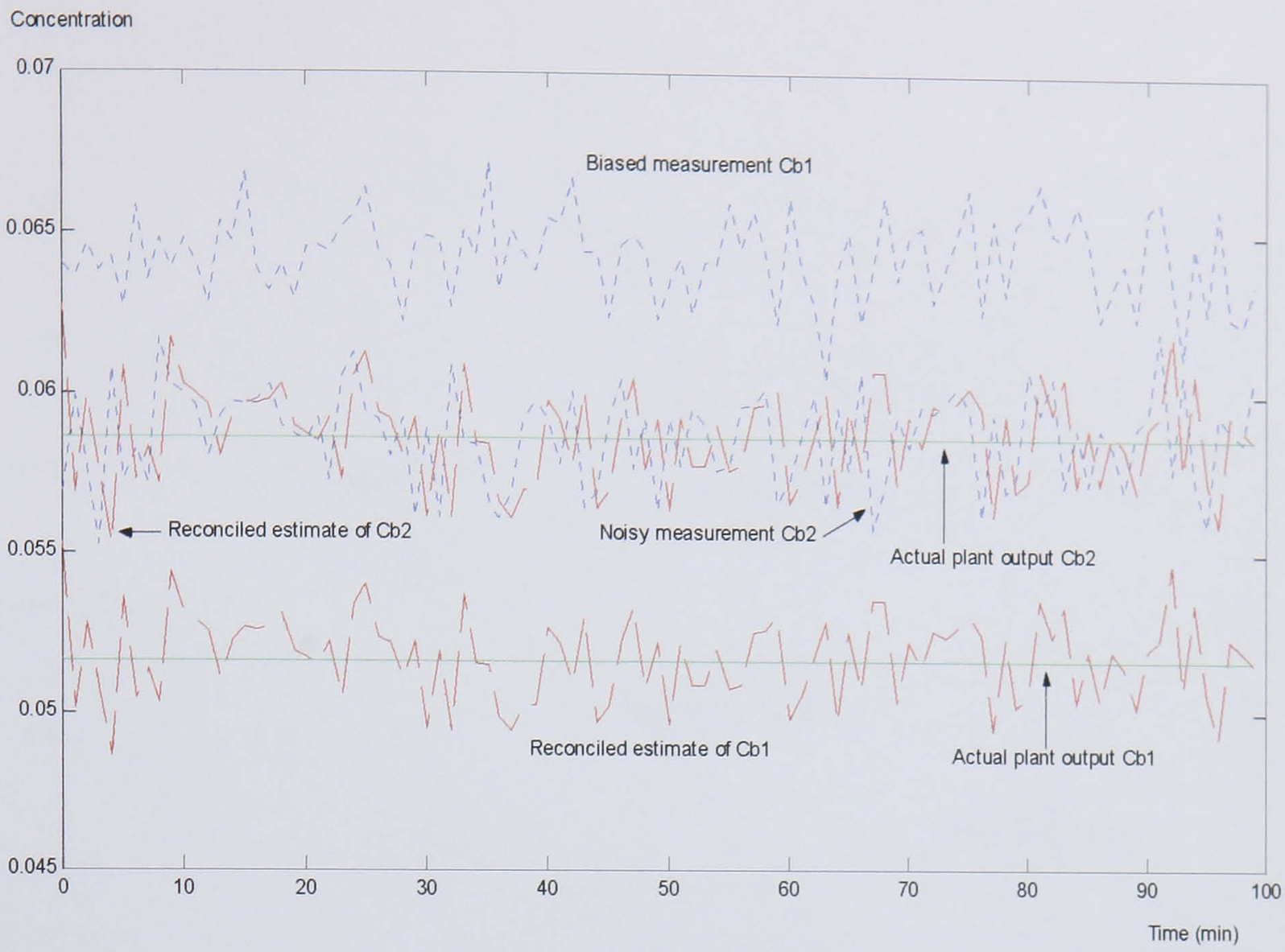


Figure (4-6): Reconciliation of measurement data while also estimating a physical parameter C_{a0} when C_{b1} is biased.

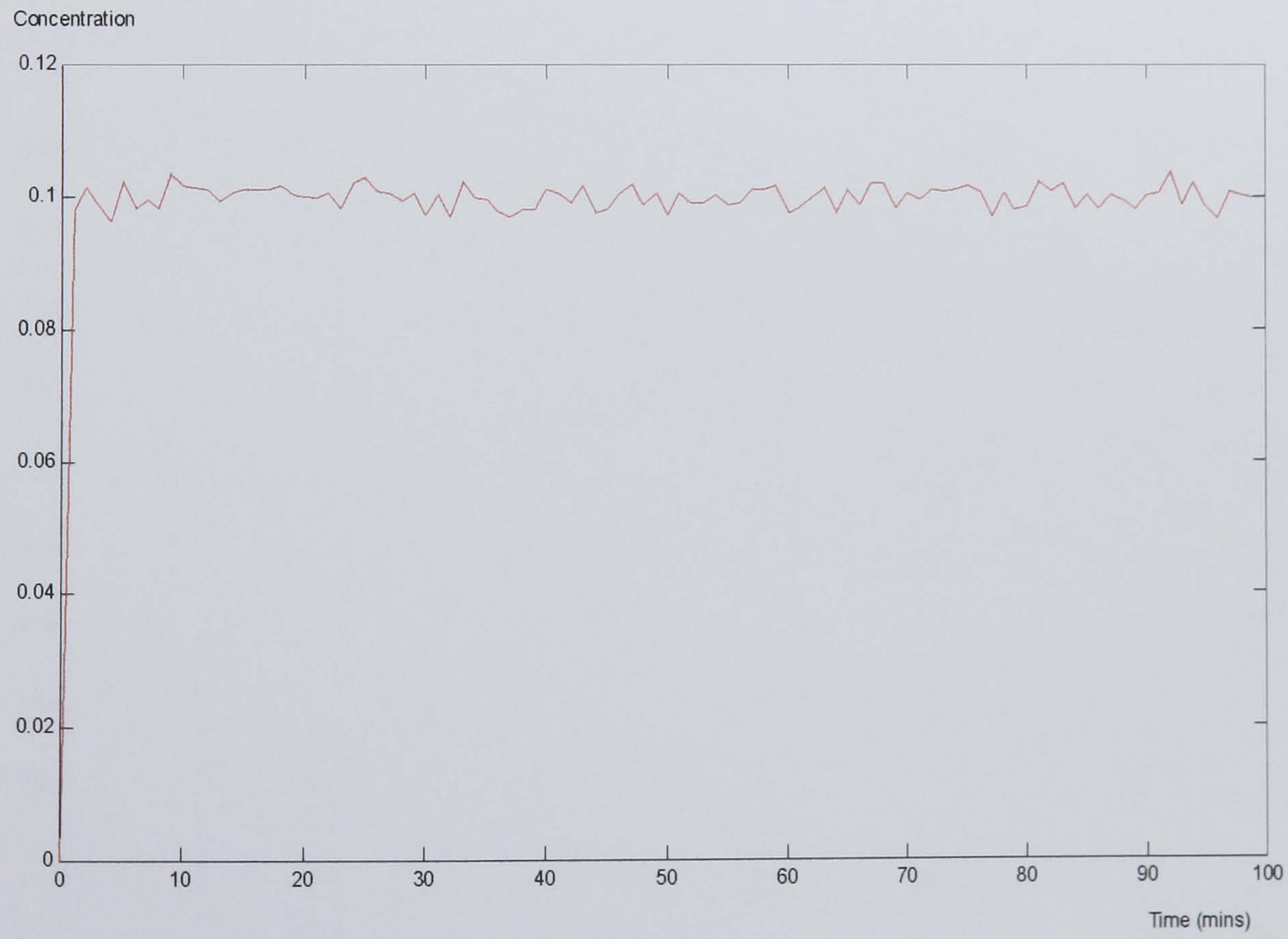


Figure (4-7): Estimate of C_{a0} when C_{b1} is biased.

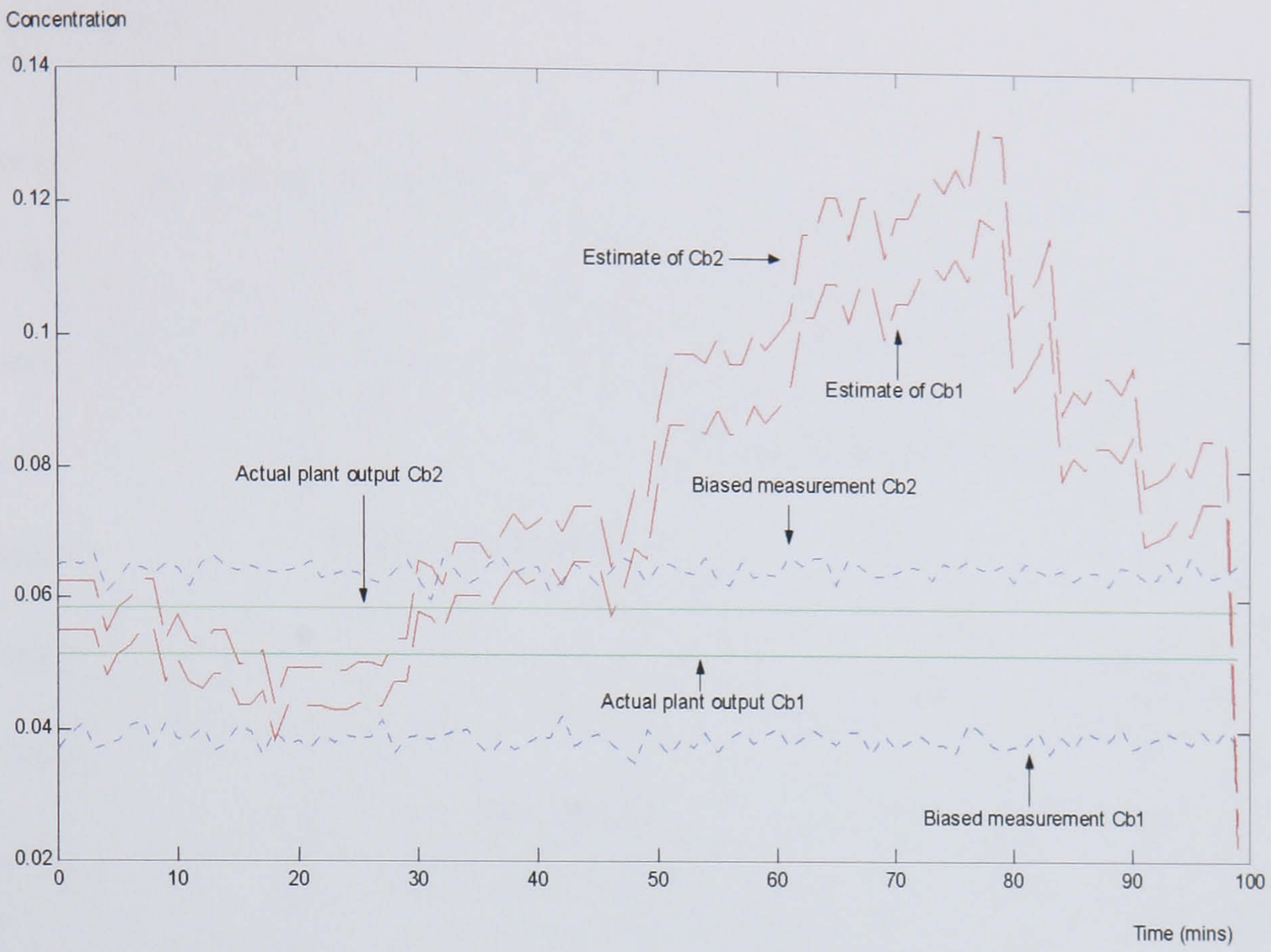


Figure (4-8): Reconciliation of measurement data while also estimating a physical parameter C_{a0} when C_{b1} and C_{b2} are biased.



Figure (4-9): Estimate of C_{a0} when there is bias on both measurements.

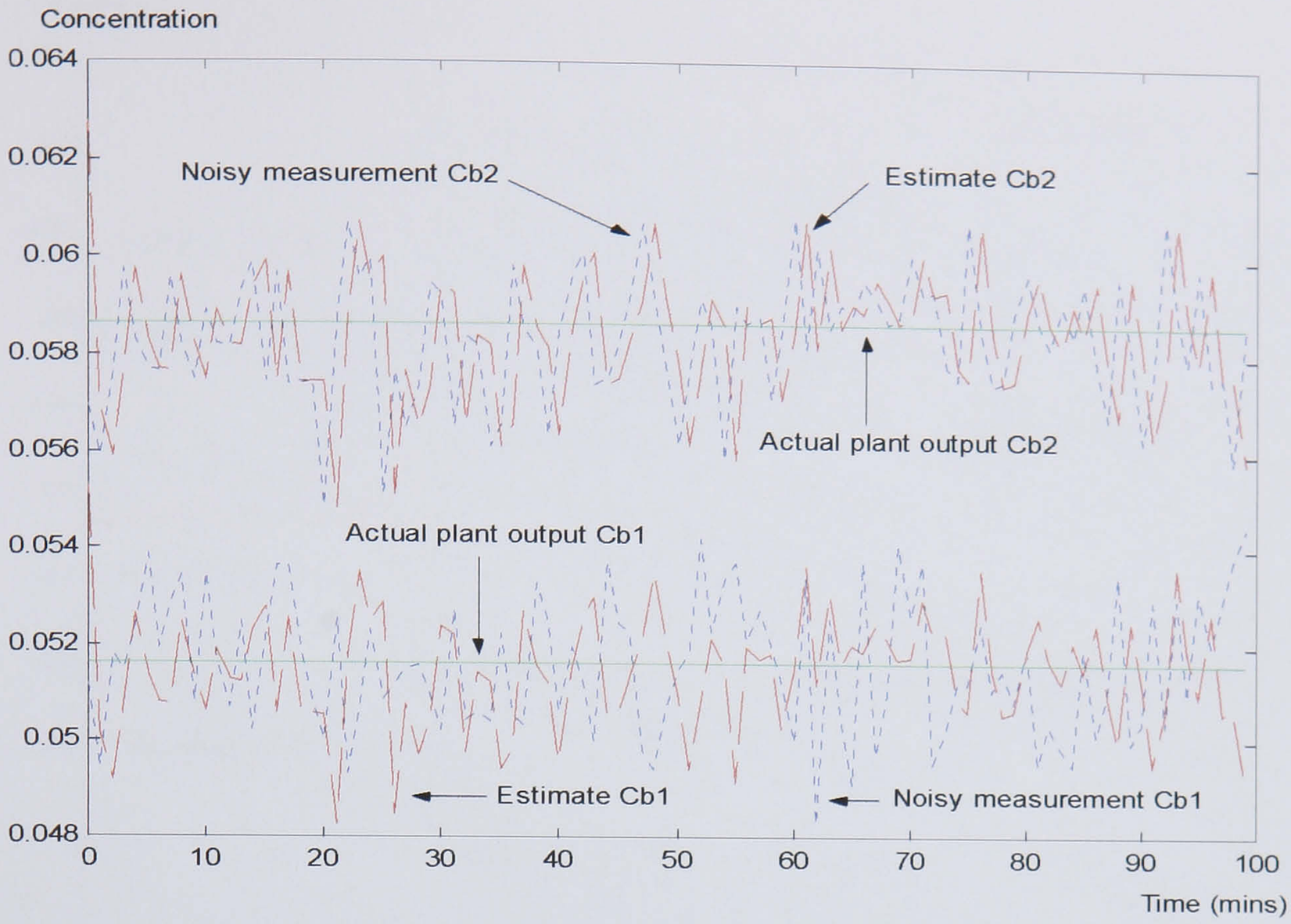


Figure (4-10): Reconciliation of measurement data while also estimating a physical parameter C_{a0} when there is no bias on the measurements.

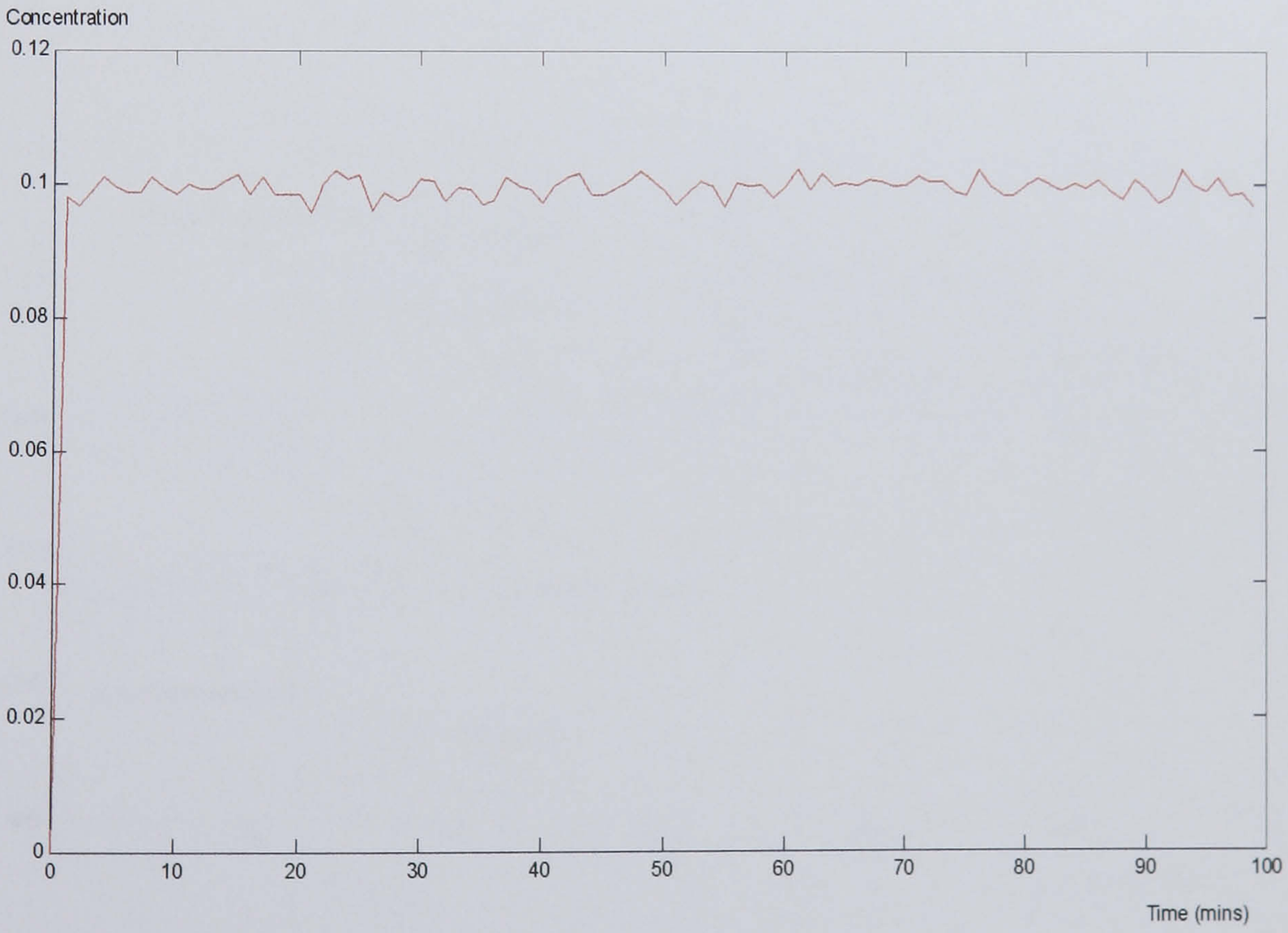


Figure (4-11): Estimate of C_{a0} when there is no bias on either of the measurements.

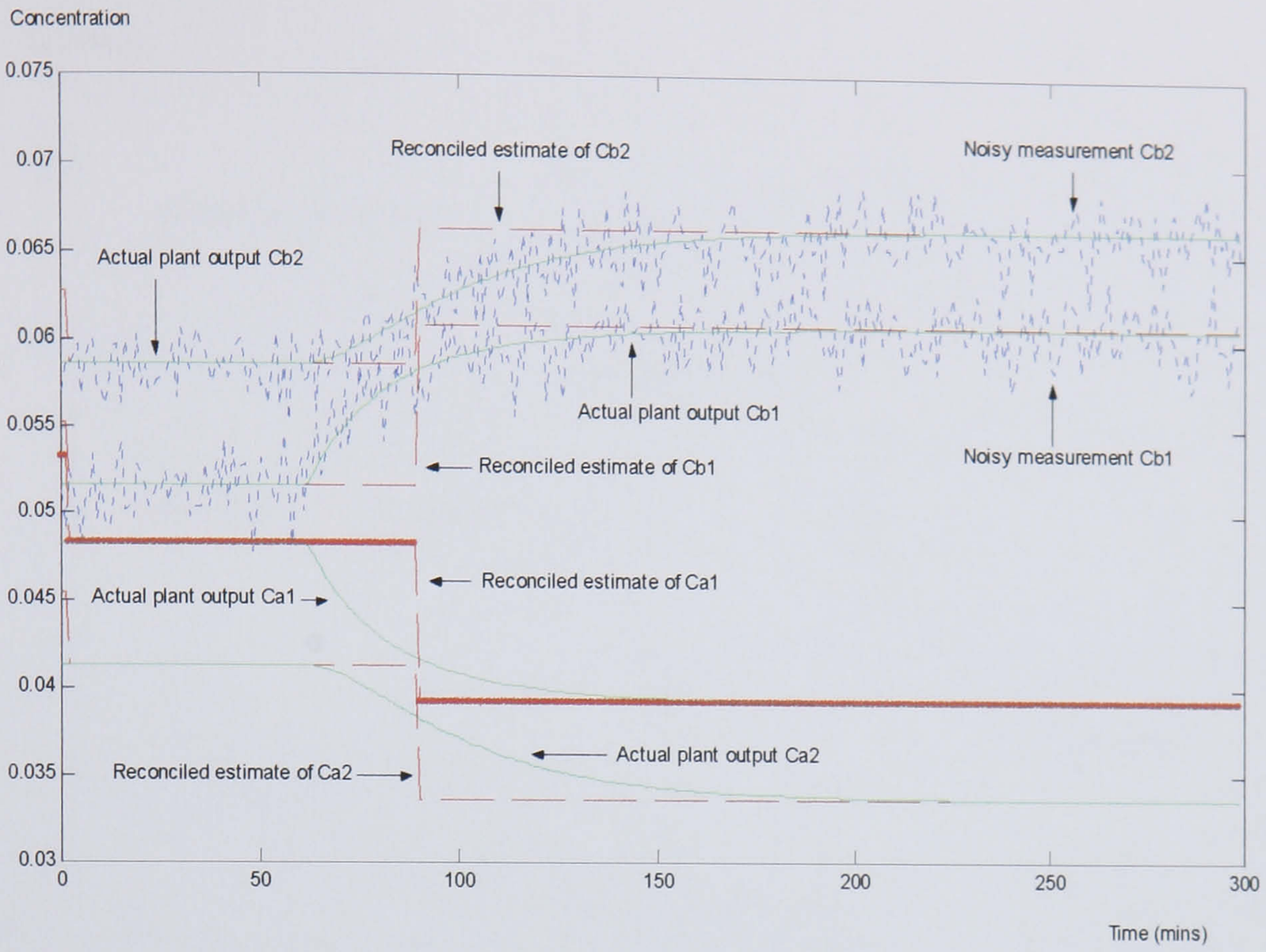


Figure (4-12): Behaviour in the presence of a transient - none of the measurements are biased.

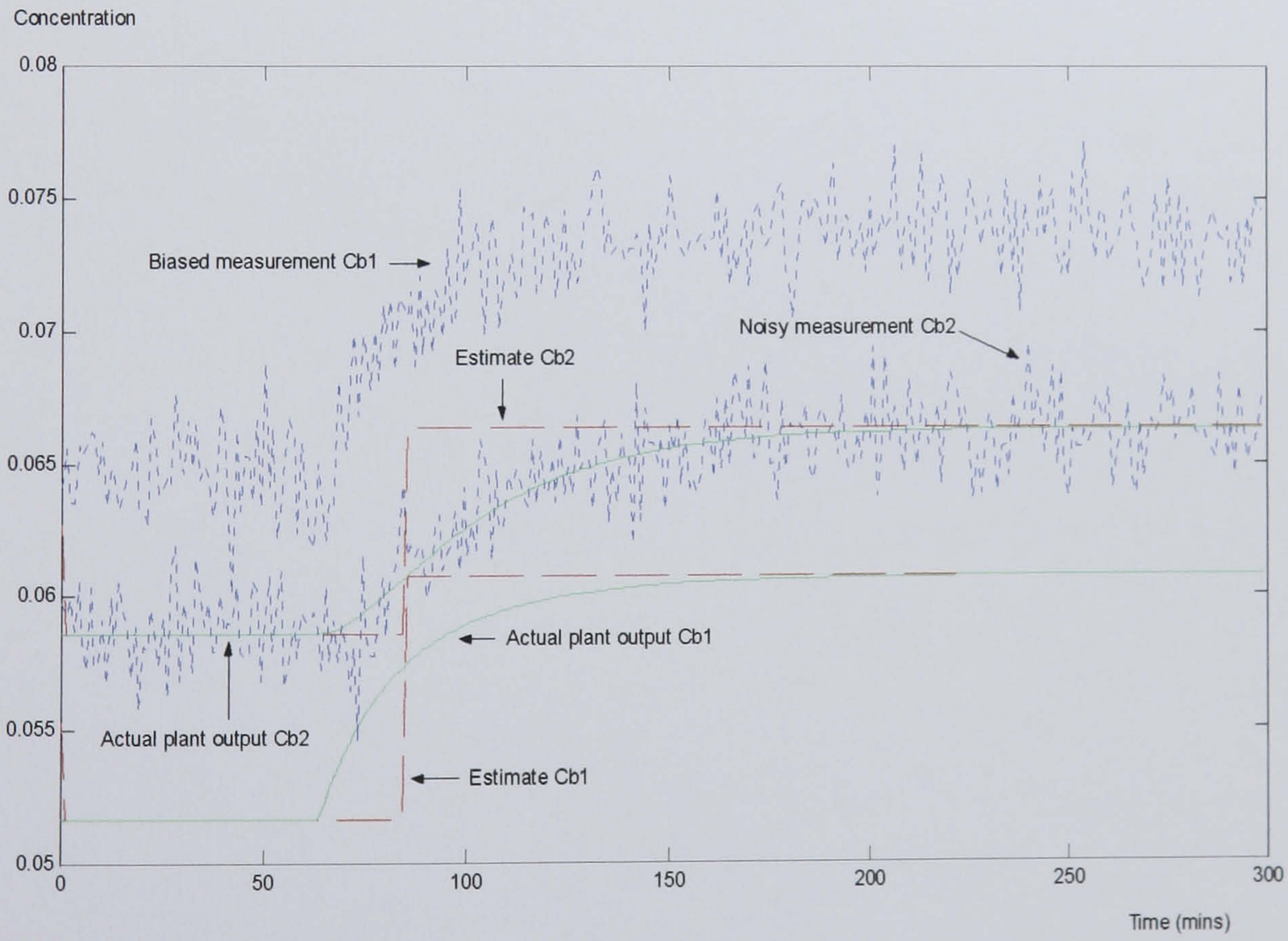


Figure (4-13): Behaviour in the presence of a transient - C_{b1} is biased.

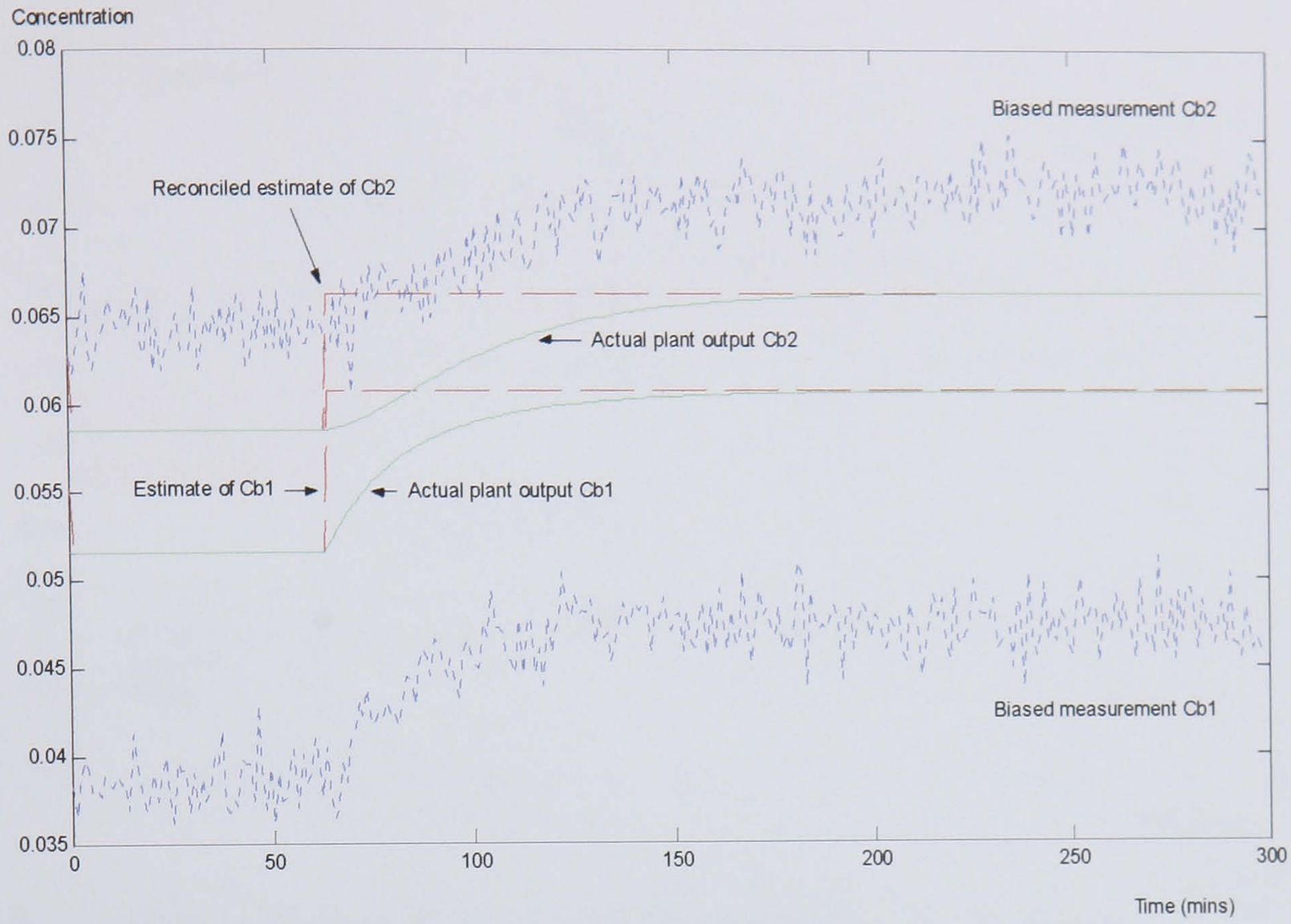


Figure (4-14): Behaviour in the presence of a transient - both measurements are biased.

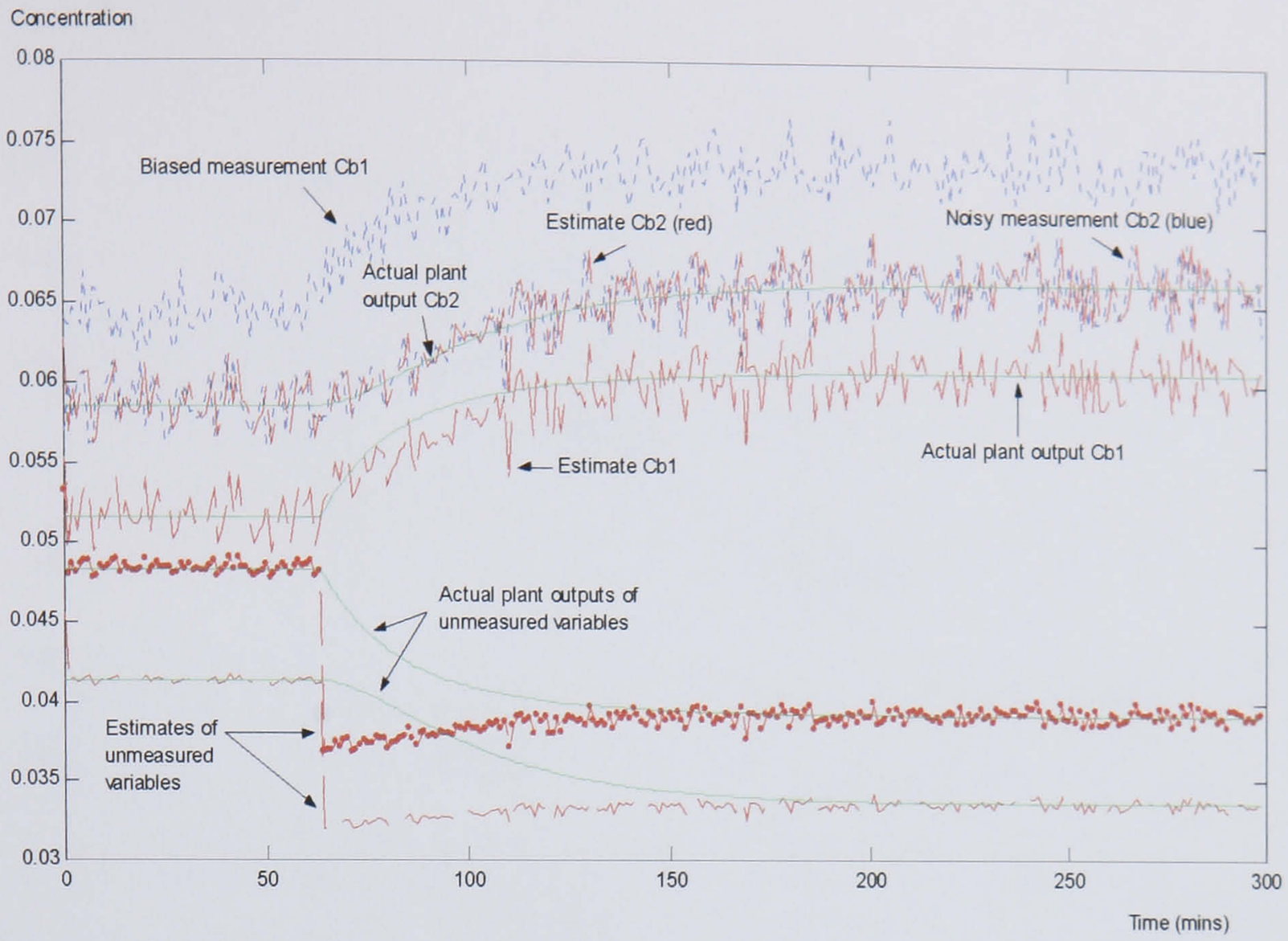


Figure (4-15): Behaviour in the presence of a transient when C_{a0} is estimated and C_{b1} is biased.

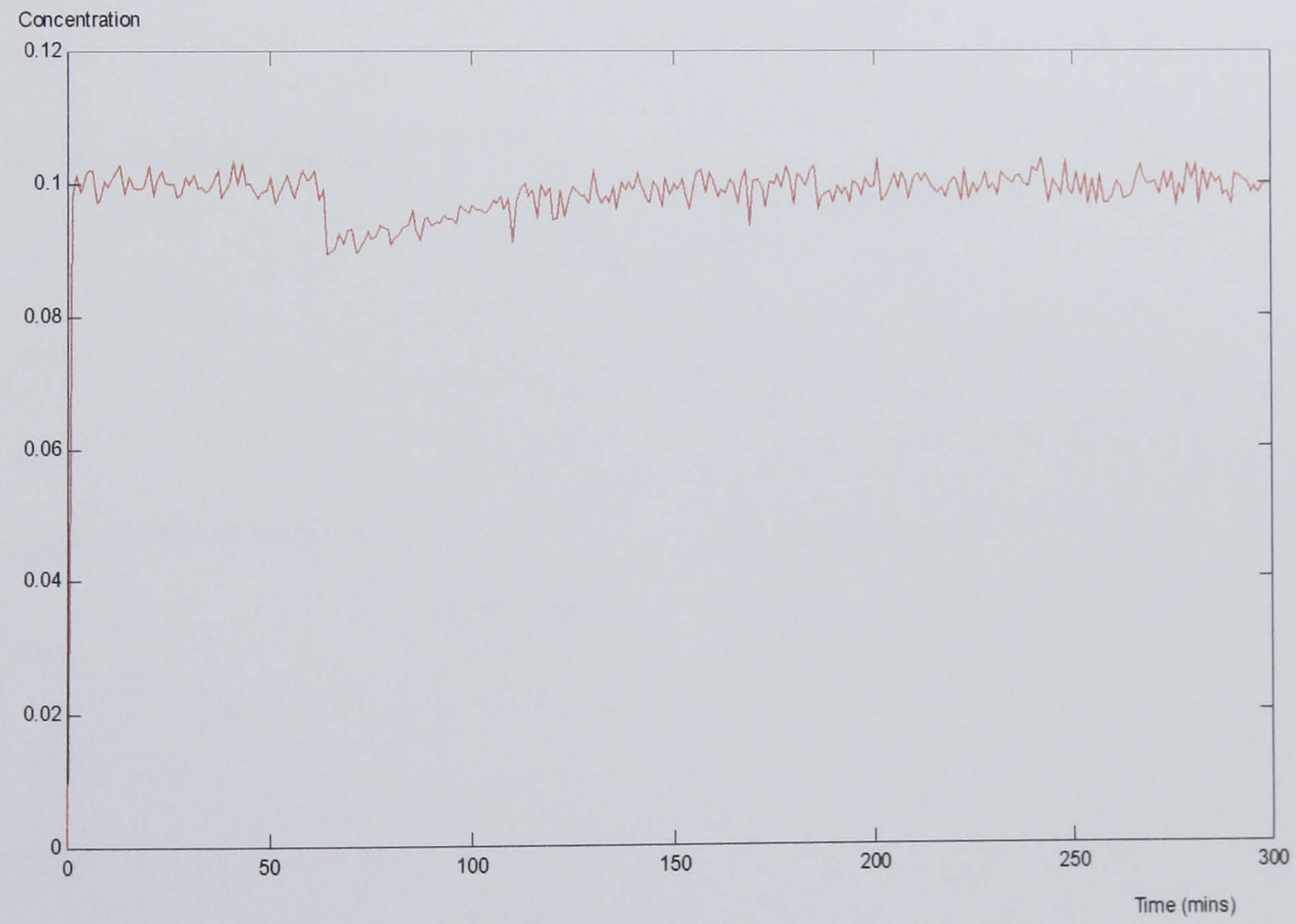


Figure (4-16): Estimate of C_{a0} in the presence of a transient and bias on C_{b1}.

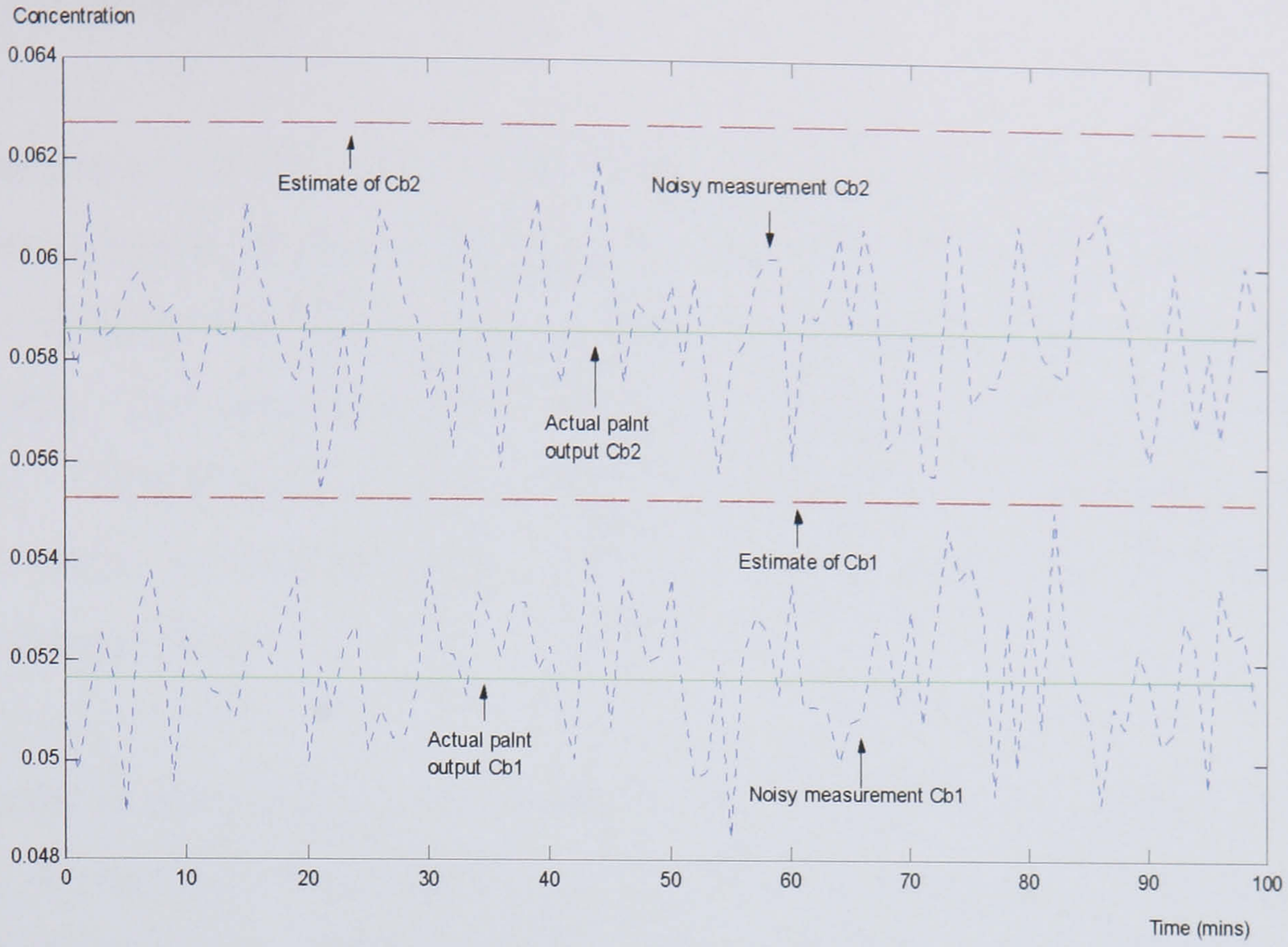


Figure (4-17): Estimates of C_{b1} and C_{b2} when parametric differences exist between the static and dynamic models (C_{a0} in static model = 0.07).

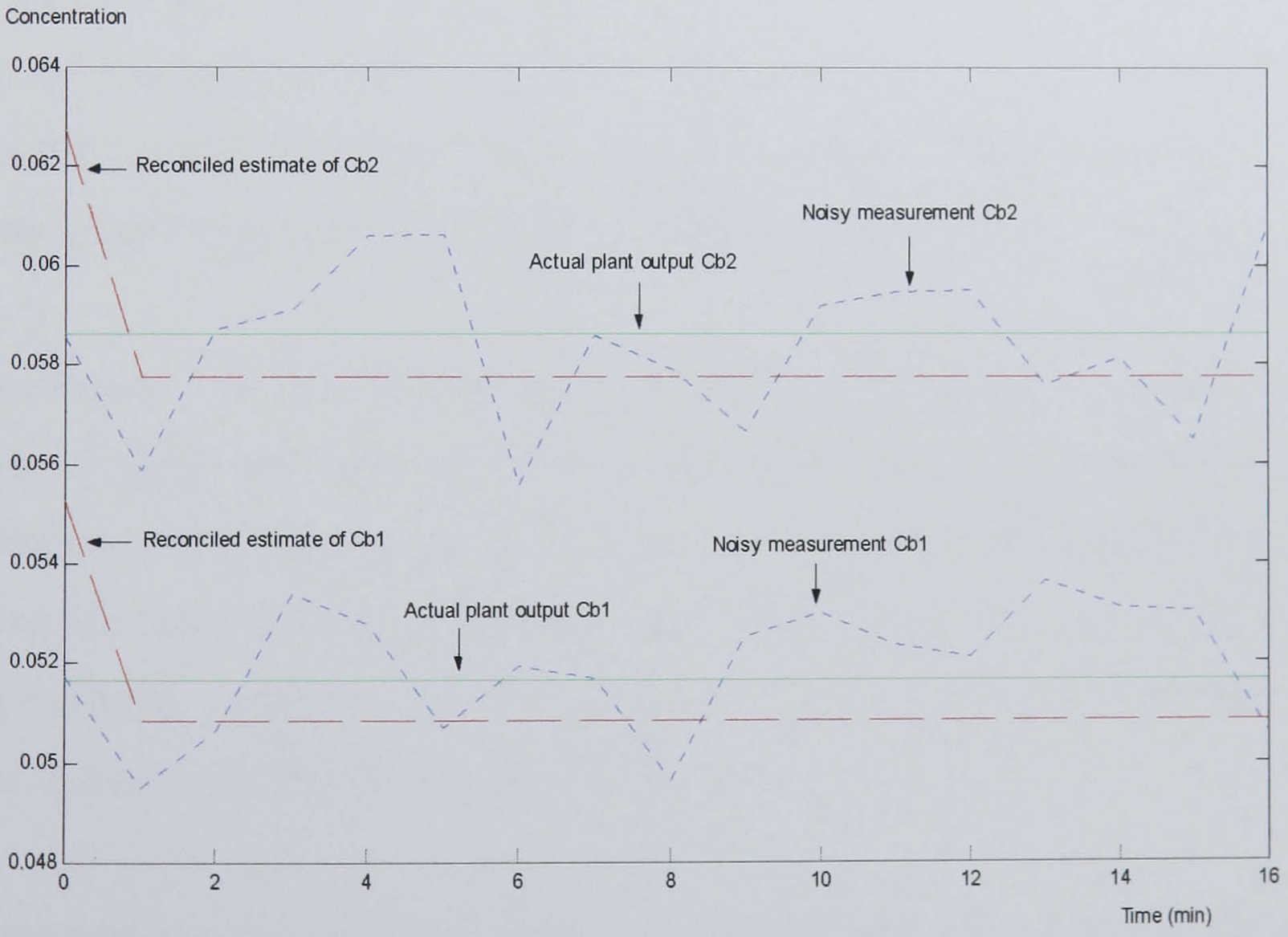


Figure (4-18): Estimates of C_{b1} and C_{b2} when parametric differences exist between the static and dynamic models (C_{a0} in static model = 0.099).

4.4 CONCLUSIONS

The simulations carried out show that under normal operating conditions, the algorithm correctly reconciles process data in the majority of cases. Even in the presence of systematic bias on either or both of the measurements, the results are encouraging. The correct estimates are achieved within a maximum of two SDR iterations. Considering that the bias imposed ranged from –50% to +50% of the nominal values, it is safe to say that in terms of systematic bias, the SDR module was rigorously tested.

In the case of estimating physical parameters, the SDR module also (in the majority of cases) produced good results with correct estimation of the required parameter within one SDR iteration. This was successfully done even with systematic bias added to the measurements. The module was further tested for robustness in terms of the steady-state condition. When a transient was introduced, the algorithm coped well with the change and produced correct estimates of the measured and unmeasured variables. This was also the case when systematic bias was added. Even when estimating a physical parameter in the presence of bias and a transient, the SDR module produced good results. These tests are particularly important because even a so-called steady-state process is constantly undergoing variations about a nominal steady-state.

It is important to note that although the robustness shown by the algorithm during transients is good and necessary, it is not sufficient to rely on static data reconciliation in the case of dynamic data. The results obtained here may be specific to this case. In order to properly reconcile dynamic data a dynamic model of the process is needed to ensure that there are no significant differences in dynamics between the model and the real process.

In the presence of parametric differences between the process model and the static model used for estimation, the algorithm either did not converge or produced

erroneous results if the differences were anything above negligible. The algorithm relies heavily on the accuracy of the static model used for estimation.

In the cases where the algorithm did not perform well it may be said that the problem itself is ill conditioned or that too much is being demanded from the algorithm given the limited information available. A possible test to see whether the results would improve is to assume 3 or 4 measured variables instead of just two.

Throughout this chapter it has been assumed that if systematic bias was present, it was known in advance which measurements were affected. In chapter 7, an algorithm for bias and gross error identification is implemented. This algorithm automatically provides information on the measurements that are biased or affected by gross errors. The work presented in this chapter is extended further to cover reconciliation of dynamic data and to tackle the problem of gross errors.

4.5 SUMMARY

In this chapter an introduction to the area of data reconciliation has been presented. A review of previous work in the field of static data reconciliation has also been presented. A technique for the reconciliation of steady-state data has been implemented in software. This has been tested successfully using a dynamic model of two continuous stirred tank reactors. The results show that the technique is capable of reconciling process data as well as estimating bias, physical parameters and unmeasured variables.

A certain amount of robustness of the technique has been shown by means of simulations carried out on dynamic data. In so doing, this has also exposed some of the limitations of using static models on dynamic data. As has been mentioned earlier, errors may sometimes occur during the transients due to the change in process dynamics and the differences between the static model and the real

process. In chapter 7, dynamic data reconciliation is introduced where a dynamic model is employed for the reconciliation process. This, along with bias and gross error detection and identification methods are then applied to a model predictive control scheme in chapter 8.

CHAPTER 5

AUTOMATIC IDENTIFICATION OF STEADY-STATE.

This chapter presents a workable solution to the problem of Steady-State Detection (SSD). A brief review of the available methods is presented. One method (Cao and Rhinehart, 1995) is chosen for its simplicity and low computational effort. The algorithm is implemented as an OTISS module using C++ code and is intended to fit into a larger collection of modules which together make up an integrated dynamic data reconciliation framework. The method is applied to a dynamic model of two continuous stirred tank reactors connected in series.

5.1 STEADY-STATE DETECTION BACKGROUND

Some control and estimation techniques that use steady-state models assume that process measurements correspond to steady-state conditions. Steady-state models are widely used in model identification, optimisation and data reconciliation. Whether or not the process is at steady-state determines what treatment may be administered to it. For the purpose of data reconciliation, it is important to know when the system is at steady-state in order to be able to apply the static data reconciliation techniques outlined in chapter 4. Thus the identification of steady-state is an important task and is applicable to the compression of process data (Mo et al., 1998) and also fault diagnosis.

In any data reconciliation framework, it is useful to have an algorithm which would automatically detect when the process is at steady-state. Here a separate module is developed for use with other modules within the data reconciliation framework. This new module would serve as a point of reference for the entire data reconciliation set up and would help to decide which tools to employ in data reconciliation depending on whether or not the process is at steady-state.

5.2 REVIEW OF EXISTING METHODS

Although there are a few existing methods for steady-state identification, work in this field has been limited. A survey of methods for detecting changes in signals which are applicable to data reconciliation was published by (Basseville, 1988). Harris and Ross (1991) outlined some procedures for correlated observations. A brief review of some of the methods available for detecting changes in steady-state was presented by (Crowe, 1996).

Narasimhan et al. (1986) presented a composite statistical test which is appropriate for quasi-steady state processes that remain essentially at steady-state for long intervals of time and change quickly from one steady-state to another. It is not suitable for detection of slow drifts in the variables. The technique examines successive time periods and consists of two tests; the first test is used to establish whether or not the unknown covariance matrices are equal. The second test is made up of two parts to establish whether the means from the two successive periods are equal. The first part uses the Hotelling T^2 test, (Anderson, 1958) for the case when the covariance matrices are equal but unknown. The second part uses a technique similar to the Hotelling T^2 test (Yao, 1965) for the case when the covariance matrices are unequal and unknown. The power of the test increases with the number of variables tested together provided those variables change state together. A major drawback of this method is that it is quite involved and requires extensive computational effort.

Another method presented by Narasimhan et al. (1987), uses the Mathematical Theory of Evidence (Shafer, 1976), and is an attractive alternative to their earlier method. However, this method is only applicable if the variables to be tested are independent.

More recently, Betha and Rhinehart (1991) presented an off-line technique which performs a linear regression over a data window and then performs a t-test on the regression slope. On-line versions require considerable data storage and computational effort.

An alternative method by Crow et al. (1955) uses an F-test type statistic. A ratio of two variances calculated by two different methods using the same set of data forms the basis of the test. The first variance measure is calculated as the mean-square-deviation from the average. The average being an average of the most recent data window. The second variance measure is calculated from the mean squared differences of successive data. The idea being that if the time series is stationary i.e. the process is at steady-state, this ratio will be unity. In reality, however, the ratio will not be exactly unity due to random noise but will have a value close to unity. Alternatively, if the process is not at steady-state, the ratio of the variances will tend to be large. This method has a few disadvantages, namely:

- The user must choose the size of the data window, too large a window implies considerable computational effort and delayed recognition of changes.
- A large amount of data must be stored at each sampling interval.
- Autocorrelation in the measured signal will affect the statistic. Even if the process is at steady-state, short-lived transient fluctuations which last a few sampling intervals will always trigger a non-steady-state condition. Considerable autocorrelation will always produce misleading results.

Perhaps the most interesting of the methods reviewed by the author is that presented by Cao and Rhinehart (1995). This method is essentially the same as and inspired by that presented by Crow et al. (1955). However, there are some

notable differences. In order to avoid having to calculate variances and means over a whole data window at each sampling interval and thus reducing the computational burden considerably, the authors introduce some essential simplifications. The method is an F-test type of statistic which calculates the ratio R_i of two variances. The two variance values are calculated using different methods but employing the same set of data. This method is simple to implement and is computationally inexpensive. There are however a couple of undesirable points:

- At the critical value where the system changes state, there is usually a short period where the identifier toggles between steady-state and non-steady-state conditions. This is undesirable especially if some course of action is dependent on this information.
- Critical values for R_i differ from one measurement variable to another. Cao and Rhinehart (1997) provide tables of critical values. Some intuition is required to choose critical R_i values for all the variables to be tested.

5.3 STEADY-STATE IDENTIFICATION METHOD – CAO AND RHINEHART

The steady-state identification method developed by Cao and Rhinehart is an F-test type of statistic originally outlined by Crow et al. (1955). The test is essentially a ratio R_i of two variances calculated by different methods. The first variance measure is the mean square deviation from the average. While the second variance measure is the mean of squared differences of successive data.

The primitive way of estimating variance would be:

$$\hat{\sigma}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \overline{X}_N)^2 \qquad (5.1)$$

where X_i is the ith process variable measurement in the data window of length N. And $\overline{X}_N$ is the sample mean calculated over the N samples. The simplification (or modification to Crow et al.'s method) begins with a conventional exponentially weighted moving average, or conventional first order filter of a process variable X_i. This requires little storage and is computationally fast.

$$X_{f_i} = \lambda_1 X_i + (1-\lambda_1) X_{f_{i-1}} \tag{5.2}$$

where $0 < \lambda_1 \leq 1$ is a filter factor.

If the previous filtered value $X_{f_{i-1}}$ is used to replace the sample mean, $\overline{X}_N$, a mean square deviation can be defined as:

$$v^2 = E\left(\left(X_i - X_{f_{i-1}}\right)^2\right) \tag{5.3}$$

and can be estimated by:

$$\hat{v}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - X_{f_{i-1}})^2 \tag{5.4}$$

Assuming that $\{ X_i \}$ is uncorrelated, using the previous value of X_f, $X_{f_{i-1}}$, prevents autocorrelation between X_i and $X_{f_{i-1}}$ and allows one to easily estimate σ^2 from v^2.

Define:

$$d_i = X_i - X_{f_{i-1}} \tag{5.5}$$

If the process is at a steady-state condition and there is no correlation in the sequential measurement, then X_i and $X_{f_{i-1}}$ are independent, then the variance on d is related to the variance on X and X_f:

$$\sigma_d{}^2 = \sigma_X{}^2 + \sigma_{X_f}{}^2 \tag{5.6}$$

Further, for the exponentially weighted moving average, when $\{X_i\}$ are independent and stationary, the variance on X_f from equation (5.2) becomes:

$$\sigma_{X_f}{}^2 = \frac{\lambda_1}{2-\lambda_1}\sigma_X{}^2 \tag{5.7}$$

Equations (5.6) and (5.7) yield:

$$\sigma_X{}^2 = \frac{2-\lambda_1}{2}\sigma_d{}^2 = \frac{2-\lambda_1}{2}v^2 \tag{5.8}$$

from which the noise variance can be estimated if v^2 is known.

$$\hat{\sigma}_X{}^2 = \frac{2-\lambda_1}{2}\hat{v}^2 \tag{5.9}$$

However, Equation (5.4) is computationally expensive; so, use a filtered value instead of a traditional average:

$$v_{f,i}{}^2 = \lambda_2(X_i - X_{f_{i-1}})^2 + (1-\lambda_2)v_{f,i-1}{}^2 \tag{5.10}$$

where $0 < \lambda_2 \leq 1$ is a filter factor.

If the process is stationary:

$$E(v_{f,i}{}^2) = E((X_i - X_{f_{i-1}})^2) = v^2 \tag{5.11}$$

So, equation (5.10) is an unbiased estimate of v^2, and the variance of $v_{f,i}^{\ 2}$ is:

$$Var(v_{f,i}^{\ 2}) = \frac{\lambda_2}{2-\lambda_2} Var((X_i - X_{f_{i-1}})^2) \tag{5.12}$$

which means that equation (5.10) provides a computationally efficient, unbiased estimate of $(X_i - X_{f_{i-1}})^2$.

Then the estimate of the noise variance from this first approach will be:

$$S_{1,i}^{\ 2} = \frac{2-\lambda_1}{2} v_{f,i}^{\ 2} \tag{5.13}$$

Since equation (5.10) requires $X_{f_{i-1}}$, one would compute equation (5.10) before equation (5.2) to eliminate the need to store the previous average.

Using this method, $S_{1,i}^{\ 2}$ will be increased from its steady-state value by a recent shift in the mean.

The second method to estimate the variance will use the mean squared differences of successive data. Define:

$$\delta^2 = E((X_i - X_{i-1})^2) \tag{5.14}$$

and δ^2 could be estimated by:

$$E(S_{2,i}^{\ 2}) = \frac{1}{2} E(X_i - X_{i-1}) \tag{5.15}$$

However, equation (5.15) is computationally expensive; so, use a filtered approach:

$$\delta_{f,i}^{\ 2} = \lambda_3 (X_i - X_{i-1})^2 + (1-\lambda_3)\delta_{f,i-1}^{\ 2} \tag{5.16}$$

where $0 < \lambda_3 \leq 1$ is a filter factor.

Again, equation (5.16) provides an unbiased estimate of δ^2.

It is easily shown that the second estimate of the noise variance would be:

$$S_{2,i}{}^2 = \frac{\delta_{f,i}{}^2}{2} \tag{5.17}$$

Taking the ratio of the two estimates of variance as determined by equation (5.13) to equation (5.17):

$$R_i = \frac{S_{1,i}{}^2}{S_{2,i}{}^2} = \frac{(2-\lambda_1)v_{f,i}{}^2}{\delta_{f,i}{}^2} \tag{5.18}$$

5.3.1 Algorithm for the identification of steady-state:

Initialisation:

1- λ values and an initial sequence of measurements given,

2- Using the first 10 measurements, estimate the mean and the variance. Cao and Rhinehart suggest a sequence of 10 measurements for the initialisation,

3- Set X_f and X_{i-1} equal to estimated mean,

4- Set $v_{f,i}{}^2$ equal to estimated variance,

5- Set $\delta_{f,i}{}^2$ equal to twice the estimated variance.

At each sampling interval:

6- For each new measurement X_i, use equations (5.10), (5.2), (5.16) and (5.18) to calculate the new ratio R_i.

7- Update X_f, X_{i-1}, v^2 and δ^2.

5.3.2 Selection of λ values

λ_1, λ_2 and λ_3 are all filter factors. Choosing small filter factors can significantly reduce the noise influences on the estimates of process variances. This makes the two states: that of steady-state and that of non-steady-state quite distinct (probability density function, pdf(R_i) of steady-state and pdf(R_i) of non-steady-state), reducing both Type I and Type II errors. Type I error is the error associated with wrongly rejecting the null hypothesis (process at steady-state) when it is true. While Type II error is the error associated with wrongly accepting the null hypothesis when it is false. The above is all very well, however, small filter factors can make the R_i statistic lag behind the present process state.

5.4 FILTERING OF THE STEADY-STATE IDENTIFIER OUTPUT

One disadvantage of the method developed by Cao and Rhinehart is that at the transition between steady-state and non-steady-state condition, the identifier output sometimes toggles between steady-state and non-steady-state. This occurs when the variability measure for the measured variable straddles the critical value. This effect is undesirable because some process decisions may depend on this information.

A new way to overcome this problem and which has been applied here is to apply some sort of filtering at the transition point such that the variability measure is held constant unless the current and two previous values are either all above or below the critical value. This will have the effect of filtering the calculations making it unlikely for the steady-state identifier to toggle between steady and non-steady-state conditions. However, applying this filtering technique means having to store an extra value i.e. R_{i-2} and also extra computational effort. Further, due to this filtering, the response of the identifier will lag behind the process state. This lag can be reduced by choosing a smaller sampling interval but at the expense of increased computational effort.

5.5 IMPLEMENTATION OF THE STEADY-STATE IDENTIFICATION MODULE

For the purpose of data reconciliation, the author chose the method by Cao and Rhinehart, outlined above, for implementation as an OTISS module using C++ code. This module was implemented and tested using data from a dynamic model of two continuous stirred tank reactors (Abu-el-zeet et al., 1999a, 2000). There are two measured variables: the concentration of species B in tank 1, C_{b1} and the concentration of species B in tank 2, C_{b2}. The steady-state identification technique was first applied on C_{b1} then on C_{b2} and finally on both these measured variables simultaneously. Noise was added to the measurements using a noise source module built into the OTISS process simulator. Critical values were chosen for both variables by inspection using a trial run. The critical values used were:

$R_{CRITIC1} = 1.45,$

$R_{CRITIC2} = 1.25.$

The λ values used were: $\lambda_1 = 0.05$, $\lambda_2 = 0.005$ and $\lambda_3 = 0.005$ for both measured variables. These were obtained by trial and error.

The filtering technique outlined above was used in this case because of the importance attached to the accuracy of the steady-state identifier output. Although this results in the response of the identifier lagging the process state slightly, it was judged acceptable in this case taking the process dynamics into consideration.

5.5.1 Simulation Results

Simulations were carried out on the dynamic model of the two continuous stirred tank reactors used in chapter 4, see Figure (4-2). In all the simulation cases that

follow, the system was started from the steady-state condition given by the set-points $T_1(0) = 307$ [K] and $T_2(0) = 302$ [K], which yield steady-state values $C_{b1}(0) = 0.05165$ [Kmol/ m^3], $C_{b2}(0) = 0.05864$ [Kmol/ m^3].

To simulate a realistic series of changes of steady-states, four different step changes were applied to the process. After a period of steady-state the following changes were made:

- Change temperature T_1 from 307 [K] to 310 [K].
- Change T_1 from 310 [K] to 307 [K].
- Change the inlet concentration C_{a0} from 0.1 to 0.12.
- Change C_{a0} from 0.12 to 0.1 and T_2 from 302 [K] to 295 [K].

Between each change the process was allowed to settle to a new steady-state condition.

A simulation run was first carried out without using the filtering technique, (Figure 5-1). The measured variable tested here was C_{b1}. The toggling effect of the steady-state identifier can easily be seen especially at the transition from non-steady-state to steady-state condition. Figure (5-2) shows the variability measure or the ratio of the two variances (R_i) of the first measurement (C_{b1}). A not-at-steady-state condition is triggered when this value rises above a critical value, set at 1.45 in this case.

Next, simulation runs were carried out using the filtering technique, first with C_{b1} (Figure 5-3) as the variable tested then C_{b2} (Figure 5-4) and finally with both being tested simultaneously (Figure 5-5).

It can be seen that with all these tests, the algorithm is working very well in identifying when the process is at steady-state. There is a slight delay until the identifier realises the effect of a change but this delay is acceptable taking the process dynamics into consideration. There is little difference between Figures (5-3), (5-4) and (5-5). This is because the two measured variables are closely

related and change state together. Differences would be present if the measured variables are independent and change state independently. Comparing Figure (5-1) with the latter plots, it can be seen that the toggling effect at the transition points has been significantly reduced by use of filtering. This has not been eliminated completely (see Figure 5-4), though this would be possible but at the expense of even greater delay. As it is, the delay appears to have increased slightly as expected but again this is acceptable. It can be seen from Figure (5-5) that applying the test on two or more dependent variables at the same time produces better and naturally filtered results.

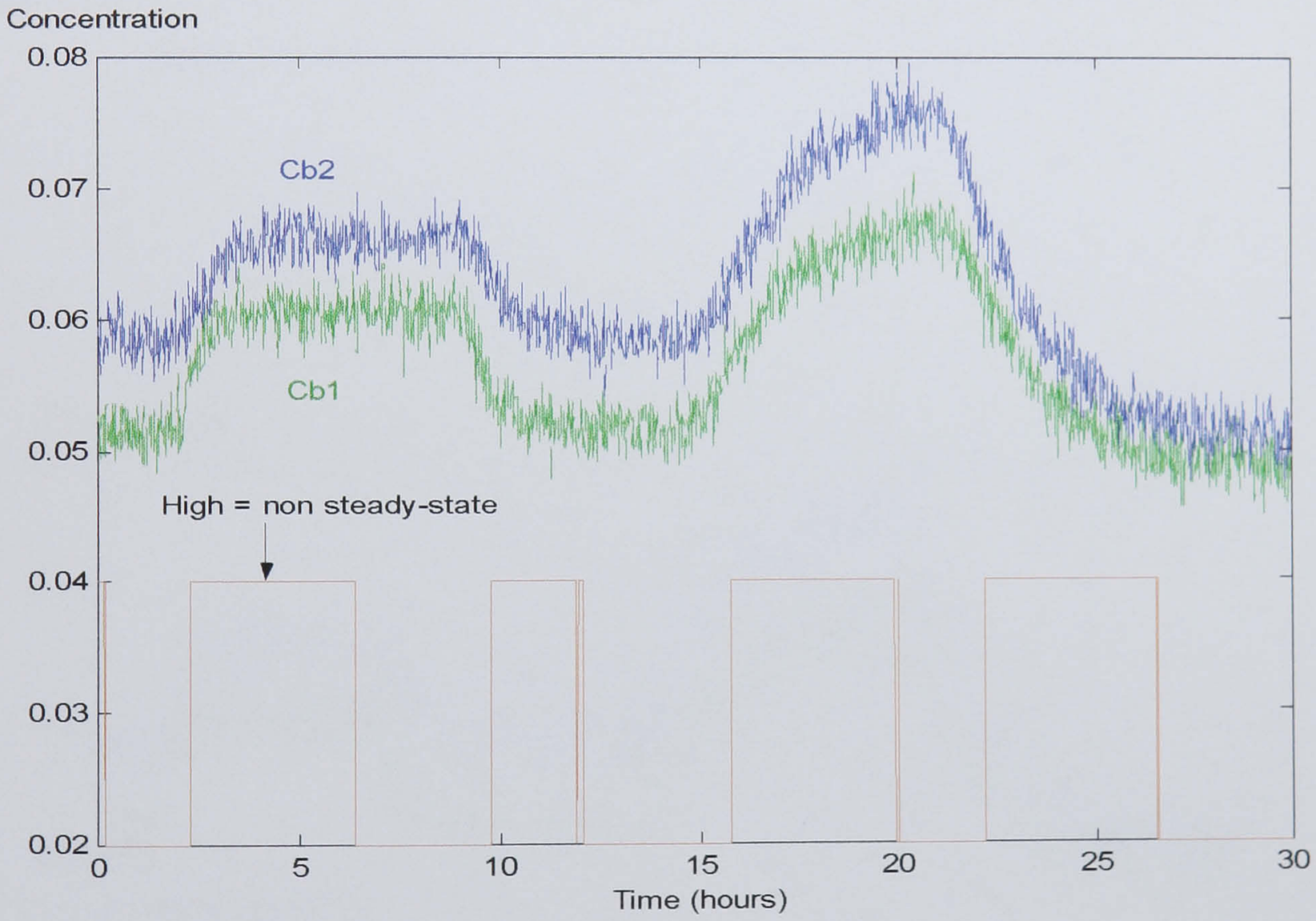


Figure (5-1): Measured process variables and output of SSD module. Tests based on measurement C_{b1} only, filtering technique not applied.

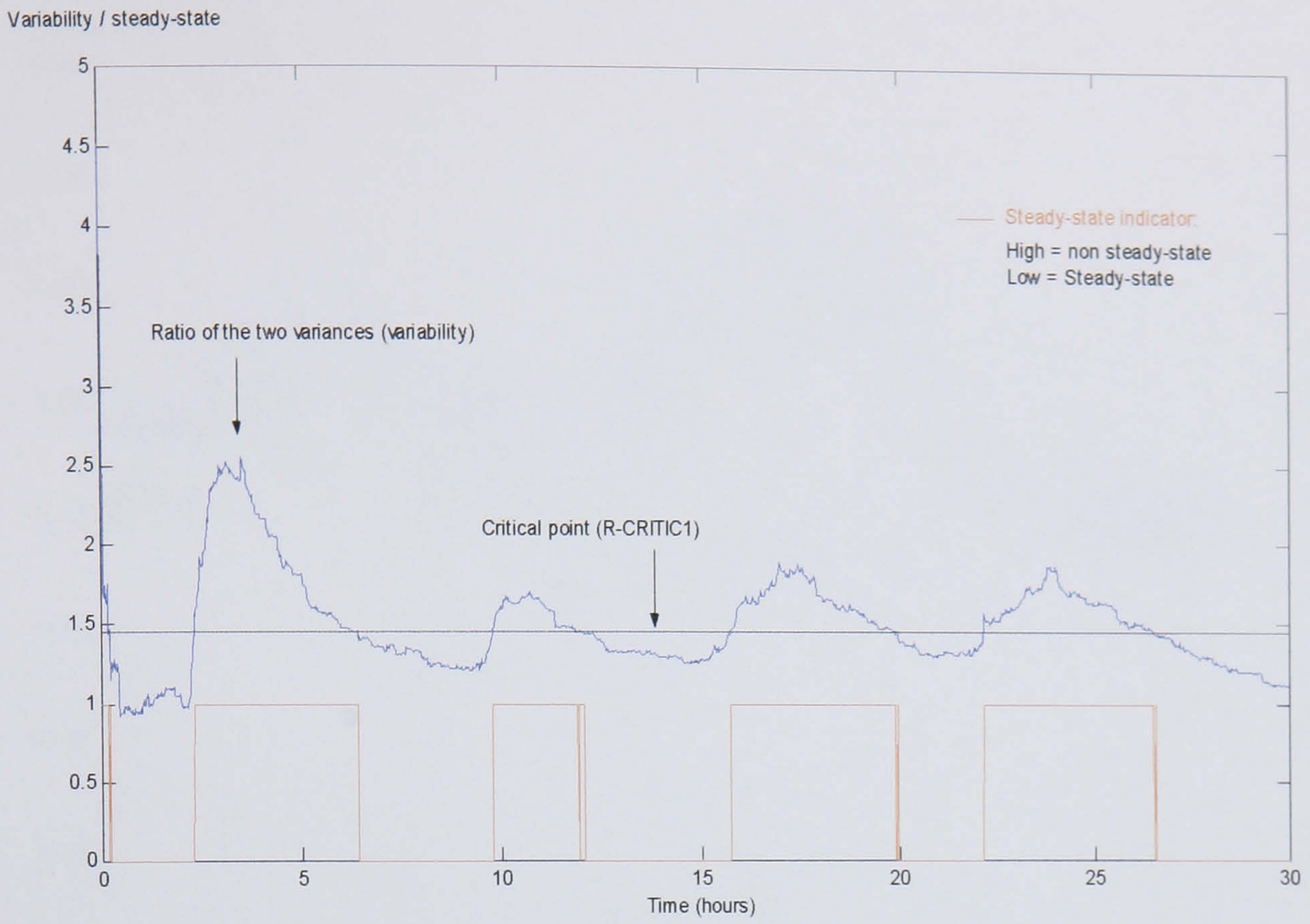


Figure (5-2): Ratio of the two variances (variability measure) related to C_{b1} along with the critical value and output of SSD module.

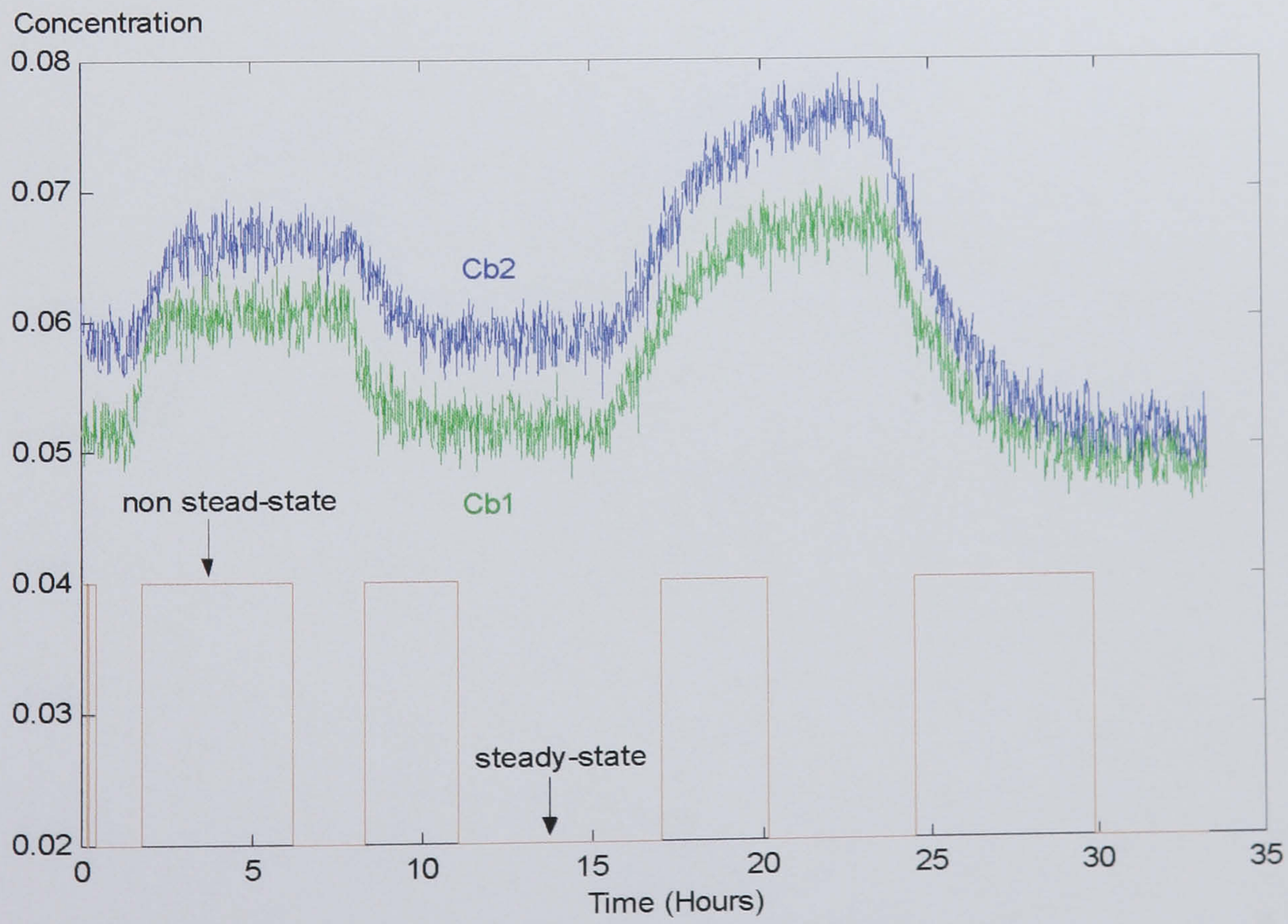


Figure (5-3): Filtering technique applied to SSD algorithm, test based on C_{b1} only.

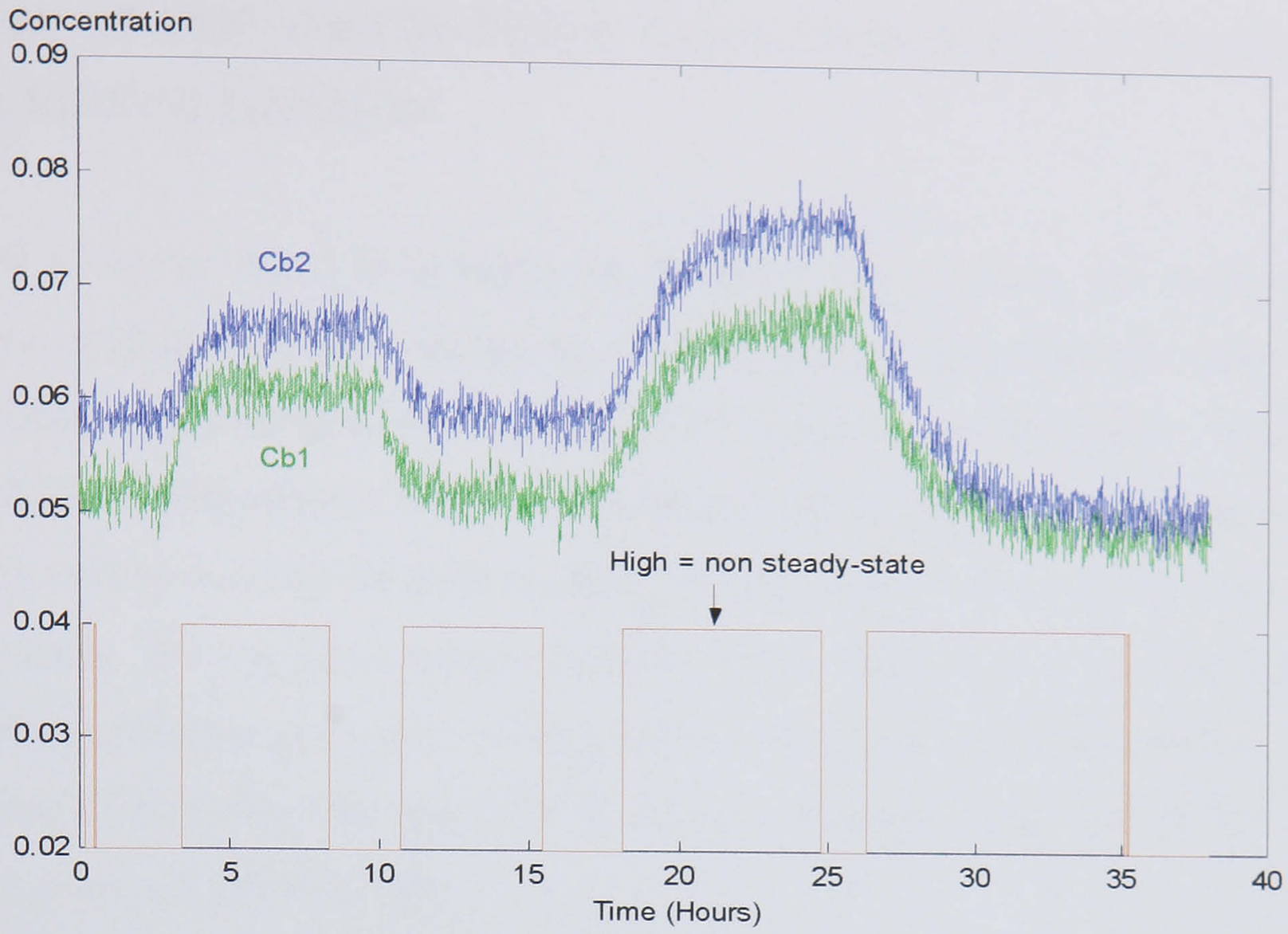


Figure (5-4): Filtering technique applied to SSD algorithm, test based on C_{b2} only.

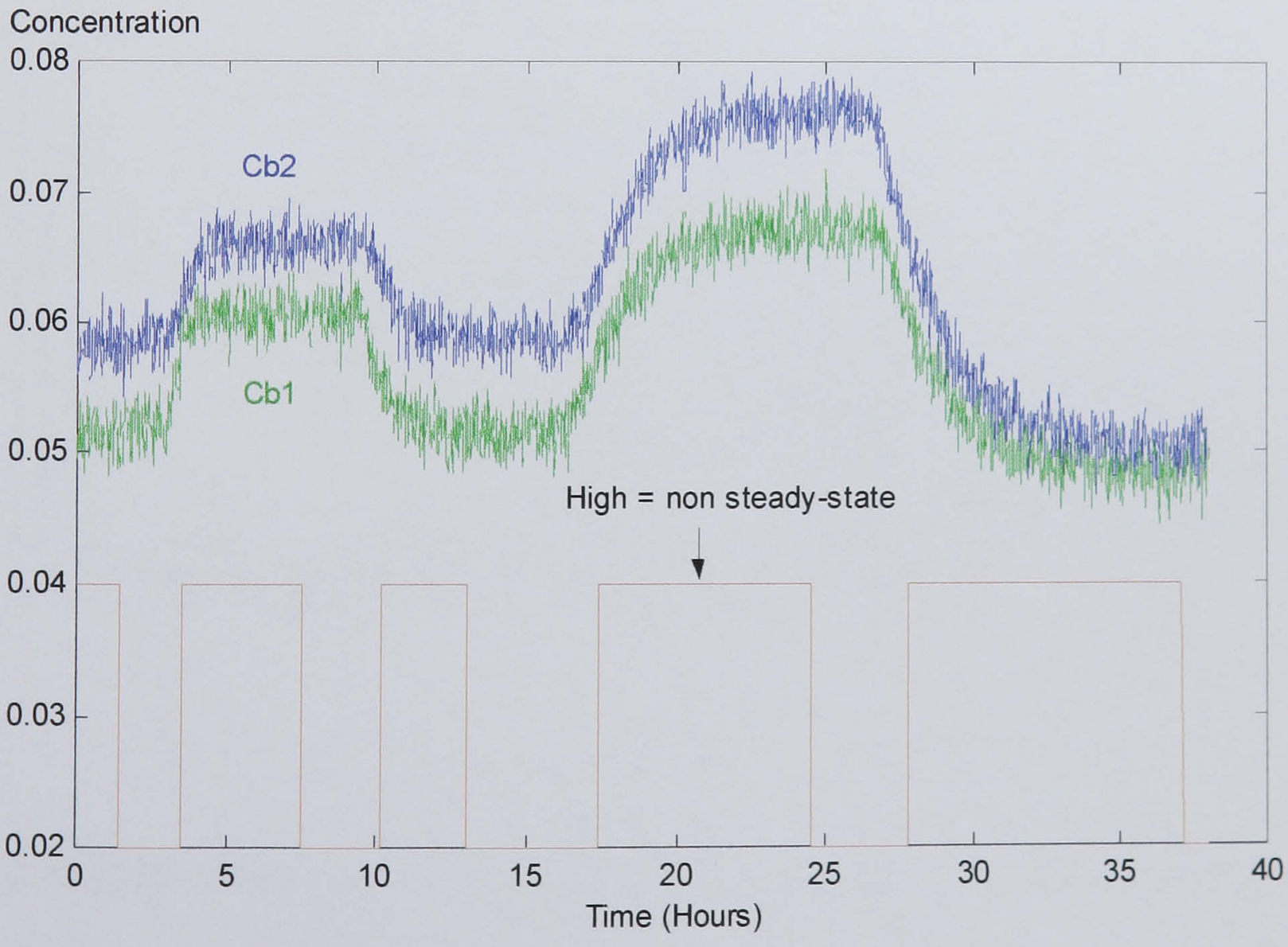


Figure (5-5): Filtering applied to SSD algorithm, test based on both C_{b1} and C_{b2}.

5.6 USE OF THE STEADY-STATE DETECTION MODULE IN STATIC DATA RECONCILIATION

As well as being tested in isolation on the two CSTR system, the steady-state detection module was also set up to work in conjunction with the static data reconciliation module described in chapter 4 (Abu-el-zeet et al., 2000). The idea being that the SDR module would either be enabled or disabled depending on the information provided by the SSD module, namely whether or not the process is at steady-state. During the transient periods where SDR would be disabled, an alternative estimator such as a moving horizon estimator could be employed. A schematic of how the SSD and SDR modules are set up within one OTISS case study is shown in Figure (5-6).

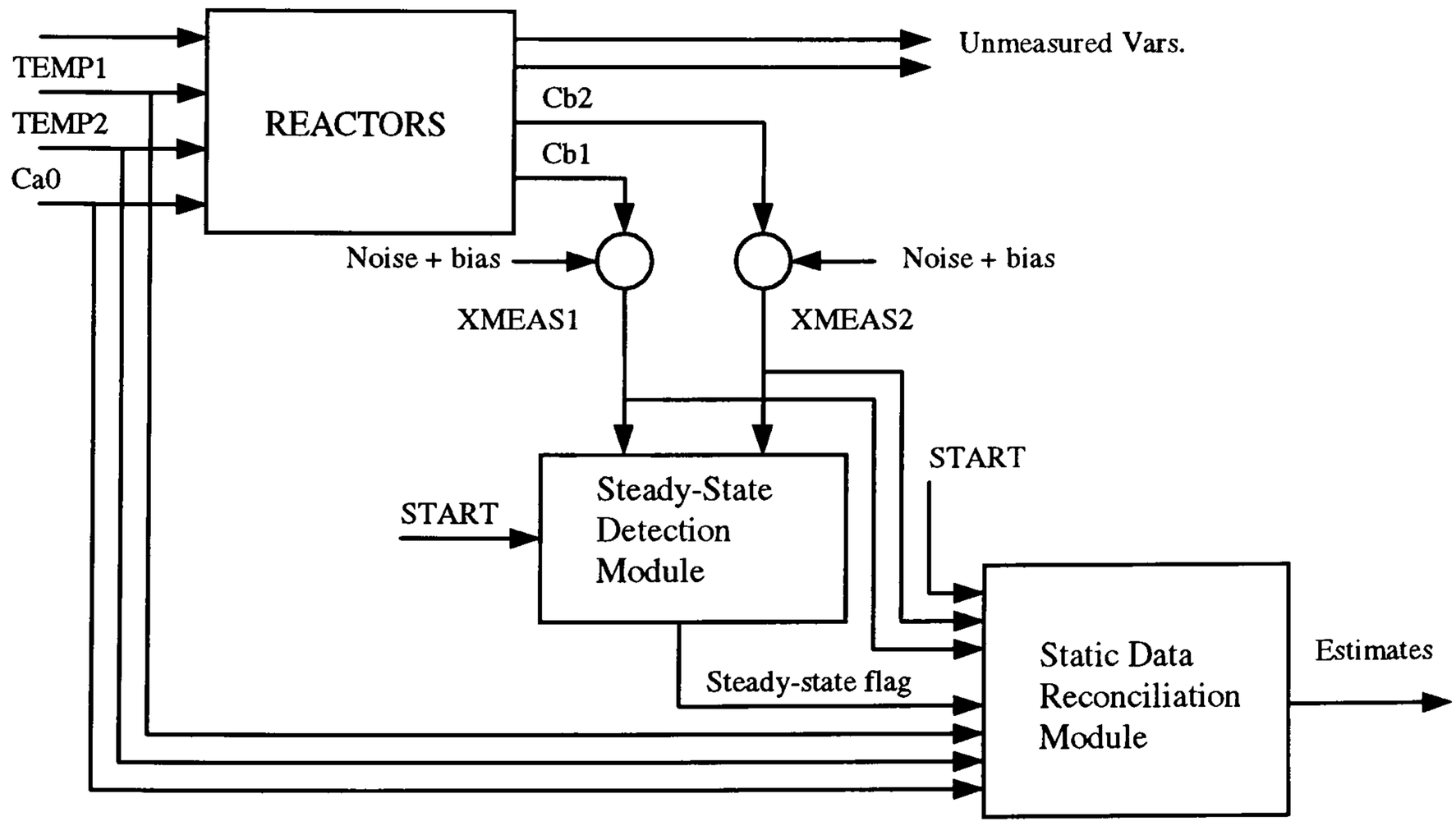


Figure (5-6): Schematic of the SDR-SSD simulation case study in OTISS.

5.6.1 Simulation Results

In order to test the capabilities of the SSD module to identify periods of steady-state and the ability to enable the SDR module, intermittent transient conditions were simulated by changing the temperature T_1 from 307 [K] to 310 [K] and then back to 307[K]. Furthermore, to test the influence of systematic bias on the SSD algorithm, bias was added to one of the measurements towards the end of the simulations. Figure (5-7) shows the simulated plant output, the noise and bias corrupted measurements of C_{b1} and C_{b2}, the estimates of C_{b1} and C_{b2} from the SDR module and the output of the SSD module determining whether or not the process is at steady-state. During the first 4.5 hours, the SSD algorithm is initializing and settling. Throughout this time it indicates that the process is not at steady-state and therefore rightly disables the SDR module. The estimates of C_{b1} and C_{b2} thus remain at the initial values. Once the SSD module detects the presence of a steady-state condition, it enables the SDR module which immediately produces correct estimates of C_{b1} and C_{b2}.

At the introduction of a transient, a delay of between 30 minutes to 1 hour can be seen before the SSD algorithm detects the change. The delay arises from the fact that the ratios of the variance measures are normally around unity when a steady-state condition prevails and will take time to increase past the critical values. It is important to point out that in this simulation, the SSD test is based on both the measured variables. This means that the delay in detecting a change is even greater due to the fact that both critical values have to be exceeded before a change in steady-state is flagged. The critical values used for this simulation are: $R_{CRITIC1} = 1.45$ and $R_{CRITIC2} = 1.25$. It can be seen in Figure (5-7) that due to this delay, the SDR module is working through a large proportion of the transient period. In fact, the SDR module in this case, has found correct estimates for the new steady-state condition before SSD detects the change.

The introduction of systematic bias after 21 hours, does not affect the SSD module. This is because the bias is only on C_{b1} which only affects the variability

measure related to that measurement. Since the SSD test is based on both measurements, the bias has no effect on the overall output of SSD. This means that the SDR module is not disabled and is able to continue estimating and compensating for the bias.

The problem of the increased delay associated with the SSD algorithm can be greatly reduced by the use of only one measurement (C_{b1}) for the SSD calculations, see Figure (5-8). It can be clearly observed that the SDR module is disabled immediately after the introduction of the transients. However, there is a considerable disadvantage to doing this and that is illustrated by the introduction of systematic bias on C_{b1} after 21 hours. The introduction of the bias causes a non-steady-state condition to be falsely declared. This is not desirable as the SDR module will be disabled throughout this period where the estimates are much needed. This potential problem emphasises the need for using two or more measurements for the SSD algorithm even though this means having to cope with extended delays in the detection of transients and steady-state conditions.

With the conflicting problems of delay and false alarms due to bias in mind, the only means left to improve the performance of the SSD module is through the selection of the critical values. Figures (5-9) and (5-10) show two combinations of critical values. Figure (5-9), where the critical values used were 1.1 and 1.1, shows no delay in detecting a movement from steady-state which is good and ensures that SDR is disabled throughout the transient period. However, it can be seen that there is an extended delay until the SSD algorithm detects a new steady-state condition and enables SDR. Figure (5-10), where the critical values used were 1.1 and 1.12, shows a similar behaviour during the first transient but during the second transient, the SDR algorithm finds a solution to the estimates before SSD detects a change. A small change in one of the critical values results in a different behaviour. In fact, due to the variations of the noise levels on the measurements from one simulation to the next, it is extremely difficult to fine tune the SSD algorithm through the selection of critical values. Therefore the results obtained in Figure (5-9) serve as an acceptable compromise where there are no

delays in detecting a non-steady-state condition and bias does not affect the SSD result but where there is considerable delay in detecting that the process has settled at a new steady-state.

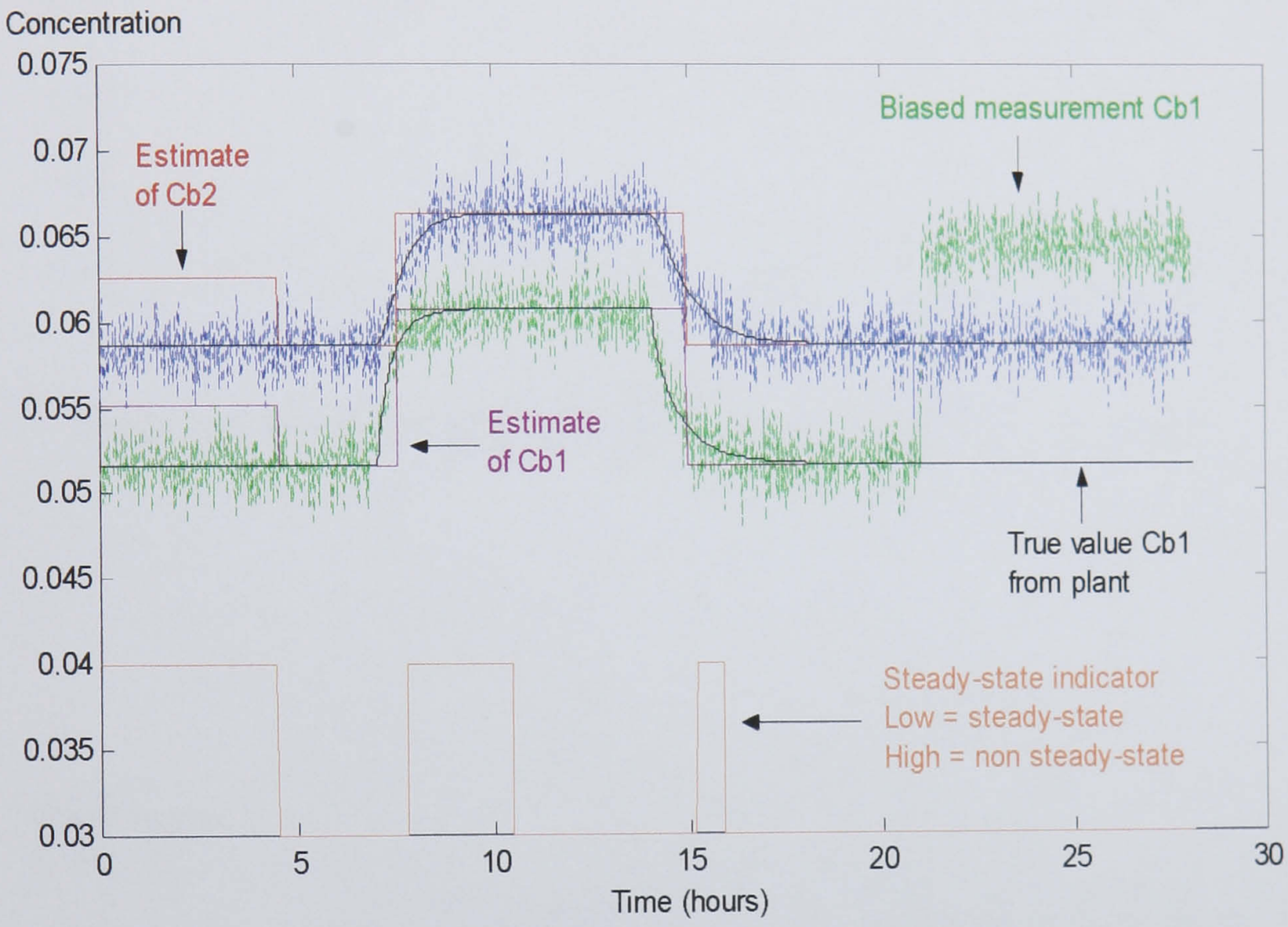


Figure (5-7): Biased measurements, estimates and SSD module output. SSD algorithm based on C_{b1} and C_{b2}, $R_{CRITIC1}$=1.45, $R_{CRITIC2}$=1.25.

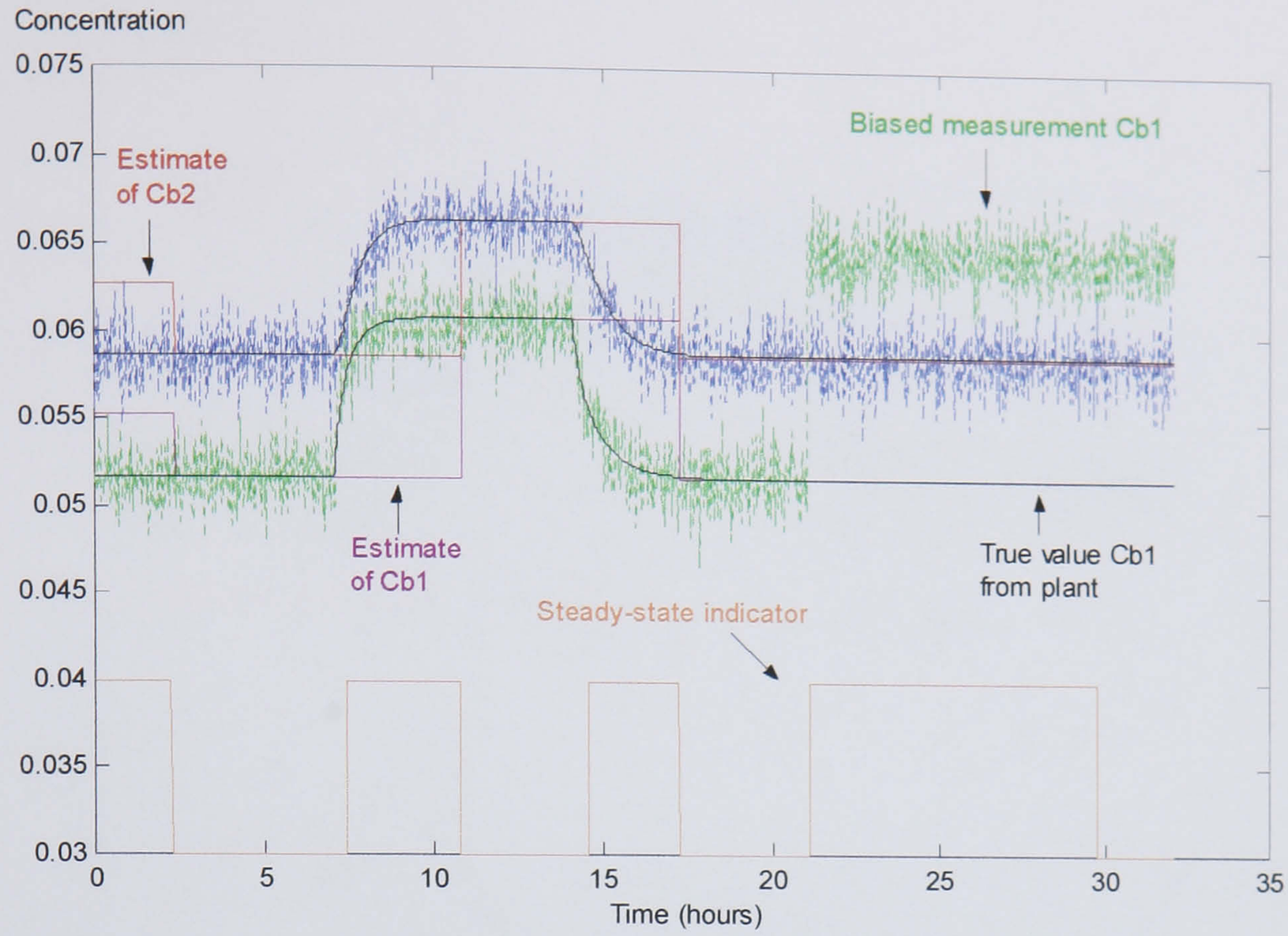


Figure (5-8): SSD algorithm based on C_{b1} only, $R_{CRITIC1}$=1.45.

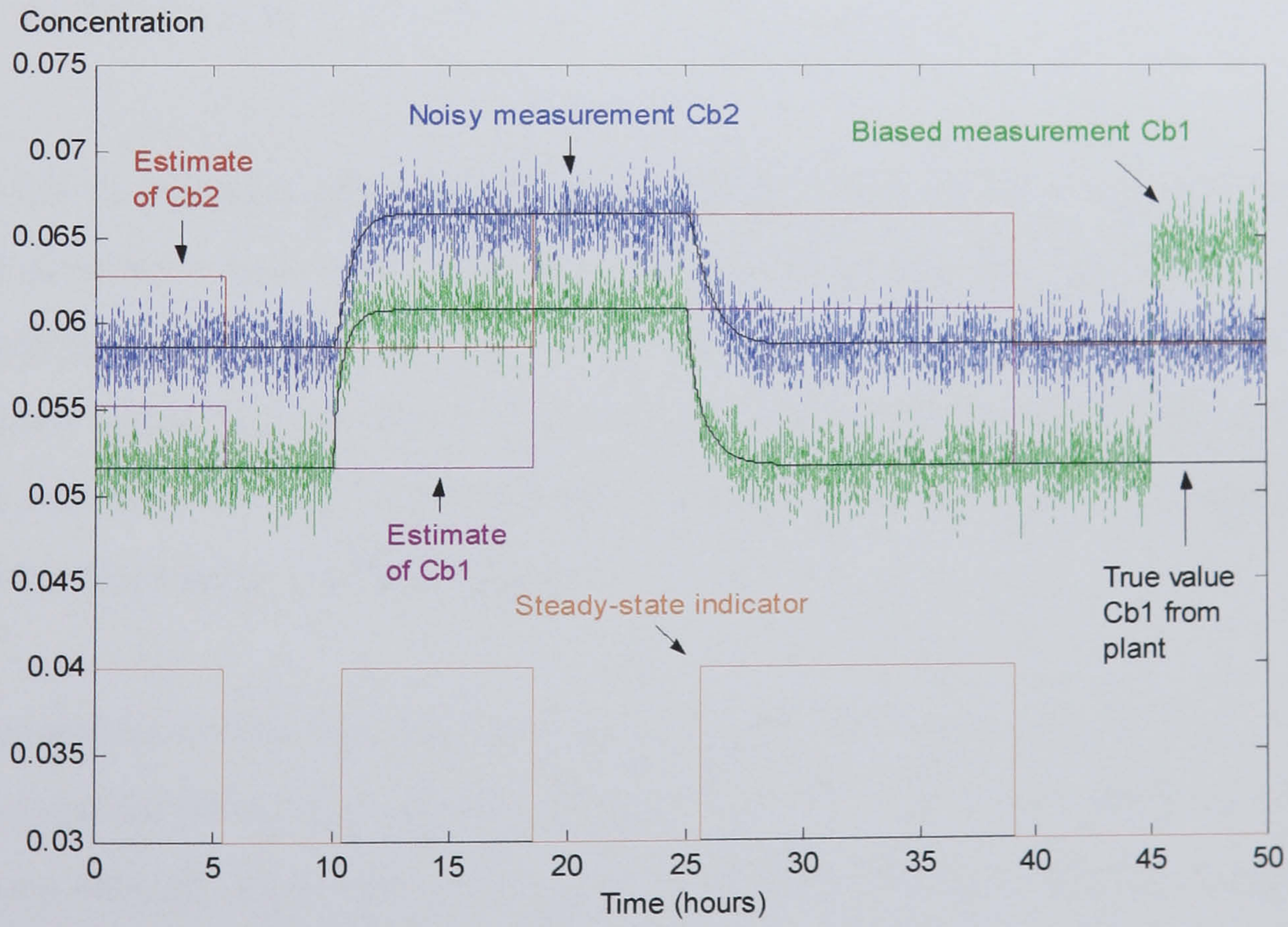


Figure (5-9): SSD algorithm based on C_{b1} and C_{b2}, $R_{CRITIC1}$=1.1, $R_{CRITIC2}$=1.1.

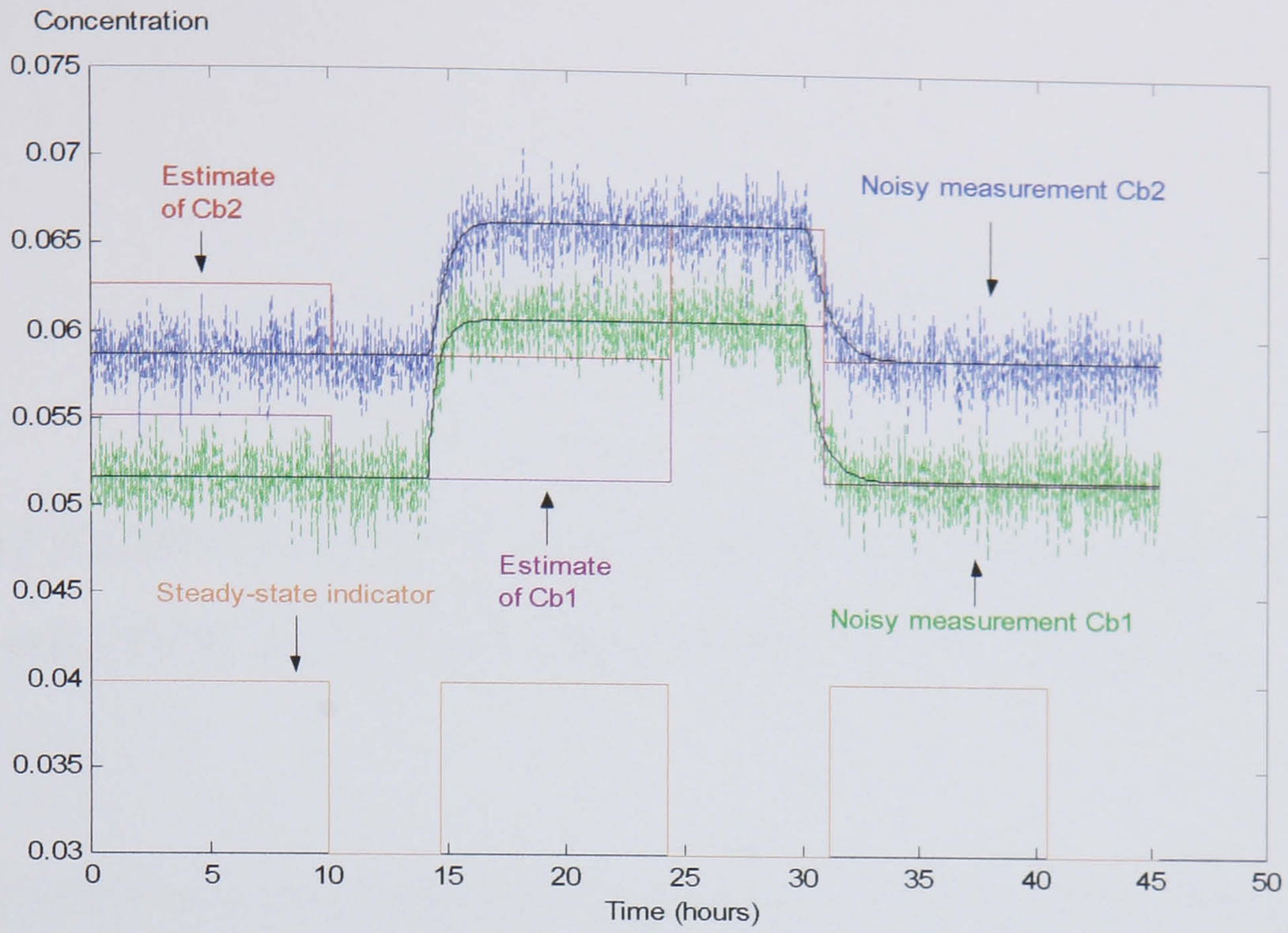


Figure (5-10): SSD algorithm based on C_{b1} and C_{b2}, $R_{CRITIC1}$=1.1, $R_{CRITIC2}$=1.12.

5.7 SUMMARY

A module to automatically identify steady-state has been implemented and tested. Simulations have been carried out on a dynamic model of two continuous stirred tank reactors connected in series. The results show that the algorithm used can accurately and efficiently detect when the process is at steady-state. Through the use of a simple filtering technique, some undesirable effects present in the original algorithm have been largely eliminated.

The steady-state detection module has been used in conjunction with the static data reconciliation module outlined in chapter 4. The results from the steady-state detection module have been used to either enable or disable the SDR module depending on whether or not the process is at steady-state.

CHAPTER 6

STEADY-STATE OPTIMISATION USING DATA RECONCILIATION AND BIAS ESTIMATION.

Data reconciliation techniques can considerably reduce the inaccuracy of process data due to measurement errors. This results in improved process knowledge and control system performance. In this chapter static data reconciliation is applied within a steady-state optimisation scheme. The optimal performance of a dynamic model of two continuous stirred tank reactors is estimated using ISOPE, the two step method introduced in chapter 2. The performance of two schemes is compared. The first scheme being a steady-state optimisation of the process, where the measurements are contaminated with noise and systematic bias. The second scheme uses static data reconciliation techniques to reconcile the process data and estimate any systematic bias before the data are fed to the optimisation algorithm.

6.1 STATIC DATA RECONCILIATION

Reliable process data are the key to the efficient operation of chemical plants. Most on-line optimisation and control activities are based on small improvements in process performance which in large plants yield considerable gains in terms of profit. However, if the measured data used for the optimal predictions is contaminated with noise and systematic bias then the errors in the data can easily exceed or mask actual changes in process performance.

Static data reconciliation (SDR) was introduced in chapter 4. A number of simulations were carried out to test the capabilities of the SDR module to reconcile process data. Here the SDR module is applied to static optimisation with the aim of identifying distinct advantages in using SDR to provide reconciled estimates of measured variables as opposed to using raw untreated data.

6.2 STEADY-STATE OPTIMISATION

The optimal set points calculated by steady-state optimisation are based on a mathematical model of the plant. However, because of differences between the model and the plant, the set-points obtained will only be optimal for the model and not the real plant. The two-step method has been proposed to take into account differences between the mathematical model and the real process. The steady-state model contains parameters which are estimated by comparing model-based and measured outputs. Then the system optimisation and parameter estimation problems are treated separately and solved repeatedly until convergence is achieved. However, as mentioned previously in chapter 2, this is not sufficient since there is interaction between the optimisation and parameter estimation problems and thus the solution obtained will, in general, be sub-optimal.

The ISOPE technique developed by Roberts (1979) and introduced in detail in chapter 2 allows for the interaction between the two problems. The interacting variables are separated and a modifier is introduced in the model-based optimisation. The role of this modifier is to take into account differences between the real process and model-based output derivatives with respect to the manipulated variables. This ensures the correct optimal operating point for the real process in spite of model-reality differences.

The ISOPE algorithm finds the values of the decision variables u_c that minimise an objective function:

$$F^*(y,u_c) \tag{6.1}$$

subject to:

$$y = K^*(u_c) \tag{6.2}$$

$$u_{\min} \le u_c \le u_{\max} \tag{6.3}$$

$$y_{\min} \le y \le y_{\max} \tag{6.4}$$

where y is a vector of measured variables, and K^* represents the real process static relationships between decision variables u_c and measured variables y. Equation (6.3) represents the constraints on the decision variables while equation (6.4) represents the constraints on the measured variables.

The algorithm finds the optimal operating conditions of the process by repeatedly solving model-based optimisation and parameter estimation problems and applying the intermediate results to the process. The technique uses a steady-state model of the process.

6.3 SIMULATION CASE STUDY

Simulations were carried out on the dynamic model of two continuous stirred tank reactors introduced in chapter 4 (see Figure 4-2). Full details of this model can be found in Garcia and Morari (1981).

In all the simulation cases that follow, the system was started from the steady-state condition given by the set-points $T_1(0)=307K$ and $T_2(0)=302K$, which yield steady-

state values $C_{b1}(0)=0.05165[Kmol/m^3]$, $C_{b2}(0)=0.05864\ [Kmol/m^3]$. The sampling time for the measurements was 1 minute. The updating time for steady-state optimisation was 60 minutes. In choosing these parameters, consideration was given to the dynamic and steady-state behaviour of the process noting that the open-loop time constant is approximately 40 minutes.

6.3.1 Implementation Issues

Two case studies were set up in OTISS™ in order to carry out the simulations related to steady-state optimisation. A schematic showing the various inputs and outputs of a basic set up involving no data reconciliation or noise is shown in Figure (6-1). The first case study involves steady-state optimisation using measurements that contain noise and systematic bias (Figure 6-2). The second case study uses data reconciliation techniques to reconcile the measurements and eliminate systematic bias prior to the optimisation stage (Figure 6-3).

The SDR module developed earlier and outlined in chapter 4 was used to provide reconciled estimates of the biased and noisy measurements. The data reconciliation problem was solved using Sequential Quadratic Programming (SQP), see Appendix A. It is assumed that the measured variables are C_{b1} and C_{b2}.

The bias added to the measurement C_{b2} is 20% of the nominal value. The values of standard deviations used were 5% of the nominal value for C_{b1} and 2% of the nominal value for C_{b2}. The objective function used for steady-state optimisation reflects the requirement to maximise the concentration of product C_{b2}.

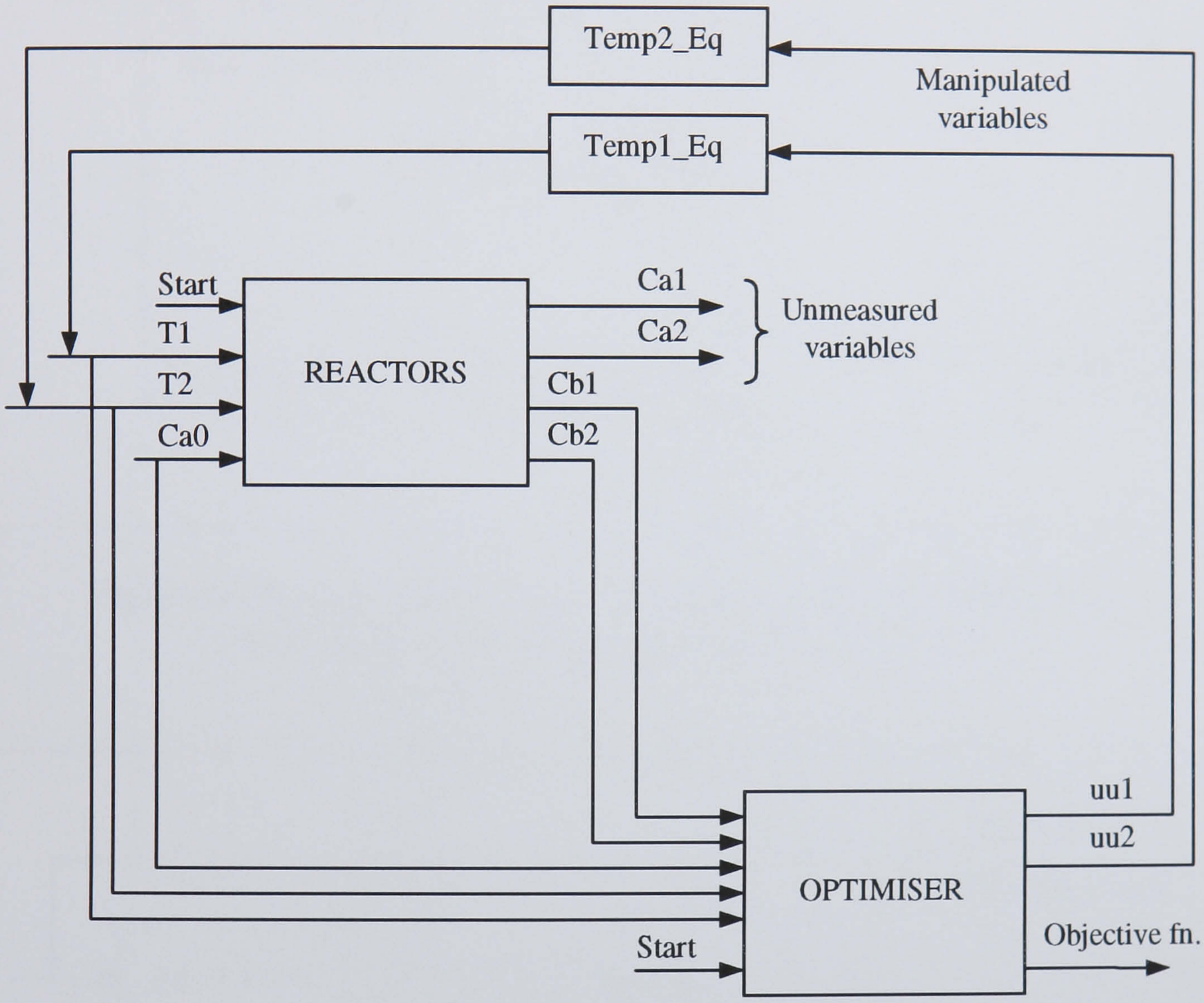


Figure (6-1): Detailed schematic of module interconnections in OTISS.

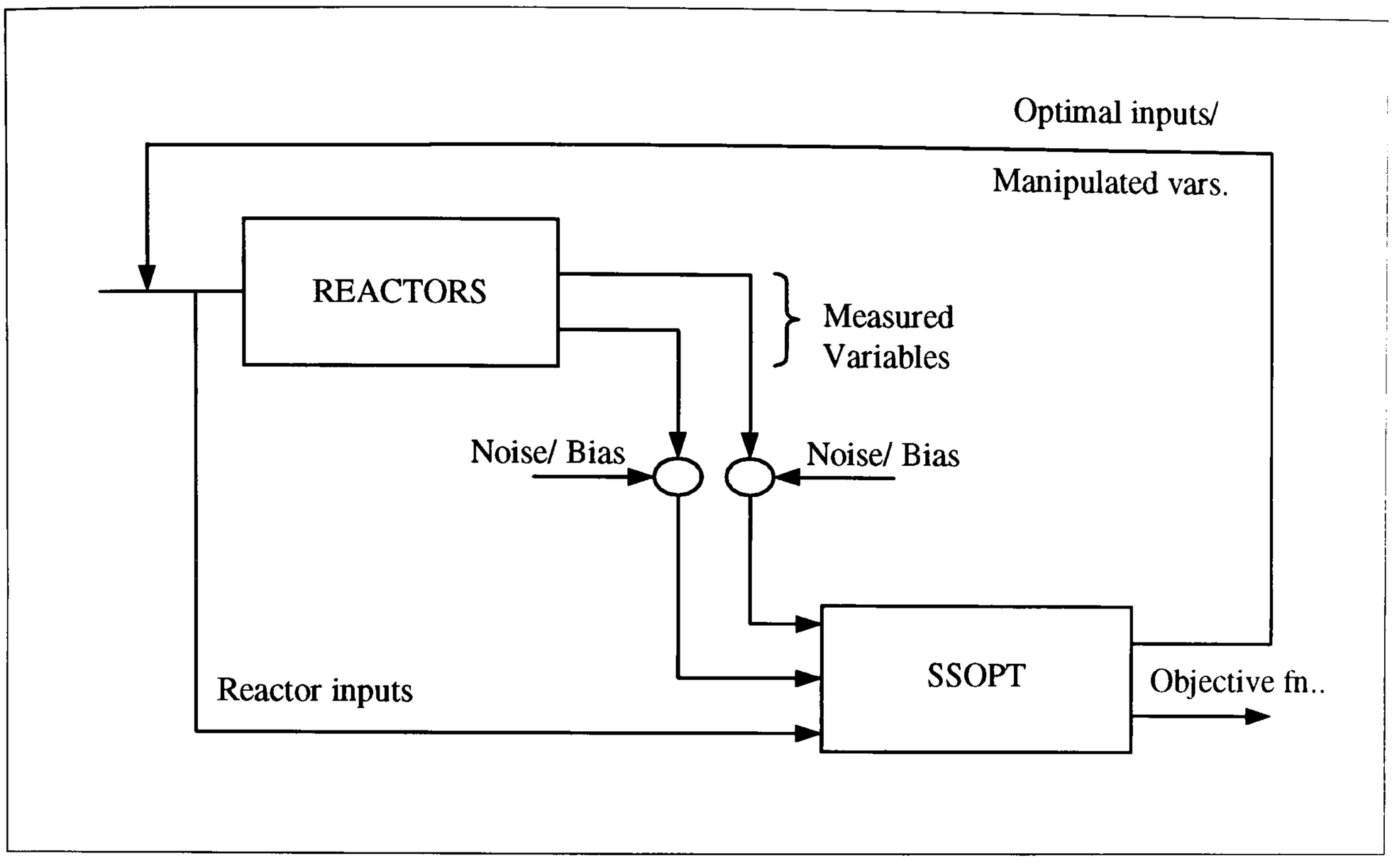


Figure (6-2): Schematic of case study (1), steady-state optimisation without the use of data reconciliation techniques.

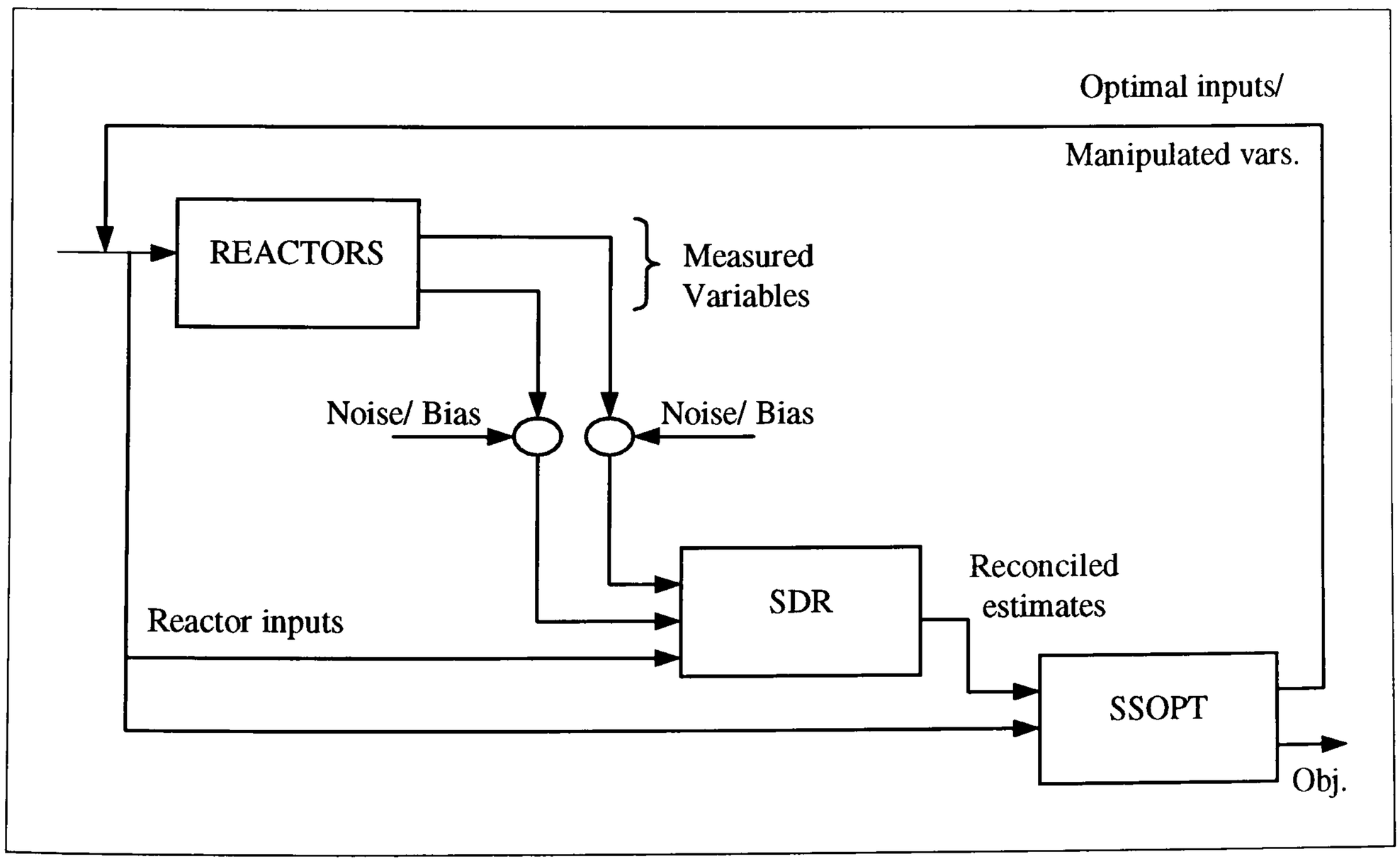


Figure (6-3): Schematic of case study (2), steady-state optimisation using SDR.

6.3.2 Discussion of Results

Figures (6-4) and (6-6) show the results from the first case with noisy measurements but no systematic bias. In this case, no data reconciliation is employed. The noisy data is fed straight to the optimisation algorithm and the optimal estimates of manipulated variables are calculated based on this flawed information. Figure (6-4) shows the noise corrupted measurements as well as the real output of the plant. Figure (6-6) shows the manipulated variables which are updated by the steady-state optimisation algorithm.

The above results should be compared with Figures (6-5) and (6-7) where static data reconciliation is used to filter the noisy measurements. Figure (6-5) shows the noisy measurements, the real output of the plant and the estimated measurements from the static data reconciliation module which are used by the ISOPE algorithm. Note that it is difficult to distinguish the reconciled measurements from the real output of the plant as they are almost superimposed. Figure (6-7) shows the manipulated variables. The response is much smoother than when static data reconciliation was not employed. The value of C_{b2} increases smoothly and quickly towards the optimum.

Figures (6-8) and (6-10) show the results when measurement C_{b2} contains systematic bias and there is no data reconciliation used. This can be compared with Figures (6-9) and (6-11) respectively which show the results when the static data reconciliation module is working in conjunction with the optimisation algorithm. Again, due to the use of reconciled data for the optimisation, the resulting optimal manipulated variables yield a much better response in terms of accuracy and maintaining an optimal profile. It can be observed that Figure (6-9) shows correct estimation and elimination of the bias.

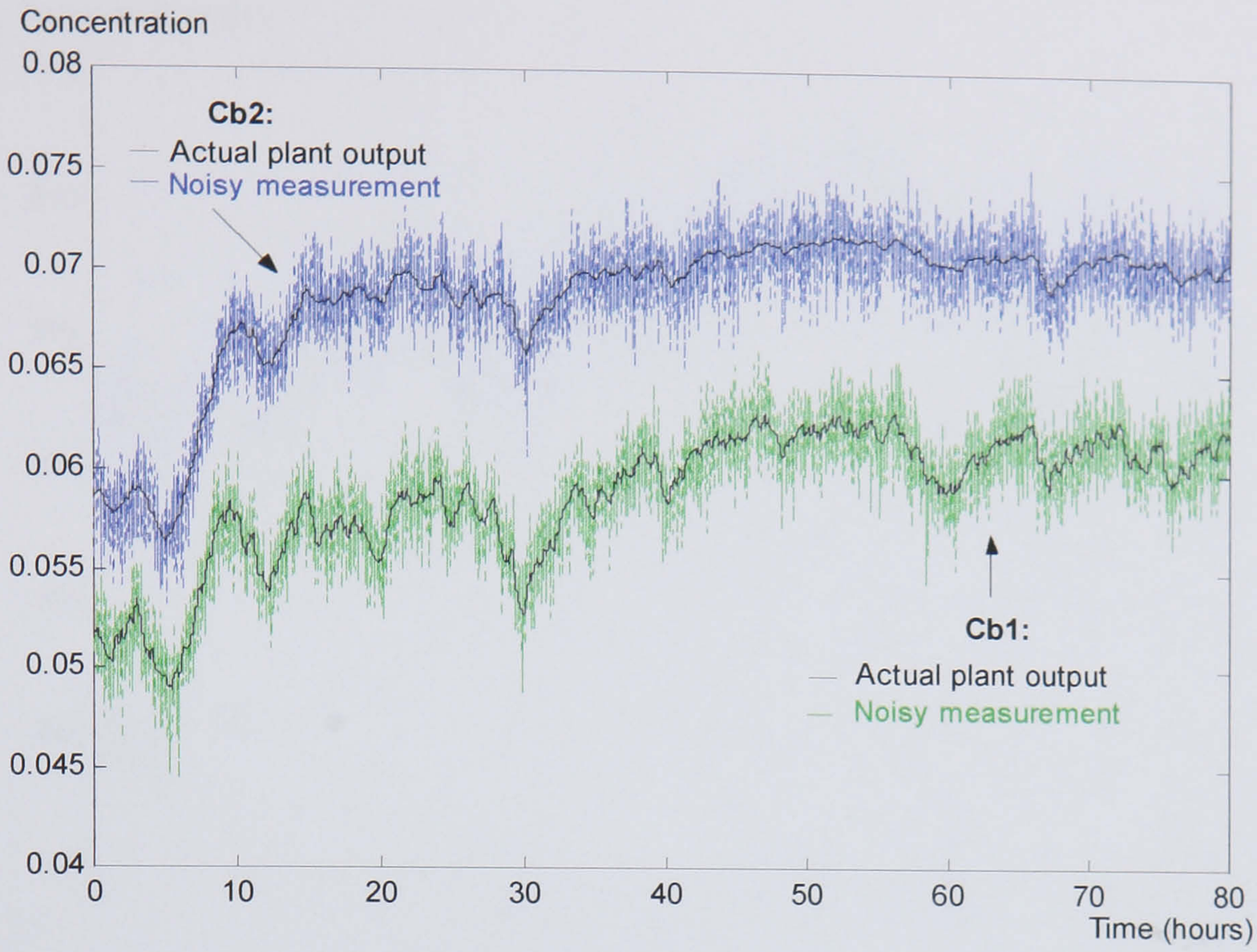


Figure (6-4): Noisy measurements and real output of the plant - case with steady-state optimisation and no data reconciliation.

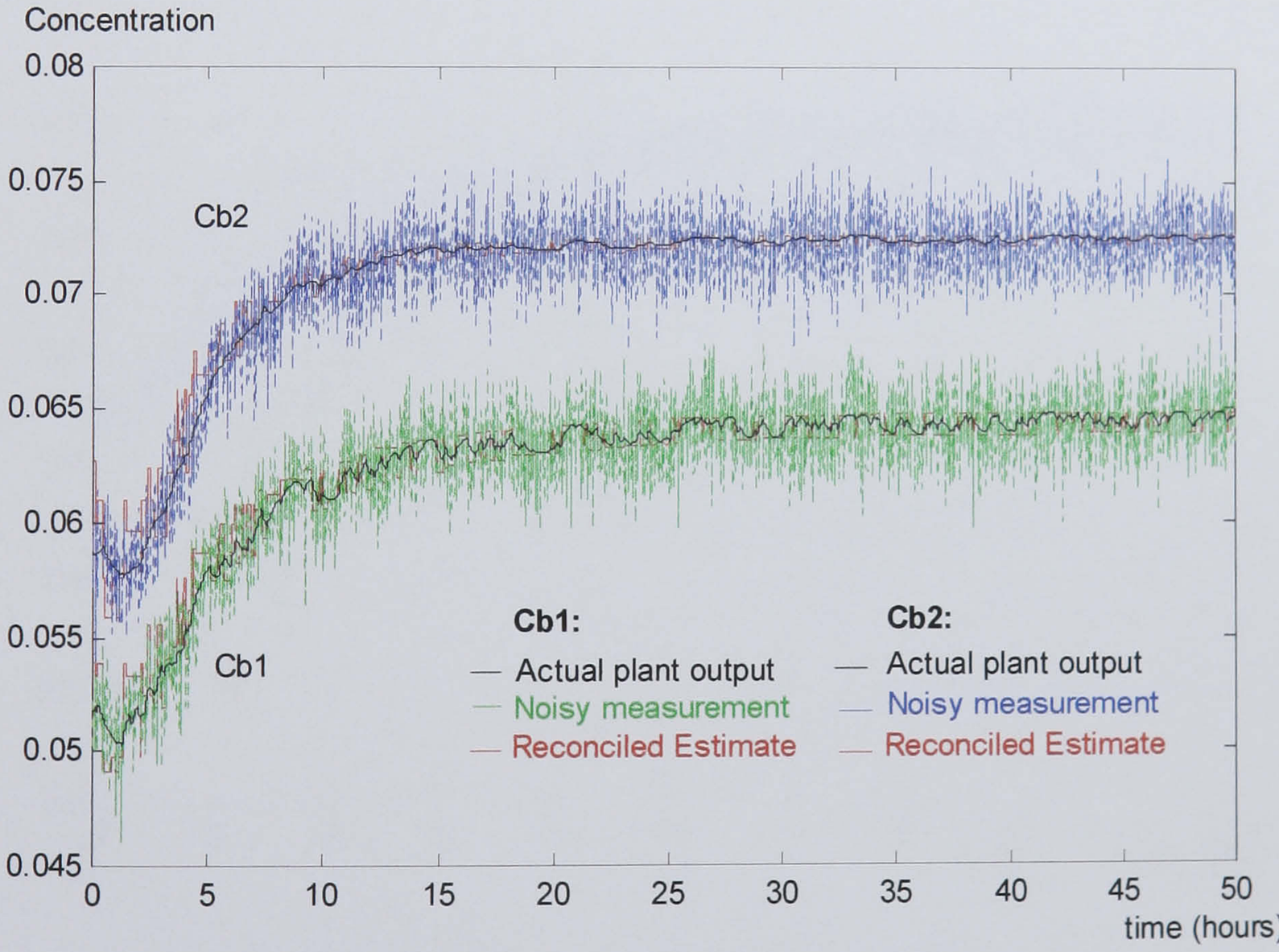


Figure (6-5): Noisy measurements, reconciled measurements and real output of the plant - case with steady-state optimisation using data reconcilaition.

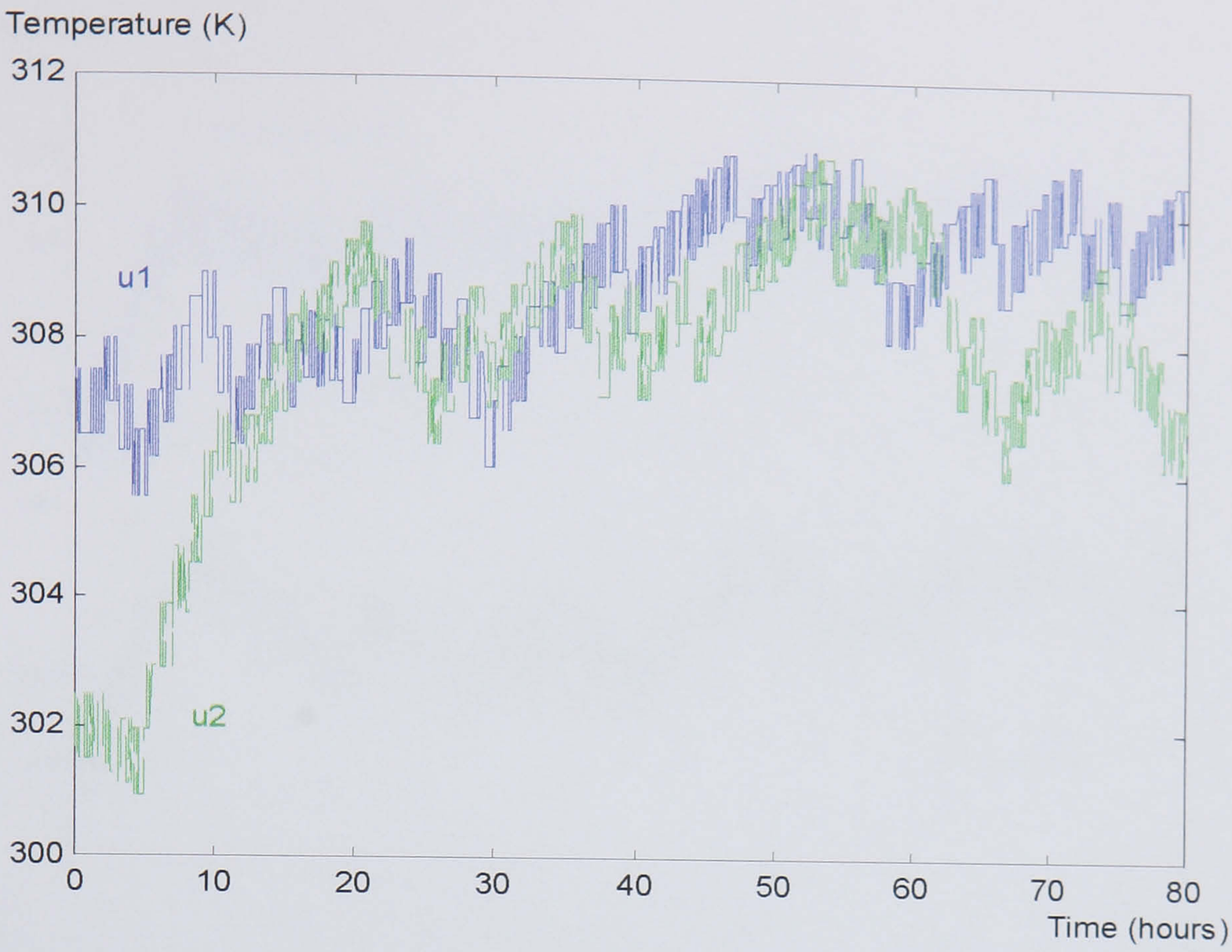


Figure (6-6): Manipulated variables -
case with steady-state optimisation and no data reconciliation.

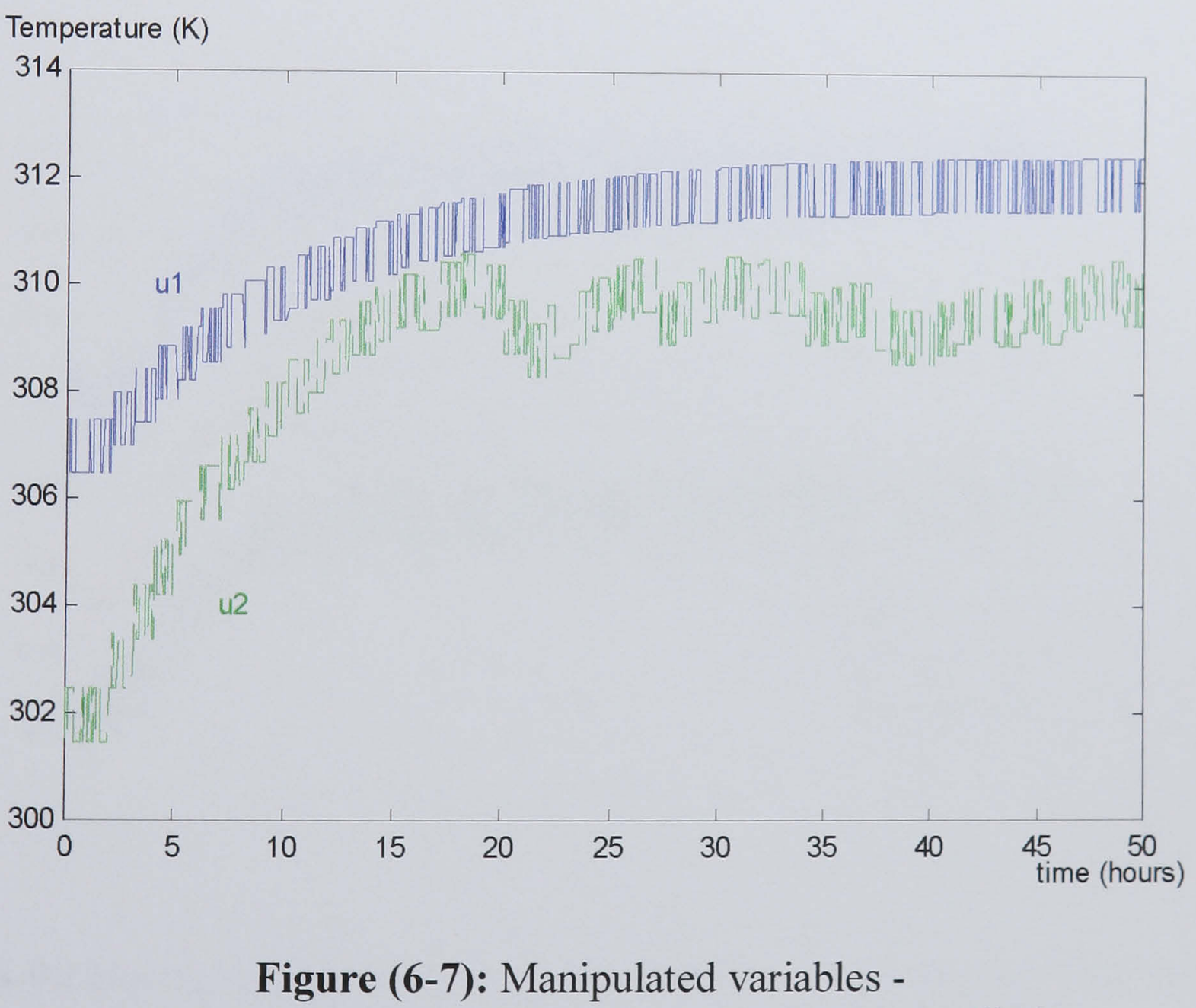


Figure (6-7): Manipulated variables -
case with steady-state optimisation using data reconciliation.

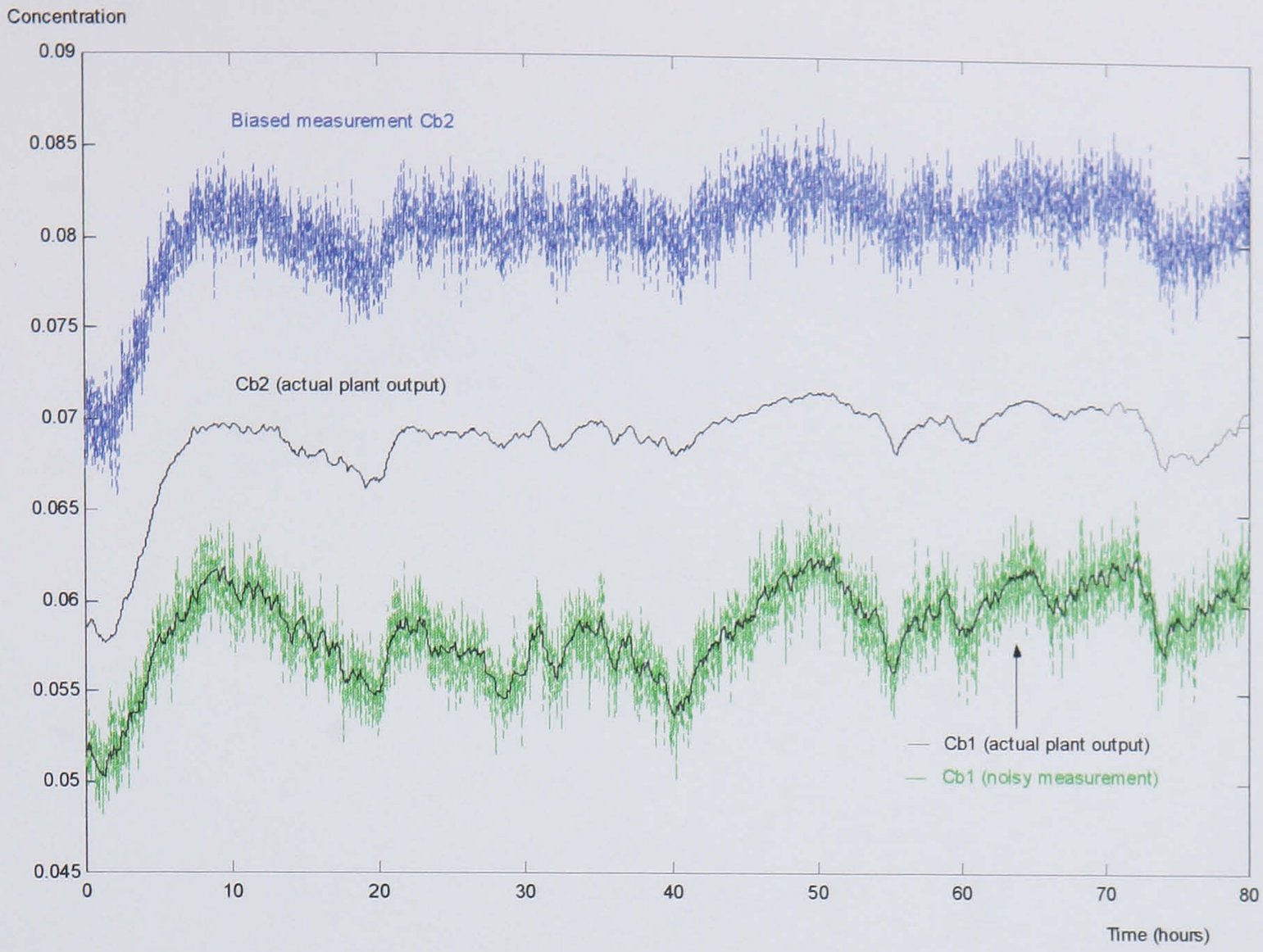


Figure (6-8): Noisy measurements and real output of the plant - case with steady-state optimisation and no data reconciliation when C_{b2} contains systematic bias.

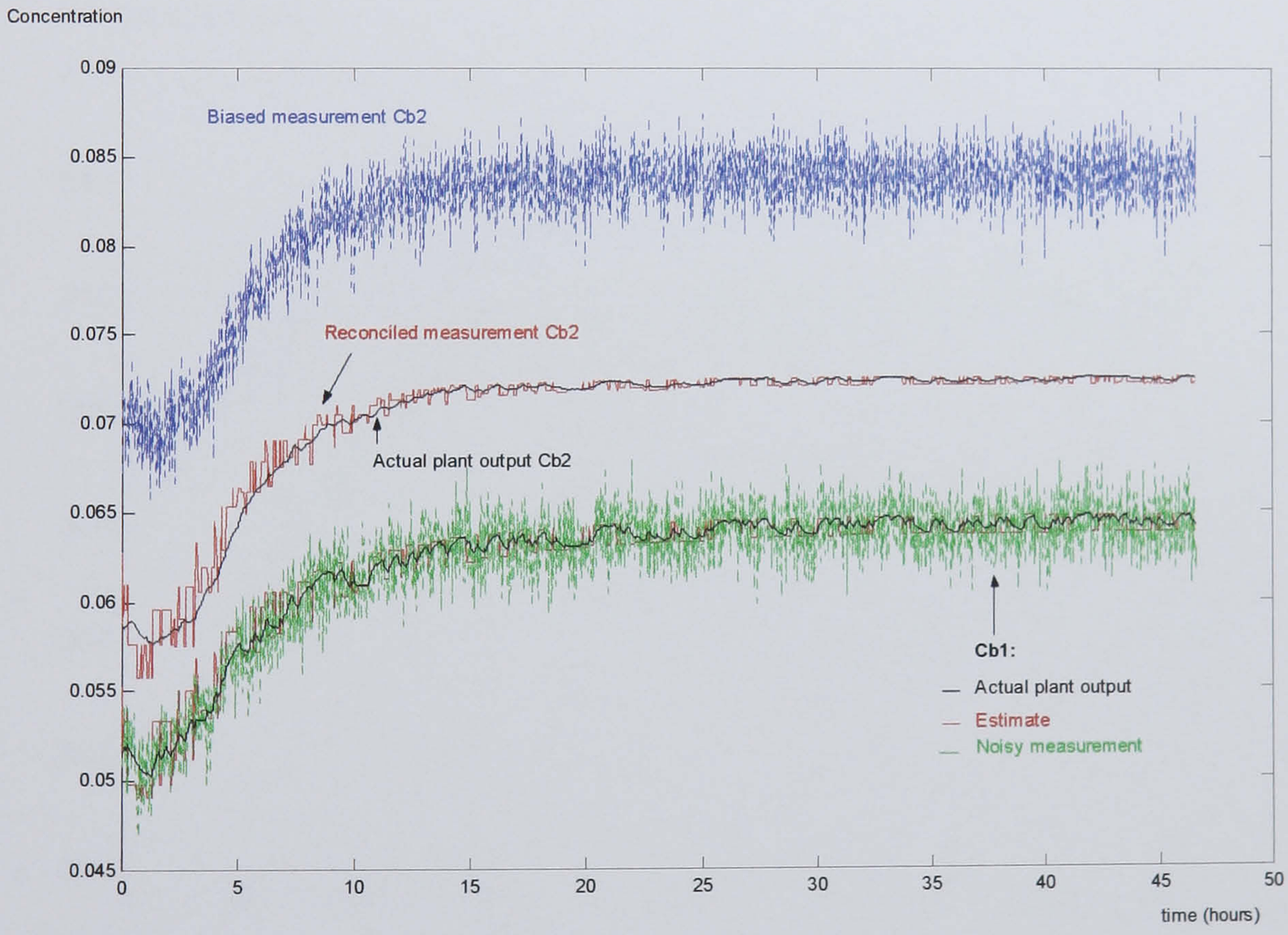


Figure (6-9): Noisy measurements, reconciled estimates and real output of the plant - case with steady-state optimisation using data reconciliation when C_{b2} contains systematic bias.

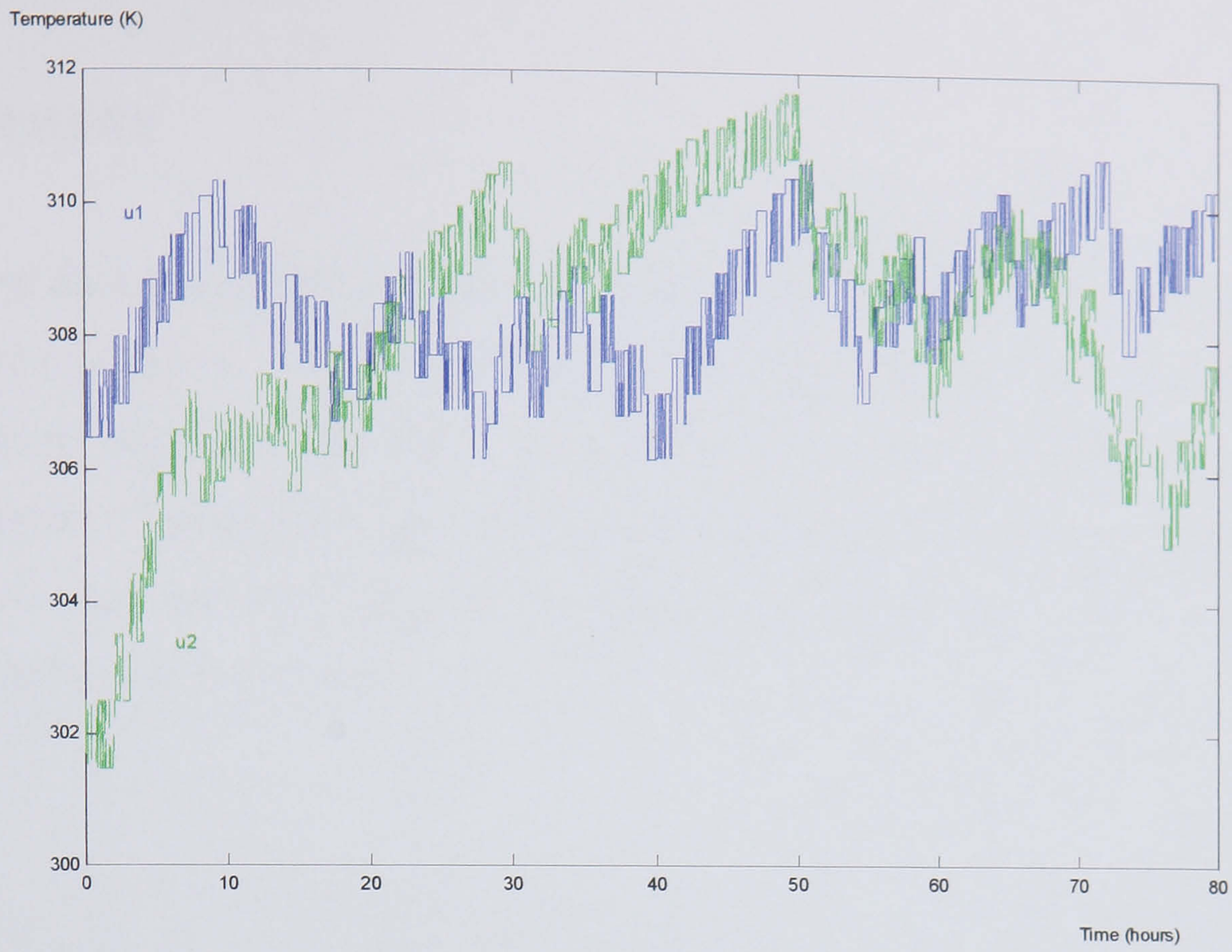


Figure (6-10): Manipulated variables - case with steady-state optimisation and no data reconciliation when C_{b2} contains systematic bias.

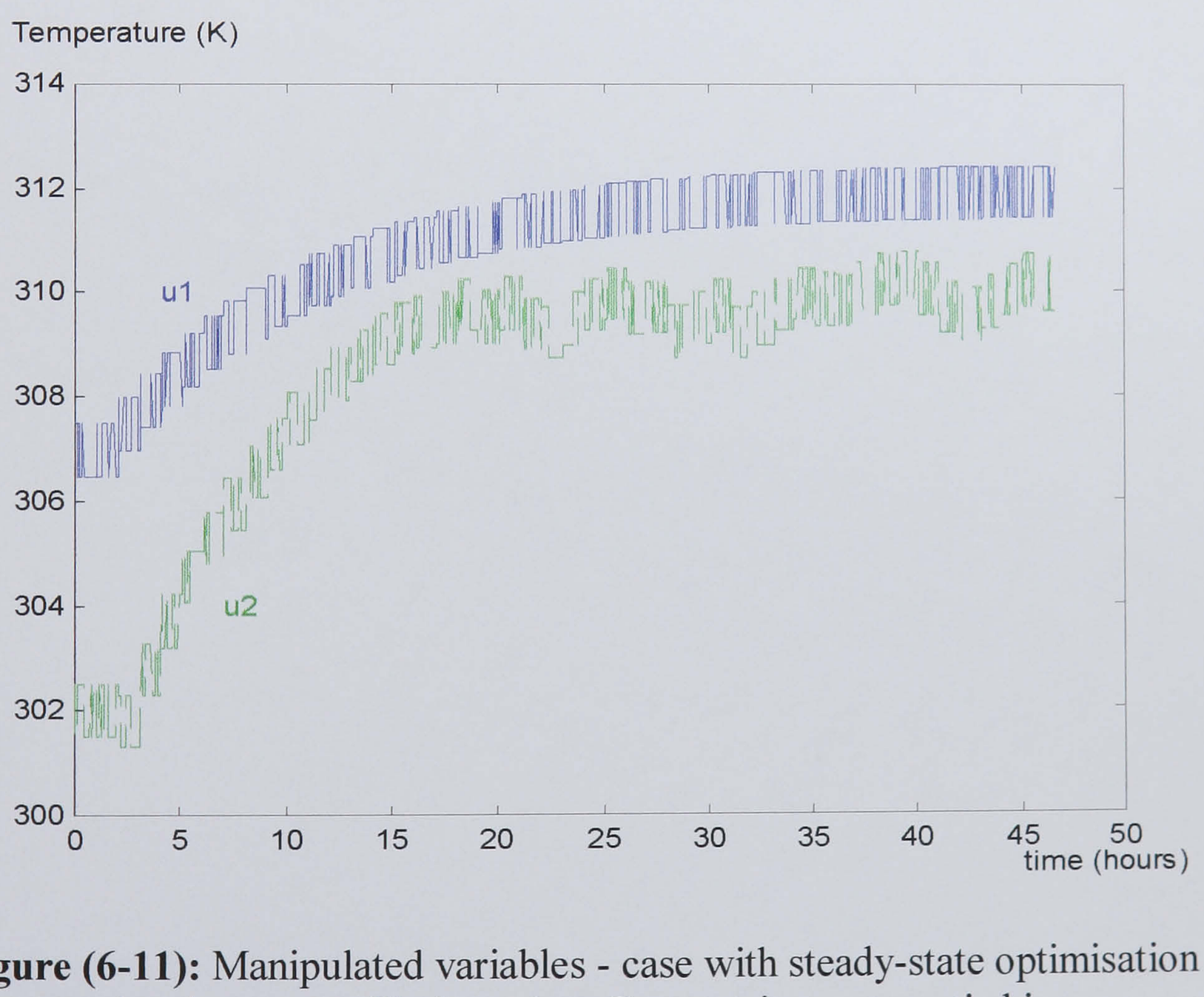


Figure (6-11): Manipulated variables - case with steady-state optimisation using data reconciliation when C_{b2} contains systematic bias.

6.4 SUMMARY

The use of data reconciliation and bias estimation has been shown to improve static optimisation schemes. Where corrupted data was used directly for optimisation, the results were not desirable even when there was no systematic bias in the measurements. Using static data reconciliation to adjust the measurements prior to optimisation has improved the results considerably. Systematic bias has also been correctly estimated and eliminated.

CHAPTER 7

BIAS AND GROSS ERROR DETECTION IN DYNAMIC DATA RECONCILIATION

7.1 INTRODUCTION

It has been shown in chapter 4 that the use of static models in the reconciliation of dynamic data can sometimes produce poor results. During transient periods the change in process dynamics and the resulting differences between the static model and the real process can be a limiting factor in the success of any data reconciliation scheme.

In this chapter a moving horizon estimator implemented in software is used for the reconciliation of dynamic process data. The algorithm uses a dynamic model of the process which is a dynamic simulation of two continuous stirred tank reactors used previously in chapters 4,5 and 6. The capabilities of the estimator are extended such that the identification and elimination of gross errors (outliers) and systematic biases are possible. In chapter 4, where a static data reconciliation module was developed, the estimation of systematic bias was possible but not its identification. There, it was a prerequisite to the estimation procedure that the measurements affected by bias be known *a priori*. Here the algorithm takes care of the identification process.

Prior to the introduction of the proposed algorithms, a general review of previous work carried out on dynamic data reconciliation is first presented. Following an introduction and formulation of the moving horizon estimator, separate reviews on the fields of gross error and bias detection and identification are presented.

These reviews relate to previous work published and which deal with static or dynamic processes.

7.2 DYNAMIC DATA RECONCILIATION

Methods for reconciling steady-state process data are well developed (Bodington, 1995). However, even so called 'steady-state' processes are never truly at steady-state. They continually undergo variations about a nominal steady-state condition (Narasimhan and Mah, 1988). Therefore, dynamic models would undoubtedly be a far better representation of the real process. Moreover, some chemical processes are intrinsically dynamic and in some chemical processes disturbances with dynamic effects may occur frequently (Becerra et al., 1998c). For the reasons outlined above and for the fact that steady-state conditions are a particular case in a dynamic model, it is desirable to develop dynamic data reconciliation strategies.

Process data from systems governed by dynamic equations can be reconciled using the Kalman filter or the Extended Kalman Filter (EKF). Unfortunately, chemical engineering systems often operate dynamically in highly nonlinear regions where the EKF may be inaccurate. In addition, the Kalman filter may not be adequate in the presence of inequality constraints (Liebman et al., 1992).

A number of researchers have addressed the dynamic data reconciliation problem. Some have presented algorithms which simultaneously deal with dynamic data reconciliation and gross error or bias detection. Darouach and Zasadzinski (1991) were some of the first researchers to address the issue. They presented an on-line estimation algorithm for linear dynamic systems. Their algorithm involves a recursive solution technique in weighted least squares. Liebman et al. (1992) presented a method for nonlinear dynamic data reconciliation using Nonlinear Programming (NLP) techniques. Some extensions for the treatment of biased measurements were also discussed. Rollins and Devanathan (1993) proposed a backward difference approximation technique that is computationally

simpler than that of Darouach and Zasadzinski (1991). Estimation accuracy is improved by means of averaging two estimates for all time instants.

Ramamurthi et al. (1993) presented a successively linearised horizon-based estimator (SLHE) for dynamic data reconciliation in closed-loop systems. They compare the performance of SLHE with the Extended Kalman Filter (EKF) and nonlinear programming (NLP) approaches. Albuquerque and Biegler (1995) proposed a method for dynamic data reconciliation which works by discretising the set of ordinary differential equations using a one-step integration method and then uses the Sequential Quadratic Programming (SQP) method to solve the resulting NLP. Karjala and Himmelblau (1996) proposed a procedure for dynamic reconciliation of data using recurrent neural networks and the EKF.

Albuquerque and Biegler (1996) presented a study on data reconciliation and gross error detection for dynamic systems. Bagajewicz and Jiang (1997) gave a brief review of data reconciliation using both steady-state and dynamic models. They also proposed an integral method that performs dynamic data reconciliation on linear systems. The result is a computationally inexpensive analytical solution. They also discussed gross errors and proposed a method to detect bias.

Becerra et al. (1998c and 1999e) presented a method for dynamic data reconciliation using sequential modular simulators using a bank of extended Kalman filters. Becerra et al. (1999d) proposed a dynamic data reconciliation method for nonlinear systems described by differential-algebraic models using the EKF.

7.3 MOVING HORIZON ESTIMATION

The moving horizon estimation problem consists of calculating the best estimates of the measured variables, unmeasured states, physical parameters and measurement bias, given a sequence of *nh* measurements and a dynamic model of

the process. A number of researchers have worked on moving horizon estimation. Liebman et al. (1992) presented a moving horizon dynamic data reconciliation algorithm which uses SQP (see Appendix A) and achieves simultaneous solution and optimisation by means of orthogonal collocation on finite elements. Ramamurthi et al. (1993) use local linearisation of the nonlinear dynamic model equations to define a two level strategy for the estimation of inputs and outputs using SQP. Henson and Seborg (1997) provide a review of moving horizon estimation along with a mathematical formulation. The following is a formulation by Becerra (1999).

7.3.1 Formulation of the moving horizon estimator

The moving horizon estimation problem may be defined as a nonlinear dynamic optimisation problem with a discrete time performance index and continuous time model and constraints. It is assumed that the measurements are sampled with a sampling time T_s. The process model is represented as:

$$\dot{x}(t) = f_x(x(t), u(t), p, t) \tag{7.1}$$

where $x \in \Re^{n_x}$ is a differential state vector, $u \in \Re^{n_u}$ is a given input vector, $p \in \Re^{n_p}$ is a vector of physical parameters, f_x is a mapping of n_x state equations and t denotes continuous time.

Assume that the model outputs are given by:

$$y(t) = c(x(t), u(t), b, t) \tag{7.2}$$

where $y \in \Re^{n_y}$ is the vector of model outputs, $b \in \Re^{n_b}$ is a vector of bias parameters and c is a mapping of n_y output equations. It is assumed that system (7.1) is observable through (7.2).

Assume that a sequence of nh recent output measurements is available: $\{y_m(t_0), y_m(t_0+T_s),\cdots, y_m(t_f)\}$, where time t_f is assumed to be present time. Assume also that the input variable $u(t)$ is known during the period $t \in [t\ , t_f]$. The moving horizon estimation problem is:

$$\min_{x_0,p,b} \quad J = \sum_{k=0}^{nh-1} L(y(t_0+kT_s), y_m(t_0+kT_s), b, k) \tag{7.3}$$

subject to:

$$\dot{x}(t) = f_x(x(t),u(t),p,t) \qquad t \in [t_0,t_f] \tag{7.4}$$

$$x(t_0) = x_0 \tag{7.5}$$

$$y(t) = c(x(t),u(t),b,t) \qquad t \in [t_0,t_f] \tag{7.6}$$

$$\psi(y(t),x(t),u(t),p,t) \le 0 \qquad t \in [t_0,t_f] \tag{7.7}$$

where $y_m \in \Re^{n_y}$ is the vector of measured outputs, k is a sampling index, L is a weighting function, ψ is a mapping of n_ψ inequality constraints, $t_f = t_0 + (nh-1)T_s$.

The purpose of the solution is to find the following estimates at present time: $\hat{y}(t_f), \hat{x}(t_f), \hat{b}(t_f)$ and $\hat{p}(t_f)$.

The weighting function L in a moving horizon estimation problem may be defined as follows:

$$L(y,y_m,b,k) = \frac{1}{2}(y-(y_m-Sb))^T V^{-1}(y-(y_m-Sb)) \tag{7.8}$$

where $S \in \Re^{n_y \times n_b}$ is a bias distribution matrix, $V \in \Re^{n_y \times n_y}$ is the covariance matrix of the measured variables y_m. Note that bias is not necessarily estimated in all measured variables.

In order to reduce the dynamic optimisation problem defined above to a nonlinear programming problem, it is necessary to discretise the continuous equations. This may be done using 4th order Runge Kutta steps. However, the integration step h will not necessarily be the same as the measurement sampling time T_s (it would be normal to expect that $h \leq T_s$). Assume that the integration time is chosen such that $T_s = n_i h$, where n_i is the number of integration steps per sampling period. Given that it is assumed that the input variable $u(t)$ is known during the period $t \in [t_0, t_f]$, then the following input sequence is also known: $\{u(t_0), u(t_0 + h), u(t_0 + 2h), \cdots, u(t_f)\}$.

Define the following decision vector:

$$X = \begin{bmatrix} x_0 \\ p \\ b \end{bmatrix} \tag{7.9}$$

where $X \in \Re^{n_x + n_p + n_b}$.

Define the following vector of inequality constraints:

$$\Psi = \begin{bmatrix} \psi(y(t_0), x(t_0), u(t_0), p, t_0) \\ \psi(y(t_0 + h), x(t_0 + h), u(t_0 + h), p, t_0 + h) \\ \psi(y(t_0 + 2h), x(t_0 + 2h), u(t_0 + 2h), p, t_0 + 2h) \\ \vdots \\ \psi(y(t_f), x(t_f), u(t_f), p, t_f) \end{bmatrix} \leq 0 \tag{7.10}$$

where $\Psi \in \Re^{(n_i+1)n_\psi}$.

Then the moving horizon estimation problem defined above can be reduced to the following nonlinear programming (NLP) problem:

$$\min_{X} \quad J(X) \tag{7.11}$$

subject to:

$$\Psi(X) \leq 0 \tag{7.12}$$

Notice that given the decision vector X and the input sequence $\{u(t_0), u(t_0+h), u(t_0+2h), \cdots, u(t_f)\}$ it is possible to integrate the model differential equation (7.4) to obtain the state sequence $\{x(t_0), x(t_0+h), x(t_0+2h), \cdots, x(t_f)\}$. With X and the state sequence it is possible to calculate, via the output equation (7.6), the output sequence $\{y(t_0), y(t_0+h), y(t_0+2h), \cdots, y(t_f)\}$. Given X, the measured output sequence $\{y_m(t_0), y_m(t_0+T_s), \cdots, y_m(t_f)\}$ and the computed output sequence $\{y(t_0), y(t_0+T_s), \cdots, y(t_f)\}$, it is possible to compute $J(X)$. Thus given X it is possible to compute $J(X)$ and $\Psi(X)$.

The solution to the above nonlinear programming problem can be obtained using a standard SQP algorithm, see appendix A. Furthermore, given that the objective $J(X)$ is often chosen to be a sum of quadratic functions such as (7.8), then a nonlinear least squares algorithm is probably a good choice for the solution.

7.4 GROSS ERROR DETECTION AND IDENTIFICATION

Gross errors are usually caused by non-random events where the measurement bears little or no relation to the true measurement value. Gross errors can be subdivided into *measurement-related* errors such as malfunctioning sensors and *process related* such as process leaks. Although a number of researchers regard measurement bias as a type of gross error, it will be regarded as a different type of error which will be treated separately in this text.

The techniques used to process measurement data have been identified in chapter 4 (Figure 4.1) as three distinct steps: (1) variable classification: (2) gross error detection; and (3) coaptation and data reconciliation. The problem of gross error detection has received considerable attention from researchers in the field. This undoubtedly stems from the importance attached to this problem in the context of successful data reconciliation.

In general, data reconciliation schemes assume that the error is normally distributed. A gross error severely violates that assumption. It is therefore paramount that gross errors are identified and removed from the data prior to (or simultaneously with) the data reconciliation step. If this is not done, the resulting variable estimates would contain significant errors, with the entering gross error accounted for in some or perhaps all of the estimates. This effect is referred to as *smearing* (Liebman, 1991). In practice, the gross error detection and the data reconciliation steps are often used iteratively. Data reconciliation is applied first. Then, the resulting residuals between the measurements and the estimates are analysed for gross errors. If a gross error is suspected, appropriate adjustments are made and the data reconciliation step is repeated (Liebman et al., 1992).

A number of methods for gross error detection have been developed. The main approaches are listed below:

1. Classical Hypothesis testing
   (a) The Global Test (1963, 1975).
   (b) The Nodal Test (NT) also known as the constraint test (1963, 1976).
   (c) The Measurement Test (MT), (1982).
2. Generalized Likelihood Ratio (GLR), (1987).
3. Bayesian Approach (1987).
4. Unbiased Estimation Technique (UBET), (1992).
5. Principal Component Test (PCT), (1994).
6. Dynamic Integral Measurement Test (DIMT), (1997).

The Global Test was first published by Almasy and Sztano (1975) although Reilly and Carpani had previously presented the method at a conference in 1963. The Nodal Test was published by Mah et al. (1976) but again this had previously been presented by Reilly and Carpani at the same conference. The Measurement Test (MT) was developed by Mah and Tamhane (1982). The main drawback of the Global Test and the Nodal Test is that they both require a separate gross error identification strategy following the detection of such errors. The Measurement Test does not require a separate identification strategy but does require that process data reconciliation be carried out. Romagnoli and Stephanopoulos (1981) published a method to detect and rectify gross errors based on the serial deletion of one or more observations from the set of measurements. This is useful for the detection and identification of multiple gross errors (Narasimhan and Mah, 1987).

Serth and Heenan (1986) proposed seven tests including their Iterative Measurement Test (IMT), the Modified IMT (MIMT) and the Screened Combinatorial method (SC) and compared their performance to those of the Measurement Test (MT) and the Nodal Test. For comparison they applied these methods to a simulated industrial steam-metering system. While reporting that the MT and the NT performed poorly in this particular application, Serth and Heenan concluded that a useful approach to gross error detection is to use a combination of the different methods so as to exploit the strengths of each.

Narasimhan and Mah (1987) developed the Generalized Likelihood Ratio (GLR) test for identifying and estimating gross errors. They applied their method to a steady-state process and rather than using serial elimination they use serial compensation. A major feature of the GLR method is that it can differentiate between different types of errors i.e. whether outliers or systematic biases.

Rosenberg et al. (1987) developed two composite tests: the Dynamic Measurement Test (DMT) and the Extended Measurement Test (EMT) for the detection of gross errors. They compared the performance of these composite

tests to those of the Global Test, and the MT. The authors conclude that the performance of DMT and EMT is more superior when compared to the MT. The word *'dynamic'* in DMT does not imply the possible application of this method to dynamic data - it only refers to the fact that the test itself is dynamic in that the candidate set is not fixed. The candidate set is enlarged at each step until all suspected measurements are included.

Narasimhan and Mah (1988) applied their GLR method (Narasimhan and Mah 1987) to a dynamic process. Narasimhan and Mah (1989) considered cases where process variables are not measured directly and in which unmeasured variables are present in the constraints.

Kao et al. (1992) proposed a composite test procedure for detecting and identifying gross errors in dynamic systems. This was an extension to the technique they had proposed two years previously for the steady-state case and makes use of the MT. Harikumar and Narasimhan (1993) proposed two methods for gross error detection that make use of their results (Narasimhan and Harikumar, 1993) on incorporating bounds. One of the methods makes use of bounded information while the other uses the GLR method. Yang et al. (1995) presented a method that uses a combination of the Iterative Measurement Test (IMT) and the Nodal Test. Albuquerque and Biegler (1996) presented a study on data reconciliation and gross error detection for dynamic systems. In particular, they developed a method for variable classification and concluded that this was closely linked to gross error detection.

Tong and Crowe (1996) presented a sequential principal component test for gross error detection by combining principal component analysis and sequential analysis. Kim et al. (1997) improved Serth and Heenan's MIMT gross error detection algorithm by using Non-Linear Programming (NLP) techniques making the scheme applicable to highly nonlinear processes at steady-state. Chen and Romagnoli (1998) developed a method that carries out simultaneous dynamic data reconciliation and gross error detection based on a combination of cluster analysis

techniques and dynamic optimisation. Similarly, Chen et al. (1998b) published an integrated method for outlier detection and data reconciliation using quantile probability plots (Johnson and Wichern, 1992).

Bagajewicz and Jiang (1998) presented a method applicable to linear dynamic processes. This was an extension to their integral dynamic data reconciliation method (Bagajewicz and Jiang, 1997) to allow multiple gross error estimation. Finally, Sanchez et al. (1999) proposed a method for the simultaneous detection of outliers and systematic bias but only applicable to steady-state processes.

A number of good review papers are available on the subject (Mah, 1982; Mah, 1987; Crowe, 1996). There are also good review sections in other papers (Romagnoli and Stephanopoulos, 1981; Crowe et al., 1983; Serth and Heenan, 1986; Narasimhan and Mah, 1988; and Bagajewicz and Jiang, 1988).

As can be clearly seen from the literature, considerable effort has been expended on developing methods for gross error identification in steady-state chemical processes. But, as mentioned previously, a 'steady-state' process is constantly undergoing variations about a nominal steady-state which means that it is never truly in steady-state. Therefore, a dynamic process model is a better representation of the real process (Narasimhan and Mah, 1988).

Only a handful of researchers have addressed the problem of gross error detection in dynamic process data. The method proposed by Chen and Romagnoli (1998) which is based on the moving horizon concept is adopted in this work with a slight modification.

7.4.1 Gross error detection and identification algorithm

By making use of cluster analysis techniques, Chen and Romagnoli (1998) propose a method which successfully distinguishes outliers from normal data.

They use a clustering technique proposed by Yin and Chen (1994) in which each object is assigned to the cluster of its nearest neighbour within a certain distance. The method is straightforward, the formulation of which is reproduced here.

Given a set of nh objects $y_1, y_2, \cdots, y_{nh}$, in a d dimensional space which refers to the number of measurement variables, the Mean Minimum Distance (MMD) is defined as:

$$MMD = \frac{1}{nh}\sum_{i=1}^{nh} \min_{j=i}\left[\left(\sum_{k=1}^{d}\left(y_{ik} - y_{jk}\right)^2\right)^{\frac{1}{2}}\right] \tag{7.13}$$

Because in practice the variations of individual measurements may be different, it is necessary to weight each variable by its own variance. If this is not done, the result may be that some outliers might end up hidden within a smoother variable containing normal variations of noisy variables. Thus equation (7.13) should be rewritten as:

$$MMD = \frac{1}{nh}\sum_{i=1}^{nh} \min_{j=i}\left[\left(\sum_{k=1}^{d}\frac{\left(y_{ik} - y_{jk}\right)^2}{v_k}\right)^{\frac{1}{2}}\right] \tag{7.14}$$

where v_k is the kth diagonal element of the covariance matrix V.

In order to incorporate outlier information into the data reconciliation procedure, the objective function of dynamic data reconciliation is modified as

$$J = \sum \frac{1}{2}\{W_i[y_i - \bar{y}_i]\}^T V^{-1}\{W_i[y_i - \bar{y}_i]\} \tag{7.15}$$

where $\bar{y}_i$ is the estimate of the measurement y_i, W_i is the trust degree of y_i and is defined as:

$$W_i = \begin{cases} 1 & if\ DIST_i \le 2 * MMD \\ \dfrac{2 * MMD}{DIST_i} & if\ DIST_i > 2 * MMD \end{cases} \qquad (7.16)$$

Chen and Romagnoli define $DIST_i$ as being equal to the minimum distance between measurement y_i and any other measurements in the moving window. This is where this algorithm is slightly modified. The definition of $DIST_i$ is modified to being the distance from measurement y_i to the mean of all measurements $y_1, y_2, \cdots y_{nh}$ in the data window.

The reason for this modification is intuitive. Consider the situation where measurement y_i is an outlier and assume that in the present data window a measurement y_{i-k} has the same value or a value close to y_i. Using Chen and Romagnoli's definition of $DIST_i$, the algorithm would fail to detect this outlier. However, using the modified version which uses the mean of all the measurements in the data window as a basis for calculating $DIST_i$, the outlier y_i would be detected and the appropriate weight W_i will be used.

7.5 DETECTION AND IDENTIFICATION OF SYSTEMATIC BIAS

Systematic biases occur when measurement devices provide consistently erroneous values, either too high or too low, and may be caused by incorrect installation or calibration of the measurement systems. It is important that data containing such bias is identified and either treated or removed prior to the data reconciliation stage. If the measurements are adjusted in the presence of such biases, all of the adjustments will be greatly affected by them and would not be reliable indicators of the true state of the process.

As mentioned previously, scanning the literature reveals that a number of researchers regard systematic biases as being a type of gross error (Chen and Romagnoli, 1998). Therefore some have included the treatment for systematic

biases within their gross error detection algorithms (Narasimhan and Mah, 1987; Keller et al., 1994). Others have opted to treat systematic biases as separate from gross errors (outliers), (McBrayer and Edgar, 1995). This is the approach taken in this thesis.

Surprisingly few researchers have explicitly addressed the problem of identification of systematic bias. Most of the limited previous work has focused on steady-state processes. Narasimhan and Mah (1987) applied their Generalized Likelihood Ratio (GLR) method to measurement bias and noted that in this case the GLR test reduces to the Measurement Test (MT), (Mah and Tamhane, 1982). They also pointed out that simulation studies by Rosenberg et al. (1987) indicated that methods based on the MT gave the best performance for identifying measurement bias. Rollins and Davis (1992, 1993) and Keller et al. (1994) worked with linear systems. Rollins and Davis (1992) presented equations that help identify biased measurements and process leaks. They named this the Unbiased Estimation Technique (UBET) and looked at the linear steady-state case. The basic goal of UBET is to find unbiased estimates for process variables when gross errors in the measurements exist. Further to their UBET technique presented in 1992, Rollins and Davis (1993) looked at the issue of unknown variances and covariances. Keller et al. (1994) proposed a method for detection, identification and estimation of gross errors in linear steady-state processes. The technique improves the GLR test (Narasimhan and Mah, 1987) for the case when several gross errors appear simultaneously and is applicable to both gross errors and systematic biases.

McBrayer and Edgar (1995) developed a method to detect and estimate bias in nonlinear dynamic processes. The technique uses the model based Nonlinear Dynamic Data Reconciliation (NDDR) method developed by Liebman (1991) and requires the examination of the resulting difference between the measured and reconciled values. Bagajewicz and Jiang (1997) proposed a method to detect bias in the context of linear dynamic systems. Sanchez et al. (1999) published a technique that simultaneously detects systematic bias and outliers. However, as

mentioned in the previous section, this method is only applicable to steady-state processes.

It can be noted that like the gross error detection problem, bias detection in the dynamic case has received limited attention from researchers in the field. In trying to implement a simultaneous strategy to deal with gross error and bias detection for dynamic data, the author's initial intention was to develop an algorithm which combines the work of Chen and Romagnoli (1998) with that of McBrayer and Edgar (1995). This particular formation was indicated as further work to be carried out by Chen and Romagnoli. However, in attempting to implement the said strategy, although inspired by McBrayer and Edgar, a simpler method for detecting and identifying bias has been found to give good results and is presented here.

7.5.1 Bias detection and identification algorithm

Following the removal of gross errors, the typical assumptions made in data reconciliation are that the measurement errors are independent, zero-mean and normally distributed (McBrayer and Edgar, 1995). These assumptions give rise to the following measurement model:

$$y_{mi} = y_{i,true} + \varepsilon_i \tag{7.17}$$

where ε_i is the random error. The residuals $(y_{mi} - \bar{y}_i)$, where $\bar{y}_i$ is the ith estimate and y_{mi} is the ith measurement, will be randomly distributed with zero-mean when this measurement model holds. However, in the presence of bias in the measurements the measurement model becomes:

$$y_{mi} = y_{i,true} + \varepsilon_i + b_i \tag{7.18}$$

and the residuals are no longer zero-mean. The mean of the residuals is ideally equal to b_i in this case. But this unfortunately is not the case because of the fact that the bias is smeared over the estimates during reconciliation.

A bias detection method based on the moving horizon strategy has been presented by McBrayer and Edgar (1995). The algorithm is not very intuitive and involves the calculation of some base statistics. These serve as a *'base case'* with which statistics from the actual data can be compared. The base statistics are calculated using base case data generated by adding Gaussian noise to the calculated estimates. To determine whether or not a bias is present, the residuals are examined. Under the null hypothesis, equation (7.17) is the correct measurement model for all y_{mi} and the following two equations must be true:

$$\sum_{j=1}^{n} e_{i,j} * \bar{y}_{i,j} = 0 \tag{7.19}$$

and

$$\hat{e}_i = 0 \qquad \forall y_{mi} \tag{7.20}$$

where $e_i = (y_{mi} - \bar{y}_i)$ is the residual and equation (7.20) means that the linear relationship between e_i and y_i is a horizontal line with intercept of zero. Suffix j refers to the measured variable while suffix i refers to the instance of variable j in the data window.

The method used by McBrayer and Edgar is summarised in Figure (7-1). In the process of implementing this algorithm, two possibly simpler algorithms were devised. The first is similar to McBrayer and Edgar's and is shown in Figure (7-2). In step 4, instead of estimating the bias as a free parameter again, it is calculated using the present data window. Furthermore, unlike the original method, once the biased measurement is identified, the bias estimate is not subtracted from the measurements prior to reconciling the data again. The bias estimate is taken into account when recalculating the estimates during the next

reconciliation step. The bias $\hat{b}$ and the standard deviation σ of the bias estimates in step 4 are calculated as follows:

$$\hat{b} = \frac{1}{nh}\sum_{i=1}^{nh}\left[(y_{mi} - \bar{y}_i)^2\right]^{\frac{1}{2}} \tag{7.21}$$

$$\sigma(\hat{b}) = \left[\frac{1}{nw-1}\sum_{s=1}^{nw}\left(\hat{b}_s - \bar{\hat{b}}\right)^2\right]^{\frac{1}{2}} \tag{7.22}$$

where nw is the number of data windows, $\bar{\hat{b}}$ is the average bias over the data windows $s = 1 \ldots nw$. In order to get a value of $\sigma(\hat{b})$ this way, a number of data windows are required. One way to overcome this and get a value of $\sigma(\hat{b})$ using one data window is to use the following equation:

$$\sigma(\hat{b}) = \sigma(y_{mi} - \bar{y}_i), \tag{7.23}$$

since $\hat{b}_i = y_{mi} - \bar{y}_i$.

This method still requires the use of base case data and statistics.

A further method developed and actually applied in this work is very intuitive and easy to implement. It works by simply assuming one of the measurements to be biased. The appropriate flags are set in order for bias on that particular variable to be estimated as a free parameter. Then the bias estimate of that variable is analysed and checked in two simple ways. The first is a check on the magnitude of the bias which is compared against a pre-set threshold value. The second test checks the bias against the standard deviation of the measurements in the current data window. Again this is checked against a preset threshold value. In order for the algorithm to flag a possible presence of bias on that particular measurement, the results from both tests must exceed their respective threshold values. If the chosen measurement is deemed to be free of bias, a different measurement is chosen and assumed to be biased and the procedure is repeated. This is done

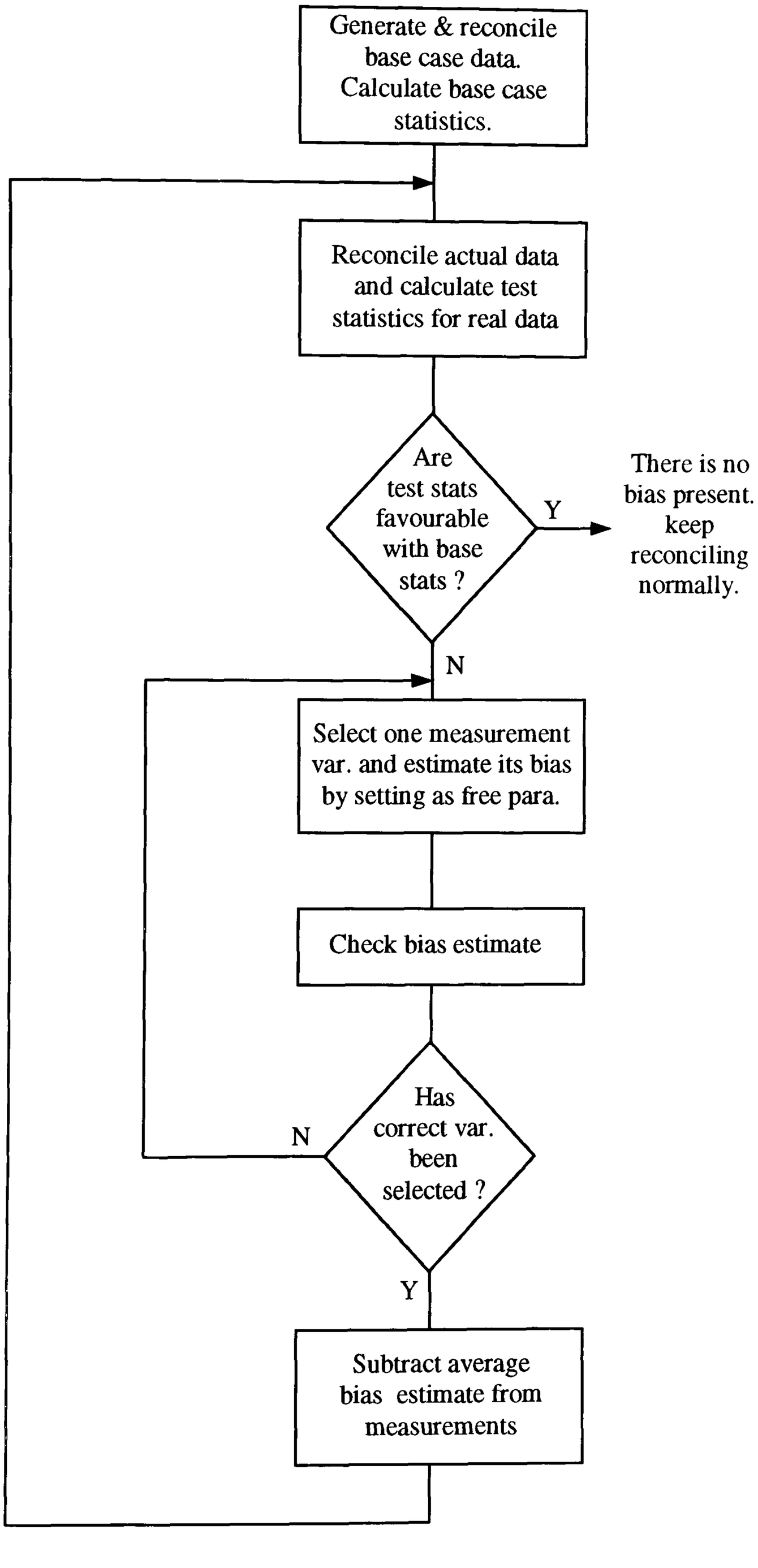


Figure (7-1): Bias detection method 1 (McBrayer and Edgar, 1995)

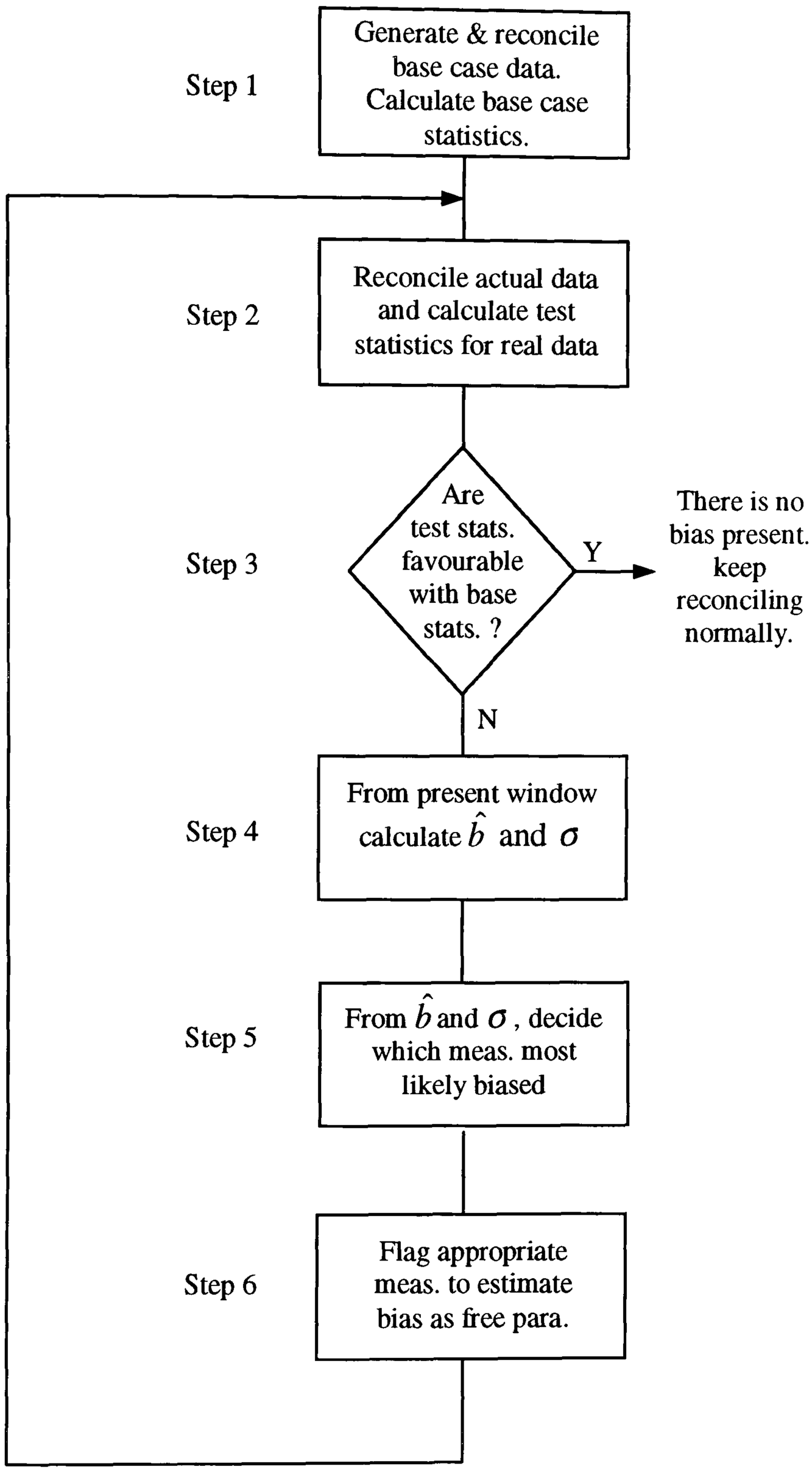


Figure (7-2): Bias detection method 2, modified version of McBrayer and Edgar.

sequentially for all the measurement variables until the biased individual (if any) is found. In other words, if the algorithm has not found a biased individual it will assume a different measurement to be biased each time the reconciliation procedure is run. The algorithm is summarised in Figure (7-3).

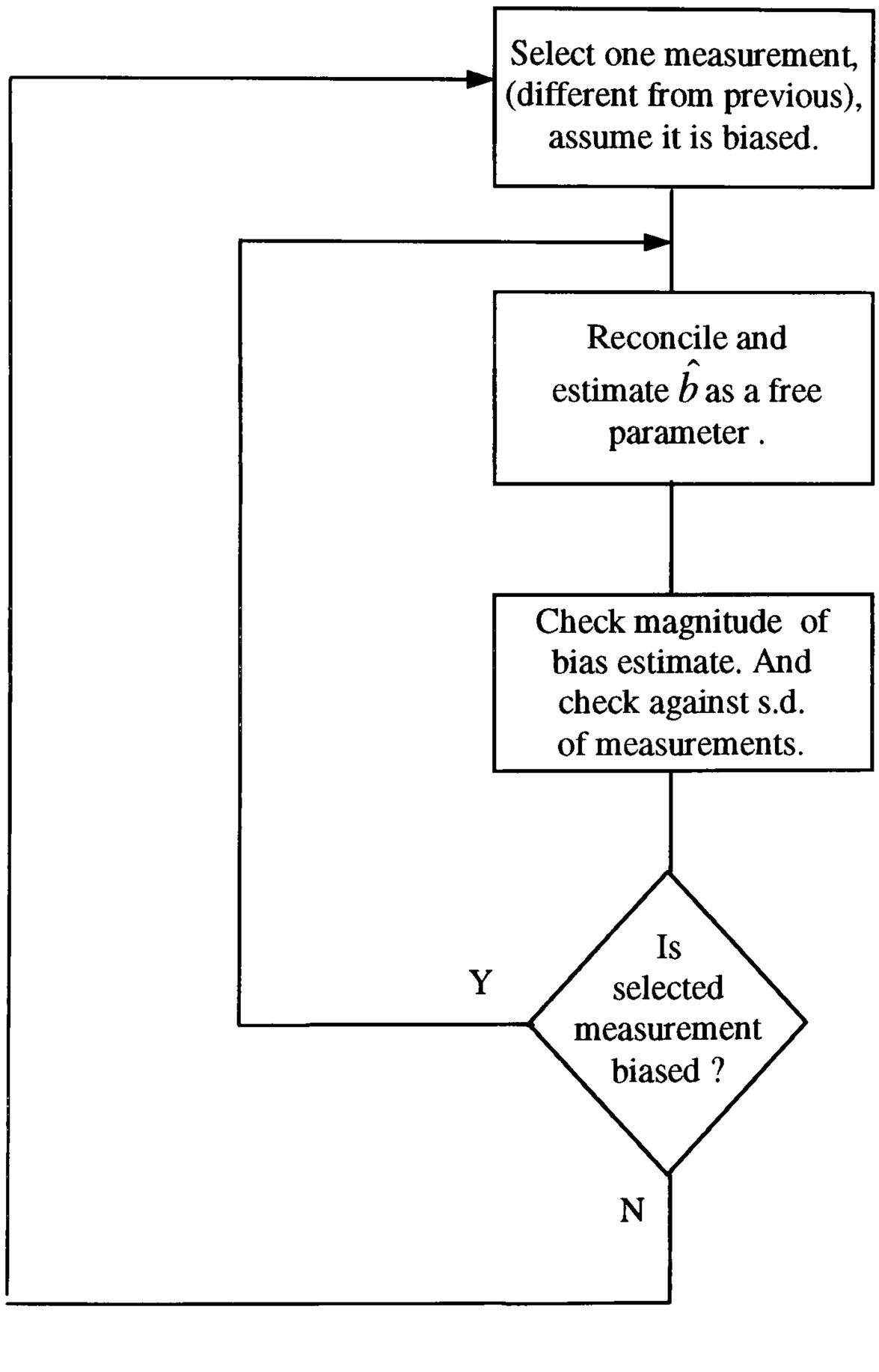


Figure (7-3): A new bias detection algorithm.

7.6 SIMULATION CASE STUDY

A dynamic data reconciliation algorithm using the moving horizon estimator has been implemented using C++ code and interfaced with the process simulation software OTISS (Becerra, 1999). Following the testing of this new module, separate algorithms for the detection of bias and gross errors have also been implemented and tested. This collection of dynamic data reconciliation modules has been applied to the dynamic model of two continuous stirred tank reactors introduced in Chapter 4 (Figure 4-1), full details of which can be found in Garcia and Morari (1981).

7.6.1 Implementation Issues

The algorithm's capabilities in identifying and estimating bias, and identifying and eliminating the effects of gross errors have been tested individually and then in combination. To do this three separate case studies have been set up. The first case study was set up to test the gross error detection algorithm. In order to properly observe the effects of gross error detection, the algorithm's built in bias estimation capability was disabled. The second case study was set up to examine the algorithm's bias detection capabilities and, for this purpose, the gross error detection capability was this time disabled. Finally, the third case study was set up to observe the behaviour of the algorithm in the presence of both gross errors and systematic bias with both gross error and bias detection functions enabled.

In order to simulate the effect of a gross error, a bias was added to the measurement in question for a short period of time and then removed. To simulate the presence of dynamic data, transients were introduced by step changes in T_1, the temperature in the first reactor. The bias detection algorithm works by applying two simple tests: the first tests the magnitude of the estimated bias and the second tests the bias against the standard deviation of the measurements. For a selected measurement to be suspected of bias, results from both tests must

exceed some pre-selected threshold values. A small absolute value of 0.00050 was selected for the first test while the second test is specifically:

$$\text{if} \quad \frac{bias}{\sigma} < 5.0$$

where σ is the standard deviation of the measurements,

then it is unlikely that bias is present on this particular measurement. The selection of the value 5.0 was inspired by the simulation results obtained by McBrayer and Edgar (1995).

The measured variables are assumed to be C_{b1} and C_{b2}, the concentrations of species B in the first and second tank respectively. While the unmeasured variables are assumed to be C_{a1} and C_{a2}, the concentrations of species A in the first and second tank respectively. The tuning parameters used for the moving horizon scheme were: data window length $nh = 15$, integration step $= 10\,s$ and the covariance matrix $V = diag(0.5, 0.5)$.

7.6.2 Results

Simulation results shown in Figures (7-4) to (7-19) correspond to the three case studies outlined above. The results from these different case studies are analysed separately below.

7.6.2.1 Gross error detection and identification

In order to appreciate the benefits of the gross error detection module, simulations have been carried out to compare the behaviour of the moving horizon estimator when the gross error detection module is enabled against when it is disabled. Figure (7-4) shows the behaviour of the moving horizon estimator in a steady-state case when the gross error detection module is disabled. A large outlier of

magnitude 0.00585 (approx. 11% of the nominal value) is present on the first measurement C_{b1}. The effect of the outlier is quite significant on the estimate of C_{b1}. The second plot in Figure (7-4) shows the measurement C_{b2}, its estimate and the true value. The 3rd and 4th plots of the same figure show the estimates of C_{a1} and C_{a2} respectively and the true values of each. Figure (7-4) should be compared to Figure (7-5) which shows the case when the gross error detection module is enabled. The outlier present is the same as in Figure (7-4). It can be observed in Figure (7-5) that as a result of the gross error detection algorithm, the estimates are more accurate and less affected by the presence of the large gross error. The first plot in Figure (7-5) shows that although the gross error is not completely removed, the algorithm does actually recognise the presence of the outlier and, through weights on the objective function, tries to limit its effect. Examination of the other three plots in the figure confirm the improvement which also translates to the variables C_{b2}, C_{a1} and C_{a2}.

Figure (7-6) shows the case when an outlier of magnitude 0.0025 (approx. 5% of the nominal value) is present on the second measurement C_{b2} with the gross error detection capability disabled. This should be compared to Figure (7-7) which shows the case when the same outlier is added but when the gross error detection capabilities are this time enabled. Again all the estimates in Figure (7-7) are more accurate than those in Figure (7-6) and the outlier has been largely eliminated.

Figure (7-8) shows the case when a large outlier of magnitude 0.007 (approx. 14% of the nominal value) is present on C_{b1} and an outlier of magnitude 0.003 (approx. 5% of the nominal value) is present on C_{b2} with the gross error detection capability disabled. This should be compared to Figure (7-9) which shows the case when the same outliers are added but when the gross error detection capability is enabled. Once again all the estimates in Figure (7-9) are more accurate than those in Figure (7-8) and the effects of the outliers have been considerably reduced.

Figure (7-10) shows the dynamic case when 3 outliers are present on C_{b1} and the gross error detection capability is disabled. This should be compared with Figure (7-11) when the same outliers are present but when the gross error detection module is enabled. The benefits of the gross error detection algorithm can again be appreciated in all the estimates. In the case of the unmeasured variables C_{a1} and C_{a2}, the estimator seems to accurately predict the final steady-state value following the step input instead of providing reasonable estimates along the transient.

7.6.2.2 Bias detection and identification

A number of simulations were carried out to test the bias detection module. The gross error detection algorithm was disabled throughout this set of simulations so that the results are not affected by it. Figure (7-12) shows a steady-state case where systematic bias of magnitude 0.0129 (approx. 25% of nominal value) is present on C_{b1} for the first 2 hours. It can be observed that throughout this period the algorithm correctly estimates all the measured and unmeasured variables. The sharp kick towards the beginning of the simulation is due to the initialisation of the estimator and the bias detection algorithm.

After two hours the bias on C_{b1} is completely removed and this state remains until shortly before 4 hours when bias on C_{b2} of magnitude 0.00585 (approx. 10% of nominal value) is added. Throughout the simulation all the variables are estimated correctly. At the transition periods when bias is either added or removed, there can be observed a severe temporary kick in almost all the estimates. This is due to the way the bias detection algorithm works. It takes the algorithm a short while to correctly find out where the bias actually is. In the meantime, because it is jumping from one variable to another with the assumption that that particular variable is biased, it produces erroneous estimates.

Figure (7-13) shows the dynamic case when systematic bias is present on C_{b1} during a transient. Again, it can be seen that the measured variables are estimated correctly. However, instead of providing estimates along the transient. the algorithm seems to predict the final steady-state value as soon as the transient enters in the case of the unmeasured variables C_{a1} and C_{a2}. Figure (7-14) shows the case when bias is present on C_{b2} which is then removed just before 4 hours. Once again, a similar behaviour in the estimates is seen and the algorithm is correctly identifying the biased measurement.

In Figure (7-15) bias is present on C_{b1} until just before 4 hours when it is removed. During this time a transient is simulated by a step change in the temperature T_1 (T_1 is changed from 307 to 310 K). Shortly after 5 hours, bias is added to C_{b2}. Just before 7 hours, another transient is simulated by a step change in T_1 (T_1 is changed from 310 to 307 K). Finally, at approximately 8 hours, the bias on C_{b2} is removed. Once again, it can be seen that the algorithm is correctly identifying the biased measurement and providing correct estimates of all the variables except during the transient for the case of unmeasured variables when the estimator provides the final steady-state value. It must be noted that the bias identification algorithm tends to become less accurate when the bias is switched repeatedly from one measurement to another. In one simulation where this was done, the algorithm failed to correctly identify the biased measurement.

7.6.2.3 Combined gross error and bias detection and identification

The final set of simulations combine the gross error and bias detection work. For the sake of comparison, simulations were carried out with both detection modules disabled in the first instance and then with both of them enabled. It is important to distinguish here between bias identification and bias estimation. The moving horizon estimator is capable of estimating bias provided it is informed in advance which measurements are biased. It is not, however, capable of bias identification without the use of the bias identification function. In the simulations in which the

bias detection capability was disabled, the bias estimation capability was kept enabled. Figures (7-16) and (7-17) show the case when bias and a number of outliers are present on C_{b1}. Figure (7-16) shows the case when both detection algorithms are disabled. Although following the correct trend, the estimates of C_{b1}, C_{a1} and C_{a2} are extremely inaccurate. However, when the detection algorithms are enabled, Figure (7-17), the estimates are greatly improved.

Figures (7-18) and (7-19) show the case when bias and a number of outliers are present on C_{b2}. Figure (7-18) shows the case when both detection algorithms are disabled. The estimate of C_{b1} can be seen to be affected by the outliers. The estimates of C_{b2} and C_{a2} are completely incorrect and in fact the C_{b2} estimate religiously follows the biased measurement. Figure (7-19), on the other hand, shows the case when both detection algorithms are enabled. The difference can be clearly appreciated in terms of the accuracy of all the variable estimates. The outliers on C_{b1} seem to be largely eliminated and the algorithm correctly identifies the presence of bias on C_{b2} and estimates it appropriately. Finally, the estimates of the unmeasured variables are accurate except during the transient where the estimator predicts the final steady-state values instead of providing estimates along the transient.

7.7 CONCLUSIONS

Here a dynamic model of the process has been used unlike the Static Data Reconciliation (SDR) case in chapter 4 where a static model was used for the reconciliation. This has the advantage that changes in the transient periods will not result in differences between the static model and the real process and therefore, more accurate estimates can be expected.

The gross error detection method proposed by Chen and Romagnoli (1998) and modified here has been shown to work in successfully identifying and eliminating outliers. For comparison, the original algorithm as suggested by Chen and Romagnoli was implemented and tested. Simulation results from this

implementation, although not presented here, have shown that the modified version of the algorithm produces considerably better results in terms of accurate identification of outliers. Although the effect of the outliers is always considerably reduced, it is not always completely eliminated even with the modified version of the algorithm.

The bias detection algorithm proposed here has been successfully implemented and has been shown to work effectively in identifying the biased measurement. Thus unlike the SDR algorithm, the moving horizon estimator with the bias detection capability enabled does not need *a priori* information on which measurement is biased. The method proposed and implemented here is intuitive and far simpler than that put forward by McBrayer and Edgar (1995). A further algorithm, which is essentially a simplified version of the method by McBrayer and Edgar has also been proposed. It may be useful as further work to implement this for the purpose of comparison.

In most of the simulations involving the bias identification algorithm, a spike can be observed in most of the estimates towards the beginning of the simulation (see for instance Figure 7-12). It may be argued that this problem might be avoided if instead of testing just one measurement for bias at each sampling time, the algorithm tests all the measurements. An algorithm was implemented specifically for this purpose, the results from which showed that the spikes could not be eliminated. Therefore the original algorithm, where at each sampling instant only one measurement is tested for bias, was used since the extra computational effort was deemed unnecessary.

An explanation for the above is that during the first 15 minutes (equal to the data window), the DDR algorithm is accumulating data. The bias detection algorithm however is working from the beginning testing each measurement in turn for bias. Once the period of data accumulation has passed the DDR algorithm kicks in. Up to this point, the bias detection algorithm will show that there is no bias on any of the measurements since the bias estimates from the DDR algorithm will still be

zero. Based on this assumption, the first set of estimates from DDR will be erroneous (see equation 7.8). In the next sampling instant when the DDR algorithm will properly estimate any bias present, the bias detection algorithm will detect the presence of a bias by making the DDR algorithm estimate the bias on all the measurements.

The above problem is not of great importance since this happens at initialisation and the algorithm quickly recovers giving correct estimates. It also occurs when there is a significant change in bias e.g. if the bias is switched from one measurement to another. A similar explanation as that given above can be used to argue the reason for this. Simulations have shown that repeated switching of the systematic bias from one measurement to another causes the detection algorithm to become inaccurate at some stage. Fortunately, in real plants, the bias is unlikely to keep moving from one measurement to another since bias is normally caused by incorrectly installed or calibrated measurement devices. What is likely to occur though, is the presence of multiple biases. The algorithm as it is can only handle single biases at a time. In fact the method proposed by McBrayer and Edgar (1995) suffers from the same problem. A possible extension of the algorithm is thus the handling of multiple biases.

The bias and gross error detection algorithms have been successfully combined producing very good results. The results have been compared with the case where these algorithms are disabled which highlights the benefits of having such algorithms within the dynamic data reconciliation scheme. The idea of combining bias and gross error detection was put forward by Chen and Romagnoli (1998) who suggested combining their outlier detection method with the bias detection algorithm by McBrayer and Edgar (1995). However, a personal communication with the former authors revealed that this had not been done.

It can be noted that during some transient simulations the estimator seems to behave as a predictor in the case of the unmeasured variables (see for instance Figures 7-13 and 7-14). Since the model used for the estimation is a dynamic one,

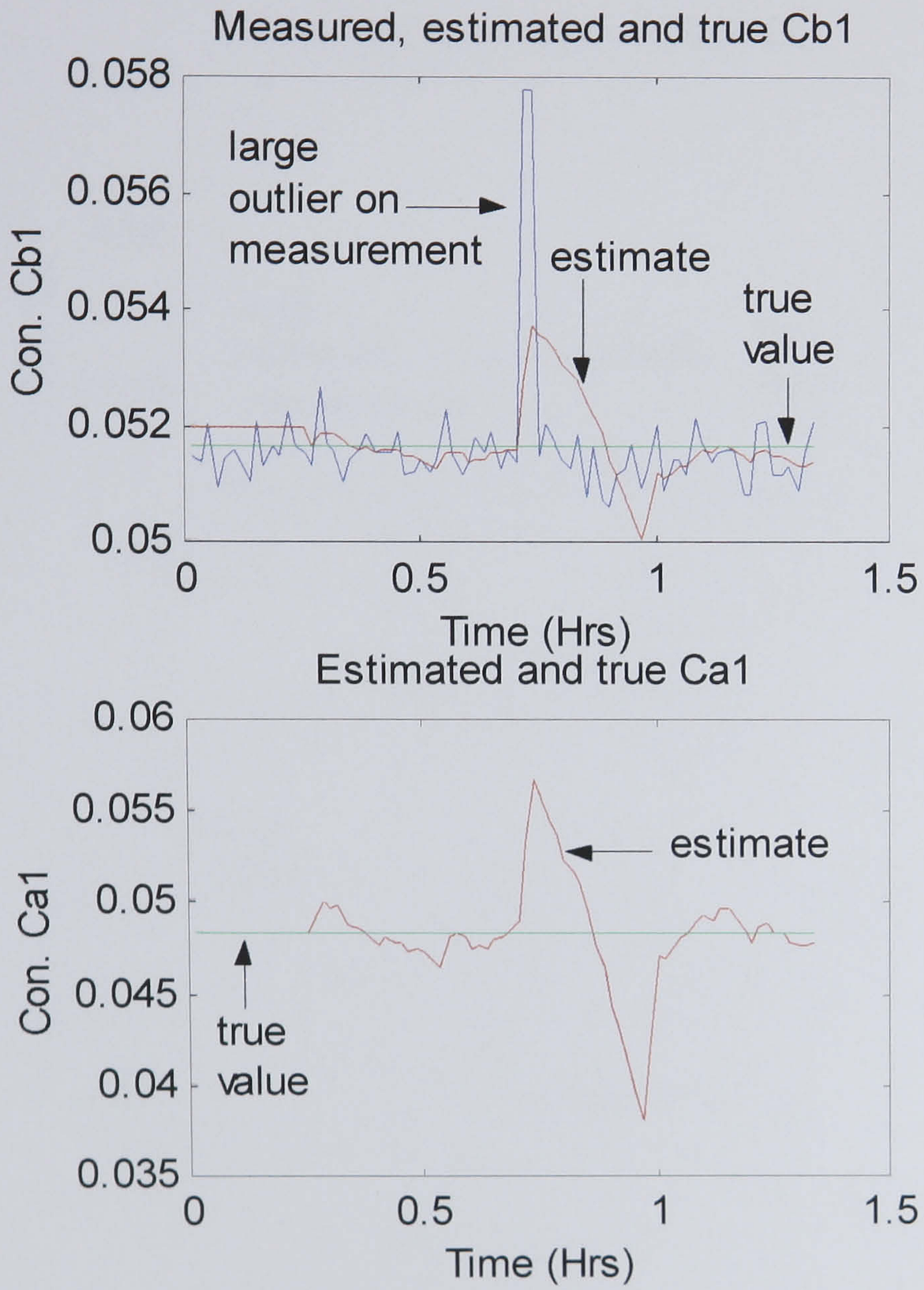
the estimator should give reasonable results along the transient similar to the actual outputs of the process. It is not know exactly why this phenomenon occurs but it appears to happen only when estimating the unmeasured variables.

7.8 SUMMARY

In this chapter, a moving horizon estimator has been used to reconcile dynamic data generated from the continuous stirred tank reactor system. This chapter is thus a natural extension to chapter 4 in which a static data reconciliation algorithm was developed and applied to the same process. The benefits of using a dynamic model for dynamic data reconciliation have been highlighted in this chapter.

Extensions to the moving horizon estimator have been made such that it is able to detect bias and gross errors simultaneously using two algorithms which are relatively easy to implement. These algorithms have been shown to work well separately and in combination through a number of simulations. Two new possible algorithms for bias detection and identification have been proposed. One has been implemented in this work while the other has been suggested for further research due to the limited time available.

In chapter 8, the dynamic data reconciliation tools developed in this chapter are used in conjunction with a model predictive control scheme.



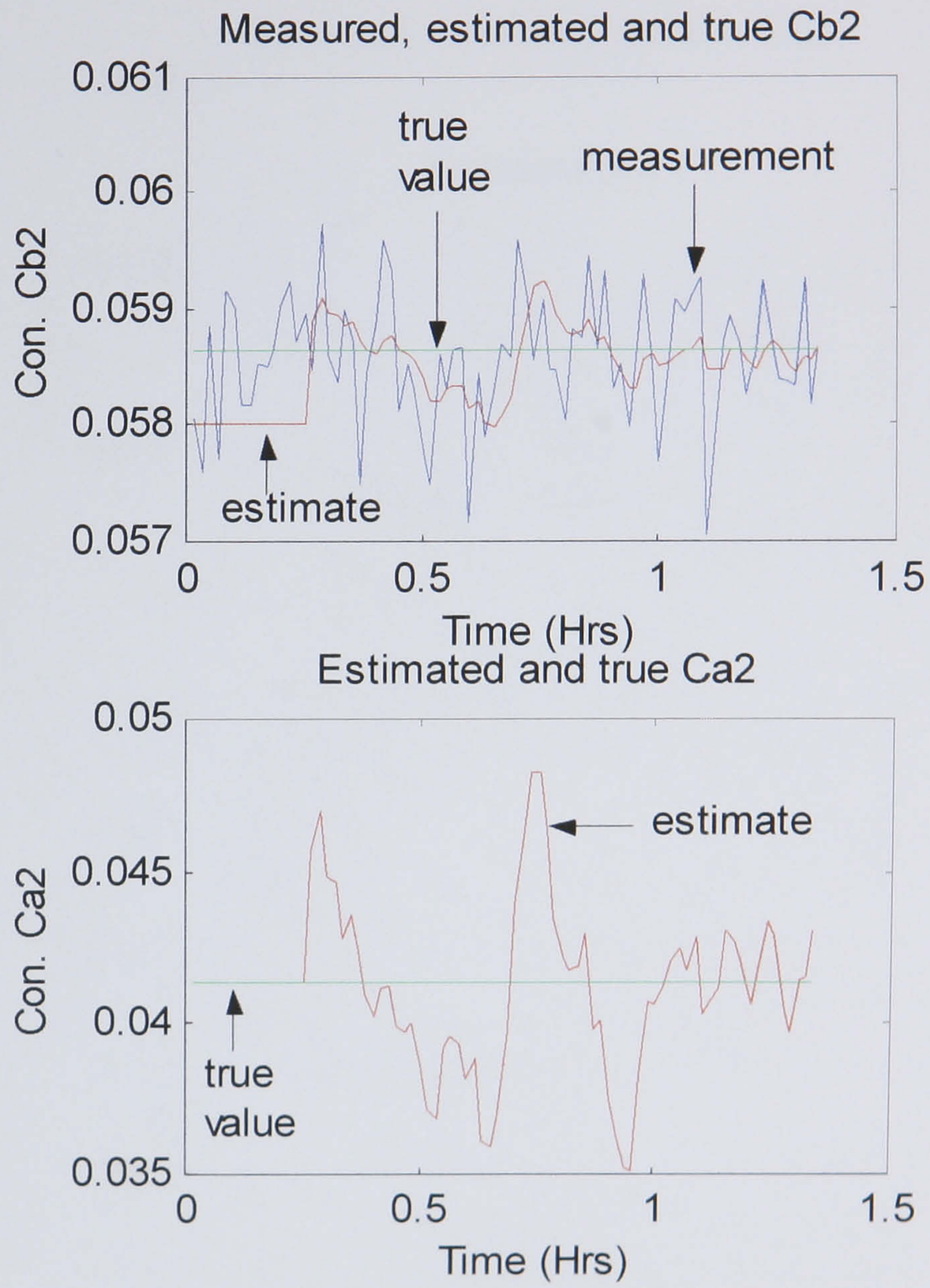


Figure (7-4): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with an outlier on C_{b1} when the gross error detection algorithm is disabled.

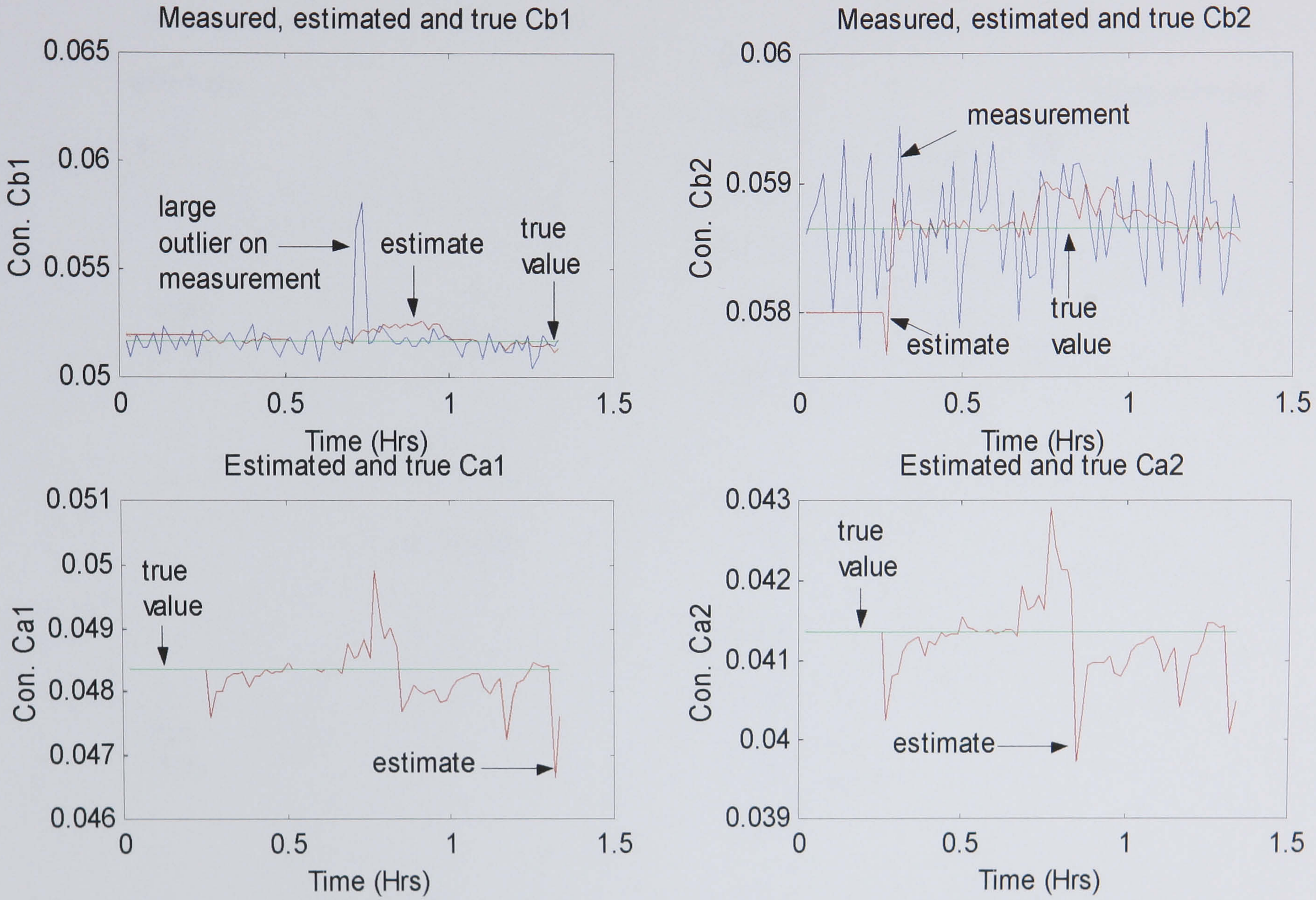
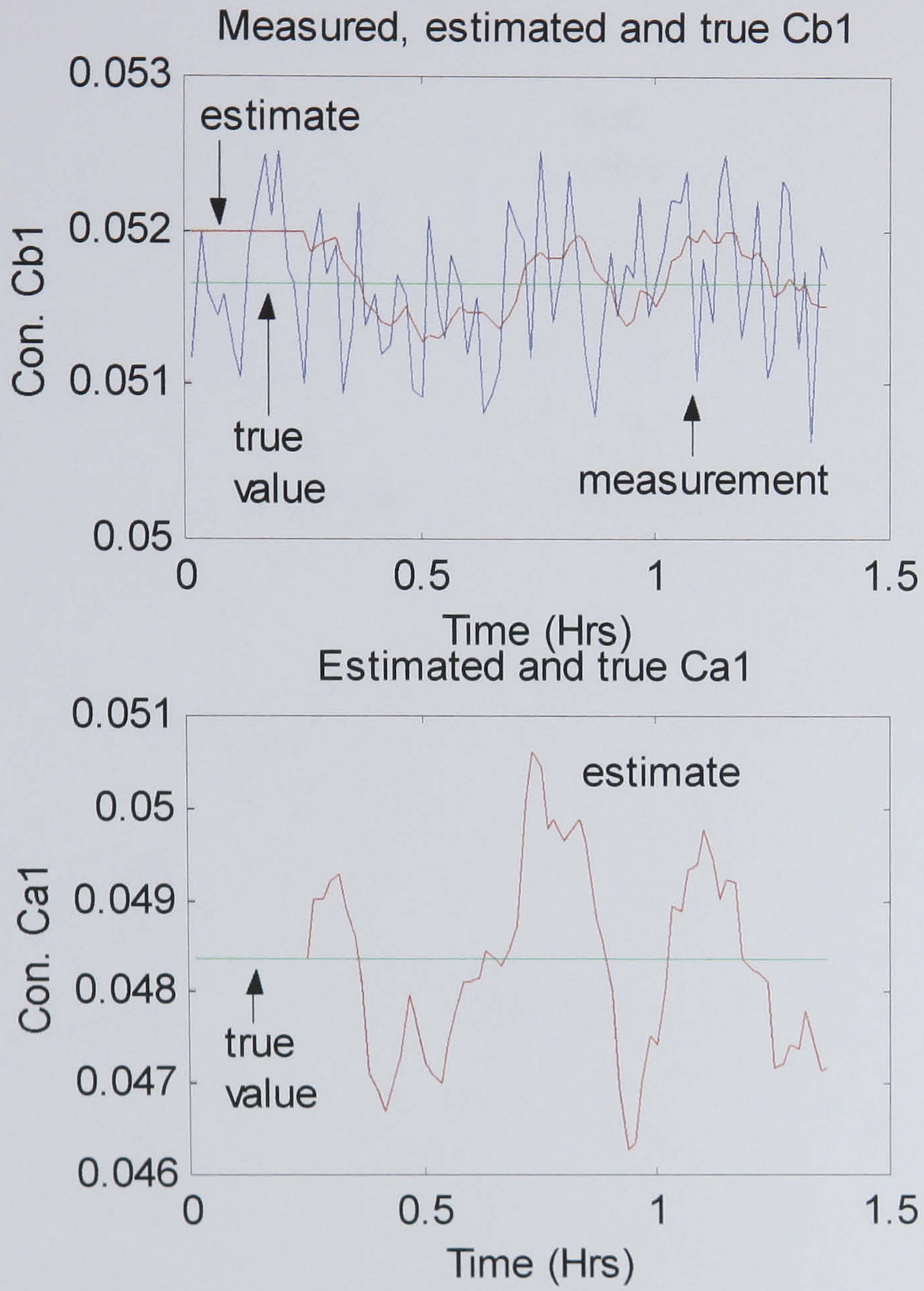


Figure (7-5): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with an outlier on C_{b1} when the gross error detection algorithm is enabled.



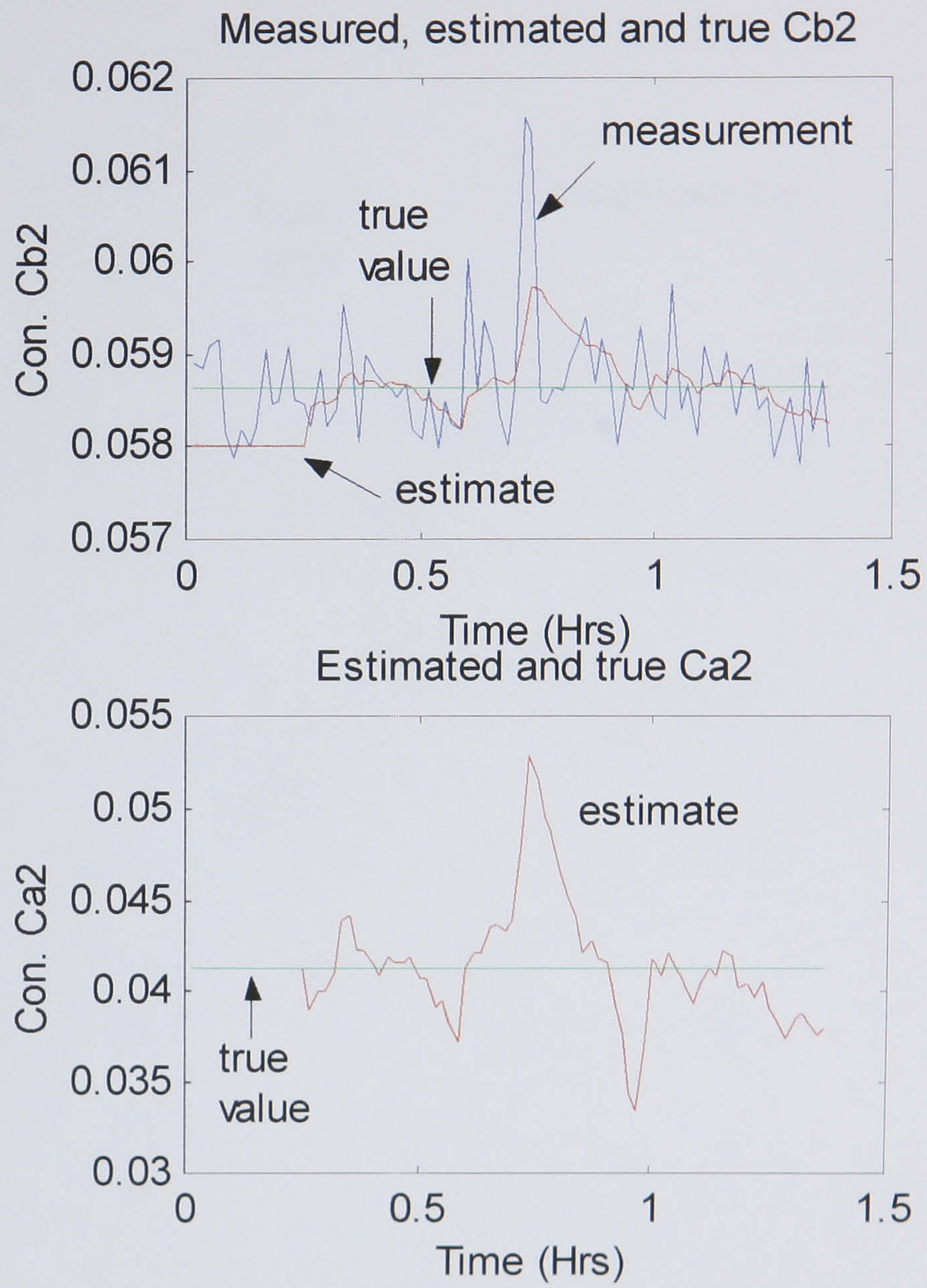
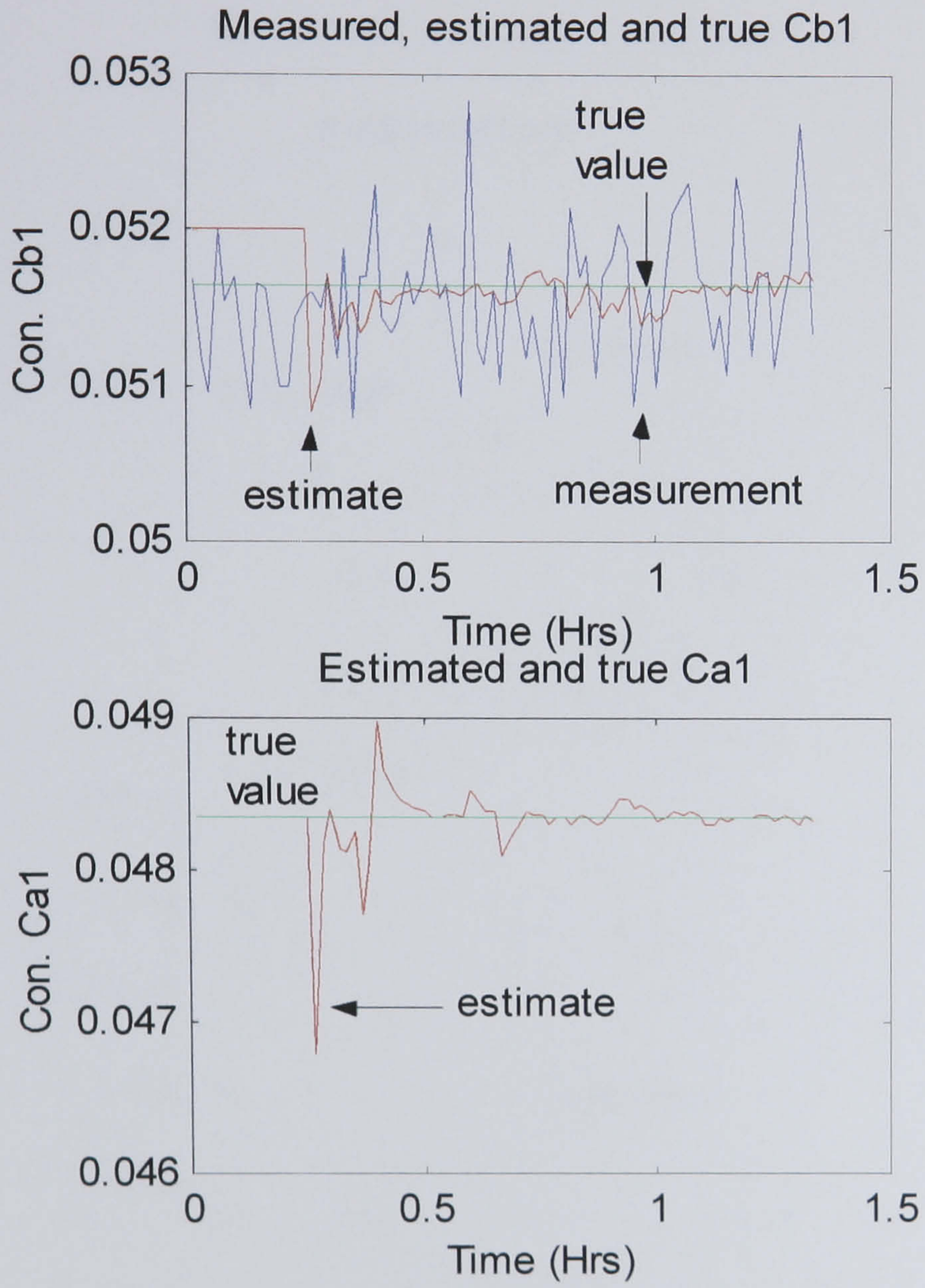


Figure (7-6): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with an outlier on C_{b2} when the gross error detection algorithm is disabled.



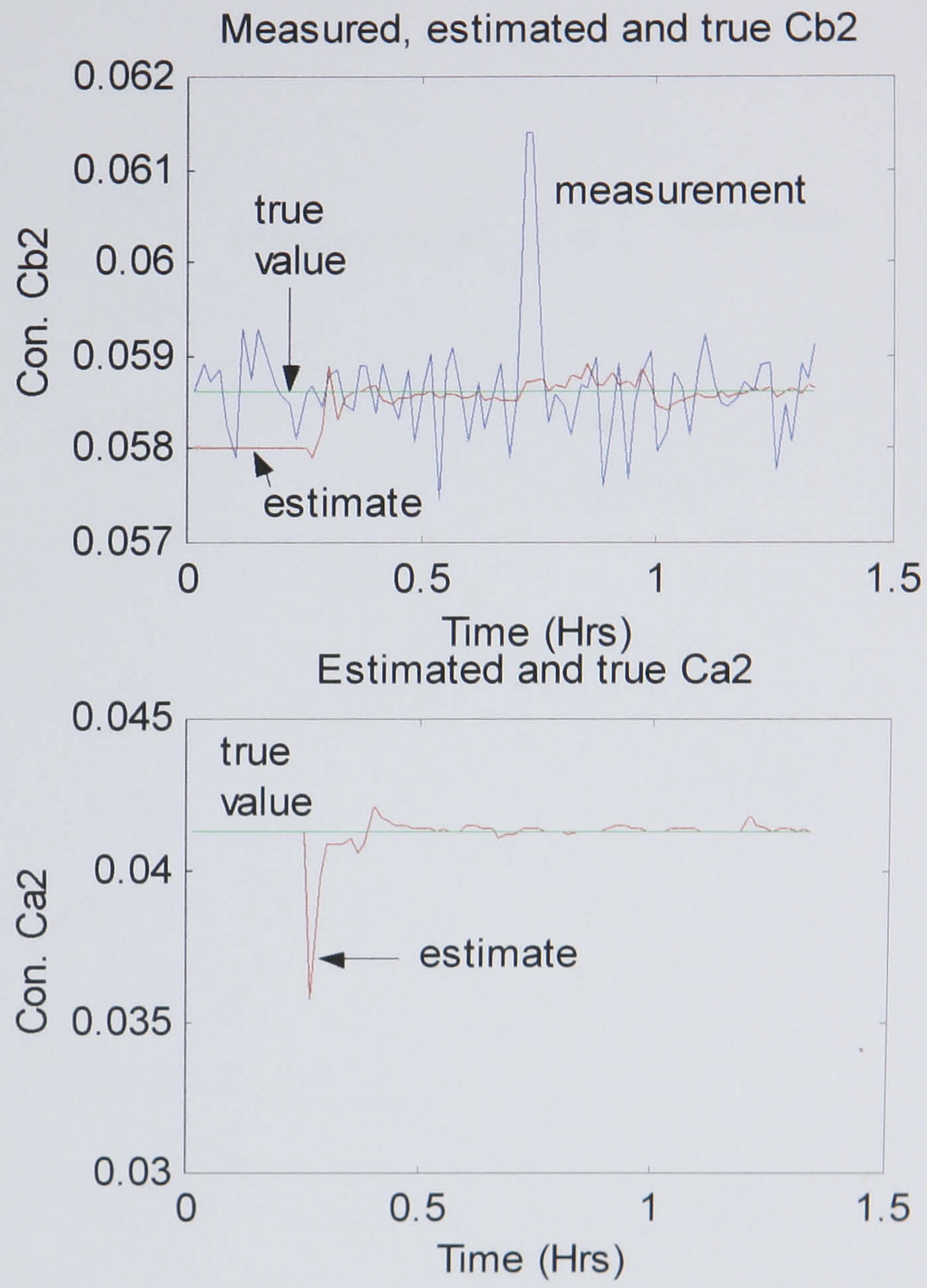
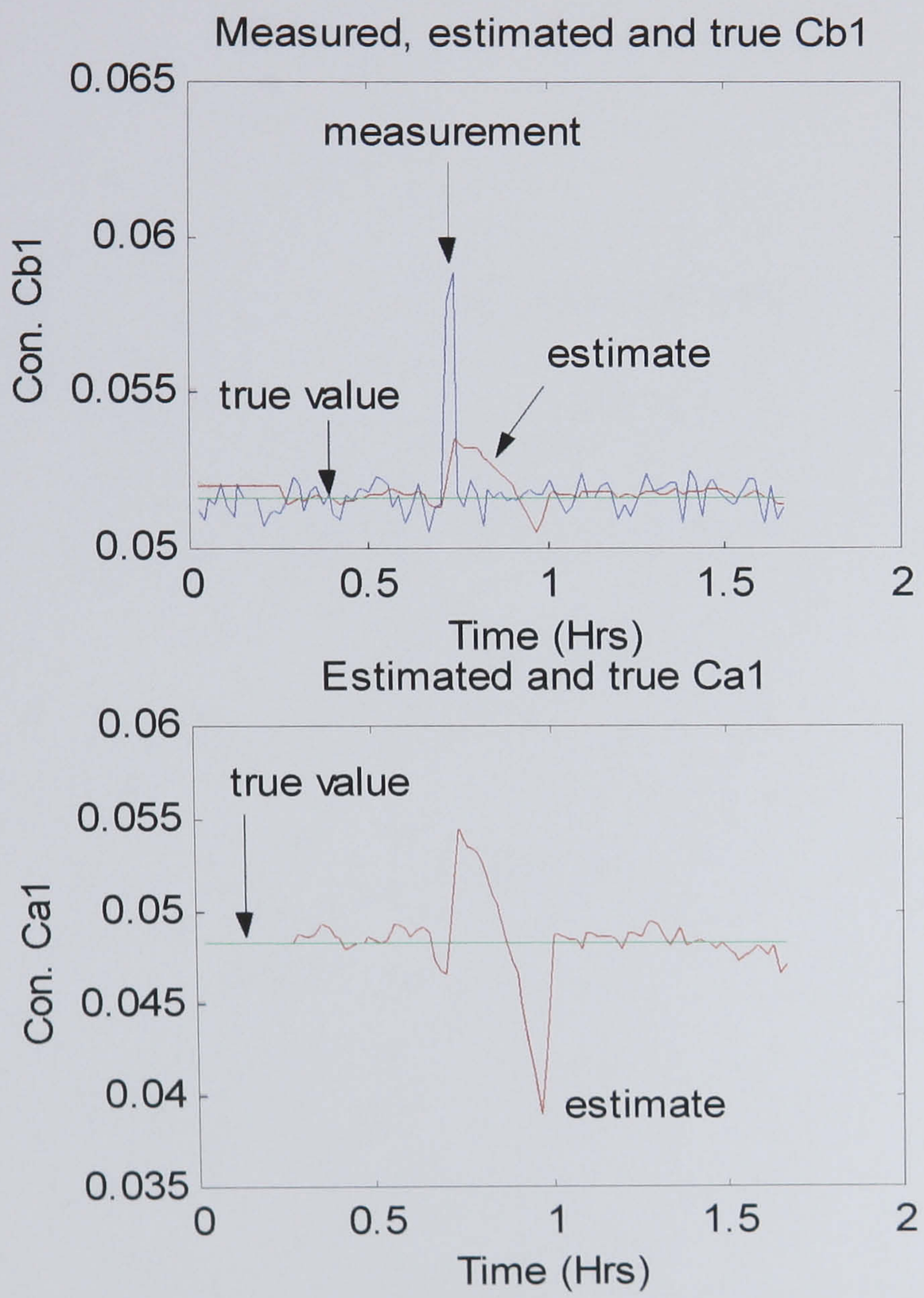


Figure (7-7): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with an outlier on C_{b2} when the gross error detection algorithm is enabled.



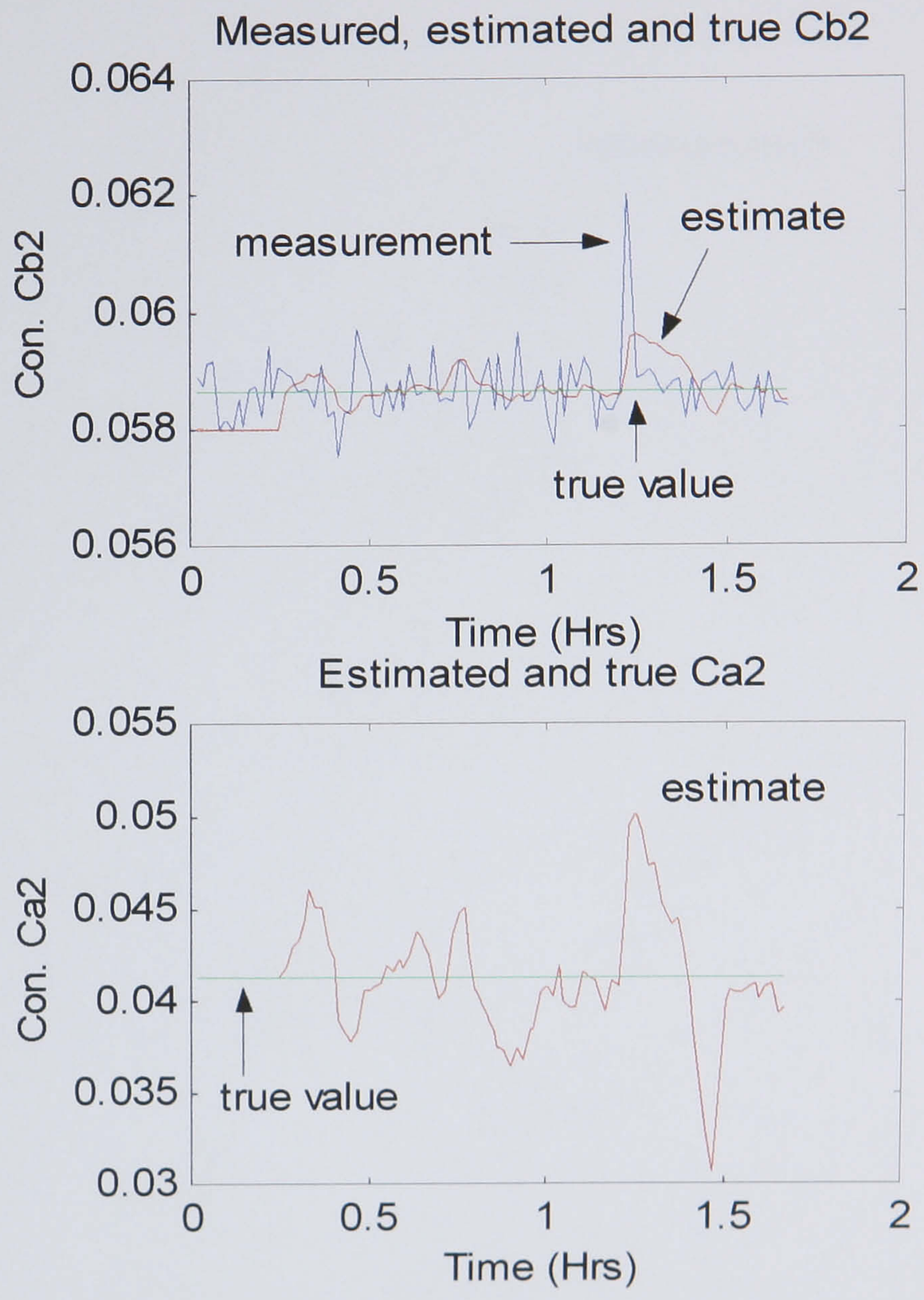


Figure (7-8): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with outliers on C_{b1} and C_{b2} when the gross error detection algorithm is disabled.

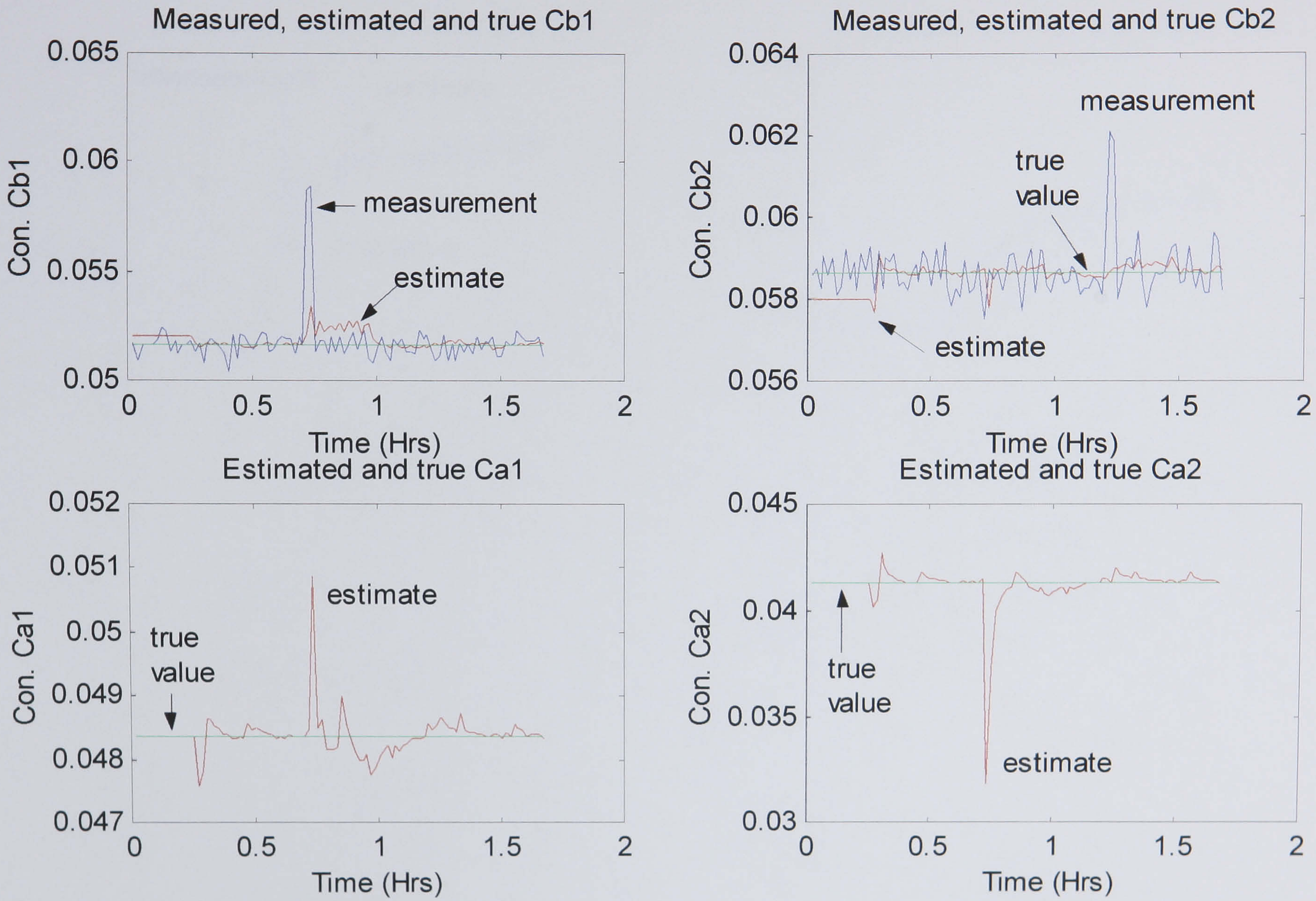
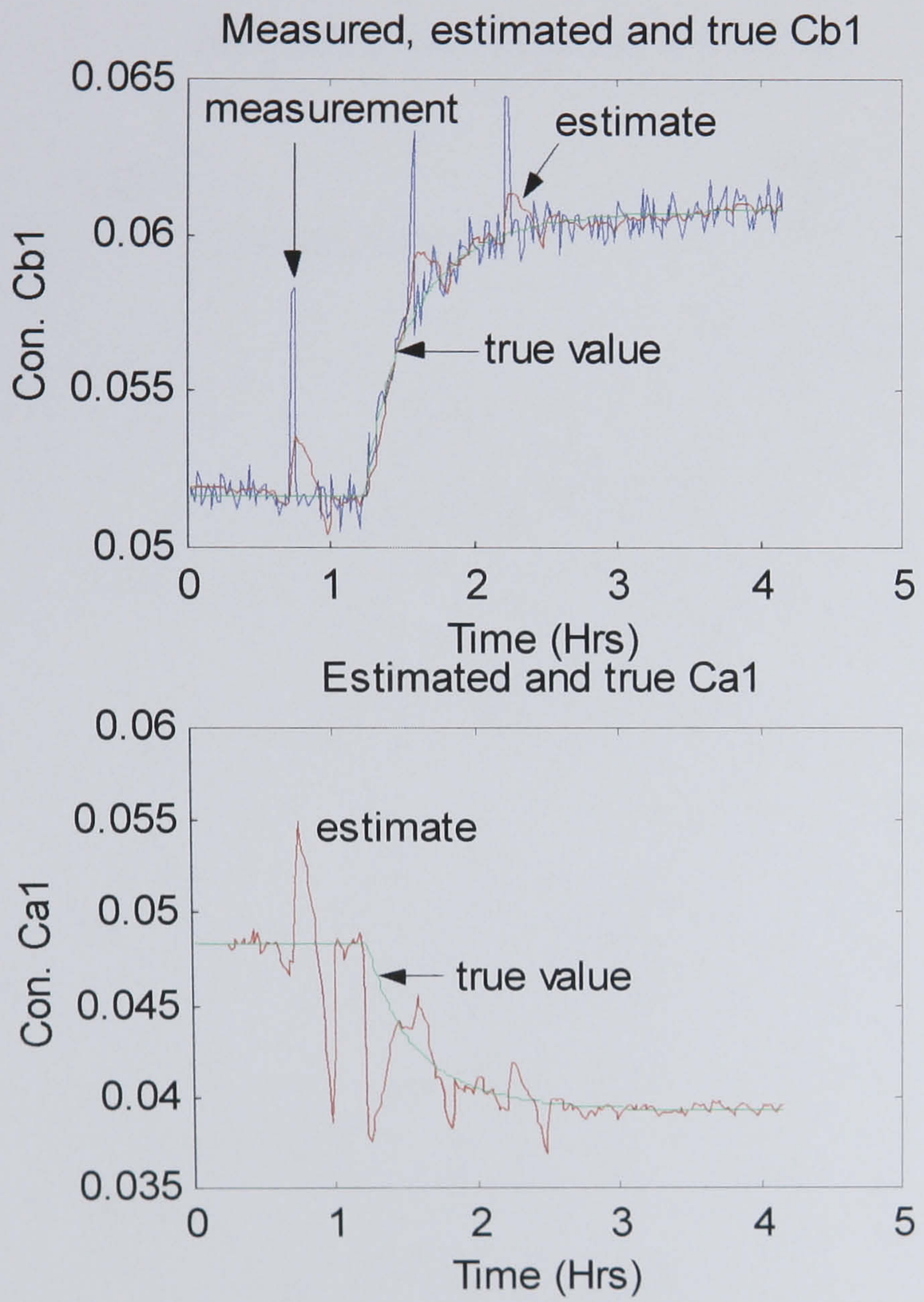


Figure (7-9): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with outliers on C_{b1} and C_{b2} when the gross error detection algorithm is enabled.



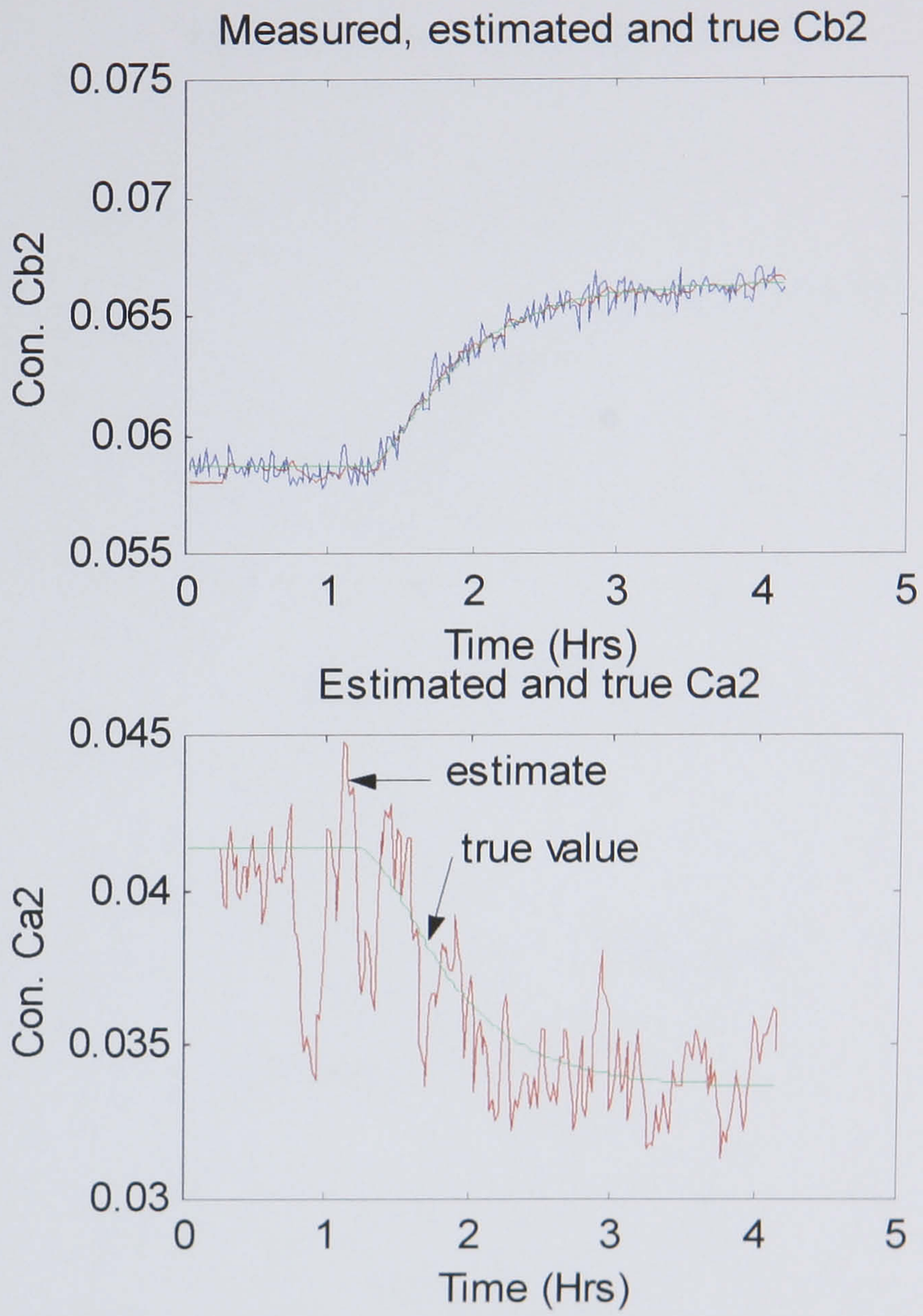


Figure (7-10): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with outliers on C_{b1} when the gross error detection algorithm is disabled.

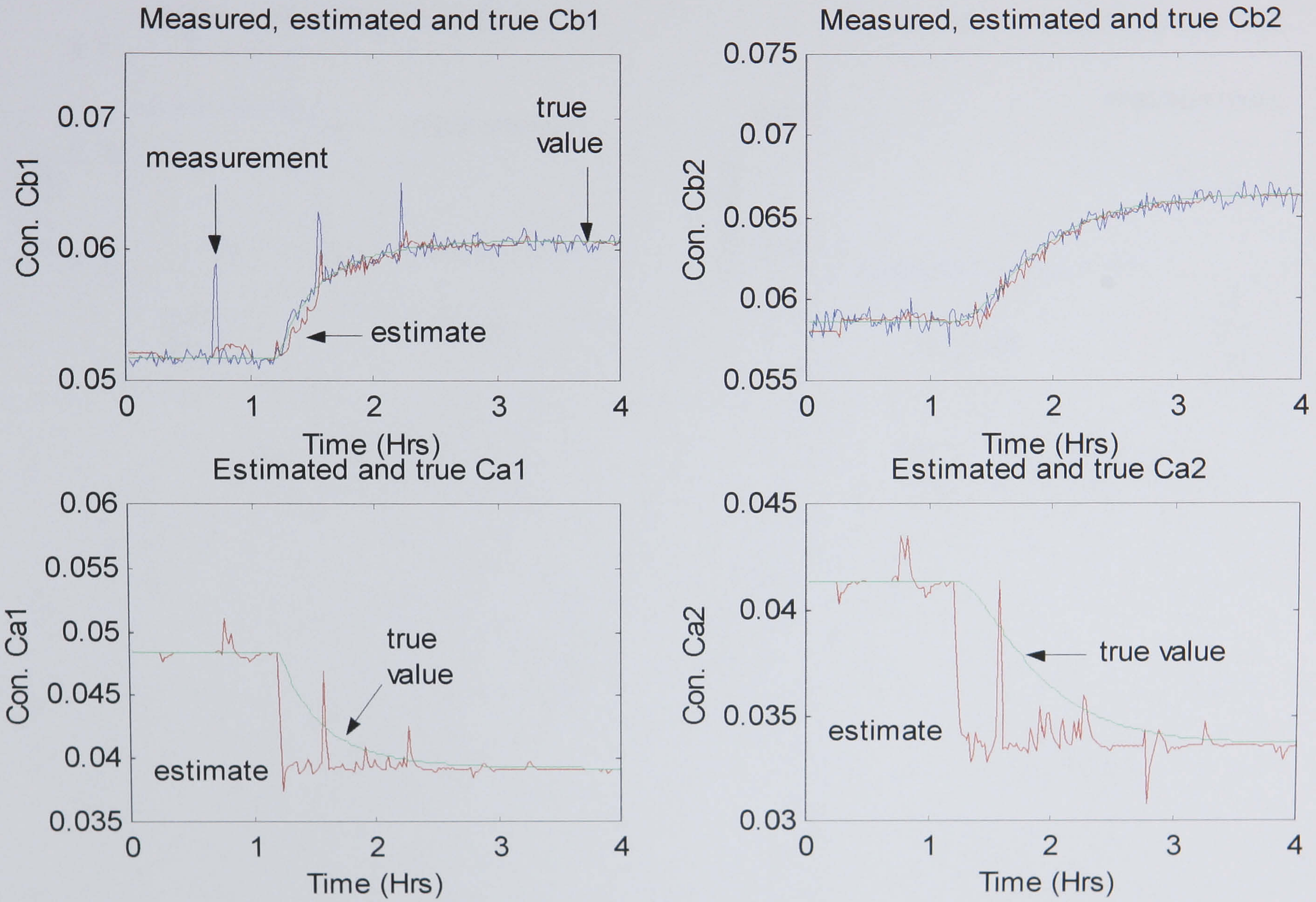


Figure (7-11): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with outliers on C_{b1} when the gross error detection algorithm is enabled.

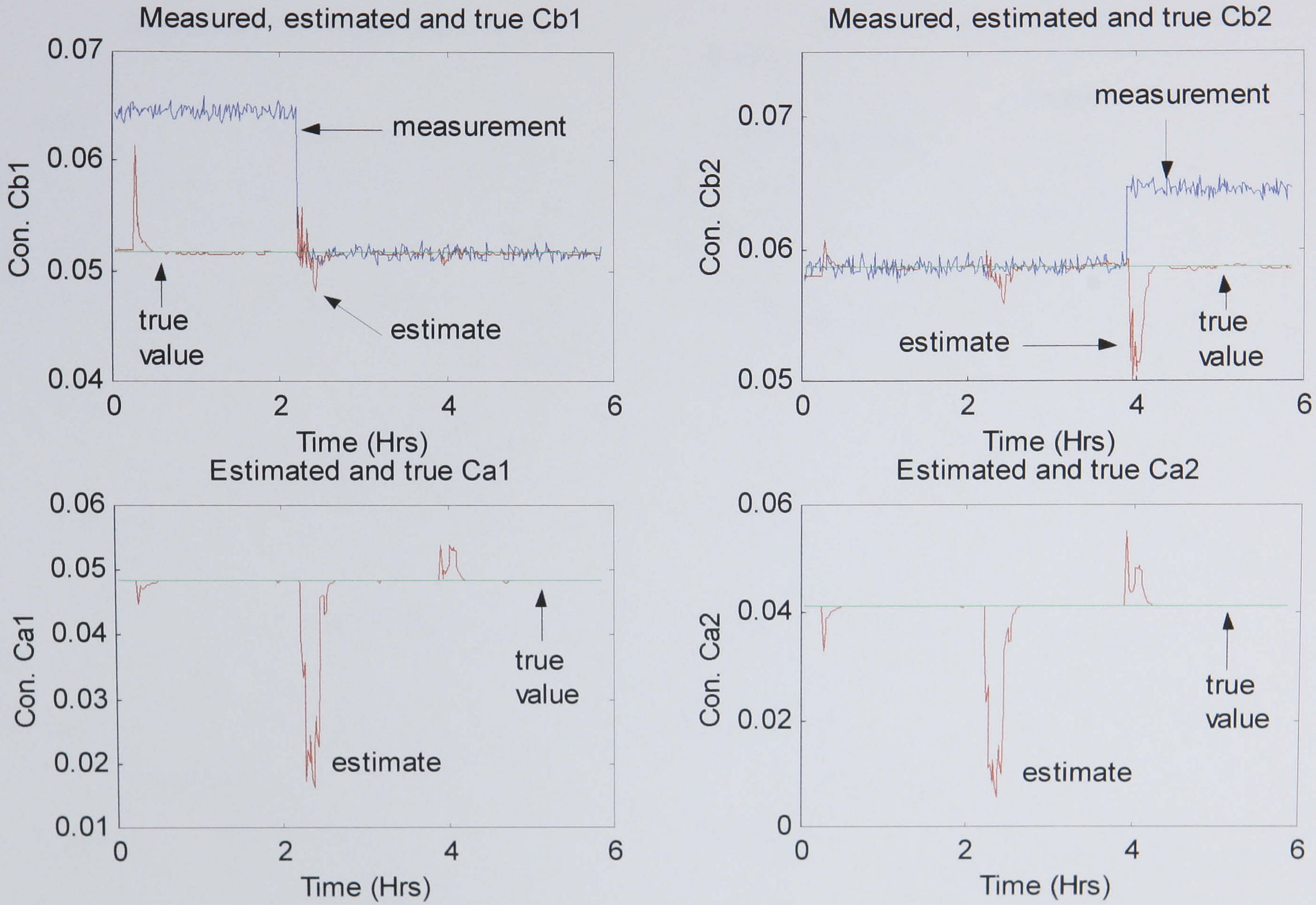


Figure (7-12): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with bias on C_{b1} then C_{b2} when the bias detection algorithm is working.

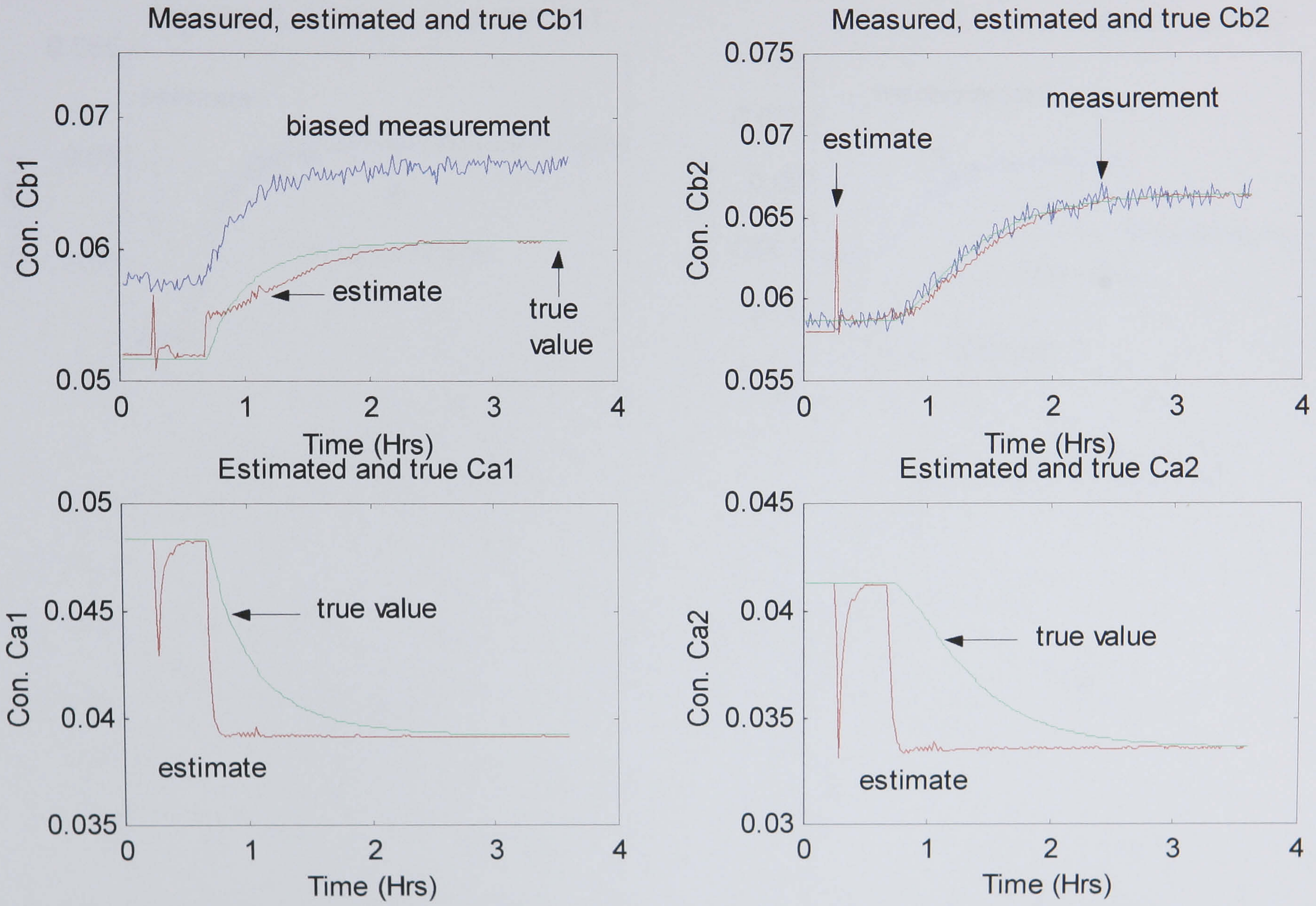


Figure (7-13): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with bias on C_{b1} when the bias detection algorithm is working.

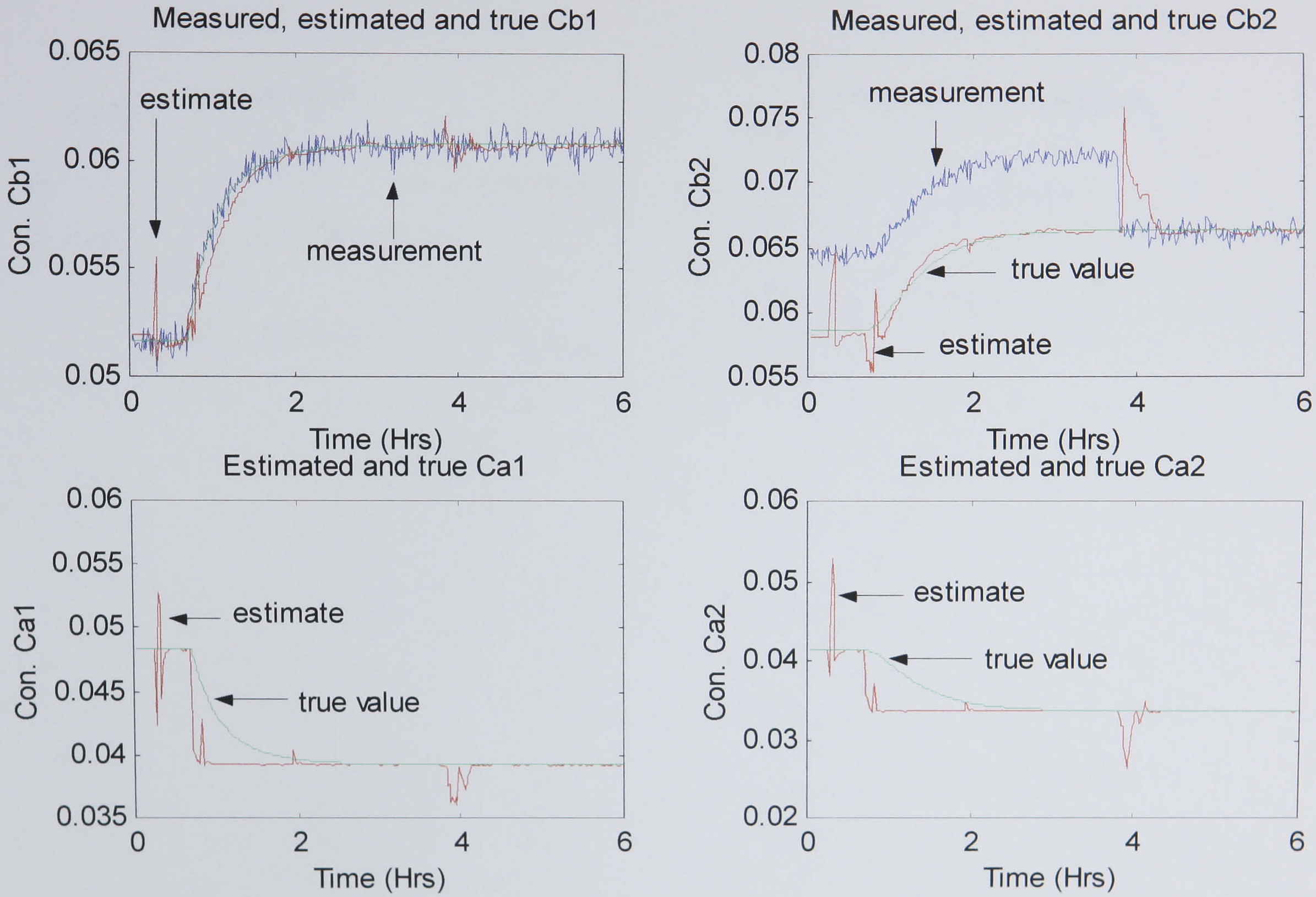
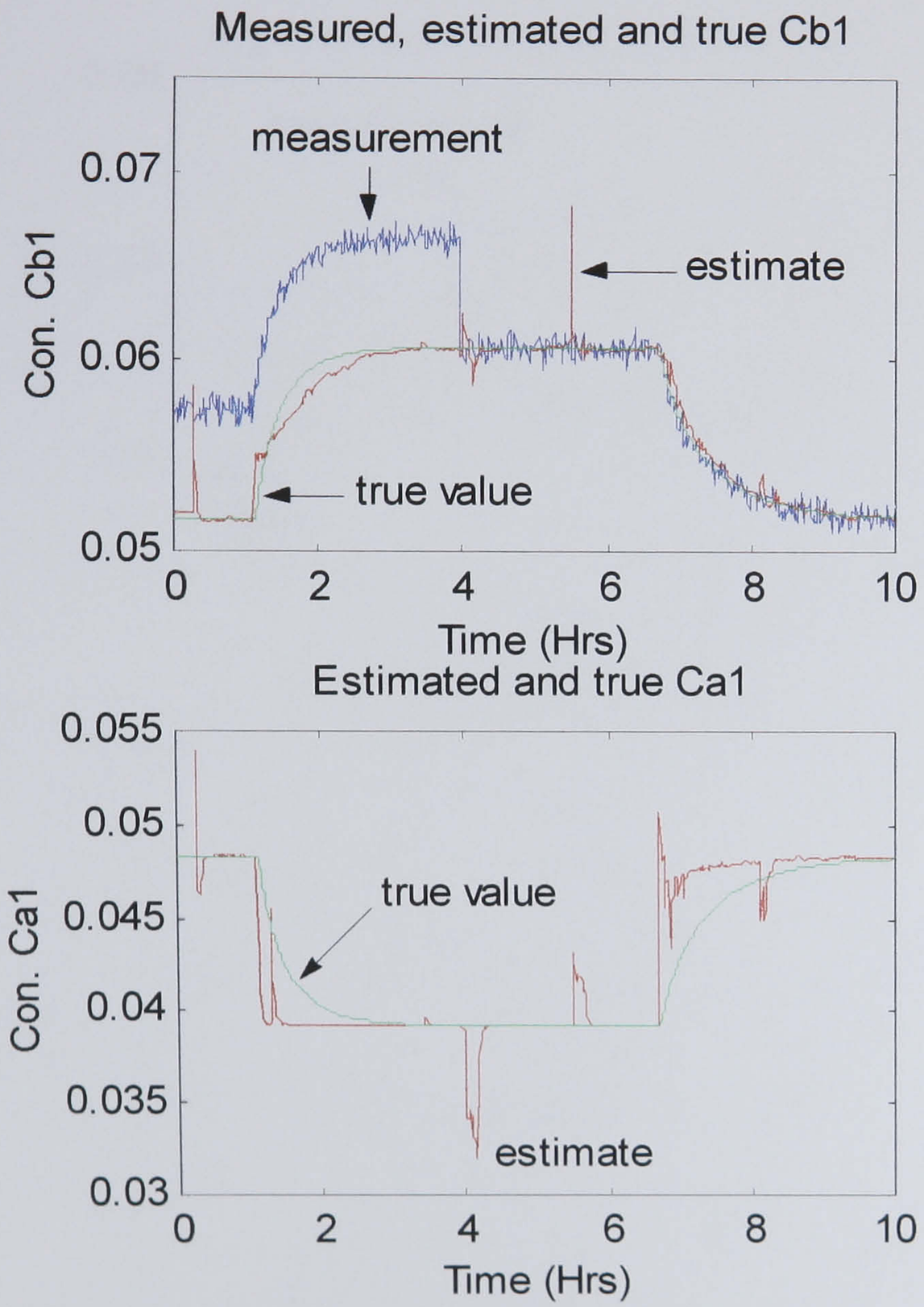


Figure (7-14): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with bias on C_{b2} which is later removed when the bias detection algorithm is working.



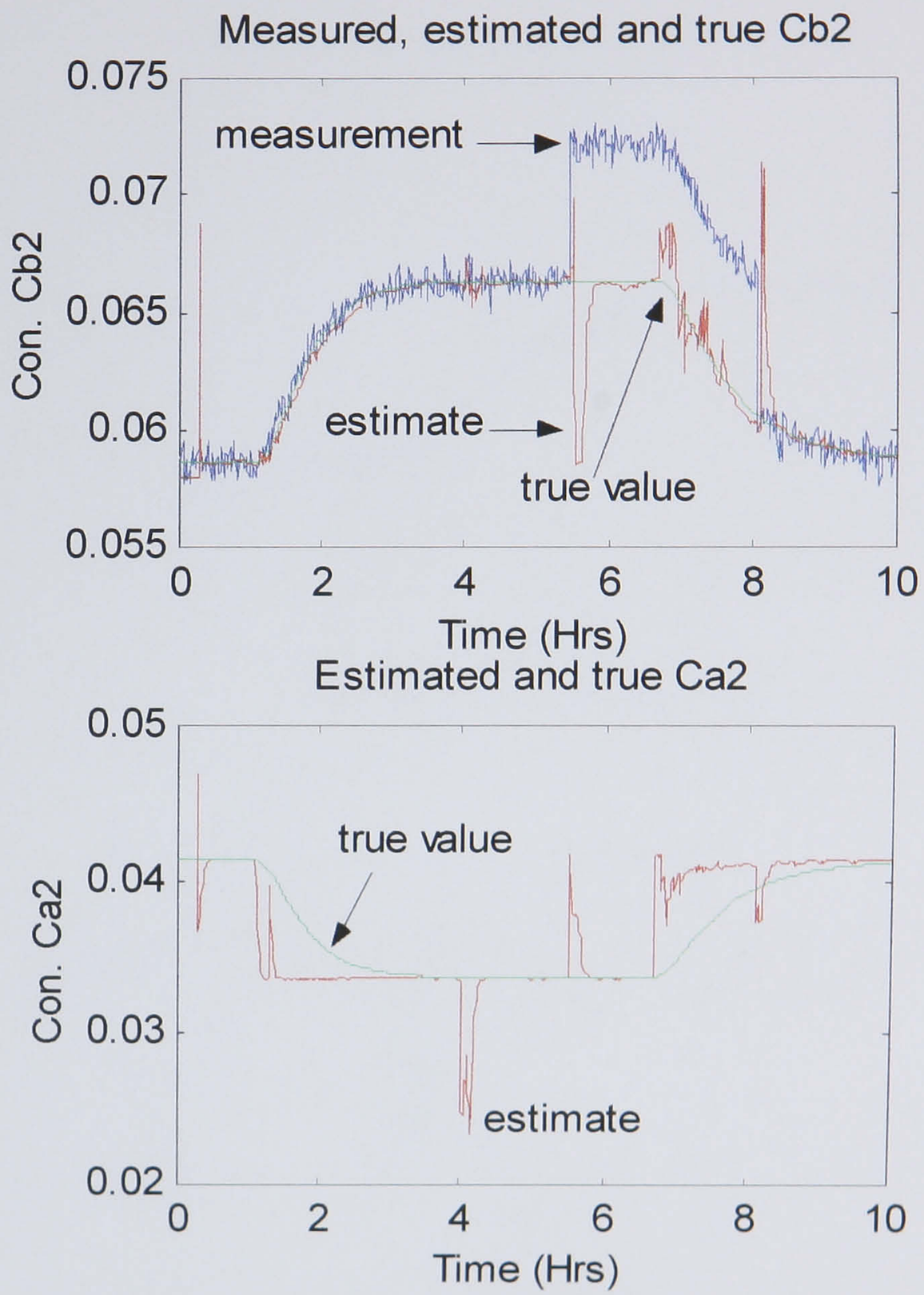


Figure (7-15): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} when the bias detection algorithm is working. Bias on C_{b1} which is then removed, then bias on C_{b2} which is later removed.

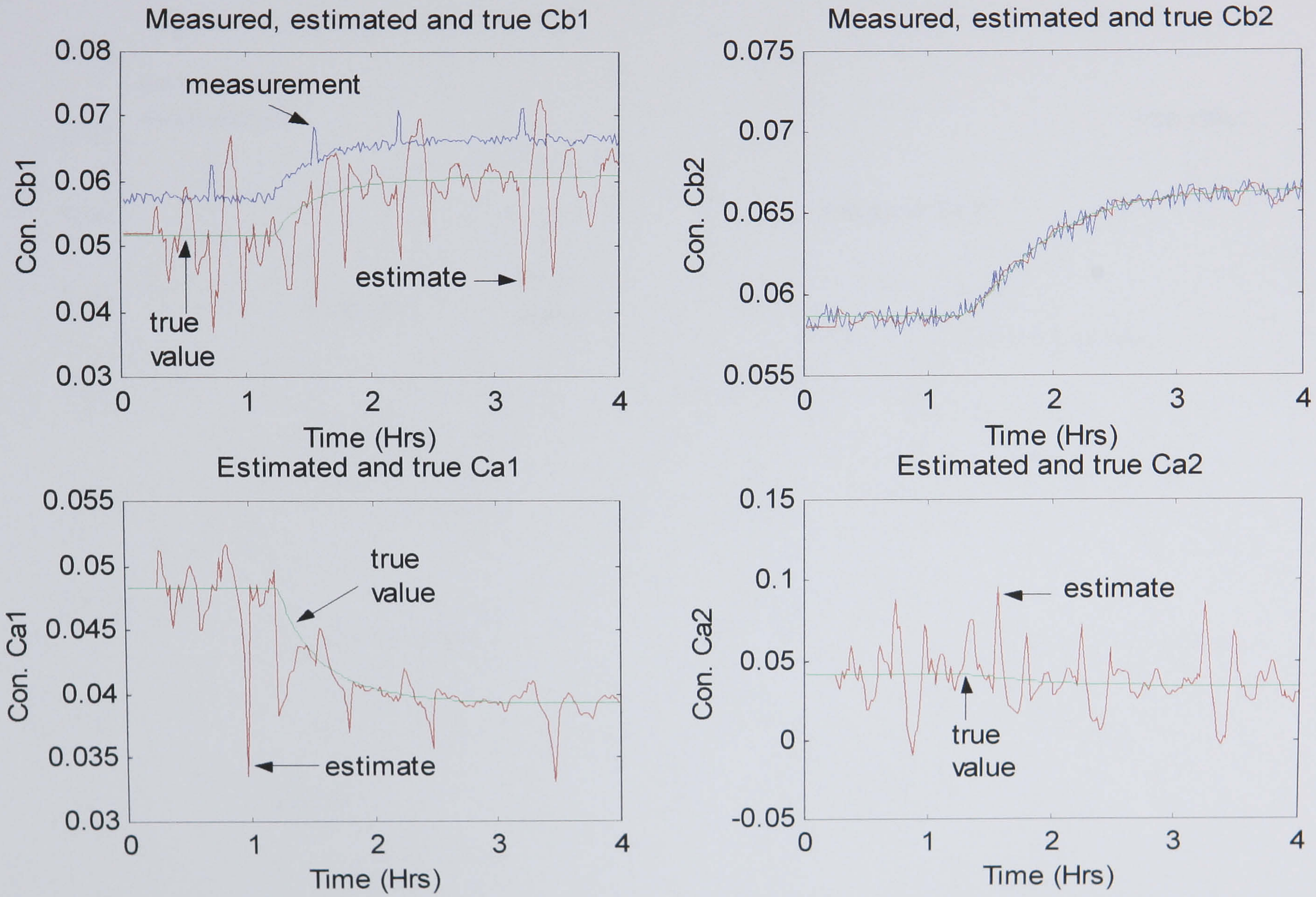


Figure (7-16): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with bias and outliers on C_{b1} when both bias and gross error detection algorithms are disabled.

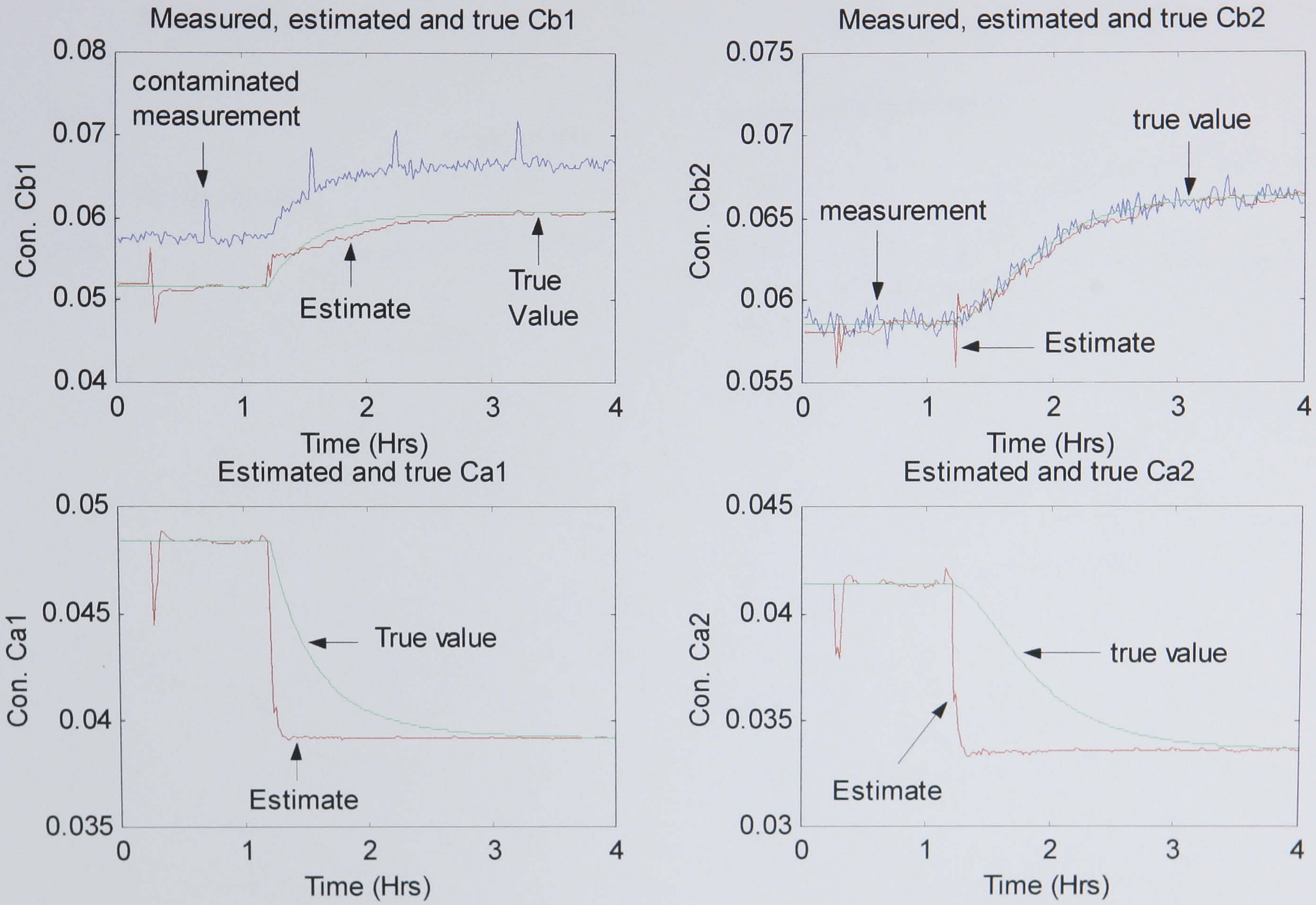


Figure (7-17): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with bias and outliers on C_{b1} when both bias and gross error detection algorithms are enabled.

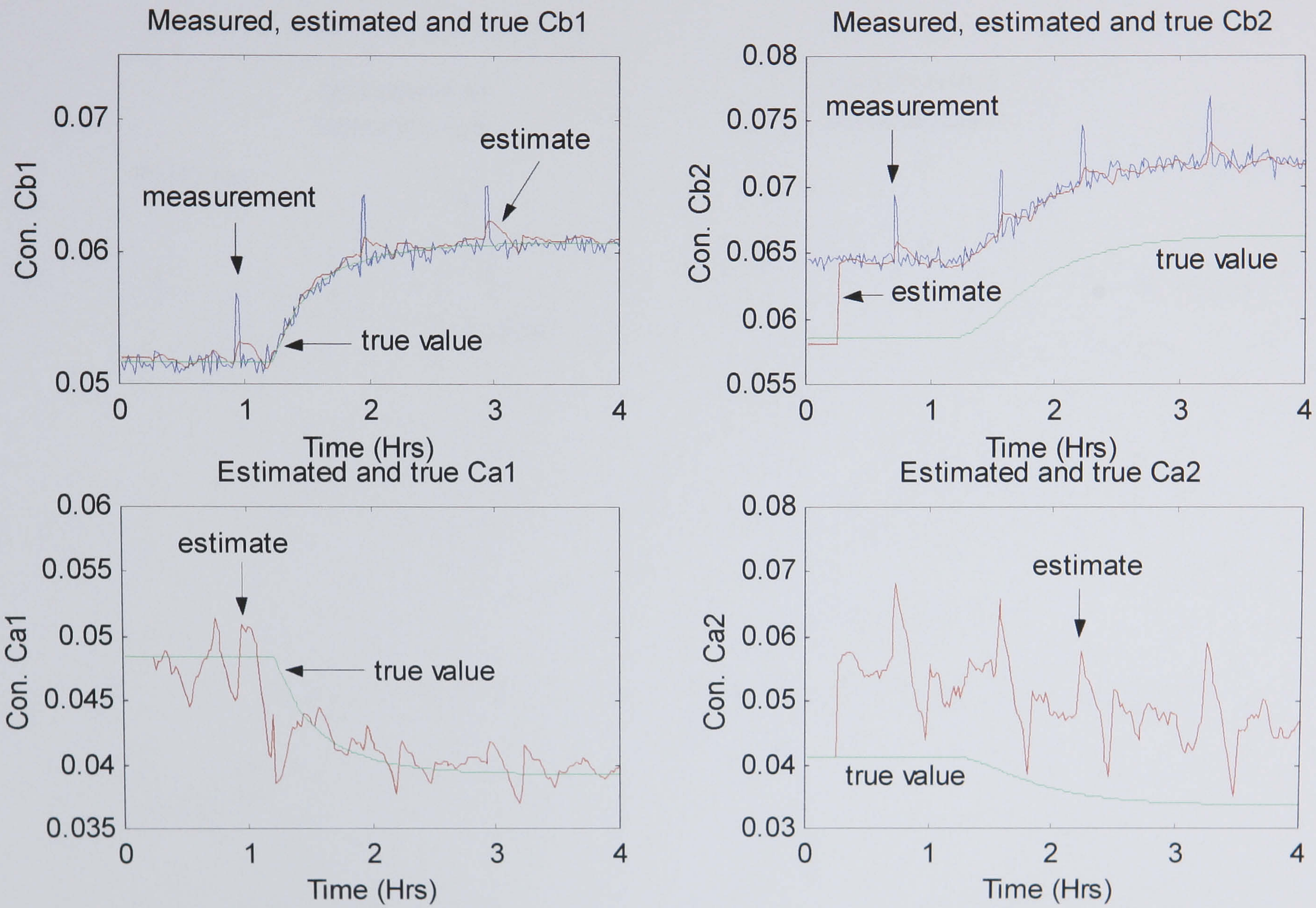


Figure (7-18): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with outliers on C_{b1} and bias and outliers on C_{b2} when both bias and gross error detection algorithms are disabled.

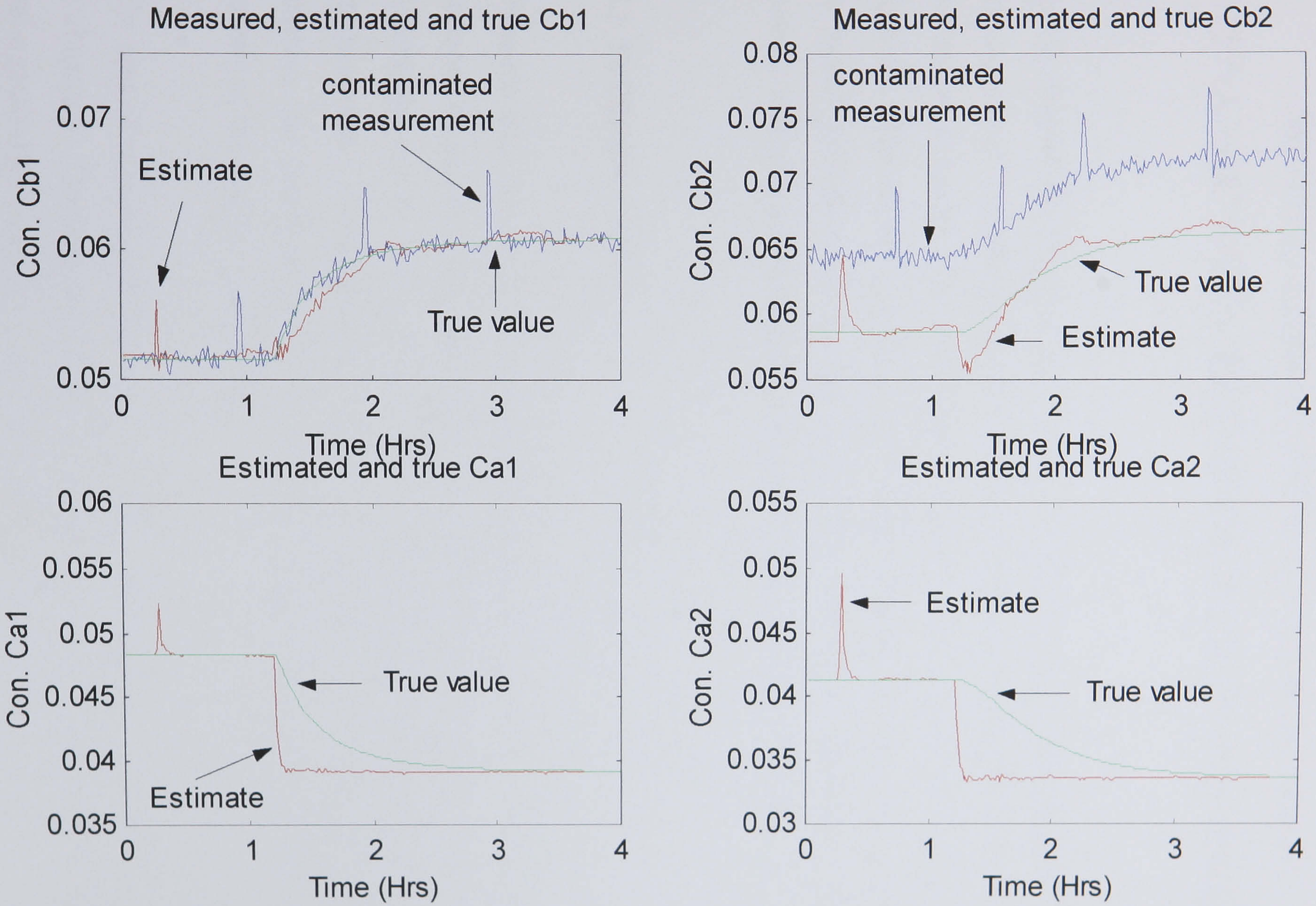


Figure (7-19): Measured variables C_{b1} and C_{b2} and unmeasured variables C_{a1} and C_{a2} with outliers on C_{b1} and bias and outliers on C_{b2} when both bias and gross error detection algorithms are enabled.

CHAPTER 8

MODEL PREDICTIVE CONTROL USING DYNAMIC DATA RECONCILIATION TECHNIQUES

In this chapter the dynamic data reconciliation techniques developed in this thesis are applied to a model predictive control scheme. Simulations are carried out on the dynamic model of two continuous stirred tank reactors used in the previous chapters. The advantages and disadvantages of using dynamic data reconciliation in model predictive control are highlighted through a comparison between a scheme that uses dynamic data reconciliation and one that does not.

8.1 INTRODUCTION

Dynamic data reconciliation has been introduced in chapter 7. There, a moving horizon estimator was used for the treatment of dynamic process data. Algorithms for the detection of systematic bias and outliers were proposed and implemented. These techniques and algorithms are now put into practice to improve the performance of the model predictive control scheme used in the control of the gasifier plant in chapter 3.

The area of model predictive control has been introduced and discussed in detail in chapter 3. The reader is therefore referred to that chapter for a thorough review and a general introduction on the subject.

8.2 SIMULATION CASE STUDIES

Two case studies have been set up in OTISS to show the effects of dynamic data reconciliation on a model predictive control scheme. In the first case study (Figure 8-1) the model predictive controller acts directly upon the biased measurements from the plant. However, in the second case study (Figure 8-2) the measurements are first treated by the dynamic data reconciliation module before being passed to the model predictive controller.

The simulations were carried out on the OTISS model of the two continuous stirred tank reactors connected in series introduced in chapter 4, Figure (4-2). The concentrations of species B in both tanks are measured such that $y_m = [C_{b1}\ C_{b2}]^T$. In all cases the system was started from the steady-state given by the set points $T_1(0) = 307K$ and $T_2(0) = 302K$, which yield steady-state values $C_{b1}(0) = 0.05165\ [Kmol/m^3]$ and $C_{b2}(0) = 0.05864\ [Kmol/m^3]$. The following tuning parameters for the predictive controller were used in the simulations: prediction horizon $N = 25$, incremental scaled state weight $Q = I$, incremental weights on the scaled manipulated variables $R = diag(20000, 20000)$. The tuning parameters used for the moving horizon scheme were: data window length $nh = 15$, integration step $= 10\,s$ and the covariance matrix $V = diag(0.5, 0.5)$.

Three different objective types were studied in each of the cases. The first objective is purely economic and reflects the desire to maximise the amount of product B in the second tank, $F(y_m, u_m) = -C_{b2}$. The second objective type is purely regulatory and reflects the desire to keep a specified measurement at a certain predefined value. In some simulations the regulated measurement was C_{b2} $(F(y_m, u_m) = (C_{b2} - 0.065)^2)$ while in others it was C_{b1} $(F(y_m, u_m) = (C_{b1} - 0.060)^2)$. The third type is a combination of economic and regulatory objectives such that $F(y_m, u_m) = (C_{b1} - 0.060)^2 - C_{b2}$. This reflects the desire to regulate the measurement C_{b1} at a value 0.060 while trying to maximise the product C_{b2}.

In order to simulate the effect of an outlier on the measurements, a bias was added to the measurement in question for a short period of time and then removed.

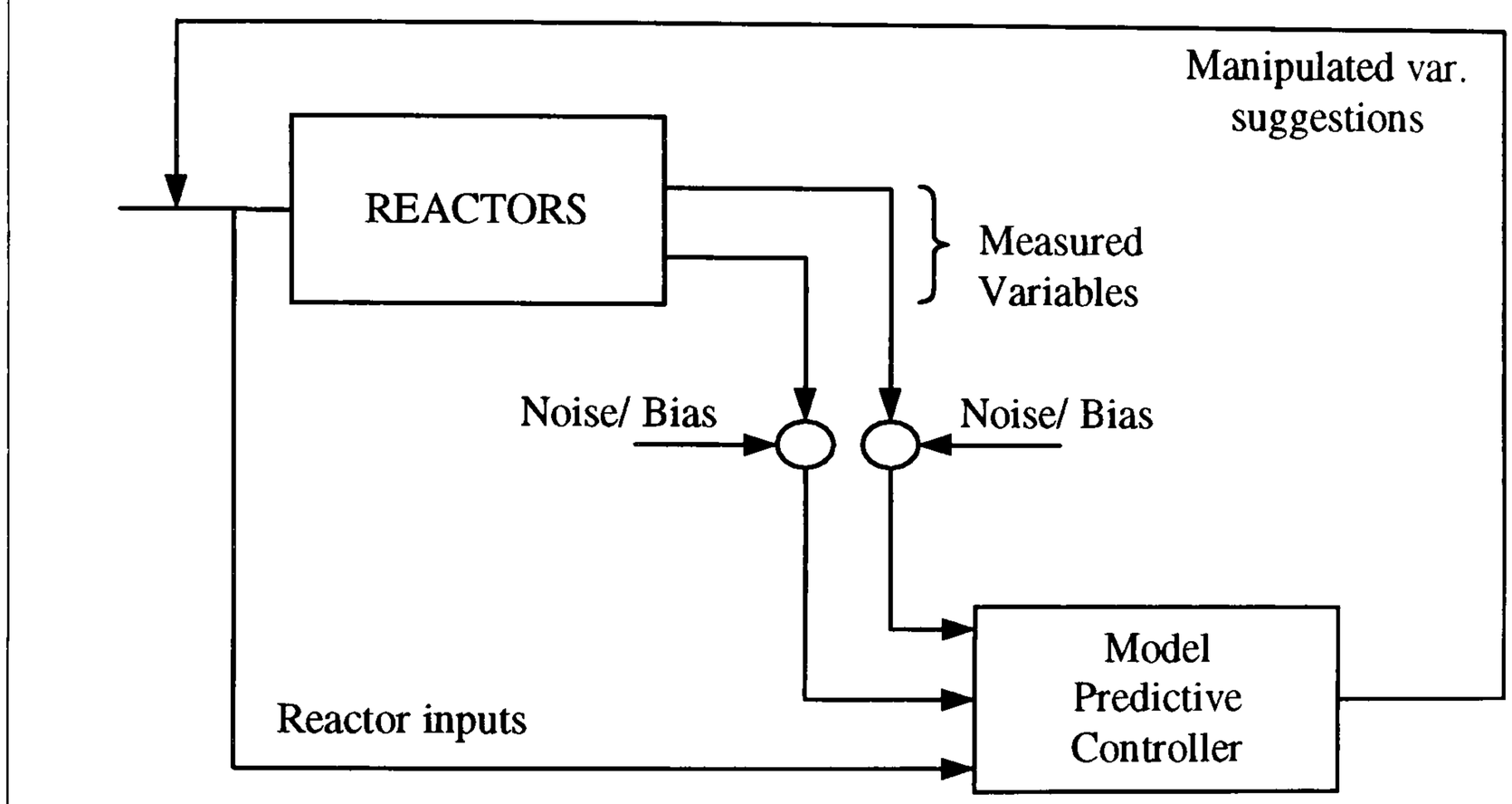


Figure (8-1): Schematic of case study (1), Model Predictive Control without the use of dynamic data reconciliation techniques.

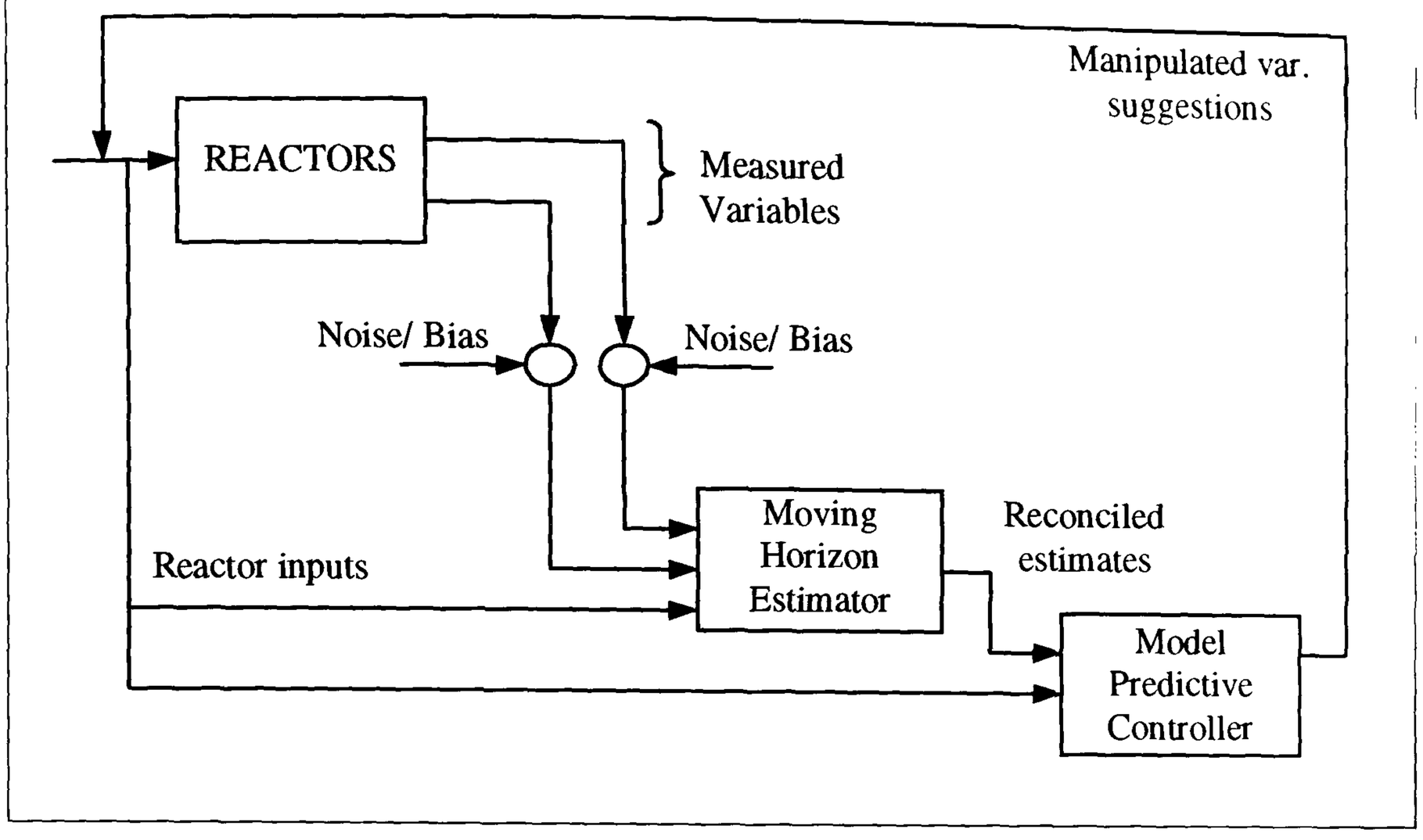


Figure (8-2): Schematic of case study (2), Model Predictive Control using dynamic data reconciliation techniques.

8.2.1 Results

An objective function of the type $F(y_m, u_m) = (C_{b2} - 0.065)^2$ was first used. This reflects the desire to regulate the measurement C_{b2} at the value 0.065. Figure (8-3) shows the measured and true values of C_{b1} and C_{b2} as well as the temperatures which are the manipulated variables of the plant. In this instance there are no bias or outliers on the measurements. It can be seen that in this case the controller manages to regulate C_{b2} about the setpoint. This behaviour should be compared to Figure (8-4) where a systematic bias of magnitude 0.00585 is present on measurement C_{b1}. The deterioration in response due to the bias is evident. Figure (8-5) shows the same case where this time the data is first reconciled prior to the model predictive control stage. It can be observed that by using data reconciliation the predictive controller is able to meet the required regulatory objective with increased accuracy and speed. The other sub-figures in Figure (8-5) show the true and estimated values of the unmeasured variables C_{a1} and C_{a2} as

well as the trends of the manipulated variables T_1 and T_2. Notice that there is an initial identification period of 2 hours when the predictive controller is not in operation.

Figure (8-6) shows the case when data reconciliation is not employed but when bias and outliers are present on the first measurement. Here, the objective function is purely economic of the form $F(y_m, u_m) = -C_{b2}$ and reflects the desire to maximise the amount of product C_{b2}. Despite the presence of large errors on C_{b1} and the fact that no data reconciliation is taking place, the controller drives the process to achieve a final value of $C_{b2} = 0.0722\ \ Kmol/m^3$ within 11 hours. This should be compared with Figure (8-7) where the same objective function is used but where dynamic data reconciliation is employed. Surprisingly, it is seen that the performance deteriorates. The final value of $C_{b2} = 0.0719$ but this is only achieved after 26 hours of simulation.

In order to see if the model adaptation capability of the predictive controller played a part in this surprising result, two simulations were done during which model adaptation was disabled after the initial identification period of 2 hours. The first simulation (Figure 8-8) was carried out using no data reconciliation techniques. The measurement C_{b1} is biased and has a number of outliers towards the beginning of the simulation. The objective is once again to maximise the amount of product C_{b2}. The trends in Figure (8-8) should be compared to those in Figure (8-9) where dynamic data reconciliation is enabled. It can be observed that in this case where the model adaptation facility is disabled, the behavior is similar except that in Figure (8-9) the response seems slightly slower.

To fully appreciate the effect of a pure economic objective and the influence of the dynamic data reconciliation modules, two simulations were carried out in the absence of bias and outliers. Figure (8-10) shows the case where data reconciliation is not employed when the objective is again to maximise the amount of product C_{b2}. Figure (8-11) on the other hand shows the same case but this time using dynamic data reconciliation techniques. Again, it can be observed

that the use of dynamic data reconciliation techniques in the case when the objective is purely economic tends to deteriorate the performance.

The third type of objective function tested is a combined economic and regulatory objective. Figures (8-12) and (8-13) show the case when the objective is to regulate measurement C_{b1} at a value 0.060 while at the same time maximising the amount of product C_{b2}. Figure (8-12) shows the case where data reconciliation is not employed. It can be observed that while C_{b2} is maximised, the controller fails to regulate the measurement C_{b1}. However Figure (8-13), in which data reconciliation is enabled, shows the fact that C_{b1} is closely following the setpoint while the controller attempts to maximise C_{b2}.

8.3 CONCLUSIONS

Three different types of objectives have been used to test the effect of employing dynamic data reconciliation techniques in a model predictive control strategy. The data reconciliation scheme used makes use of the bias and gross error detection algorithms developed in this thesis. In the case of a pure regulatory objective, the simulations show that reconciling process data prior to the predictive control stage enhances the performance considerably.

Surprisingly, however, in the case of a pure economic objective, the performance of the predictive control scheme tends to deteriorate when dynamic data reconciliation is employed. This fact may be due to the combined result of the two objective functions: the one associated with the predictive controller and that associated with the reconciliation procedure. Furthermore, the results obtained may be specific to this particular application. A mathematical investigation into the cause of this phenomenon would probably be a worthwhile exercise but is beyond the scope of this thesis. Also further investigations using an alternative process would certainly be useful.

In the case of a combined objective that includes regulatory as well as economic terms, the use of data reconciliation techniques tends to have a positive effect on the predictive control scheme. It has been shown in the simulations that the controller will hold the regulated measurement at the setpoint while attempting to meet the economic objective.

8.4 SUMMARY

In this chapter, the dynamic data reconciliation techniques developed in this thesis have been employed to enhance the performance of a model predictive control scheme. It has been shown, in this particular case, that reconciling the process data before it is used by the predictive controller improves the performance in cases where there are pure regulatory or combined objectives which include regulatory as well as economic objectives. However this is not the case when the objective is purely economic.

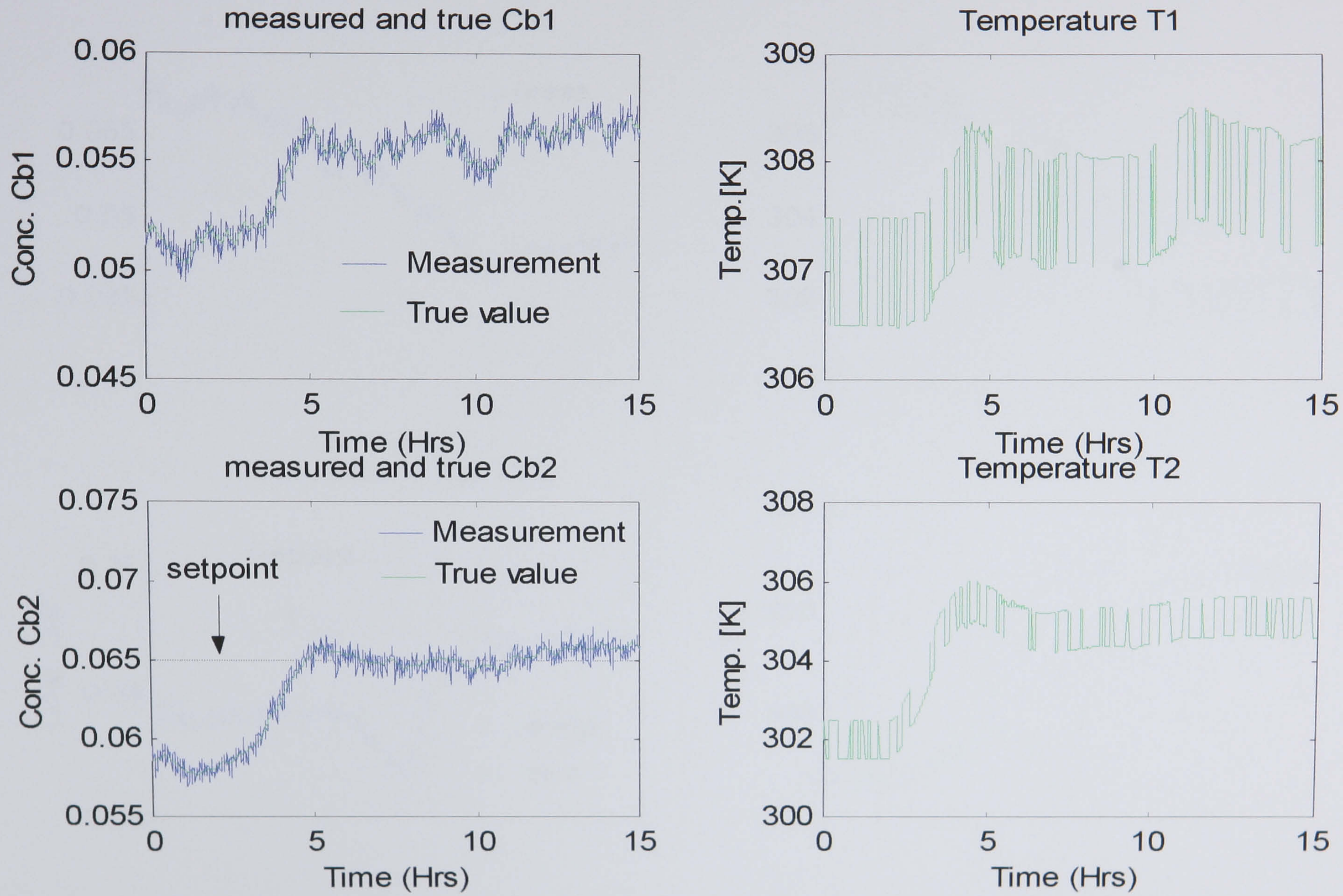
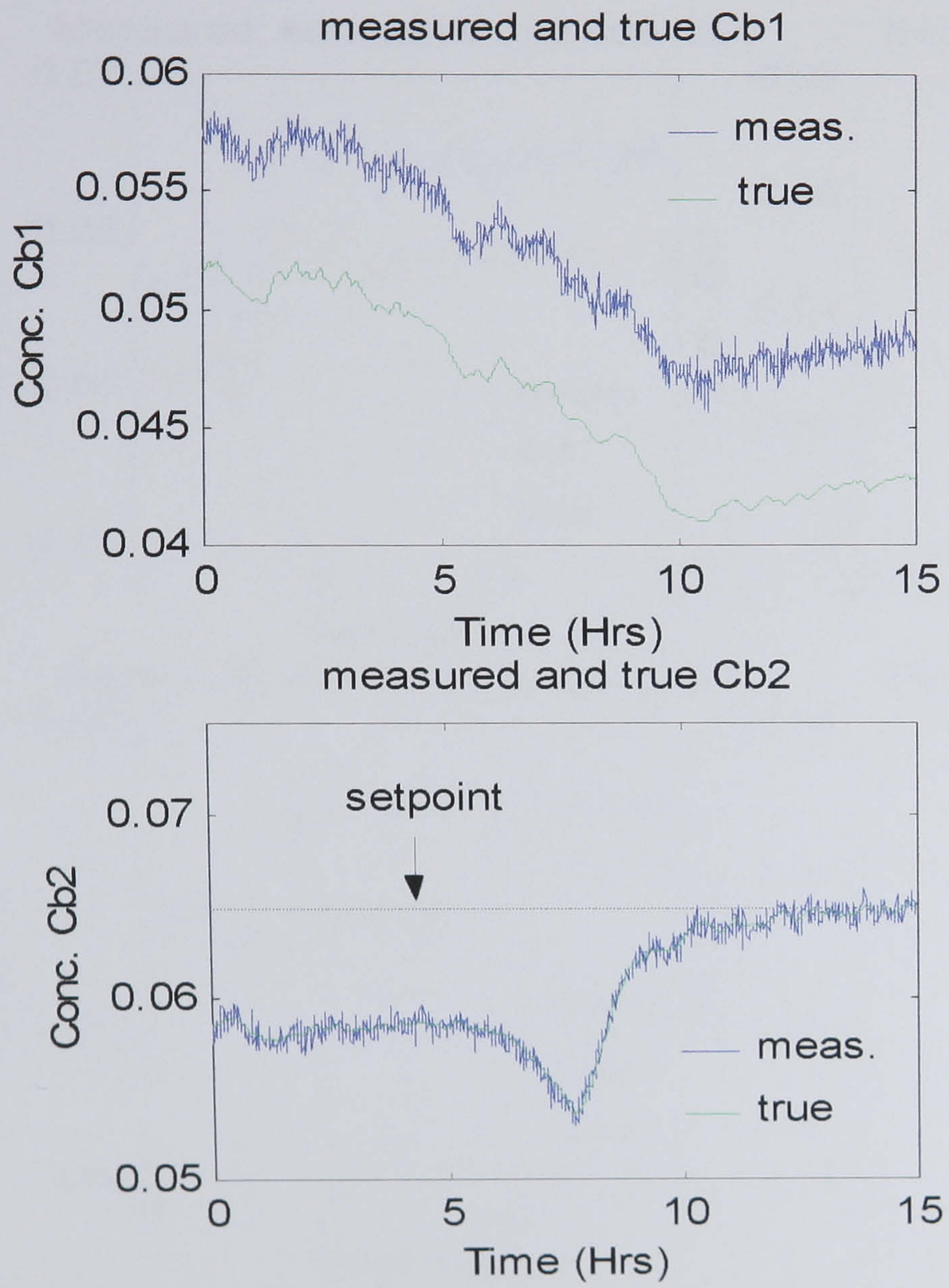


Figure (8-3): Measured and true C_{b1} and C_{b2} as well as trends of the manipulated variables T_1 and T_2 when measurements are not biased, data reconciliation is disabled, objective: regulatory (C_{b2}=0.065).



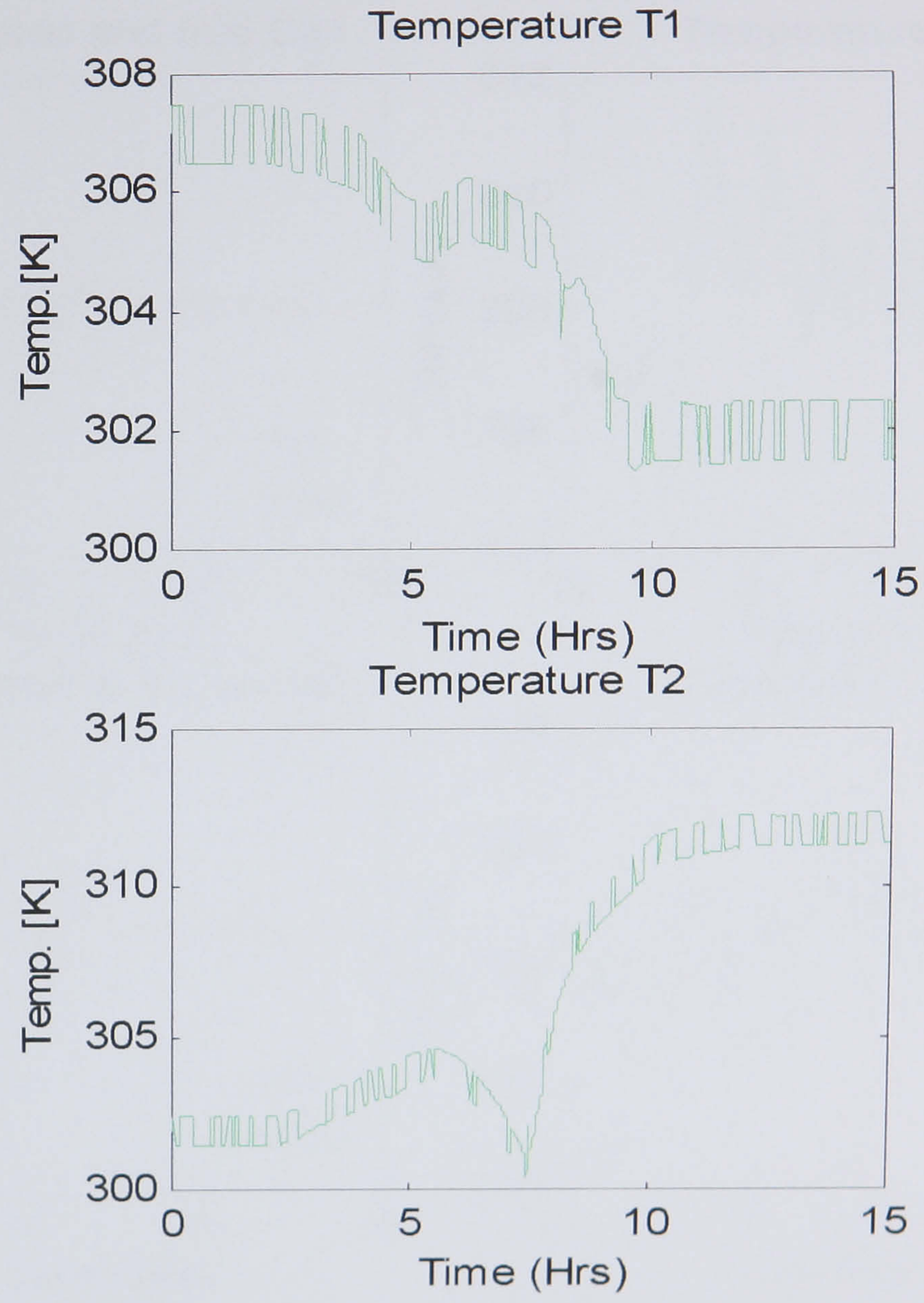


Figure (8-4): Measured and true C_{b1} and C_{b2} as well as trends of the manipulated variables T_1 and T_2 bias on C_{b1}= 0.00585, data reconciliation disabled, objective: regulatory (C_{b2}=0.065).

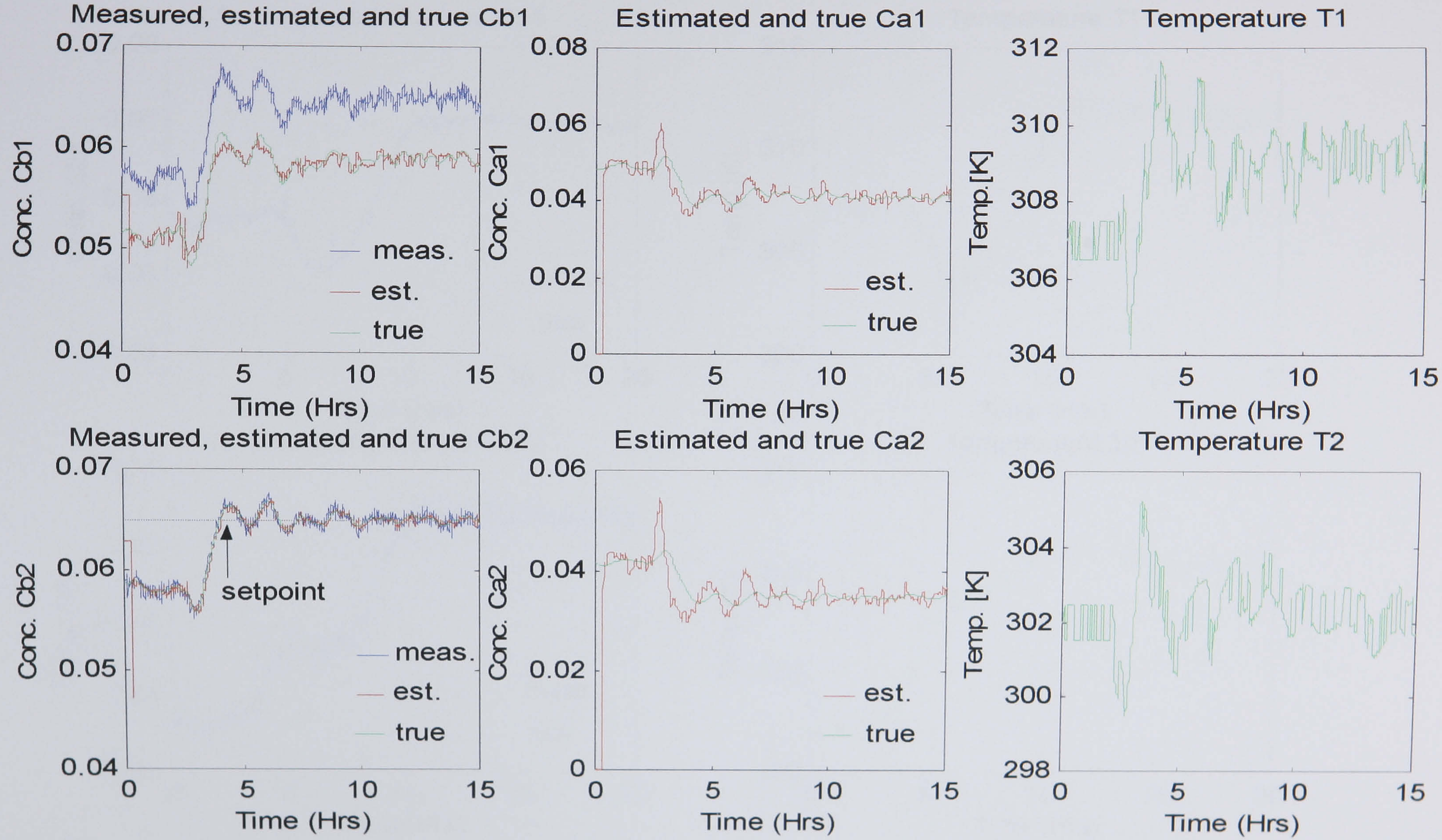


Figure (8-5): Measured, estimated and true C_{b1} and C_{b2}, estimated and true C_{a1} and C_{a2}, trends of manipulated variables T_1 and T_2, bias on C_{b1}, data reconciliation enabled, objective: regulatory (C_{b2}=0.065).

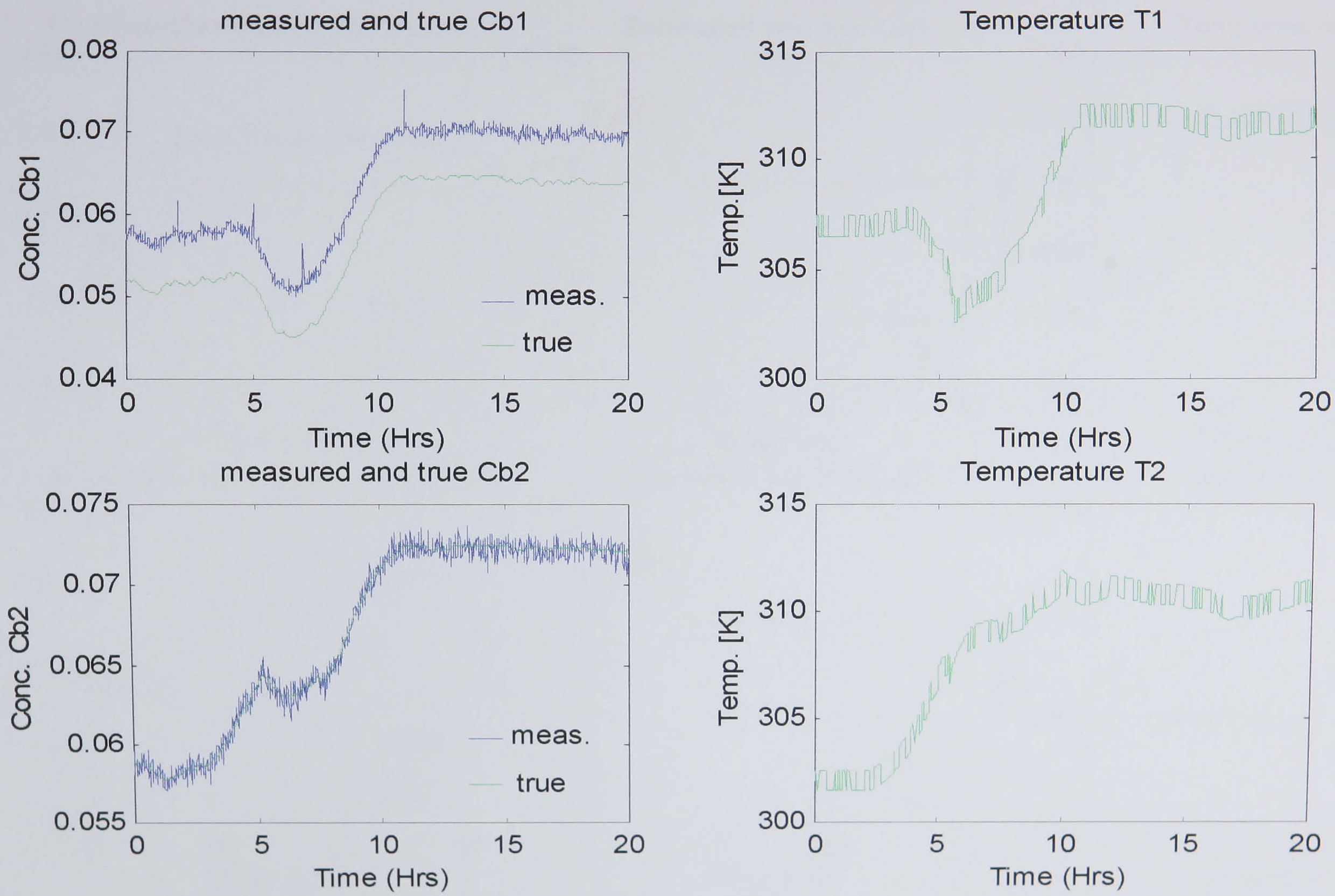


Figure (8-6): Measured and true C_{b1} and C_{b2} as well as trends of the manipulated variables T_1 and T_2 bias and outliers on C_{b1}, data reconciliation disabled, objective: to maximise C_{b2}.

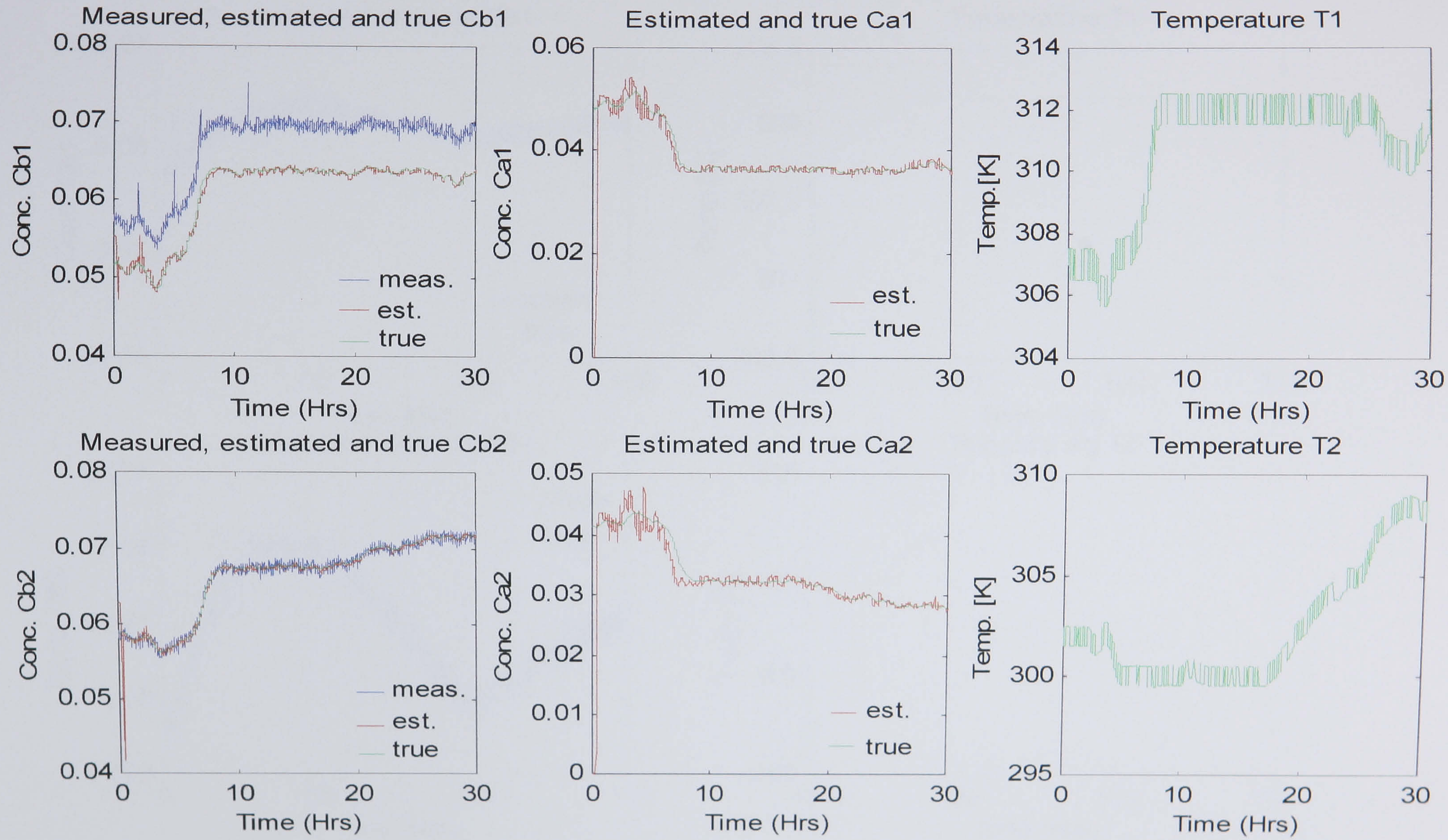
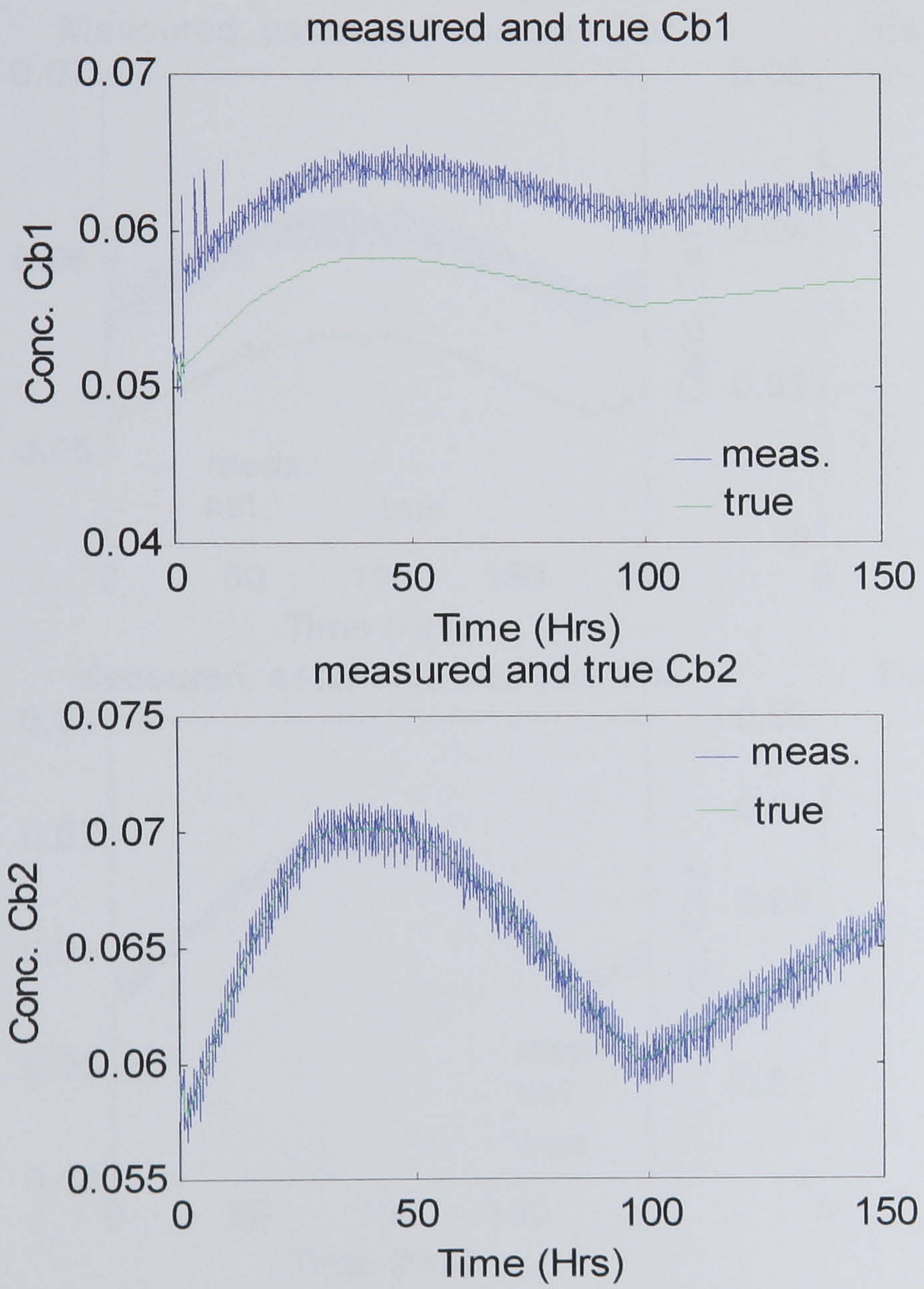


Figure (8-7): Measured, estimated and true C_{b1} and C_{b2}, estimated and true C_{a1} and C_{a2}, trends of manipulated variables T_1 and T_2, bias and outliers on C_{b1}, data reconciliation enabled, objective: to maximise C_{b2}.



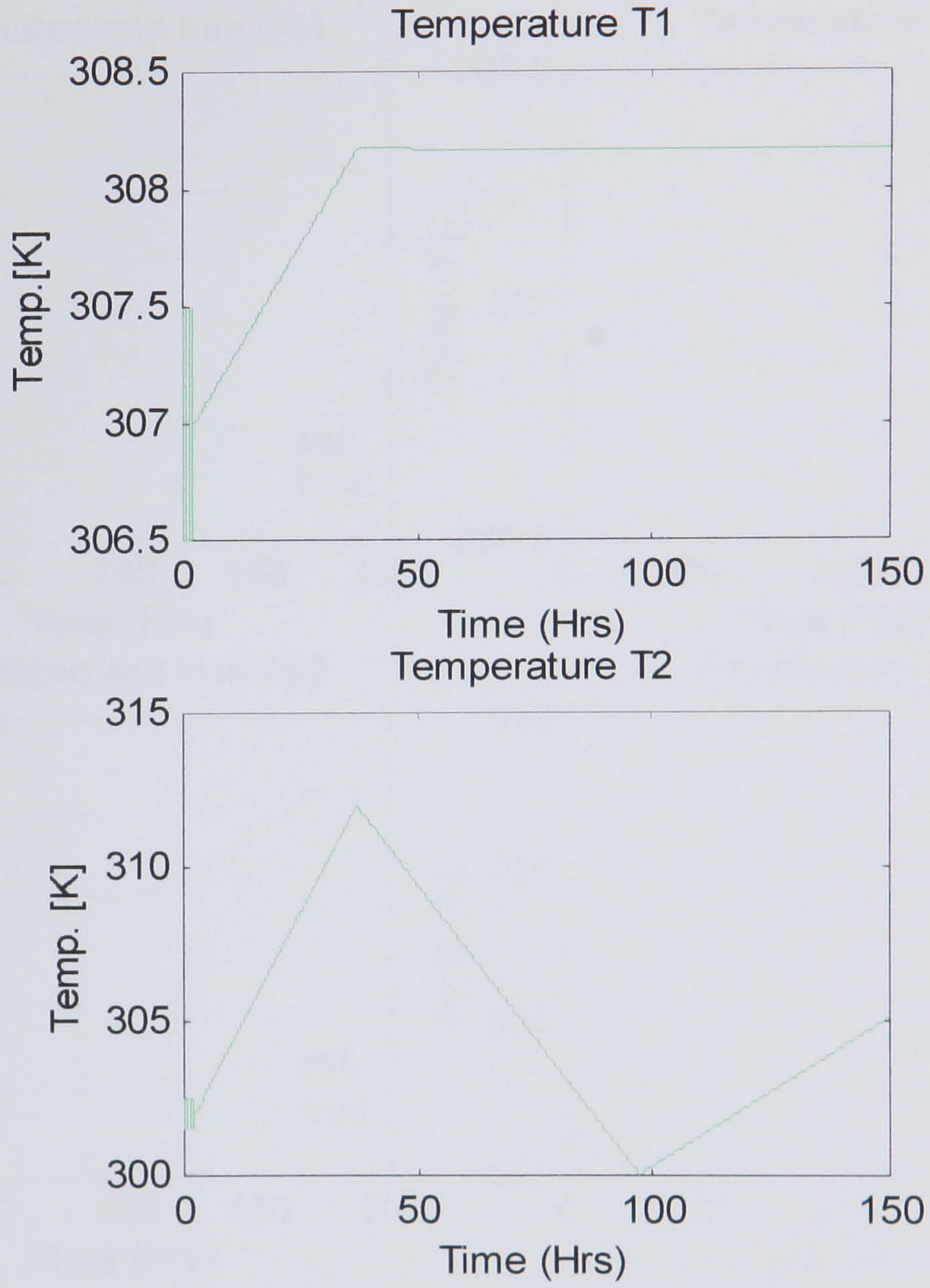


Figure (8-8): Measured and true C_{b1} and C_{b2} as well as trends of the manipulated variables T_1 and T_2, bias and outliers on C_{b1}, model adaptation disabled at 2 hours, data reconciliation disabled, objective: to maximise C_{b2}.

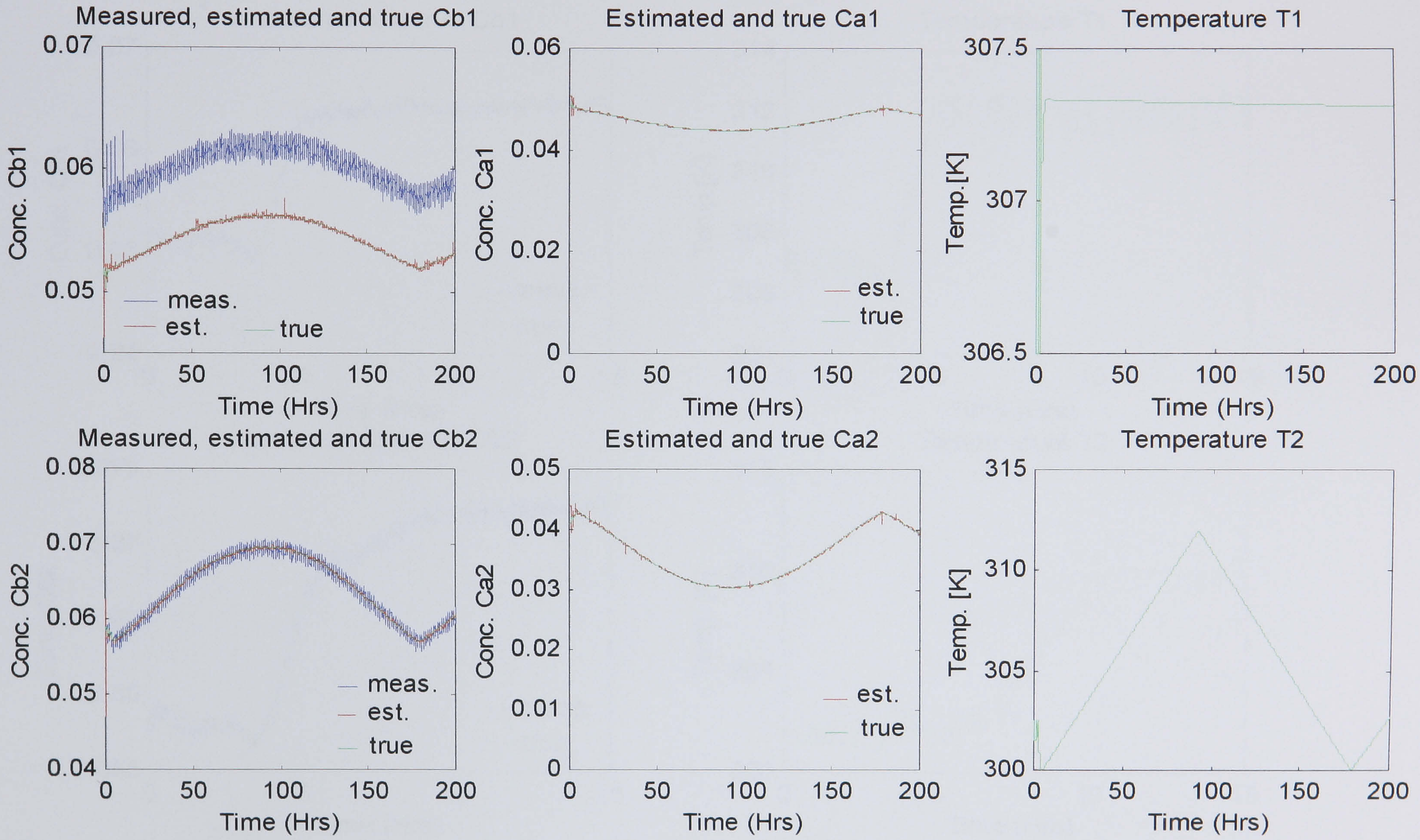
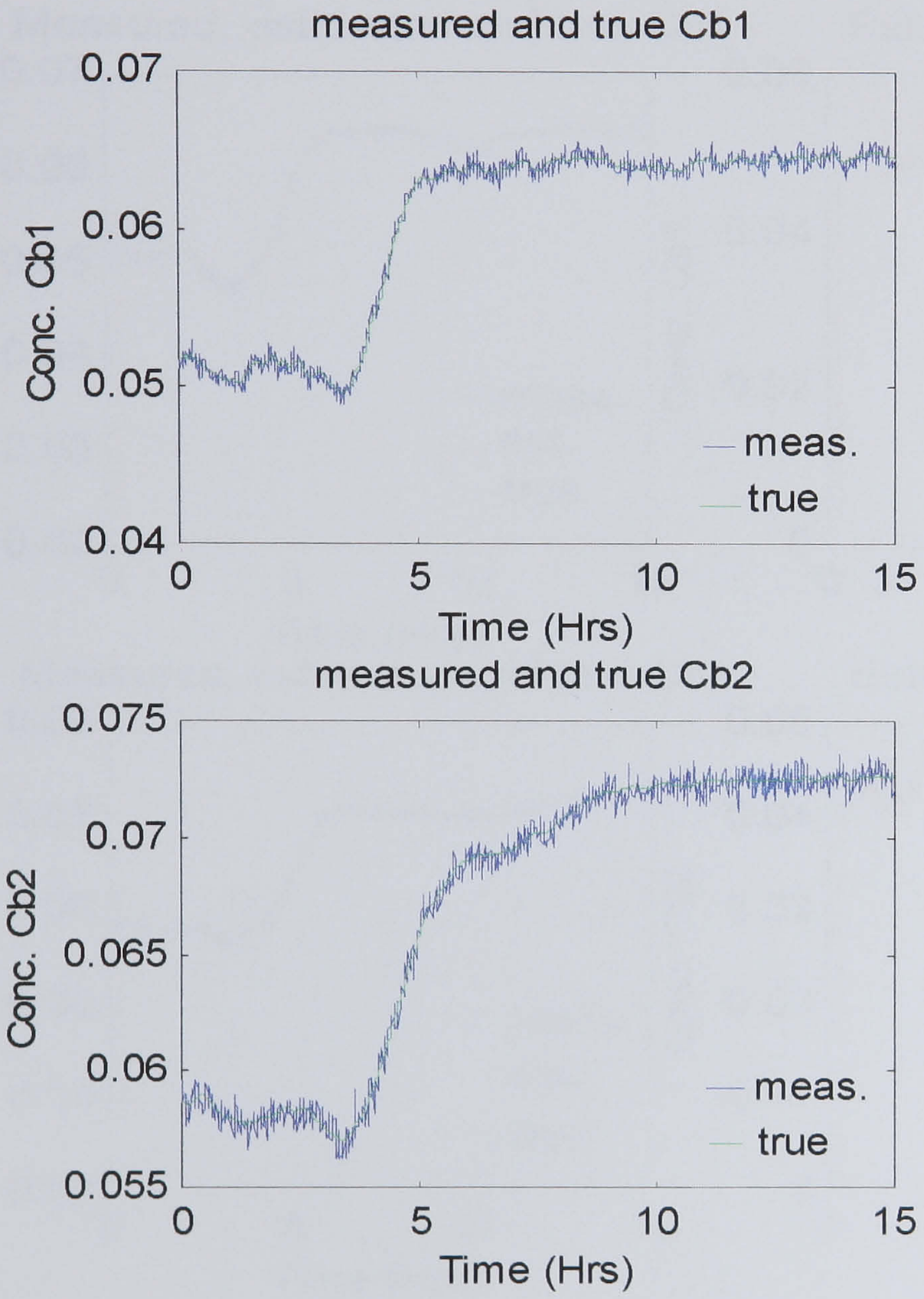


Figure (8-9): Measured, estimated and true C_{b1} and C_{b2}, estimated and true C_{a1} and C_{a2}, trends of manipulated variables T_1 and T_2, bias and outliers on C_{b1}, model adaptation disabled after 2 hours, data reconciliation enabled, objective: to maximise C_{b2}.



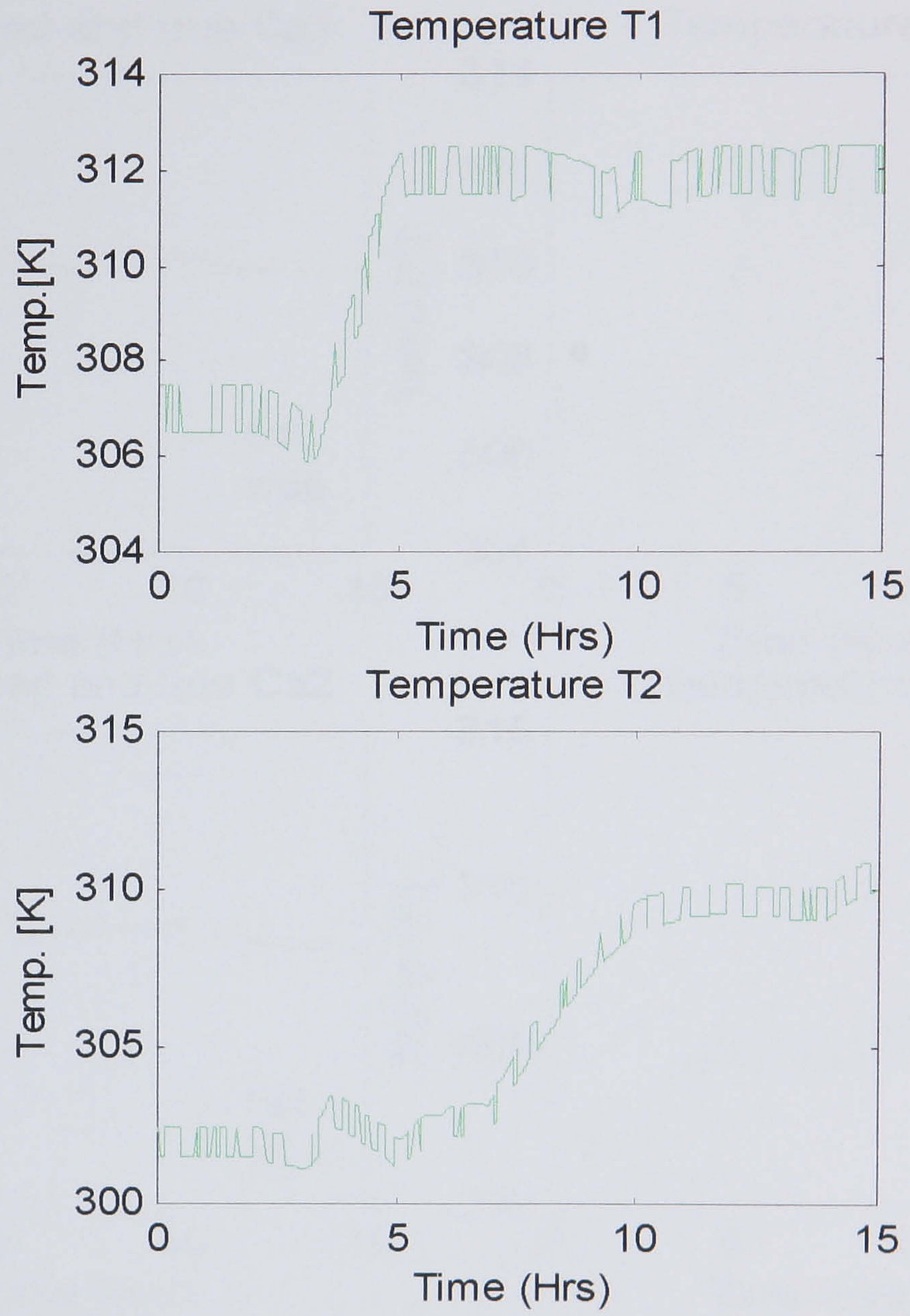


Figure (8-10): Measured and true C_{b1} and C_{b2} as well as trends of the manipulated variables T_1 and T_2, no bias or outliers present, data reconciliation disabled, objective: to maximise C_{b2}.

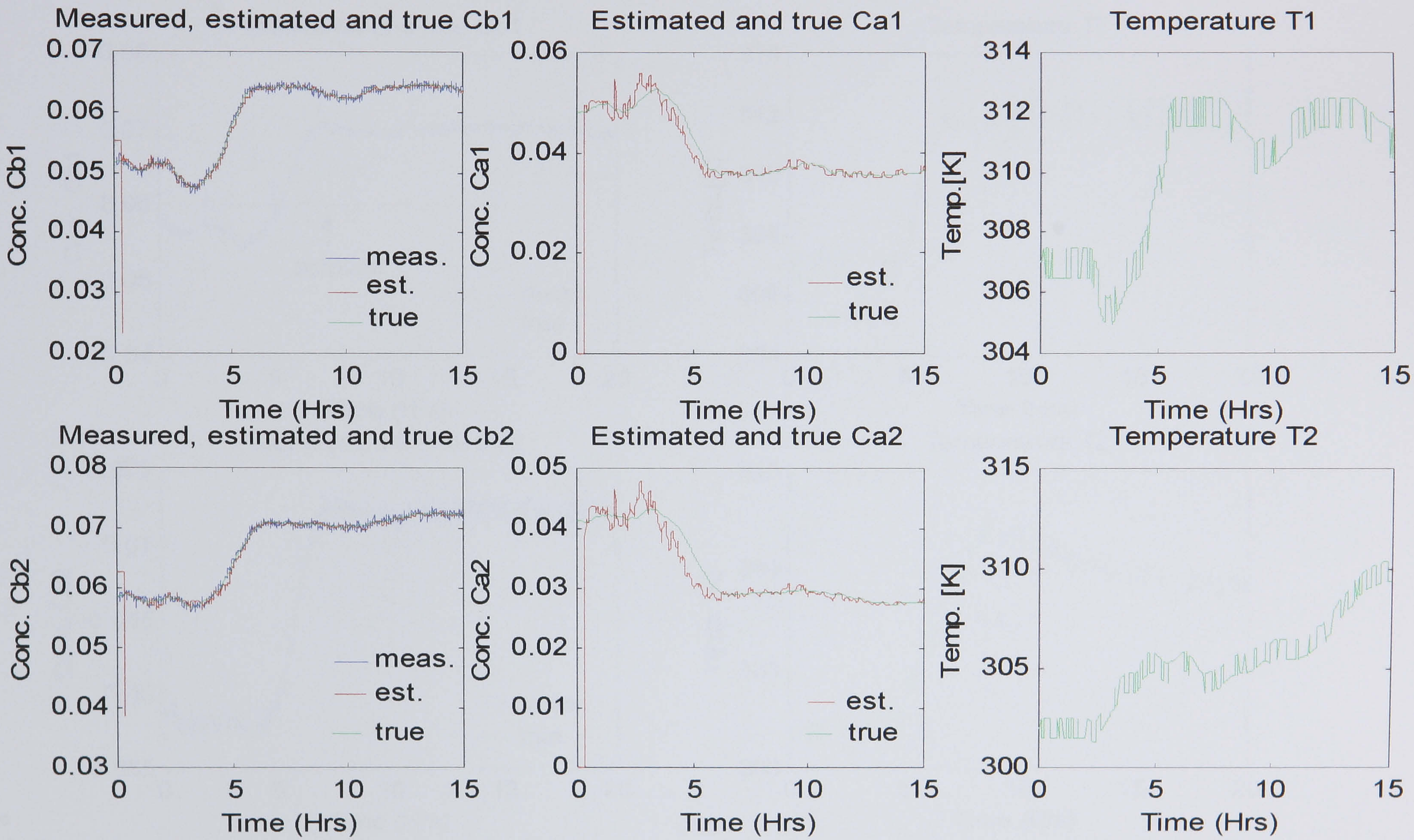
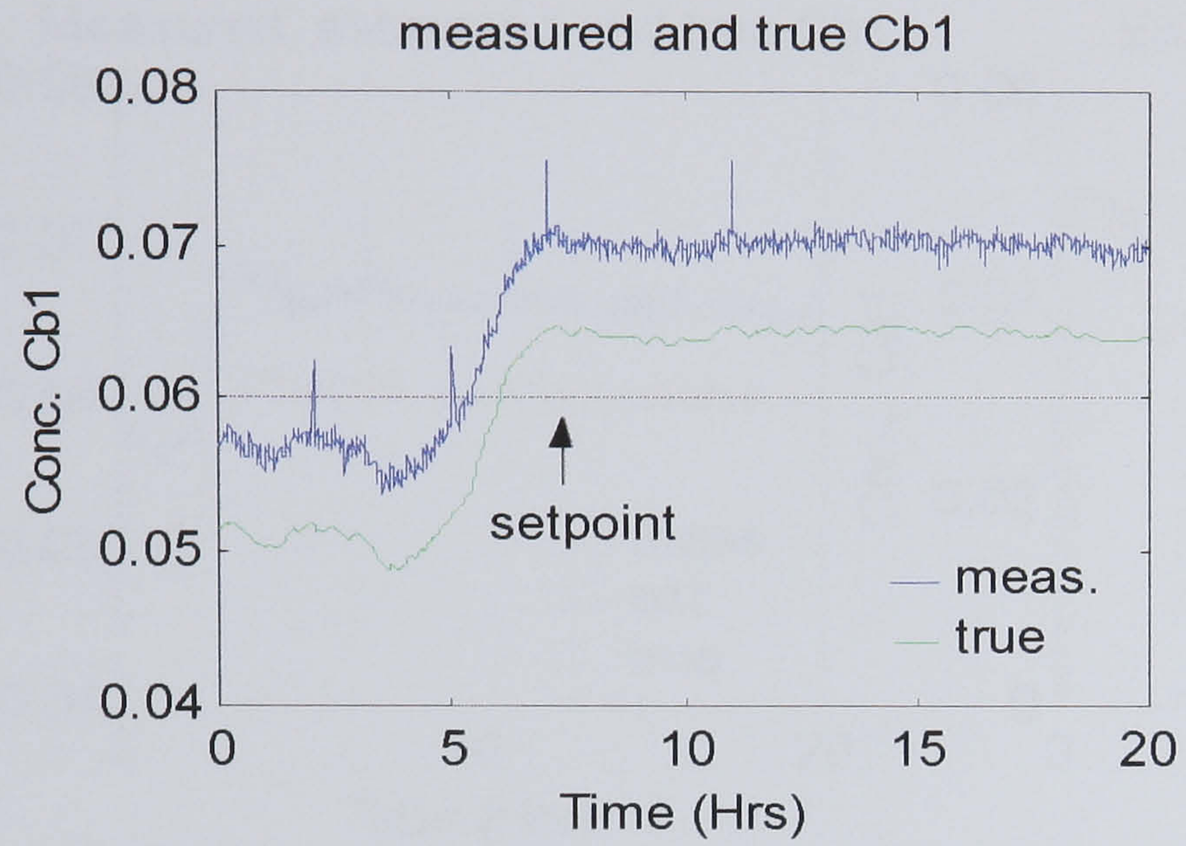
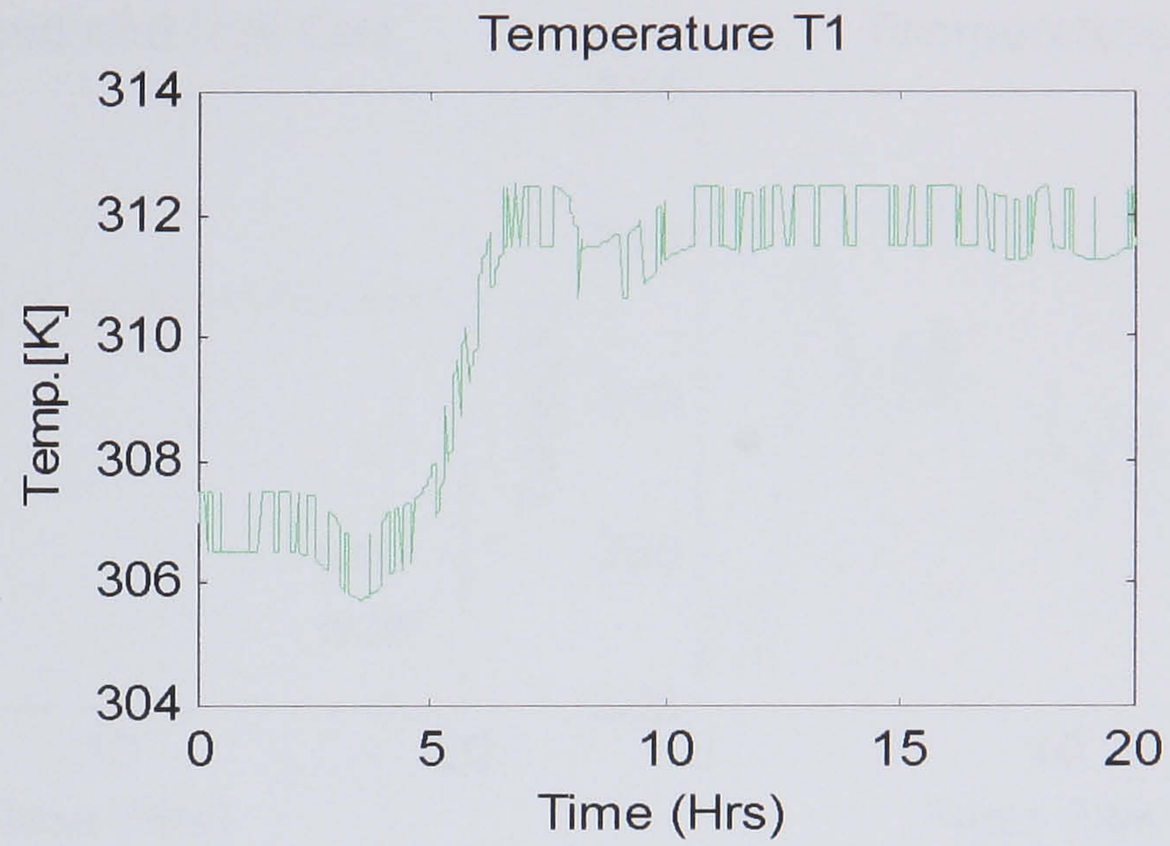
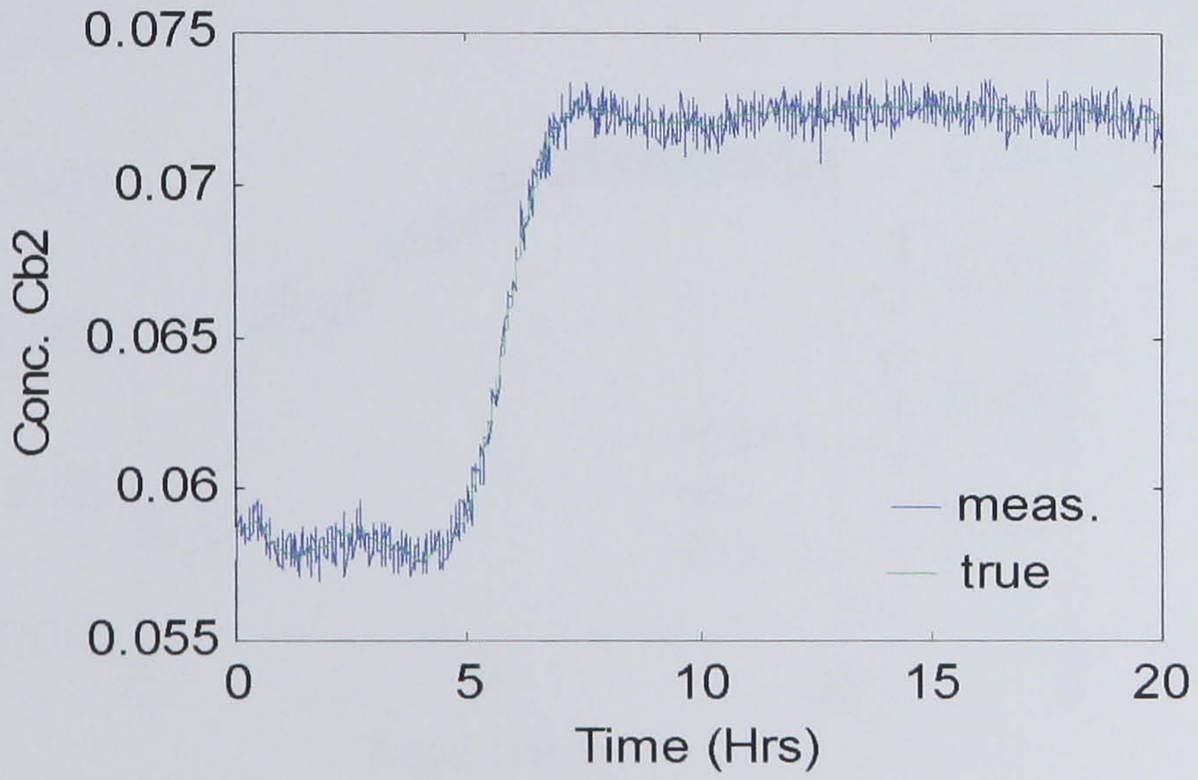


Figure (8-11): Measured, estimated and true C_{b1} and C_{b2}, estimated and true C_{a1} and C_{a2}, trends of manipulated variables T_1 and T_2, no bias or outliers present, data reconciliation enabled, objective: to maximise C_{b2}.







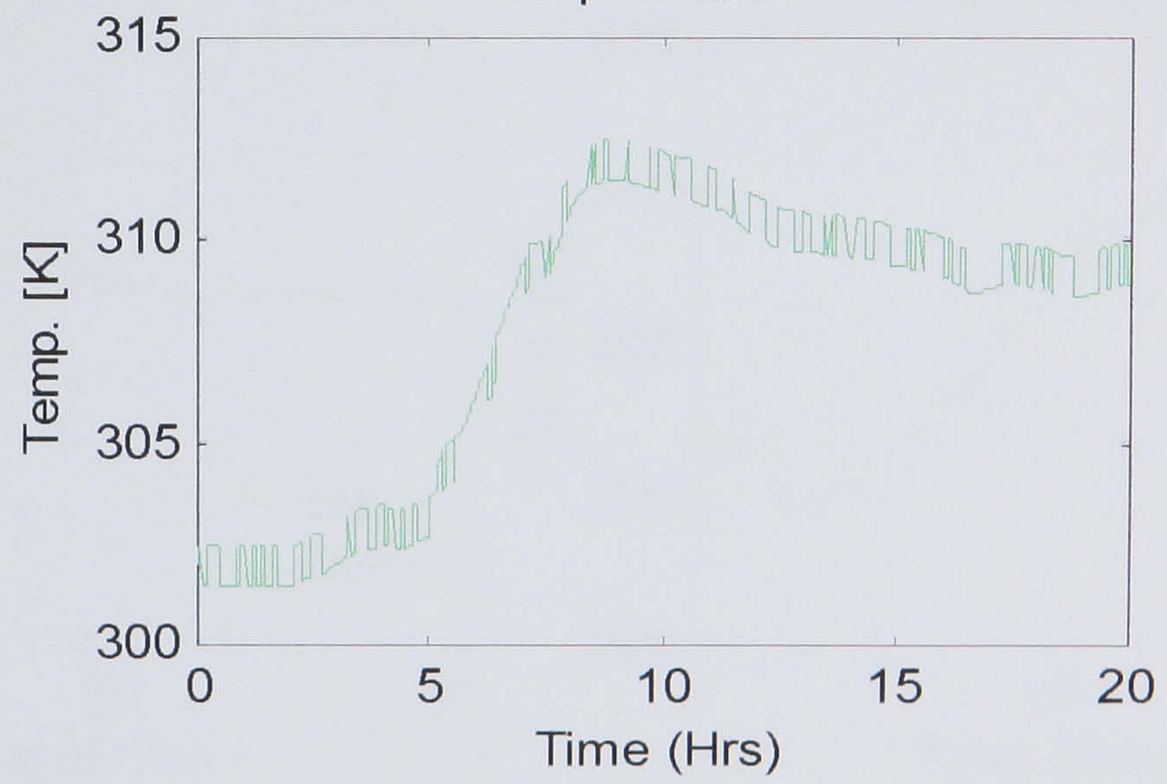


Figure (8-12): Measured and true C_{b1} and C_{b2} as well as trends of the manipulated variables T_1 and T_2, bias and outliers on C_{b1}, data reconciliation disabled, objective: combined (regulate C_{b1} at 0.060 while maximising C_{b2}).

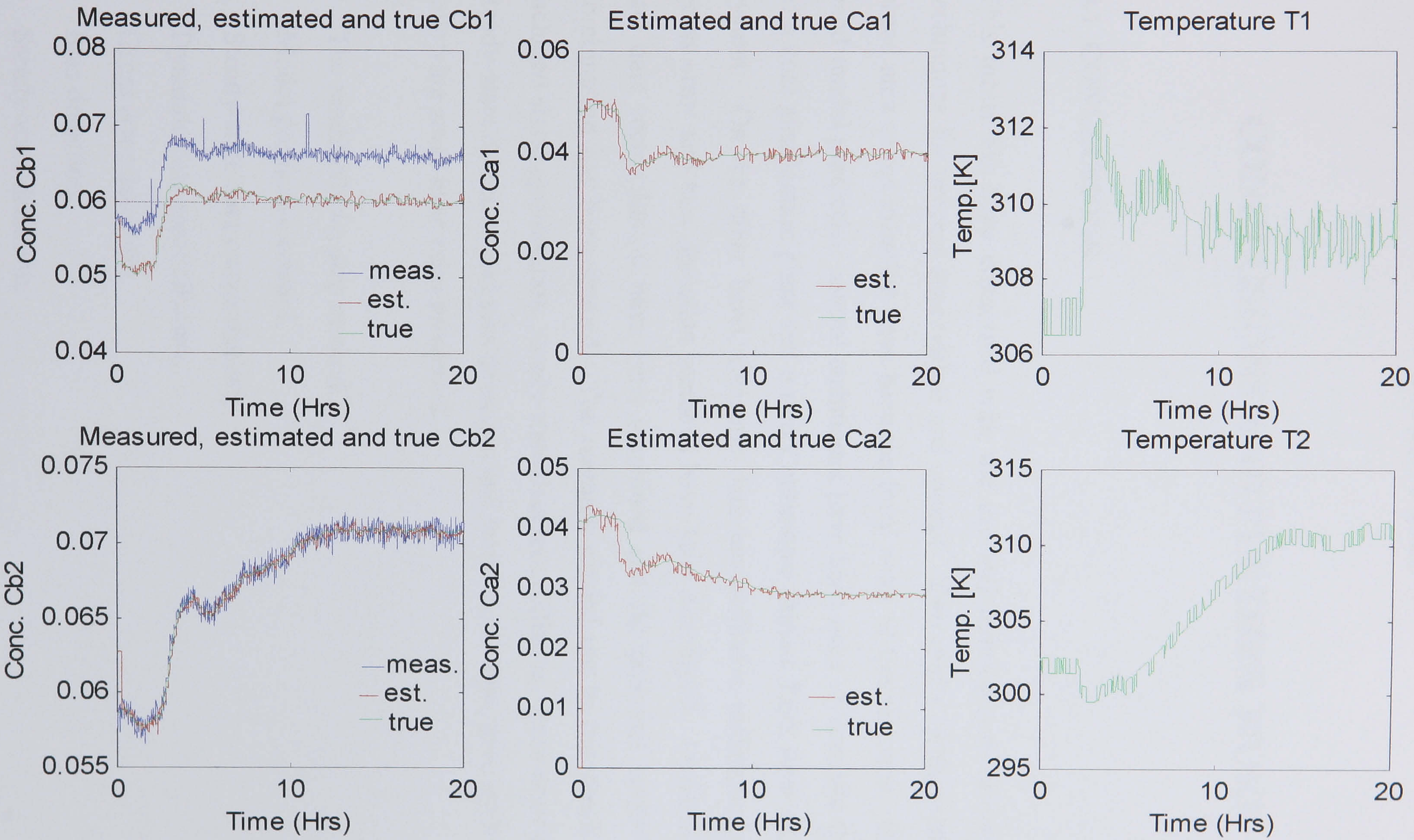


Figure (8-13): Measured, estimated and true C_{b1} and C_{b2}, estimated and true C_{a1} and C_{a2}, trends of manipulated variables T_1 and T_2, bias and outliers on C_{b1}, data reconciliation enabled, objective: combined (regulate C_{b1} at 0.060 while maximising C_{b2}).

CHAPTER 9

CONCLUSIONS AND FURTHER WORK

9.1 CONCLUSIONS

This thesis has been concerned with the use and development of optimisation techniques for process supervision and control. Two major fields related to the broad area of optimisation have been the focal point of this research. On the one hand model predictive control techniques have been used to simulate the control of a coal gasification plant and a two Continuous Stirred Tank Reactors (CSTR) system. On the other hand, dynamic data reconciliation techniques including gross error and bias detection methods have been developed. Predictive control and data reconciliation have been combined and the potential benefits of this combination has been studied. The research carried out in this thesis has also included static optimisation, steady-state data reconciliation, gross error detection, steady-state detection and bias detection and estimation. Extensive reviews of the following areas have been presented:

- The modified two step method,
- Model predictive control,
- Steady-state data reconciliation,
- Dynamic data reconciliation,
- Gross error detection,
- Bias detection,
- Steady-state detection.

Testing of the algorithms developed has been done using models of two chemical processes: a coal gasifier and a two CSTR system. Simulations were carried out using the industrial process system simulator Aspen-OTISS using custom designed C and C++ modules.

A state-space model predictive control algorithm developed at City University (Becerra et al., 1998a) has been applied to a benchmark challenge process set by ALSTOM Mechanical Engineering Centre. The process is a coal gasification plant used for the generation of environmentally clean power from coal. The challenge involved the control of models of the coal gasifier at different operating conditions where the dynamics of the process change significantly from one operating condition to the next. The scheme employed was shown to be successful for the majority of cases set by the challenge but had difficulty dealing with cases involving considerable departure of the process dynamics from the identified model.

A static data reconciliation module which uses sequential quadratic programming has been developed in C/C++ and interfaced with OTISS. This module has capabilities of estimating systematic bias, physical parameters and unmeasured variables. Simulations on a two CSTR process have shown that under normal operating conditions, the technique used accurately reconciles process data in the majority of cases and quickly finds the correct solution. This was observed even in extreme conditions when the bias on the measurements was in the range of -50% to +50% of the nominal values. The technique also proved successful in the estimation of unknown physical parameters even in the presence of systematic bias on the measurements. In the very few cases where the algorithm did not perform so well, it may be said that the problem itself is ill conditioned or that too much is being demanded from the algorithm given the limited information available.

Although the static data reconciliation module showed some robustness when applied to dynamic data, errors in the estimation can sometimes occur due to the

change in process dynamics and the consequent differences between the real process and the static model used for the reconciliation. This fact points to the need for dynamic data reconciliation techniques that use dynamic models when the process data is dynamic. A further limiting factor of the static data reconciliation module is that for proper estimation of systematic bias on the measurements, it has to be known *a priori* which measurements (if any) are biased.

In order for a data reconciliation technique to be able to choose from a static and a dynamic version depending on the state of the process, a steady-state detection algorithm was implemented. This is especially important because even a so-called *steady-state* process often departs from its normal operating point. The algorithm implemented in software and tested on the two CSTR system was suggested by Cao and Rhinehart (1995). Some slight modifications were made to overcome certain undesirable effects in the original algorithm at the transition stage between steady-state and non-steady-state conditions. Simulation results have shown that the algorithm used can accurately and efficiently detect when the process is at steady-state. Further testing carried out involved the use of the static data reconciliation module along with the steady-state detection algorithm. Information from the steady-state detection algorithm regarding the state of the process was used successfully to either enable or disable the static data reconciliation module.

The static data reconciliation module has been used to improve the performance of a static optimisation scheme. The optimisation method used was the modified two step algorithm (ISOPE), as implemented by Becerra and Roberts (2000). Simulations have shown that where corrupted data was used directly for optimisation, the results were not desirable even in the absence of systematic biases. However, using static data reconciliation to adjust the measurements and estimate systematic bias (if any) prior to optimisation improves the response considerably.

As mentioned previously static data reconciliation suffers from a major drawback and that is the possible inaccuracy of the estimates when the data being reconciled is from a dynamic process. To overcome this, a moving horizon estimation algorithm (Becerra, 1999) was used to reconcile dynamic process data. Unlike the static data reconciliation module, this estimator uses a dynamic model for the reconciliation procedure. A further drawback of the static data reconciliation algorithm is the requirement for prior information on which measurements are biased. A procedure for the detection and identification of systematic bias has been devised and implemented in software and interfaced with OTISS.

A procedure for the detection of outliers in the measurements has also been implemented and tested. The algorithm suggested by Chen and Romagnoli (1998) and which uses cluster analysis, has been modified here. A problem was foreseen in the original algorithm especially in the case where there are one or more measurements in the data window which are similar in magnitude to the outlier. It has been shown that in this case the algorithm would fail to detect the outlier. With a simple modification where averages of measurements in the data window are used, this problem has been successfully addressed. For comparison purposes, the original algorithm was implemented and it was found that the modified algorithm was considerably more accurate in detecting outliers.

Simulations from the moving horizon estimator have shown that using this type of estimator, dynamic process data can be effectively reconciled. In terms of results, the dynamic data reconciliation compares favourably against the static data reconciliation algorithm in the presence of a transient. The former enjoys the advantage that changes in the process dynamics will not result in differences between the real process and the model used for the reconciliation procedure and thus more accurate estimates can be expected.

Results have shown that through the use of a gross error (outlier) detection technique, outliers on the measurements can be successfully identified and eliminated. Furthermore, by using a simple technique bias has been successfully

detected and estimated. The bias detection technique proposed is intuitive and far simpler than that put forward by McBrayer and Edgar (1995), for example. A further algorithm which is essentially a simplified version of the method suggested by McBrayer and Edgar has also been proposed but not implemented. Tests involving the gross error and bias detection algorithms have been carried out using the two CSTR process. Simulations on the algorithms have been done separately as well as when both algorithms are enabled simultaneously. The author was inspired by the idea of combining bias and gross error detection algorithms from Chen and Romagnoli (1998) but has failed to find any researchers who have carried this out. The bias and gross error detection algorithms developed in this thesis have been successfully combined and their effects have been studied.

Simulations on the dynamic data reconciliation module when the bias and gross error algorithms are enabled have shown that repeated switching of the systematic bias from one measurement to another can cause the detection algorithm to become inaccurate at some stage. Fortunately, in real plants, the bias is unlikely to keep moving from one measurement to another since bias is normally caused by incorrectly installed or calibrated measurement devices. What is likely to occur though, is the presence of multiple biases. The bias detection algorithm as it stands can only handle single biases at a time. Other methods studied by the author suffer from the same setback, for example McBrayer and Edgar (1995).

The dynamic data reconciliation techniques developed in this thesis have been applied to a model predictive control scheme. Simulation results have shown that the use of data reconciliation techniques significantly enhances the performance of a model predictive controller in the cases where the objective is purely regulatory and when the objective is a combination of regulatory and economic terms. However, in the case where the objective is purely economic, the use of data reconciliation techniques has been found to cause the performance of a model predictive control scheme to deteriorate. This may have something to do with the overall combined result of the objectives: that relating to the optimisation and that

relating to the reconciliation procedure. A further explanation of this may be that the result is relevant to this particular application and that a generalisation may not be well founded.

In summary, static as well as dynamic data reconciliation techniques have been successfully implemented. A steady-state detection algorithm has been improved and implemented. Testing of this algorithm was done in conjunction with Static Data Reconciliation (SDR) where the SDR module was enabled and disabled depending on whether or not the process was deemed to be at steady-state. A bias identification method has been developed and used in conjunction with dynamic data reconciliation. Furthermore, an algorithm for the detection of outliers in process data has been improved and used simultaneously with dynamic data reconciliation. The bias identification and outlier detection methods have been successfully combined. All these algorithms have been tested on a two CSTR system using industrial simulation software. A model predictive control scheme has been used to control a coal gasification plant. The scheme was shown to be successful for the majority of the cases set by the challenge but had difficulty in dealing with cases involving considerable departure of the process dynamics from the identified model. Static and dynamic data reconciliation procedures developed in this thesis have been applied to static optimisation and model predictive control schemes. Finally, the aims and objectives of this work set out at the beginning of the thesis have been successfully achieved.

9.2 SUGGESTIONS FOR FURTHER RESEARCH

The following items relevant to the work in this thesis are recommended for further research:

1. The implementation and testing of the other proposed bias detection algorithm which is a simplification of the method put forward by McBrayer and Edgar (1995) is recommended.

2. The extension of the bias detection algorithm to handle multiple biases is essential.

3. The static data reconciliation algorithm has been applied to the two step method, ISOPE. A similar exercise where the dynamic data reconciliation techniques could be applied to the dynamic version of ISOPE (DISOPE) may be beneficial, in particular, for the optimisation of batch processes.

4. In applying dynamic data reconciliation techniques to model predictive control it was observed that in the case of a purely economic objective the performance of the model predictive control scheme deteriorated. In order to properly explain these results further investigation is recommended.

5. Further testing of the algorithms proposed and implemented in this thesis is required by way of simulations using different case studies.

6. The static and dynamic data reconciliation modules should be used together in a modular fashion where the steady-state detection algorithm may act as the decision unit which enables and disables the appropriate module depending on whether or not the process is at steady-state. This way, it is possible to avoid having to solve a dynamic optimisation problem when the data at hand is static. Conversely, when the data is dynamic then the use of a static data reconciliation scheme is avoided.

REFERENCES

Abu-el-zeet Z.H., Becerra V.M. and Roberts P.D. (1999a): "Automatic Identification of Steady-state", Technical Report no. 170, CERC, City University. London, UK, July 1999.

Abu-el-zeet Z.H., Becerra V.M. and Roberts P.D. (1999b): "Steady State Data Reconciliation and Estimation of Systematic Biases", Technical Report no. 171, CERC, City University, London, UK, October 1999.

Abu-el-zeet Z.H., Becerra V.M. and Roberts P.D. (2000): "Data reconciliation and steady-state detection applied to a chemical process", UKACC International Conference (Control 2000), University of Cambridge, Cambridge, U.K., 4-7th September 2000.

Albuquerque J.S. and Biegler L.T. (1995): "Decomposition Algorithms for On-Line Estimation with Nonlinear Models", Computers and Chemical Engineering, Vol. 19, No. 10, pp. 1031-1039.

Albuquerque J.S. and Biegler L.T. (1996): "Data Reconciliation and Gross Error Detection for Dynamic Systems", AIChE Journal, Vol. 42, No.10, October 1996.

Almasy G.A. and Mah R.S.H. (1984): "Estimation of Measurement Error Variances from Process Data", Ind. Eng. Chem. Process Des. Dev., 23, pp. 779-784.

Almasy G.A. and Sztano T. (1975): "Checking and Correction of Measurements on the Basis of Linear System Model", Problems of Control and Information Theory, Vol. 4(1), pp. 57-69.

Anderson T.W. (1958): "An Introduction to Multivariate Statistical Analysis", John Wiley & Sons Inc., London, pp. 101-115.

Augustin P. and Roberts P.D. (1993): "On-line hierarchical steady-state optimising control of a power plant using the ISOPE technique with dynamic information", Transactions of the Institute of Measurement and Control, Vol. 15, No. 4, pp. 170-177.

Bagajewicz M.J. (1996): "On the Probability Distribution and Reconciliation of Process Plant Data", Computers and Chemical Engineering, Vol. 20, No. 6/7, pp. 813-819.

Bagajewicz M.J. and Jiang Q. (1997): "Integral Approach to Plant Linear Dynamic Reconciliation", AIChE Journal, October 1997, Vol. 43, No. 10, pp. 2546-2558

Bagajewicz M.J. and Jiang Q. (1998): "Gross Error Modelling and Detection in Plant Linear Dynamic Reconciliation", Computers and Chemical Engineering, Vol. 22, No.12, pp. 1789-1809.

Balchen J.G., Ljungquist D. and Strand S. (1992): "State-Space Predictive Control", Chemical Engineering Science, Vol. 47, No. 4, pp. 787-807.

Bamberger W. and Isermann R. (1978): "Adaptive On-Line Steady-State Optimisation of Slow Dynamic Processes", Automatica, Vol. 14, pp. 223-230.

Basseville M. (1988): "Detecting Changes in Signals and Systems – A Survey", Automatica, Vol. 24, No. 3, pp. 309-326.

Bazaraa M.S., Sherali H.D. and Shetty C.M. (1993): "Nonlinear Programming, Theory and Algorithms", John Wiley and Sons.

Becerra V.M. (1994): "Development and Applications of Novel Optimal Control Algorithms", Ph.D. Thesis, City University, London, U.K.

Becerra V.M. (1998): *C++ implementation of Sequential Quadratic Programming*, personal communication.

Becerra V.M. (1999): "A moving horizon estimation algorithm", personal communication, technical note DDR-99-1 and accompanying C++ code, CERC, City University, London.

Becerra V.M., Abu-el-zeet Z.H. and Roberts P.D. (1998): "Use of a predictive Control Technique for the GEC-ALSTHOM MEC Gasifier Challenge", Seminar entitled 'MEC Benchmark Challenge On Gasifier Control', Wednesday 24th June 1998, Coventry University, Coventry, U.K.

Becerra V.M., Abu-el-zeet Z.H. and Roberts P.D. (1999a): "Integrating Predictive Control and Economic Optimisation", Computing and Control Engineering Journal, Vol. 10, No. 5, pp.198-208.

Becerra V.M., Abu-el-zeet Z.H. and Roberts P.D. (1999b): "Use of a Predictive Control Technique for the ALSTOM Gasifier Benchmark Challenge", Paper submitted for consideration in Journal of Systems and Control Engineering (Special Issue), Proceedings of the Institution of Mechanical Engineers, Part I.

Becerra V.M., Abu-el-zeet Z.H. and Roberts P.D. (1999c): "Integrating Predictive Control and Economic Optimisation", IEE Colloquium on Model Predictive Control: Techniques and Applications, 29th April 1999, IEE, London.

Becerra V.M. and Roberts P.D. (1996): "Dynamic integrated system optimisation and parameter estimation for discrete time optimal control of nonlinear systems", International Journal of Control, Vol. 63, No.2, pp. 257-281.

Becerra V.M. and Roberts P.D. (2000): "Formulation, implementation and application of a practical version of the modified two-step method for optimising control", Technical Report No. 173, CERC, City University, London, April 2000.

Becerra V.M., Roberts P.D. and Griffiths G.W. (1997): "Process optimisation using predictive control: Application to a simulated reaction system", Proceedings of IFAC Symposium on Advanced Control of Chemical Processes, ADCHEM '97, Banff, Canada, 9-11 June 1997, pp. 629-634.

Becerra V.M., Roberts P.D. and Griffiths G.W. (1998a): "Novel Developments in Process Optimisation using Predictive Control", Journal of Process Control, Vol.8, No. 2, pp. 117-138.

Becerra V.M., Roberts P.D. and Griffiths G.W. (1998b): "Integrated Predictive Control and Economic Optimisation: Decomposition and Hierarchical Structure". LSS'98, Patras, Greece, 15-17 July 198.

Becerra V.M., Roberts P.D. and Griffiths G.W. (1998c): "Dynamic Data Reconciliation For Sequential Modular Simulators". UKACC Control '98 International Conference,Swansea, U.K., 1-4 September 1998.

Becerra V.M., Roberts P.D. and Griffiths G.W. (1999d): "Dynamic Data Reconciliation for a Class of nonlinear differential equation models using the extended Kalman Filter", IFAC World Congress, Beijing, China, 5-9 July 1999.

Becerra V.M., Roberts P.D. and Griffiths G.W. (1999e): "Dynamic Data Reconciliation For Sequential Modular Simulators – Application to a Mixing Process", Paper sent for consideration to ECC 99, Karlrube, Germany, 31st August – 3rd September 1999.

Becerra V.M., Roberts P.D., Lin J. and Griffiths G.W. (1996), "Novel Developments in Process Optimisation using Predictive Control", Proceedings of the 13th IFAC World Congress, San Francisco, USA, 30th June - 5th July 1996, Vol. M, pp. 115-120.

Betha R.M. and Rhinehart R.R. (1991): "Applied Engineering Statistics", Marcel Dekker, New York, NY.

Biegler L.T. (1984): "Solution of Dynamic Optimization Problems by Successive Quadratic Programming and Orthogonal Collocation", Short Note, Computers and Chemical Engineering, Vol. 8, No. 3/4, pp. 243-248.

Bodington C.E. (1995): "Planning Scheduling and Control Integration in the Process Industries", Chapter 4, pp. 67-85, McGraw-Hill, London.

Brdys M. and Roberts P.D. (1987): "Convergence and Optimality of Modified two-step algorithm for integrated system optimisation and parameter estimation". International Journal of Systems Science, Vol.18, pp. 1305-1322.

Bussani G., Chiari M., Grottoli M.G., Pierucci S., Faravelli T., Ricci G. and Gioventu G. (1995): "Application of Data Reconciliation and Optimisation Procedure to Hydrogen Plant", Computers and Chemical Engineering, Vol. 19. Suppl., pp. S299-304.

Camacho E.F. and Bordons C. (1999): "Model Predictive Control", Springer-Verlag, London.

Cao S. and Rhinehart R.R. (1995): "An Efficient Method for on-line identification of steady state", Journal of Process Control, Vol.5, No. 6, pp. 363-374.

Cao S. and Rhinehart R.R. (1997): "Critical Values for a steady-state identifier", Journal of Process Control, Vol.7, No.2, short note, pp. 149-152.

Chen J. and Romagnoli J.A. (1998): "A Strategy for Simultaneous Dynamic Data Reconciliation and Outlier Detection", Computers and Chemical Engineering, Vol. 22, No. 4/5, pp. 559-562.

Chen X., Pike R.W., Hertwig T.A. and Hopper J.R. (1998a): "Optimal Implementation of On-Line Optimisation", Computers and Chemical Engineering, Vol. 22, Suppl., pp. S435-442.

Chen J., Bandoni A. and Romagnoli J.A. (1998b): "Outlier detection in process data", Computers and Chemical Engineering, Vol. 22, No. 4/5, pp. 641-646.

Clarke D.W., Mohtadi C. and Tuffs P.S. (1987): "Generalized Predictive Control - Part1, The Basic Algorithm", Automatica, Vol. 23, No. 2, pp. 137-148.

Crow E.L., Davis, F.A. and Maxfield, M.W. (1955): "Statistics Manual", Dover Publications, New York, N.Y, 63.

Crowe C.M. (1986): "Reconciliation of Process Flow Rates by Matrix Projection – Part II: The Nonlinear Case", AIChE Journal, Vol. 32, No. 4, pp. 616-623.

Crowe C.M. (1996): "Data Reconciliation – Progress and Challenges", Journal of Process Control, Vol. 6, No. 2/3, pp. 89-98.

Crowe C.M, Garcia Campos Y.A. and Hrymak A. (1983): "Reconciliation of Process Flow Rates by Matrix Projection – Part I: The Linear Case", AIChE Journal, Vol. 29, No.6, pp.881-888.

Cutler C., Morshedi A. and Haydel J. (1983): "An Industrial Perspective on Advanced Control", AIChE Annual Meeting, Washington D.C.

Darouach M. and Zasadzinski M. (1991): "Data Reconciliation in Generalized Linear Dynamic Systems", AIChE Journal, Vol. 37, No. 2, pp. 193-201.

Dorf R.C: "Modern Control Systems", Sixth Edition, Addison-Wesley. Reading. England, 1992.

Eaton J.W. and Rawlings J.B. (1992): "Model Predictive Control of Chemical Processes", Chemical Engineering Science, Vol. 47, No. 4, pp. 705-720.

Ellis J.E., Kandamby N.H. and Roberts P.D. (1993): "Rationalization and application of algorithms for integrated system optimisation and parameter estimation", International Journal of Systems Science, Vol. 24, No. 2. pp. 219-229.

Ellis J.E., Kambhampati C., Sheng G. and Roberts P.D. (1988): "Approaches to the optimizing control problem", International Journal of Systems Science, Vol. 19, No. 10, pp. 1969-1985.

Fillon M., Meyer M., Pingaud H. and Joulia X. (1995): "Data Reconciliation Based on Elemental Balances Applied to Batch Experiments", Computers and Chemical Engineering, Vol. 19, Suppl., pp. S293-S298.

Garcia C.E. and Morari M. (1981): "Optimal Operation of Integrated Processing Systems. Part I: Open-loop On-line Optimising Control", AIChE Journal, 27(6), 960-968.

Garcia C.E. and Morshedi A.M. (1986): "Quadratic Programming Solution of Dynamic Matrix Control (QDMC), Chemical Engineering Commun., Vol. 46, pp. 73-87.

Garcia C.E., Prett D.M. and Morari M. (1989): "Model Predictive Control: Theory and Practice - a survey", Automatica, Vol. 25, No. 3, pp.335-348.

Gattu G. and Zafiriou E. (1992), "Nonlinear Quadratic Dynamic Matrix Control with State Estimation", Industrial Engineering and Chemical Research, Vol. 31, No. 4, pp. 1096-1104.

Gawthrop P.J., Demircioglu H. and Siller-Alcala I.I. (1998): "Multivariable Continuous-Time Generalised Predictive Control: A State-Space Approach to Linear and Nonlinear Systems", IEE Proceedings - Control Theory Applications, Vol. 145, No. 3, pp. 241-250.

Ham P.G., Cleaves G.W. and Lawlor J.K. (1979): "Operation Data Reconciliation: An Aid to Improved Plant Performance", 10th World Petroleum Congress, Romania, 1979.

Harikumar P. and Narasimhan S. (1993): "A Method to Incorporate Bounds in Data Reconciliation and Gross Error Detection – II. Gross Error Detection Strategies", Computers and Chemical Engineering, Vol. 17, No. 11, pp. 1121-1128.

Harris T.J. and Ross W.H. (1991): "Statistical Process Control Procedures for Correlated Observations", The Canadian Journal of Chemical Engineering, Vol. 69, pp. 48-57.

Henson M.A. and Seborg D.E. (1997): "Nonlinear Process Control", Prentice Hall, Upper Saddle River, New Jersey, pp. 325-326.

Hlavacek V. (1977): "Analysis of a Complex Plant-Steady State and Transient Behavior", Computers and Chemical Engineering, Vol. 1, pp. 75-100.

Hodouin D., Gelpe T. and Everell M.D. (1982): "Sensitivity Analysis of Material Balance Calculations – an Application to a Cement Clinker Grinding Process", Powder Technology, 32, pp. 139-153.

Hodouin D., Garon M., Remillard M. and Therien M. (1988): "Assessment of Precious Metals Distribution in Lac Mattagami Flotation Plant by Computer Mass Balance Calculation", CIM Bulletin, Vol. 81, November, No. 919, pp.62-67.

Holly W., Cook R. and Crowe C.M. (1989): "Reconciliation of Mass Flow Rate Measurements in a Chemical Extraction Plant", The Canadian Journal of Chemical Engineering, Vol. 67, pp. 595-601.

Islam K.A., Weiss G.H. and Romagnoli J.A. (1994): "Non-Linear Data Reconciliation for an Industrial Pyrolysis Reactor", Computers and Chemical Engineering, Vol. 18, Suppl., pp. S217-S221.

Johnson R.A. and Wichern D.W. (1992): "Applied Multivariate Statistical Analysis", Prentice- Hall International (UK), pp. 153-158.

Kalman R.E. (1960a): "Contributions to the theory of optimal control", Bol. Soc. Mat. Mexicana, Vol. 5, pp. 102-119.

Kalman R.E. (1960b): "A new approach to linear filtering and prediction problems", Trans. ASME J. Basic Eng., Vol. 82, pp. 34-35.

Kao C.S., Tamahane A.C. and Mah R.S.H. (1992): "Gross Error Detection in Serially Correlated Process Data", Ind. Eng. Chem. Res., 31, pp.254-262.

Karjala T.W. and Himmelblau D.M. (1996): "Dynamic Rectification of Data Via Recurrent Neural Nets and the Extended Kalman Filter", AIChE Journal, Vol. 42, No. 8, pp. 2225-2239.

Keller J.Y., Darouach M. and Krzakala G. (1994): "Fault Detection of Multiple Biases or Process Leaks in Linear Steady State Systems", Computers and Chemical Engineering, Vol. 18, No. 10, pp. 1001-1004.

Kim I.W., Kang M.S., Park S. and Edgar T.F. (1997): "Robust Data Reconciliation and Gross Error Detection: The Modified MIMT Using NLP", Computers and Chemical Engineering, Vol. 21, No.7, pp. 775-782.

Kretsovalis A. and Mah R.S.H. (1987): "Effect of Redundancy on Estimation Accuracy in Process Data Reconciliation", Chemical Engineering Science, Vol. 42, No. 9, pp. 2115-2121.

Kuehn D.R. and Davidson H. (1961): "Computer Control, II Mathematics of Control", Chemical Engineering Progress, Vol. 57, No.6, pp.44-47.

Kwon W.H. and Byun D.G. (1989): "Receding Horizon tracking control as a predictive control and its stability properties", International Journal of Control, Vol. 50, No. 5, pp1807-1824.

Lee J.H. and Ricker N.L. (1994): "Extended Kalman Filter Based Nonlinear Model Predictive Control", Industrial Engineering and Chemical Research, Vol. 33, No. 6, pp. 1530-1541.

Liebman M.J. (1991): "Reconciliation of Process Measurements Using Statistical and Nonlinear Programming Techniques", PhD. Thesis, University of Texas at Austin, May 1991.

Liebman M.J., Edgar T.F. and Lasdon L.S. (1992): "Efficient Data Reconciliation and Estimation For Dynamic Processes Using Nonlinear Programing Techniques", Computers and Chemical Engineering, Vol. 16, No. 10/11, pp. 963-986.

Lundstrom P., Lee J.H., Morari M. and Skogestad S. (1995): "Limitations of Dynamic Matrix Control", Computers and Chemical Engineering, Vol. 19, No. 4, pp. 409-421.

MacDonald R.J. and Howat C.S. (1988): "Data Reconciliation and Parameter Estimation in Plant Performance Analysis", AIChE Journal, Vol. 34, No.1, pp. 1-8.

Madron F. (1992): "Process Plant Performance, Measurement and Data Processing for Optimization and Retrofits", Ellis Horwood, Chichester, U.K.

Mah R.S.H. (1982): "Design and Analysis of Process Performance Monitoring Systems", In Chemical Process Control II, Engineering Foundation, New York, 1982, pp.525, Proceedings of the Engineering Foundation Conference, January 18-23, 1981, Sea Island, Georgia, Editors: T.F. Edgar and D.E. Seborg.

Mah R.S.H. (1987): "Data Screening", Paper presented at the FPCAPO Conference, Park City, Utah.

Mah R.S.H. (1990): "Chemical Process Structures and Information Flows". Butterworths, Stoneham, MA, Chapters 8 and 9.

Mah R.S.H., Stanley G.M. and Downing D.M. (1976): "Reconciliation and Rectification of Process Flow and Inventory Data", Ind. Eng. Chem., Process Des. Dev., Vol. 15, No.1, pp. 175-183.

Mah R.S.H., Tamhane A.C. (1982): "Detection of Gross Errors in Process Data", AIChE Journal, Vol. 28, No.5, pp. 828-830.

Mahfouf M., Linkens D.A. and Asbury A.J. (1997): "Generalised Predictive Control (GPC): A Powerful Control Tool in Medicine", IEE Proceedings - Control Theory Applications, Vol. 144, No. 1, pp. 8-14.

Mayne D.Q. and Michalska H. (1990): "Receding Horizon Control of Nonlinear Systems", IEEE Transactions on Automatic Control, Vol. 35, No. 7, pp.814-824.

Mayne D.Q., Rawlings J.B., Rao C.V. and Scokaert P.O.M. (2000): "Constrained model predictive control: Stability and optimality" - survey paper, Automatica, Vol. 36, pp. 789-814.

McBrayer K. and Edgar T.F. (1995): "Bias Detection and Estimation in Dynamic Data Reconciliation", Journal of Process Control, Vol. 5, No. 4, pp. 285-289.

Meyer M., Koehret B. and Enjalbert M. (1993): "Data Reconciliation on Muticomponent Network Process", Computers and Chemical Engineering, Vol. 17, No. 8, pp. 807-817.

Mo K.J., Eo S., Shin D., Yoon E.S. (1998): "Qualitative Interpretation and Compression of Process Data Using Clustering Method", Computers and Chemical Engineering, Vol. 22, Suppl., pp. S555-S562.

Narasimhan S. (1984): "Detection of Steady States and its Application to Process Data Reconciliation", M.S. Thesis, Northwestern University.

Narasimhan S. and Harikumar P. (1993): "A Method to Incorporate Bounds in Data Reconciliation and Gross Error Detection – I. The Bounded Data Reconciliation Problem", Computers and Chemical Engineering, Vol. 17, No. 11, pp. 1115-1120.

Narasimhan S. and Mah R.S.H. (1987): "Generalized Likelihood Ratio Method for Gross Error Identification", AIChE Journal, Vol. 33, No. 9, pp. 1514-1521.

Narasimhan S. and Mah R.S.H. (1988): "Generalized Likelihood Ratios for Gross Error Identification in Dynamic Processes", AIChE Journal, Vol. 34. No. 8. pp. 1321-1331.

Narasimhan S. and Mah R.S.H. (1989): "Treatment of General Steady State Process Models in Gross Error Identification", Computers and Chemical Engineering, Vol. 13, No.7, pp. 851-853.

Narasimhan S., Mah R.S.H., Tamhane A.C., Woodward J.W. and Hale J.C. (1986): "A Composite Statistical Test for Detecting Changes of Steady States", AIChE Journal, September 1986, Vol. 32 No. 9 pp1409 – 1418.

Narasimhan S., Chen Shan Kao and Mah R.S.H. (1987): "Detecting Changes of Steady States Using the Mathematical Theory of Evidence", AIChE Journal, November 1987, Vol. 33, No. 11, pp 1930 – 1932.

Pai C.C.D. and Fisher G.D. (1988): "Application of Broyden's Method to Reconciliation of Nonlinearly Constrained Data", AIChE Journal, Vol. 34, No.5. May 1988.

Pike A.W., Grimble M.J., Johnson M.A., Ordys A.W. and Shakoor S. (1996): "Predictive Control", The Control Handbook, CRC Press Inc., Edited by Levine W.S.

Pontryagin L.S., Boltyanskii V.G., Gamkrelidze R.V. and Mishchenko E.F. (1962): "The mathematical theory of optimal processes", New York, Wiley-Interscience.

Qin S.J. and Badgwell T.A. (1996): "An Overview of Industrial Model Predictive Control Technology", CPC V, Tahoe City, (http://www.che.utexas.edu/~qin/cpcv/cpcv14.html).

Ramamurthi Y., Sistu P.B. and Bequette B.W. (1993): "Control-Relevant Dynamic Data Reconciliation and Parameter Estimation", Computers and Chemical Engineering, Vol. 17, No. 1, pp. 41-59.

Richalet J., Rault A., Testud J.L. and Papon J. (1978): "Model Predictive Control: Applications to Industrial Processes", Automatica, Vol. 14, pp. 413-428.

Roberts P.D. (1979): "An Algorithm for Steady-State System Optimisation and Parameter Estimation", International Journal of Systems Science, Vol.10, No.7. pp. 719-734.

Roberts P.D. (1992): "Optimal control of nonlinear systems with model-reality differences", Proceedings of the 31st Conference on Decision and Control, Tucson, Arizona, December 1992.

Roberts P.D. (1995): "Coping with model-reality differences in industrial process optimisation - A review of integrated system optimisation and parameter estimation (ISOPE)", Computers in Industry, 26, pp. 281-290.

Roberts P.D. (1997): "Dynamic Data Reconciliation Using Sequential Modular Models and its application to real time model tuning". EPSRC Research proposal Case for Support, Control Engineering Research Centre, City University, London.

Roberts P.D. (1999): "A Brief Overview of Model Predictive Control". Computing and Control Engineering Journal, Vol. 10, No. 5, pp. 186-188.

Roberts P.D. and Williams T.W.C. (1981): "On an algorithm for combined system optimisation and parameter estimation", Automatica, Vol. 17. No.1. pp. 199-209.

Rollins D.K., and Davis J.F. (1992): "Unbiased Estimation of Gross Errors in Process Measurements", AIChE Journal, April 1992, Vol. 38, No. 4, pp. 563-572.

Rollins D.K., and Davis J.F. (1993): "Gross Error Detection When Variance-Covariance Matrices are Unknown", AIChE Journal, August 1993, Vol. 39. No. 8, pp. 1335-1341.

Rollins D.K. and Devanathan S. (1993): "Unbiased Estimation in Dynamic Data Reconciliation", AIChE Journal, Vol. 39, No. 8, pp. 1330-1334.

Romagnoli J.A. and Sanchez M.C. (2000): "Data Processing and Reconciliation for Chemical Process Operations", Academic Press, 2000.

Romagnoli J.A. and Stephanopoulos G. (1981): "Rectification of Process Measurement Data in the Presence of Gross Errors", Chenical Engineering Science, Vol. 36, No. 11, pp. 1849-1863.

Rosenberg J., Mah R.S.H. and Iordache C. (1987): "Evaluation of Schemes for Detecting and Identifying Gross Errors in Process Data", Ind. Eng. Chem. Res., 26, pp. 555-564.

Rossiter J.A., Kouvaritakis B. and Gossner J.R. (1996): "Guaranteeing Feasibility in Constrained Stable Generalised Predictive Control", IEE Proceedings - Control Theory Applications, Vol. 143, No. 5, pp. 463-469.

Rouhani R. and Mehra R.K. (1982): Model Algorithmic Control (MAC); Basic Theoretical Properties", Automatica, Vol. 18, No. 4, pp. 401-414.

Sanchez M., Romagnoli J., Jiang Q. and Bagajewicz M. (1999): "Simultaneous Estimation of Biases and Leaks in Process Plants", Computers and Chemical Engineering, Vol. 23, pp. 841-857.

Sandoz D.J. (1998): "Perspectives on the Industrial Exploitation of Model Predictive Control", The 1998 UKACC Lecture, The Royal Society, London, UK. 24th March 1998.

SAST Ltd. (1993): "OTISS User's Manual", Waterway House. The Ham, Brentford, Middlesex, UK, (now Aspentech (U.K.) Ltd.).

Serth R.W. and Heenan W.A. (1986): "Gross Error Detection and Data Reconciliation in Steam-Metering Systems", AIChE Journal, Vol. 32. No. 5. pp. 733-742.

Shafer G. (1976): "A Mathematical Theory of Evidence", Princeton University Press, Princeton, New Jersey.

Sistu P.H. and Bequette B.W. (1991): "Nonlinear Predictive Control of Uncertain Processes: Application to a CSTR", AIChE Journal, Vol. 37, No. 11.

Takiyama H., Naka Y., O'Shima E. and Adriani A. (1991): "Sensor-Based Data Reconciliation Method and Application to the Pilot Plant". Journal of Chemical Engineering of Japan, Vol. 24, No. 3, pp. 339-346.

Tamhane A.C. and Mah R.S.H. (1985): "Data Reconciliation and Gross Error Detection in Chemical Process Networks", Technometrics, November 1985, Vol. 27, No. 4, pp. 409-422.

Tjoa I.B. and Biegler L.T. (1991): "Simultaneous Strategies for Data Reconciliation and Gross Error Detection of Nonlinear Systems", Computers and Chemical Engineering, Vol. 15, No. 10, pp. 679-690.

Tong H. and Crowe C.M. (1996): "Detecting Persistent Gross Errors by Sequential Analysis of Principal Components", Computers and Chemical Engineering, Vol. 20, Suppl., pp. S733-S738.

Vaclavek V. (1969): "Studies on System Engineering – III Optimal Choices of the Balance Measurements in Complicated Chemical Engineering Systems", Chemical Engineering Science, Vol. 24, pp. 947-955.

Weiss G.H., Romagnoli J.A. and Islam K.A. (1996): "Data Reconciliation – An Industrial Case Study", Computers and Chemical Engineering, Vol. 20, No. 12, pp. 1441-1449.

Yang Y., Ten R. and Jao L. (1995): "A Study of Gross Error Detection and Data Reconciliation in Process Industries", Computers and Chemical Engineering, Vol. 19, Suppl., pp. S217-S222.

Yao Y. (1965): "An Approximate degrees of freedom solution to the multivariate Behrens-Fisher problem", Biometrica, 52, 1 and 2, pp. 139-147.

Yin P.Y and Chen L.H. (1994): "A new non-iterative approach for clustering". Pattern Recognition Letters Vol.15, pp. 125-133.

Zhang H. and Roberts P.D. (1990): "On-line steady-state optimisation of nonlinear constrained processes with slow dynamics", Transactions of the Institute of Measurement and Control, Vol. 12, pp. 251-261.

APPENDIX

A. SEQUENTIAL QUADRATIC PROGRAMMING

Sequential Quadratic Programming (SQP) methods employ Newton's method (or quasi-Newton methods) to directly solve the Karush-Kuhn-Tucker conditions (Bazaraa et al, 1993) for the original problem. SQP methods, also known as Successive, or Recursive, Quadratic Programming, basically linearise inequality and equality constraints and construct a convex quadratic objective function from gradients of the objective and constraint functions. Solution of the resulting Quadratic Program (QP) determines the search direction while a one-dimentional minimisation along this direction locates the next point (Biegler, 1984). Here only the linearised sets of equality constraints are solved by QP. As SQP converges to the minimum, the solution of the linearised sets converges to the solution of the equality constraints.

The SQP approach begins by initialising the vector of optimisation variables to the user-supplied initial guess. Then, the initial approximation to the Hessian matrix is set to the identity matrix and the gradients of all functions are calculated (Liebman 1991).

A quadratic approximation to the objective function is used, along with first-order Taylor-series approximations to all constraints, to form a QP at each iteration. This results in the following QP approximation to the Nonlinear Program (NLP):

$$\min_{s} \quad \nabla\Phi^T s + \frac{1}{2} s^T B s \tag{A.1}$$

subject to

$$f_j + \nabla f_j{}^T s = 0 \tag{A.2}$$

$$g_j + \nabla g_j{}^T s \geq 0 \qquad \text{(A.3)}$$

where

s = vector of components of the search direction

B = symmetric positive-definite approximation of the Hessian of the corresponding Lagrangian function.

Once the QP approximation has been solved, the resulting estimates are tested for optimality (Kuhn-Tucker conditions). A line search is used to find the optimal step size in the calculated direction to obtain the new estimates. The approximate Hessian is updated and the next iteration begins.

LIST OF PUBLICATIONS

The following is a list of journal and conference publications as well as internal university technical reports to date.

1. Integrating Predictive Control and Economic Optimisation, Becerra V.M., Abu-el-zeet Z.H. and Roberts P.D., *Computing and Control Engineering Journal*, Vol. 10, No. 5, pp. 198-208, October 1999.

2. Use of a predictive Control Technique for the GEC-ALSTHOM MEC Gasifier Challenge, Becerra V.M., Abu-el-zeet Z.H. and Roberts P.D., *Seminar entitled 'MEC Benchmark Challenge On Gasifier Control'*, Wednesday 24th June 1998, Coventry University, Coventry, U.K.

3. Integrating Predictive Control and Economic Optimisation, Becerra V.M., Abu-el-zeet Z.H. and Roberts P.D., *IEE Colloquium on Model Predictive Control: Techniques and Applications*, 29th April 1999, IEE, London.

4. Data reconciliation and steady-state detection applied to a chemical process, Abu-el-zeet Z.H., Becerra V.M. and Roberts P.D., *UKACC International Conference (Control 2000)*, University of Cambridge, Cambridge, U.K., 4-7th September 2000.

5. Automatic Identification of Steady-state, Abu-el-zeet Z.H., Becerra V.M. and Roberts P.D., *Technical Report no. 170, CERC, City University*, London, UK, July 1999.

6. Steady-state Data Reconciliation and Estimation of Systematic Biases, Abu-el-zeet Z.H., Becerra V.M. and Roberts P.D., *Technical Report no. 171, CERC, City University*, London, UK, October 1999.