INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

On-line Identification of a Two-Stage Thermo-Mechanical Pulping Process

by

Deng Kailin, M. Sc.

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

Master of Engineering

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering

Carleton University

Ottawa, Ontario

Canada

© Copyright

April 2000

**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-52388-8

Canada

The undersigned hereby recommend to
The Faculty of Graduate Studies and Research
Acceptance of the thesis,

On-line Identification of a Two-Stage Thermo-Mechanical Pulping Process

Submitted by

Deng Kailin, M. Sc.
in partial fulfillment of the requirements for
the degree of Master of Engineering

Thesis Supervisor

Chair, Department of Mechanical and Aerospace Engineering

Carleton University
April 1, 2000

Abstract

The research work of the thesis involves the modeling, simulation, off-line identification and on-line identification of a two-stage Thermo-mechanical pulping (TMP) process. The main components of the TMP process are the refiners. Experiments were run and data was measured and collected from the refiners. The off-line least-squares algorithm and the on-line directional forgetting algorithm are used to fit the models of the TMP process to the collected data. The parameters of the TMP process are time varying, and therefore on-line identification is needed. For the purpose of on-line identification, the thesis presents the standard exponential forgetting algorithm and a novel directional forgetting algorithm. Comparisons between these two algorithms are also presented. The standard exponential forgetting algorithm works well only if the process is properly excited all the time. However, it is not the case of TMP process. In the TMP process, the input signals are not persistently exciting, windup phenomenon will occur and the estimates of the system parameters are unreliable. The results of comparisons between the standard exponential forgetting algorithm and the novel directional forgetting algorithm demonstrate that the directional forgetting algorithm should be chosen to implement the on-line identification of the TMP process.

Acknowledgements

I would like to thank my supervisor, Professor H. M. Schwartz for his research guidance throughout the thesis. His help is much appreciated.

I am also grateful to Dr. Liyu Cao, a post doctor fellow in our group. His support, idea and encouragement gave me the motivation to complete this thesis.

Dedication

To my parents Deng Jianhong and Zhang Jilian,
To my wife Jin Feiman,
For their love, faith and encouragement in all these years.

Contents

Title Page	i
Acceptance Sheet	ii
Abstract	iii
Acknowledgements	iv
Dedication	v
Contents	vi
List of Figures	vii
1 Introduction	1
1.1 System Identification	1
1.2 Previous Work	3
1.3 About This Thesis	5
2 Least-Squares Methods	7
2.1 Introduction	7
2.2 ARX Model	7
2.3 The Least-squares Algorithm	8
2.4 Asymptotic Properties of the Least-squares Algorithm	12
2.5 Example of Applying Least-squares Algorithm to Estimate the Parameter of A Linear System	15
2.6 Instrumental Variable Method	18
2.7 Weighted Least Squares	25

3	Off-line Identification of TMP Process	26
3.1	Introduction	26
3.2	Model Structure and Identification Methods	27
3.3	Experimental Data Sets	31
3.4	Identification Results and Discussions	37
4	Exponential Forgetting Algorithm	47
4.1	Introduction	47
4.2	The Recursive Least-squares Algorithm	47
4.3	Exponential Forgetting Algorithm	49
4.4	Examples of Applying Exponential Forgetting Algorithm	56
5	Directional Forgetting Algorithm	62
5.1	Introduction	62
5.2	Decomposition of Information Matrix	63
5.3	Simulation Example	68
6	On-line Identification of TMP Process	72
6.1	Introduction	72
6.2	Model Structures, Data and Algorithm Formulae	73
6.3	Identification of Parameters for the Primary Refiner	74
6.4	Identification of Parameters for the Secondary Refiner	89
7	Comparison Between DF and EF	99
7.1	Introduction	99
7.2	Comparison Between DF and EF in terms of Parameter Converging Behaviors	100
7.3	Comparison Between DF and EF in terms of Ability of Tracking Sudden Parameter Changes	111

8 Conclusion	122
References	123
Appendixes	126
A.2.1. Matlab program for example 2-1	126
A.2.2. Matlab program for example 2-2	127
A.3.1. Off-line identification of the primary refiner based on 1996 data set c	129
A.3.2. Off-line identification of the primary refiner based on 1996 data set a	130
A.3.3. Off-line identification of the primary refiner based on 1997 data set b	131
A.3.4. Off-line identification of the secondary refiner based on 1997 data set a	132
A.3.5. Off-line identification of the secondary refiner based on 1997 data set c	133
A.4.1. Matlab program for example 4-1	134
A.4.2. Simulated data set for example 4-1	135
A.4.3. Matlab program for example 4-2	136
A.4.4. Simulated data set for example 4-2	137
A.5.1. Matlab program for example 5-1	138
A.5.2. Simulated data set for example 5-1	139
A.6.1. On-line identification of the primary refiner based on 1996 data set c. Static model	140
A.6.2. On-line identification of the primary refiner based on 1996 data set c. Dynamic model	141
A.6.3. On-line identification of the primary refiner based on 1996 data set a. Static model	143
A.6.4. On-line identification of the primary refiner based on 1997 data set b. Static model	144
A.6.5. On-line identification of the secondary refiner based on 1997 data set a. Static model	145

A.6.6. On-line identification of the secondary refiner based on 1997 data set a.	
Dynamic model	146
A.6.7. On-line identification of the secondary refiner based on 1997 data set c.	
Static model	148
A.7.1. Comparison between DF and EF in terms of parameter converging behaviors based on simulated data set.	150
A.7.2. Simulated data set for A.7.1.	152
A.7.3. Comparison between DF and EF in terms of ability of tracking sudden parameter changes based on simulated data set	153
A.7.4. Simulated data set for A.7.3.	155

List of Figures

1.1 Systematic diagram of two-stage thermo-mechanical pulping system	6
2.1 Example of least-squares algorithm	17
2.2 Example of instrumental-variable method	24
3.1 1996 data set a of primary refiner	32
3.2 1996 data set c of primary refiner	33
3.3 1997 data set b of primary refiner	34
3.4 1997 data set a of secondary refiner	35
3.5 1997 data set c of secondary refiner	36
3.6 Least-squares identification for primary refiner based on 1996 data set c	40
3.7 Least-squares identification for primary refiner based on 1996 data set a	41
3.8 Least-squares identification for primary refiner based on 1997 data set b	42
3.9 Least-squares identification for secondary refiner based on 1997 data set a ...	45
3.10 Least-squares identification for secondary refiner based on 1997 data set c	46
4.1 Constant regression vector causes windup	58
4.2 Exponential forgetting algorithm works well with persistently exciting	61
5.1 Novel directional forgetting algorithm can prevent windup	71

6.1	On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set c, zero initial conditions	78
6.2	On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set c, nonzero initial conditions	79
6.3	On-line Identification of TMP process, the primary refiner, dynamic model, based on 1996 data set c, zero initial conditions	80
6.4	On-line Identification of TMP process, the primary refiner, dynamic model, based on 1996 data set c, nonzero initial conditions	81
6.5	On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set a, zero initial conditions	84
6.6	On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set a, nonzero initial conditions	85
6.7	On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set a, comparison of prediction error of motor load	86
6.8	On-line Identification of TMP process, the primary refiner, static model, based on 1997 data set b, zero initial conditions	87
6.9	On-line Identification of TMP process, the primary refiner, static model, based on 1997 data set b, nonzero initial conditions	88
6.10	On-line Identification of TMP process, the secondary refiner, static model, based on 1997 data set a, zero initial conditions	91
6.11	On-line Identification of TMP process, the secondary refiner, static model, based on 1997 data set a, nonzero initial conditions	92
6.12	On-line Identification of TMP process, the secondary refiner, dynamic model, based on 1997 data set a, zero initial conditions	93
6.13	On-line Identification of TMP process, the secondary refiner, dynamic model, based on 1997 data set a, nonzero initial conditions	94
6.14	On-line Identification of TMP process, the secondary refiner, static model, based on 1997 data set c, zero initial conditions	97

6.15	On-line Identification of TMP process, the secondary refiner, static model, based on 1997 data set c, nonzero initial conditions	98
7.1	Comparison between DF and EF, the simulated data set	103
7.2	Comparison between DF and EF, identification results	104
7.3	Comparison between DF and EF in terms of windup phenomenon	105
7.4	The trace of covariance matrix $P(t)$ in detail	106
7.5	Comparison between DF and EF, on-line identified value of parameter $\alpha(t)$	107
7.6	Comparison between DF and EF, on-line identified value of parameter $\beta(t)$	108
7.7	Comparison between DF and EF, on-line identified value of parameter $K(t)$	109
7.8	Comparison between DF and EF, estimated motor load	110
7.9	Real data sets of the TMP process	114
7.10	Whole-simulated data set	115
7.11	Identification results based on the whole-simulated data set	116
7.12	Comparison between DF and EF in terms of windup phenomenon	117
7.13	The trace of covariance matrix $P(t)$ in detail	118
7.14	On-line identified value of parameter $\alpha(t)$	119
7.15	On-line identified value of parameter $\beta(t)$	120
7.16	On-line identified value of parameter $K(t)$	121

Chapter 1

INTRODUCTION

1.1 System Identification

System identification deals with the problem of building mathematical models of dynamical systems based on observed data from the systems. The main concepts for the problem of system identification are as follows.

The system

A system is a broad concept, which plays an important role in modern science. Generally speaking, in the field of system identification, the term of system is referred to an object or a process in which variables of different kinds interact and produce observable signals. External signals that can be manipulated by the observer are called inputs. Others are called disturbances and can be divided into those that are directly measured and those that are only observed through their influence on the output. People build mathematical models of a system by determining the relationship between the input and the output signals. The term of dynamical system means, for a system, the current output values depend not only on the current external stimuli but also on their earlier values.

This thesis deals with a system of a two-stage thermo-mechanical wood pulping (TMP) process. Wood pulp can be manufactured by a number of various methods. They may be chemical or mechanical. The main components in the two-stage thermo-mechanical pulping process are the

refiners driven by electrical motors. The refiners are characterized as disc refiners that can either be single disc or double disc. The refiner plate can be either conical or flat shaped. Single disc refiners have a rotating disc and a stationary one. The double disc refiners have two counter rotating discs. Thermo-mechanical pulping is a process in which the wood chips are subjected to both mechanical and thermodynamic forces setting up between the discs. In the first stage of refining, wood chips are fed into the primary refiner where they are broken into tiny particles and fibers. The infeed dilution water is added into the refiner to control the pulp consistency. The pulp refined by the primary refiner is further refined in the secondary refiner where the dilution water is added again in order to obtain the pulp of required quality. One can find a detailed description of two-stage refining process in [1]. A schematic diagram of the two-stage pulping system is shown in Figure 1-1.

The model

The model describes how the variables of a system relate to each other. Models may come in various shapes and be phrased with varying degrees of mathematical formalism. In the engineering applications, the most common mathematical model structures are difference equations, such as ARX and ARMAX models, as well as the linear state-space models. Artificial Neural Networks and Fuzzy models are also used recently. In this thesis, nonlinear black box models for the electrical motors of the refiners in the TMP process are used. The models are parametric models whose parameters need to be determined by the system identification techniques. To deal with the problem of system identification for the TMP process, the production rate $R_c(t)$ and the infeed dilution water flow rate $R_w(t)$ are the input signals. The electrical motor load $L(t)$ is the output signal.

The identification methods

A large variety of identification methods have been proposed in the literature. The basic idea of system identification is to adjust the system parameters within a given model until its predicted output coincides as well as possible with the measured output. The purpose of various methods of system identification is to determine the parameters for the given models, that is, to fit the given model to the data measured. System identification has its roots in standard statistical techniques. Many of the basic identifying routines have direct interpretations known as statistical

methods such as Least Squares and Maximum Likelihood. The literature on system identification is extensive. The theoretical foundation of system identification is presented by Ljung [24] and Soderstrom and Stoica [25]. To follow the development in the field, the IFAC series of Symposia on System Identification (Budapest, 1991, Copenhagen, 1994) is a good source. This thesis uses the least-squares algorithm to implement the off-line identification of TMP process. It also presents the instrumental-variable method. For on-line identification, the thesis presents the standard exponential forgetting algorithm and a novel directional forgetting algorithm, which was proposed and developed by Cao and Schwartz [20]. Comparisons between the standard exponential forgetting algorithm and the novel directional forgetting algorithm are also presented.

The data

The experimental data sets for the identification of TMP process were measured and collected from the Avenor Paper Mill, Gatineau, Quebec. The data sets were collected from the operating refiners without affecting the pulp production and with no modifications to the facilities. There are six data sets. Among them, three were collected on June 21, 1996 and other three were collected on June 9, 1997. One calls them 1996 data set a, b, c and 1997 data sets a, b, c, respectively. Each data set has the data for the primary refiner as well as for the secondary refiner.

1.2 Previous Work

The thermo-mechanical pulping process is a process in which the wood chips are subjected to both mechanical and the thermodynamic forces setting up between the discs. R.A. Leask described the complex process in [1]. Miles and May described the flow of pulp in the refining zone by analyzing the forces that impact on the pulp and developed a set of equations in [4]. In Miles' and May's model, the variables such as steam velocity, pressure gradients, pulp velocity, etc. were involved. These variables are unavailable in the data sets collected from Avenor Paper Mill. To model the TMP process, Schwartz et al developed a linear model for the motor load of the refiners based on the available information in the data sets collected from Avenor Paper Mill [9][10]. In 1998, Schwartz, Cao and Phung [11] proposed a nonlinear model to cope with the

nonlinear property of the refiners. They also proposed a dynamic model structure to describe the transient behavior of TMP process.

The standard exponential forgetting (EF) algorithm is the basic method to identify the system parameters recursively. The EF algorithm works in such a way that it discounts old data and adds the new information about the system dynamics at each update step. The new information is carried by the regression vector $\varphi(t)$. If the process is properly excited all the time, the regression vector $\varphi(t)$ will carry sufficient information, the old data will be replaced by new information gradually. However, this is not the case of TMP process. In TMP process, some input variables are kept constant for a quite long period of time while the refiners are in operation, the regression vector $\varphi(t)$ does not carry sufficient information. In this case, the old data will be discounted continuously regardless of whether there is sufficient information. Thus, only a part of the old data can be replaced by new data, the others will be discounted step by step. As a result, some eigenvalues of information matrix $R(t)$ will tend to be zero, the covariance matrix $P(t)$ will continuously grow and will tend to become unbounded ($P(t)$ is the inverse of $R(t)$). This is the phenomenon called estimator windup. The estimator windup occurs whenever the system is not persistently exciting. The occurrence of windup means that the estimator is sensitive to the noise and the estimated parameters are unreliable. To avoid windup, several methods have been proposed. To prevent the windup phenomenon, which is the character the exponential forgetting algorithm, a more suitable forgetting philosophy should be to forget old data only when it can be replaced by new data. According to such a philosophy, Kulhavy and Karny[12], and Kulhavy[15] proposed a directional forgetting algorithm based on the Bayesian estimation approach. This algorithm can prevent estimator windup but some eigenvalues of the information matrix $R(t)$ may become unbounded (Bittanti et al [17]). This means the algorithm may lose its tracking capability in some directions. Parkum et al [18] proposed the selective forgetting algorithm in which the eigenvalues of the covariance matrix $P(t)$ are updated directly. The updating is performed in terms of eigenvalues' directions and each eigenvalue is bounded at any time. A drawback of the selective forgetting algorithm is that the eigenvalues and eigenvectors have to be computed at each update, which will greatly increase the computational requirement compared to the standard exponential forgetting algorithm. In 1998, Cao and Schwartz [20] proposed and developed a new directional forgetting algorithm, which does not need to compute

eigenvalues and eigenvectors at each update to determine the forgetting direction. They replaced the forgetting factor λ by the forgetting matrix $F(t)$ which is able to discount the old data contained in the information matrix $R(t-1)$ according to the direction of the regression vector $\varphi(t)$.

1.3 About This Thesis

The models and the results of identification for the TMP process are intended to be the guides for the pulp mill plant operators so that they can control the properties of the refined pulp more efficiently. Focusing on the recursive identification of the TMP process, this thesis introduces, analyzes and implements several different algorithms based on the real data as well as the simulated data. By demonstrating and analyzing the simulation results, the thesis points out that the novel directional forgetting algorithm is a suitable choice for the TMP process on-line identification.

The thesis has eight chapters. Chapter 1 is the introduction. In Chapter 2, the theoretical foundation of least-squares algorithm is presented; its properties are analyzed; and a simulation example is presented. As an improvement to the least-squares algorithm, the instrumental-variable method is also presented. A corresponding simulation example is presented too. Chapter 3 implements the off-line identification of the TMP process by the least-squares algorithm, which demonstrates that the parameters of the TMP process are time-varying and therefore the on-line identification is needed. Chapter 4 introduces the theory of the standard exponential forgetting algorithm, its properties are analyzed, and the problem of estimator windup is presented and analyzed. In Chapter 5, the novel directional forgetting algorithm is presented and analyzed. Corresponding simulation examples are also presented. The on-line identification of the TMP process by the novel directional forgetting algorithm is implemented in Chapter 6. The analyses based on the identification results are presented. In Chapter 7, some results of comparison between the standard exponential forgetting algorithm and the novel directional forgetting algorithm are presented. The analyses based on the simulation results point out that the novel directional forgetting algorithm is a suitable algorithm for on-line identification of TMP process. The final chapter, Chapter 8 is the conclusion.

Figure 1-1. Schematic Diagram of Two-stage Thermo-mechanical Pulping System

Chapter 2

LEAST-SQUARES METHODS

2.1 Introduction

As mentioned in Chapter 1, system identification has its roots in standard statistical techniques. Many of the basic identifying routines have direct interpretations known as statistical methods such as Least Squares and Maximum Likelihood. The literature on System Identification is extensive. To follow the development in the field, the IFAC (The International federation of Automatic Control) series of Symposia on System Identification (Budapest, 1991, Copenhagen, 1994) is a good source. To follow the "mainstream" of the field, this chapter will mainly deal with the Linear Least-Squares Algorithm. The ARX model, the least-squares criterion, and the asymptotic properties of the least-squares algorithm are discussed. Examples of applying least-squares algorithm to estimate the parameter vector of a linear system are presented. As a generalization and improvement of the least-squares method, the instrumental-variable method is discussed and an example is also presented.

2.2 ARX Model

Denote the system's input and output at time t by $u(t)$ and $y(t)$, respectively. The most popular relationship between the input and output is the linear difference equation

$$y(t) + a_1 y(t-1) + \dots + a_j y(t-j) = b_1 u(t-1) + \dots + b_k u(t-k). \quad (2-1)$$

The system has been presented in discrete time since observed data are always collected by sampling. It is thus more straightforward to relate observed data to discrete time models. For

convenience, assume the sampling interval to be one time unit. One can see model (2-1) as a way of determining the next output value based on previous observations:

$$y(t) = -a_1 y(t-1) - \dots - a_j y(t-j) + b_1 u(t-1) + \dots + b_k u(t-k). \quad (2-2)$$

The total terms' number in the right side is $n = j + k$. To simplify (2-2), introduce the parameter vector as

$$\theta = [a_1 \quad a_2 \quad \dots \quad a_j \quad b_1 \quad b_2 \quad \dots \quad b_k]^T \quad (2-3)$$

and the regression vector as

$$\varphi(t) = [-y(t-1) \quad \dots \quad -y(t-j) \quad u(t-1) \quad \dots \quad u(t-k)]^T. \quad (2-4)$$

Consequently, (2-2) can be written as

$$y(t) = \varphi^T(t) \theta. \quad (2-5)$$

Such model structures as (2-5) are linear in θ . They are known in statistics as linear regression models. "Regress" here implies the fact that one tries to calculate $y(t)$ by "going back" to $\varphi(t)$ which contains old data of the variable. Such a model structure has the standard name ARX-model (Auto-regression with exogenous inputs).

2.3 The Least Squares Algorithm

Suppose for a given system that one does not know the values of the parameters in θ , but the input and output data have been recorded over a time interval, $1 \leq t \leq N$:

$$Z^N = \{u(1) \quad y(1) \quad \dots \quad u(N) \quad y(N)\}. \quad (2-6)$$

The problem is to find an estimated $\hat{\theta}$ of the parameter vector θ in such a way that the outputs computed from the model (2-5) agree as closely as possible with the measured output $y(t)$ in the sense of least squares. That is, the parameter vector θ should be chosen to minimize the least-squares loss function

$$V(\theta, N) = \frac{1}{2} \sum_{t=1}^N [y(t) - \varphi^T(t) \theta]^2 \quad (2-7)$$

based on the input and output data Z^N , namely, the measurements $y(1), \varphi(1), \dots, y(N), \varphi(N)$. From these measurements, a system of linear equations can be obtained as

$$\begin{aligned} y(1) &= \varphi^T(1)\theta \\ y(2) &= \varphi^T(2)\theta \\ &\vdots \\ y(N) &= \varphi^T(N)\theta. \end{aligned} \tag{2 - 8}$$

In matrix notation above linear equations can be written as

$$Y = \Phi \theta \tag{2 - 9}$$

where

$$Y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \tag{2 - 10}$$

is an $N \times 1$ vector, and

$$\Phi = \begin{bmatrix} \varphi^T(1) \\ \vdots \\ \varphi^T(N) \end{bmatrix} \tag{2 - 11}$$

is an $N \times n$ matrix. A simple way to find θ from equation (2-9) would be to choose the number of measurements, N , to be equal to n so that Φ becomes a square matrix. If this matrix is nonsingular, equation (2-9) could easily be solved for θ . In practice, however, such an approach does not work well because the effects of measurement noise, disturbances and model misfit would lead to wrong results. With additional data it should be possible to get an improved estimate. When $N > n$, the linear system of equations (2-9) becomes overdetermined. An exact solution for such a system of equations will not exist.

Now define the equation errors as

$$\varepsilon(t) = y(t) - \varphi^T(t)\theta \tag{2 - 12}$$

which describes the deviation in the data from the deterministic system. For the whole system of equations, the errors can be written in matrix form as

$$\varepsilon = \begin{bmatrix} \varepsilon(1) \\ \vdots \\ \varepsilon(N) \end{bmatrix} = \begin{bmatrix} y(1) - \varphi^T(1)\theta \\ \vdots \\ y(N) - \varphi^T(N)\theta \end{bmatrix} = Y - \Phi\theta. \quad (2-13)$$

The least squares estimate of θ is defined as the vector $\hat{\theta}$ that minimizes the least-squares loss function

$$\begin{aligned} V(\theta, N) &= \frac{1}{2} \sum_{i=1}^N [y(i) - \varphi^T(i)\theta]^2 \\ &= \frac{1}{2} \sum_{i=1}^N \varepsilon^2(i) = \frac{1}{2} \|\varepsilon\|^2 = \frac{1}{2} \varepsilon^T \varepsilon \\ &= \frac{1}{2} [Y - \Phi\theta]^T [Y - \Phi\theta]. \end{aligned} \quad (2-14)$$

One can find the value of $\hat{\theta}$ by setting the derivative of $V(\theta, N)$ to zero:

$$0 = \frac{dV(\theta, N)}{d\theta} = -Y^T \Phi + \theta^T (\Phi^T \Phi) \quad (2-15)$$

which can be written as

$$(\Phi^T \Phi)\theta = \Phi^T Y \quad (2-16)$$

The identifiability of the parameter vector θ is determined by the singularity of $\Phi^T \Phi$. $\Phi^T \Phi$ is nonsingular if and only if Φ has full rank. If Φ does not have full rank, namely $\text{rank}(\Phi) < n$, $\Phi^T \Phi$ is singular. In such a case, equation (2-16) will have infinitely many solutions.

Singularity of matrix $\Phi^T \Phi$

Assume Φ is an $N \times n$ matrix, $N > n$. Then $\Phi^T \Phi$ is nonsingular if and only if Φ has full rank, that is $\text{rank}(\Phi) = n$.

Proof

“If” part of the proof can be expressed as: Φ has full rank means that the column vectors in Φ are linearly independent. For any $x \neq 0$, $\Phi x \neq 0$ exists. Consequently, $(\Phi x)^T(\Phi x) > 0$, $x^T(\Phi^T \Phi)x > 0$, which means matrix $\Phi^T \Phi$ is positive definite. Naturally, it is nonsingular. The “only if” part of the proof is based on rank of matrix products. $\Phi^T \Phi$ is nonsingular means $\text{rank}(\Phi^T \Phi) = n$. On the other hand, one knows that

$$\text{rank}(C) \leq \min\{\text{rank}(A), \text{rank}(B)\} \text{ if } C = AB. \quad (2 - 17)$$

Similarly,

$$\text{rank}(\Phi^T \Phi) \leq \min\{\text{rank}(\Phi), \text{rank}(\Phi^T)\} \quad (2 - 18)$$

Consequently, from the inequality

$$n = \text{rank}(\Phi^T \Phi) \leq \text{rank}(\Phi) \leq n, \quad (2 - 19)$$

one can get

$$\text{rank}(\Phi) = n. \quad (2 - 20)$$

When $\Phi^T \Phi$ is nonsingular, the least squares solution of the overdetermined linear equation (2-9) is given by [23]

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T Y \quad (2 - 21)$$

and its equivalent form is

$$\hat{\theta} = \left[\sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \sum_{t=1}^N \varphi(t) y(t), \quad (2 - 22)$$

or

$$\hat{\theta} = \left[\frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t) y(t). \quad (2 - 23)$$

2.4 Asymptotic Properties of the Least-Squares Algorithm

Consider applying the least-squares estimate (2-23) to the following dynamic model

$$y(t) = \varphi^T(t)\theta_0 + v(t). \quad (2-24)$$

where θ_0 is the "true value" of the parameter vector θ . Assume $v(t)$ is a stationary stochastic process(also called noise) that is independent of the input signal $u(t)$. One can see from equation (2-24) that

$$\begin{aligned} y(t-1) &= \varphi^T(t-1)\theta_0 + v(t-1) \\ y(t-2) &= \varphi^T(t-2)\theta_0 + v(t-2) \\ &\vdots \end{aligned} \quad (2-25)$$

From equation (2-4) one can get that

$$\begin{aligned} \varphi(t) &= [-y(t-1)\dots -y(t-j) \quad u(t-1)\dots u(t-k)]^T \\ &= [-(\varphi^T(t-1)\theta_0 + v(t-1))\dots -(\varphi^T(t-j)\theta_0 + v(t-j)) \quad u(t-1) \quad u(t-k)]^T \\ &\vdots \end{aligned} \quad (2-26)$$

Therefore, the term $\varphi(t)$ is a stochastic process that is affected by the noise $v(t-1)$, $v(t-2)$, ..., and $v(t-j)$. With such consideration, replace $y(t)$ in (2-23) by (2-24) and the least squares estimate (2-23) can be written as

$$\begin{aligned} \hat{\theta}_N^{LS} &= \left[\frac{1}{N} \sum_{t=1}^N \varphi(t)\varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t)y(t) \\ &= \left[\frac{1}{N} \sum_{t=1}^N \varphi(t)\varphi^T(t) \right]^{-1} \left[\frac{1}{N} \sum_{t=1}^N \varphi(t)\varphi^T(t)\theta_0 + \frac{1}{N} \sum_{t=1}^N \varphi(t)v(t) \right] \\ &= \theta_0 + \left[\frac{1}{N} \sum_{t=1}^N \varphi(t)\varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t)v(t). \end{aligned} \quad (2-27)$$

The desired properties of the least squares algorithm would be that the estimated values of parameter vector $\hat{\theta}_N^{LS}$ are close to the "true value" of θ_0 , and they converge to the "true value" of

θ_0 as N tends to infinity. That is, $E \hat{\theta}_N^{LS} = \theta_0$. Where E represents mathematical expectation.

However, one can see from equation (2-27) that the estimated $\hat{\theta}_N^{LS}$ will be affected by a bias term given by

$$\bar{\theta}_N = \hat{\theta}_N^{LS} - \theta_0 = \left[\frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t) v(t) \quad (2-28)$$

As analysis in section 2.3, $\Phi^T \Phi$ should be nonsingular so that the inverse exists and the least-squares algorithm can be implemented. With such condition, the $\hat{\theta}_N^{LS}$ is asymptotically unbiased or consistent (namely, $\hat{\theta}_N^{LS}$ tends to θ_0 as N tends to infinity) if and only if

$$E \varphi(t) v(t) = 0 \quad (2-29)$$

However, as expressed in equations (2-25), (2-26) and (2-27), $y(t-1) \dots, y(t-j)$ are affected by $v(t-1), \dots, v(t-j)$, the noise $v(t-1), \dots, v(t-j)$ are correlated with $\varphi(t)$. In most cases, the noise $v(t)$ is correlated with $v(t-1), \dots, v(t-j)$ and therefore is correlated with $\varphi(t)$, and equation (2-29) does not hold. Thus the least squares estimated $\hat{\theta}_N^{LS}$ will be biased from the "true value" of θ_0 . An important exception is when $v(t)$ is white noise with zero mean, that is, $v(t)$ is a sequence of uncorrelated random variables, it will be uncorrelated with all past data and in particular with $\varphi(t)$. In such a case, it is easy to see that

$$E \varphi(t) v(t) = E \varphi(t) E v(t) = 0 \quad (2-30)$$

Therefore,

$$\lim_{N \rightarrow \infty} \hat{\theta}_N^{LS} = \theta_0 \quad (2-31)$$

will hold, and the estimate is consequently unbiased.

Taking expectation with respect to the white noise, which has zero mean and variance λ , one can form the covariance matrix of the parameter estimate as [24]

$$P_N = E \bar{\theta}_N \bar{\theta}_N^T$$

$$\begin{aligned}
&= E \left\{ \left[\frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t) v(t) \right\} \left\{ \left[\frac{1}{N} \sum_{s=1}^N \varphi(s) \varphi^T(s) \right]^{-1} \frac{1}{N} \sum_{s=1}^N \varphi(s) v(s) \right\}^T \\
&= R_N^{-1} \left[\frac{1}{N} \sum_{t,s=1}^N \varphi(t) \varphi^T(s) \right] E[v(t)v(s)] R_N^{-1} \\
&= \lambda R_N^{-1}
\end{aligned} \tag{2 - 32}$$

Provided R_N is non-singular, where

$$R_N = \frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \tag{2 - 33}$$

is called the information matrix.

A very important and useful aspect of the expression (2-32) is that it can be estimated from observed data. Having processed N data points and the estimated parameter vector, one can use following equations to estimate the covariance matrix of the parameter [24]:

$$\hat{P}_N = \hat{\lambda}_N \left[\frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \tag{2 - 34}$$

$$\hat{\lambda}_N = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t) = \frac{1}{N} \sum_{t=1}^N \left[y(t) - \varphi^T(t) \hat{\theta} \right]^2 \tag{2 - 35}$$

The approximation improves as $N \rightarrow \infty$. A number of general properties are useful for system identification[24]:

1. The covariance decays like $1/N$, so the parameters approach the limiting value at the rate of $1/N^{1/2}$.
2. The covariance is proportional to the Noise-To-Signal ratio. That is, it is proportional to the noise variance and inversely proportional to the input power.
3. The covariance does not depend on the input's or noise's signal shapes, only on their variance/covariance properties.
4. Experiment design, i.e., the selection of the input u , aims at making the information matrix R "as large as possible".

2.5 An Example of Applying Least-Squares Algorithm to Estimate the Parameters of A Linear System

Suppose a process is described by

$$y(t) - 1.85y(t-1) + 0.855y(t-2) = u(t-1) + u(t-2) . \quad (2 - 36)$$

The Z-transfer function of the system is

$$\frac{Y(z)}{U(z)} = \frac{z + 1}{z^2 - 1.85z + 0.855} . \quad (2 - 37)$$

The linear regression form is

$$\hat{y}(t) = \varphi^T(t)\theta . \quad (2 - 38)$$

The regression vector and the parameter vector are

$$\varphi(t) = [-y(t-1) \quad -y(t-2) \quad u(t-1) \quad u(t-2)]^T , \text{ and} \quad (2 - 39)$$

$$\theta_0 = [-1.85 \quad 0.855 \quad 1 \quad 1]^T . \quad (2 - 40)$$

The components of the parameter vector

$$a_1 = -1.85, \quad a_2 = 0.855, \quad b_1 = 1, \quad b_2 = 1$$

need to be identified.

Using a square wave with period 10 as the input signal, the signal shifts between -1 and $+1$. If there is no effect of noise, the output can be obtained by

$$y(t) = 1.85y(t-1) - 0.855y(t-2) + u(t-1) + u(t-2) . \quad (2 - 41)$$

The least-squares method (2-23) applied to model (2-41) with the number of data points $N = 300$, yields the estimated parameter vector exactly equal to that of the original model, that is

$$a_1 = -1.8500 \text{ with variance } 0.0000,$$

$$a_2 = 0.8550 \text{ with variance } 0.0000,$$

$$b_1 = 1.0000 \text{ with variance } 0.0000, \text{ and}$$

$$b_2 = 1.0000 \text{ with variance } 0.0000.$$

Again, using the same square wave as the input signal and adding the effect of noise, the observed data actually have been generated by

$$y(t) = 1.85y(t-1) - 0.855y(t-2) + u(t-1) + u(t-2) + \varepsilon(t), \quad (2-42)$$

where $\varepsilon(t)$ is a white noise sequence with zero mean and variance one. The least-squares algorithm (2-23) was applied to model (2-42) with the number of data points $N = 300$, the components of the estimated parameter vector and the corresponding variances are:

$$\begin{aligned} a_1 &= -1.8268 \text{ with variance } 3.5344 \times 10^{-4}, \\ a_2 &= 0.8332 \text{ with variance } 3.5721 \times 10^{-4}, \\ b_1 &= 0.9415 \text{ with variance } 6.3680 \times 10^{-3}, \text{ and} \\ b_2 &= 1.0513 \text{ with variance } 6.1937 \times 10^{-3}. \end{aligned}$$

Therefore, the estimated model of the system is

$$\frac{Y(z)}{U(z)} = \frac{0.9415z + 1.0153}{z^2 - 1.8268z + 0.8332}. \quad (2-43)$$

If one takes the number of data points $N = 5000$, the components of the estimated parameter vector and the corresponding variances would be

$$\begin{aligned} a_1 &= -1.8546 \text{ with variance } 2.3040 \times 10^{-5}, \\ a_2 &= 0.8602 \text{ with variance } 2.3040 \times 10^{-5}, \\ b_1 &= 1.0061 \text{ with variance } 4.5369 \times 10^{-4}, \text{ and} \\ b_2 &= 0.9902 \text{ with variance } 4.2025 \times 10^{-4}. \end{aligned}$$

Then, the estimated model of the system becomes

$$\frac{Y(z)}{U(z)} = \frac{1.0061z + 0.9902}{z^2 - 1.8546z + 0.8602} \quad (2-44)$$

One can see, with $N \rightarrow \infty$, the estimate is improved. To check the accuracy of the estimation, the responses to unit-step input signal for models (2-37), (2-43) and (2-44) are shown in Figure 2-1.

Figure 2-1. Example of Least-Squares Algorithm

2.6 Instrumental Variable Methods

Disadvantage of Least-Squares Method

The Least-Squares Method is easy to apply and needs very limited computation. This is the reason for wide application of this method in industry. However, the drawback of the method is also obvious: the estimated parameter converges to the "true value" of parameter vector only under restrictive conditions. From section 2.2, 2.3 and 2.4, the least-squares estimate of parameter vector is given by

$$\hat{\theta}_N^{LS} = \left[\frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t) y(t). \quad (2-45)$$

Since the observed data actually have been generated by

$$y(t) = \varphi^T(t) \theta_0 + v(t), \quad (2-46)$$

the difference between the estimated parameter and "true value" of parameter is

$$\hat{\theta}_N^{LS} - \theta_0 = \left[\frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t) v(t). \quad (2-47)$$

When N tends to infinity this difference becomes

$$\lim_{N \rightarrow \infty} \left(\hat{\theta}_N^{LS} - \theta_0 \right) = \left[E \varphi(t) \varphi^T(t) \right]^{-1} \left[E \varphi(t) v(t) \right]. \quad (2-48)$$

Thus the least-squares estimated parameter (2-45) will have an asymptotic bias, or expressed in another way, will not converge to the "true value" of parameter θ_0 unless

$$E \varphi(t) v(t) = 0. \quad (2-49)$$

However, analysis in section (2.4) has shown, $\varphi(t)$ depends on the output of the system thus implicitly on the past values of $v(t)$. Equation (2-49) can be satisfied only if $v(t)$ is white noise. This condition is quite restrictive since one cannot be sure that the measurement noise $v(t)$ is white.

Recall the ARX-model described by equation (2-1). When one tries to calculate $y(t)$ by

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) - \dots - a_j y(t-j) + b_1 u(t-1) + \dots + b_k u(t-k), \quad (2-50)$$

one cannot measure the past outputs $y(t-1)$, $y(t-2)$, ..., $y(t-j)$. One can only measure the outputs that are corrupted by noise. Even if the noise is white, then one gets the following set of measurements:

$$\begin{aligned} y_m(t-1) &= y(t-1) + e(t-1), \\ y_m(t-2) &= y(t-2) + e(t-2), \\ &\vdots \\ y_m(t-j) &= y(t-j) + e(t-j), \end{aligned} \quad (2-51)$$

where $e(t)$ is white noise. The subscript m of y_m denotes the measured output y . In this case, the true model structure of the sampled system is

$$\begin{aligned} y(t) &= -a_1[y(t-1) + e(t-1)] - a_2[y(t-2) + e(t-2)] - \dots - a_j[y(t-j) + e(t-j)] \\ &\quad + b_1u(t-1) + b_2u(t-2) + \dots + b_ku(t-k). \end{aligned} \quad (2-52)$$

If one uses the model structure

$$y(t) = \varphi^T(t)\theta \quad (2-53)$$

to calculate the output $y(t)$, the true model for a system where only corrupted measurements are available becomes

$$\begin{aligned} y(t) &= -a_1y(t-1) - a_2y(t-2) - \dots - a_jy(t-j) \\ &\quad + b_1u(t-1) + b_2u(t-2) + \dots + b_ku(t-k) \\ &\quad - a_1e(t-1) - a_2e(t-2) - \dots - a_je(t-j) \end{aligned} \quad (2-54)$$

where

$$\begin{aligned} \varphi(t) &= [-y(t-1) \quad -y(t-2) \quad \dots \quad -y(t-j) \quad u(t-1) \quad u(t-2) \quad \dots \quad u(t-k)]^T \\ \theta &= [a_1 \quad a_2 \quad \dots \quad a_j \quad b_1 \quad b_2 \quad \dots \quad b_k]^T. \end{aligned} \quad (2-55)$$

Consequently, the equation error is

$$v(t) = -a_1e(t-1) - a_2e(t-2) - \dots - a_je(t-j). \quad (2-56)$$

Therefore, in the bias term given by equation (2-28)

$$\bar{\theta}_N = \hat{\theta}_N^{LS} - \theta_0 = \left[\frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t) v(t) \quad (2-57)$$

the mathematical expectation

$$\begin{aligned} E[\varphi(t)v(t)] &= E \begin{bmatrix} -y(t-1) - e(t-1) \\ \dots \\ -y(t-j) - e(t-j) \\ u(t-1) \\ \dots \\ u(t-k) \end{bmatrix} [-a_1 e(t-1) - a_2 e(t-2) - \dots - a_j e(t-j)] \\ &= \begin{bmatrix} a_1 e^2(t-1) \\ \dots \\ a_j e^2(t-j) \\ 0 \\ \dots \\ 0 \end{bmatrix} \end{aligned} \quad (2-58)$$

is obviously not zero, the least-squares estimate is consequently biased. The idea of how to overcome this shortcoming of least-squares method is the motivation for introducing the instrumental variable method.

Instrumental -Variable Method

Suppose one can select a certain finite-dimensional vector sequence $\{\xi(t)\}$ derived from the past data Z^{t-1} that satisfied

$$E\xi(t)v(t) = 0 \quad (2-59)$$

where $v(t)$ is not necessarily a sequence of white noise. Then the estimate of θ determined by the following equation [24]

$$\hat{\theta}_N^{IV} = \left[\frac{1}{N} \sum_{t=1}^N \xi(t) \varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \xi(t) y(t) \quad (2-60)$$

would be the best estimate of parameter vector θ_0 based on the observed data provided the indicated inverse exists. This method is called Instrumental-Variable Method (IV). The elements of $\xi(t)$ are called Instruments or Instrumental Variables.

Choice of Instruments

One needs to generate a vector sequence $\{\xi(t)\}$ derived from the past data Z^{t-1} in order to satisfy

$$E\xi(t)\varphi^T(t) \text{ is nonsingular and} \quad (2-61)$$

$$E\xi(t)v(t) = 0. \quad (2-62)$$

For this purpose, suppose a system is described by an ARX model

$$y(t) + a_1 y(t-1) + \dots + a_n y(t-n) = b_1 u(t-1) + \dots + b_m u(t-m) + v(t) \quad (2-63)$$

and the observed data are actually generated by

$$y(t) = \varphi^T(t)\theta_0 + v(t). \quad (2-64)$$

If the inverse

$$R^{-1}(t) = \left[\frac{1}{N} \sum_{t=1}^N \varphi(t)\varphi^T(t) \right]^{-1} \quad (2-65)$$

exists, a natural way is to generate the instruments similar to (2-63) so as to insure (2-60). At the same time not let the instruments be influenced by $\{v(t)\}$ so that to satisfy (2-62), that is

$$\xi(t) = K(q)[-x(t-1) \quad -x(t-2) \dots -x(t-n) \quad u(t-1) \dots u(t-m)]^T. \quad (2-66)$$

Where q denotes the forward or backward shift operator: $qu(t) = u(t+1)$, $q^{-1}u(t) = u(t-1)$. Here K is a linear filter and $x(t)$ is generated from the input through a linear system

$$x(t) = \frac{M(q)}{N(q)}u(t), \quad (2-67)$$

where

$$N(q) = 1 + n_1 q^{-1} + \dots + n_n q^{-n}, \text{ and} \quad (2-68)$$

$$M(q) = m_0 + m_1 q^{-1} + \dots + m_m q^{-m}. \quad (2 - 69)$$

If the input is generated in open loop so that $\xi(t)$ does not depend on the noise $v(t)$ in the system, then equation (2-62) holds. Also, since both the φ -vector and ξ -vector are generated from the same input sequence (φ contains additional effects from v), it is reasonable to expect that equation (2-61) is valid. A simple and appealing choice of instruments is to apply the least-squares method to (2-63) first. Then use the least-squares estimated model for $M(q)/N(q)$ and calculate $x(t)$ in (2-67). Finally, the instrumental variables $\xi(t)$ are determined by (2-66) with $K(q) = 1$.

Example 2-2. Instrumental -Variable Algorithm application

Suppose a process is described by

$$y(t) - 0.90y(t-1) + 0.20y(t-2) = u(t-1) + u(t-2). \quad (2 - 70)$$

Using a square wave with period 10 as the input signal, which shifts between -1 and $+1$. The output data are generated by

$$y(t) = 0.90y(t-1) - 0.20y(t-2) + u(t-1) + u(t-2) + v(t). \quad (2 - 71)$$

Instead of white noise, $v(t)$ is a sequence of uniformly distributed random numbers on the interval $(0.0, 1.0)$. With the number of data points $N = 100$, three steps of the instrumental -variable algorithm are implemented as follows.

Step 1. The Z-transfer function of the system is

$$\frac{Y(z)}{U(z)} = \frac{z+1}{z^2 - 0.90z + 0.20}. \quad (2 - 72)$$

Write the model structure as a linear regression form

$$y(t) = [-y(t-1) \quad -y(t-2) \quad u(t-1) \quad u(t-2)][a_1 \quad a_2 \quad b_1 \quad b_2]^T. \quad (2 - 73)$$

The components of the parameter vector,

$$a_1 = -0.90, \quad a_2 = 0.20, \quad b_1 = 1, \quad b_2 = 1$$

need to be identified. Estimate θ by least-squares method. The estimated parameter vector $\hat{\theta}_N^{(1)}$ and the corresponding estimated model, as well as the corresponding transfer function are

$$\hat{\theta}_N^{(1)} = [-0.9301 \quad 0.1366 \quad 1.2447 \quad 0.8084]^T \quad (2 - 74)$$

$$\frac{Y(z)}{U(z)} = \frac{1.2447z + 0.8084}{z^2 - 0.9301z + 0.1366} \quad (2 - 75)$$

$$\frac{M(q)}{N(q)} = \frac{1.2447q^{-1} + 0.8084q^{-2}}{1 - 0.9301q^{-1} + 0.1366q^{-2}}. \quad (2 - 76)$$

Step 2. Generate the instruments by

$$x^{(1)}(t) = \frac{M(q)}{N(q)} u(t) = \frac{1.2447q^{-1} + 0.8084q^{-2}}{1 - 0.9301q^{-1} + 0.1366q^{-2}} u(t), \quad (2 - 77)$$

$$\xi^{(1)}(t) = [-x^{(1)}(t-1) \quad -x^{(1)}(t-2) \quad u(t-1) \quad u(t-2)]^T. \quad (2 - 78)$$

Then determine the IV estimate (2-60) for $\hat{\theta}_N^{(2)}$ using the instruments $\xi^{(1)}$, the results and the corresponding estimated model as well as the transfer function are

$$\hat{\theta}_N^{(2)} = [-0.8833 \quad 0.1486 \quad 1.1083 \quad 0.9558]^T \quad (2 - 79)$$

$$\frac{Y(z)}{U(z)} = \frac{1.1083z + 0.9558}{z^2 - 0.8833z + 0.1486} \quad (2 - 80)$$

$$\frac{M(q)}{N(q)} = \frac{1.1083q^{-1} + 0.9558q^{-2}}{1 - 0.8833q^{-1} + 0.1486q^{-2}} \quad (2 - 81)$$

Step 3.

Repeat the process in step 2 to obtain a third estimated parameter vector $\hat{\theta}_N^{(3)}$ and the corresponding estimated model as follows

$$\hat{\theta}_N^{(3)} = [-0.8967 \quad 0.1651 \quad 1.0916 \quad 0.9362]^T \quad (2 - 82)$$

$$\frac{Y(z)}{U(z)} = \frac{1.0916z + 0.9362}{z^2 - 0.8967z + 0.1651} \quad (2 - 83)$$

The responses to unit-step input of the original model as well as all three estimated models are shown in Figure 2-2. From Figure 2-2 one can see that the unit-step response of the third estimated or second IV estimated model is almost completely the same as that of the original system. The obvious improvement from estimated models (2-75) to (2-80) to (2-83) can also be seen.

Figure 2-2. Example of Instrumental - Variable Method

2.7 Weighted Least Squares

It may happen that measurements at different time instants are considered to be of varying reliability. The reason may be that the degree of noise corruption changes or that certain measurements are less representative for the system's properties. In such a case less reliable measurements can be associated with less weight in the criterion:

$$V_N(\theta, Z^N) = \sum_{t=1}^N \beta(N, t) [y(t) - \varphi^T(t)\theta]^2. \quad (2 - 84)$$

From which, one can obtain

$$\hat{\theta}_N^{LS} = \left[\sum_{t=1}^N \beta(N, t) \varphi(t) \varphi^T(t) \right]^{-1} \sum_{t=1}^N \beta(N, t) \varphi(t) y(t), \quad (2 - 85)$$

where, with the $n \times n$ information matrix

$$R(N) = \sum_{t=1}^N \beta(N, t) \varphi(t) \varphi^T(t) \quad (2 - 86)$$

and the n -dimensional column vector

$$f(N) = \sum_{t=1}^N \beta(N, t) \varphi(t) y(t) \quad (2 - 87)$$

(2-80) can be written as

$$\hat{\theta}_N^{LS} = R^{-1}(N) f(N). \quad (2 - 88)$$

Equations (2-85), (2-86), (2-87) and (2-88) are used in deriving the Recursive Least-Squares Algorithm [24].

Chapter 3

OFF-LINE IDENTIFICATION OF TMP PROCESS

3.1 Introduction

In this chapter, the Least-Squares Algorithm is implemented to identify the system parameters of a two-stage thermo-mechanical pulping (TMP) process. In such a process the raw materials, wood chips, are refined by thermal and mechanical methods. The main facilities of the TMP process are refiners that are driven by electrical motors. In the first stage of refining, wood chips are fed into the primary refiner where they are broken down into tiny particles and fibers. Dilution water is added into the refiner to control the pulp consistency. The pulp refined by the primary refiner is further refined in the secondary refiner where the dilution water is added again in order to obtain the pulp of required quality. The TMP process is quite complicated, one can find a detailed description of two-stage refining process in [1]. In the identification process, linear regression models and input control variables are used to identify the system parameters. The motor load of refiners is predicted. Experimental data is collected from operational refiners without affecting the pulp production and with no modifications to the facilities.

The pulp mill operators, who are in charge of pulp production and need to maintain the required pulp quality, have to be able to estimate the results of adjusting or changing input control variables. In this way they can efficiently control the pulp properties specified by the Canadian Standard Freeness (CSF), long fiber content (LFC) and shive content (Shive). Since the pulp

quality measurement unit (PQM) results are only available approximately 30 to 45 minutes after the adjustment [5], from which CSF, LFC and Shive are collected, the modeling and predicting would be the powerful technique for them to control the pulp with desired quality.

Substantial work on the modeling of the TMP process has been presented by a number of researchers. Miles and May presented a physical-based model with a set of ordinary differential equations [4] [6]. In their model the variables such as steam velocity, pressure gradients, pulp velocity are involved. Under the conditions of this research, such variables cannot be obtained from the distributed control system (DCS) and the pulp quality measurement unit (PQM), from which the experimental data was collected. Therefore, the models based on the data collected from the DCS and PQM have to be developed differently. In [10], linear models for the motor load based on the data obtained from DCS were developed. The relationship between the pulp properties to the control variables was also established. In [11], nonlinear models to cope with the nonlinear properties of the refining process were developed. In these two papers, the system parameters were identified off-line and the motor load was predicted by the least-squares method. The research in those two papers has shown that the TMP process parameters are time-varying.

3.2 Model Structures and Identification Methods

The models and the results of the simulation are intended to be the guides for the pulp mill plant operators so that they can control the properties of the refined pulp more efficiently. For such purposes, the static and dynamic models were developed in [11].

The data that are used to measure the properties of refined pulp and can be collected from the PQM are the Canadian Standard Freeness (CSF), the long fiber content (LFC), and the shive content (Shive). Based on the experimental results, they were all found to be linearly related to the specific energy [7]. Furthermore, the specific energy is a function of the motor load and the production rate, and the motor load is a function of infeed dilution water flow rate, production rate and plate gap [7] [9] [10]. The following equation can show the relationship:

$$\text{pulp property} = f(\text{specific energy}) = f(L, R_c, R_w, gf)$$

Where $L(t)$, $R_c(t)$, $R_w(t)$ and $g(t)$ present the motor load, the production rate, the infeed dilution water flow rate and the plate gap, respectively. From such a relationship one can see, controlling the motor load, the production rate, the infeed dilution water flow rate and the plate gap means controlling the properties of refined pulp. In [11], the motor load $L(t)$ is modeled as a function of production rate $R_c(t)$ and infeed dilution water flow rate $R_w(t)$ with the restriction of constant plate gap. At constant production rate, the motor load increases proportionally to the inverse of refining zone gap. However, this relationship does not hold below a certain level of the refining gap [5]. When the gap is less than this level, the motor load will decrease as the gap decreases. This phenomenon makes it difficult to adjust the motor load through controlling the refining zone gap. Because of this, the refining zone gap is not used as control variable. In this case, in fact, when the production rate and the infeed dilution water flow rate are decided, then the motor load and the pulp properties are fixed. This functional relationship can be shown as the following equation:

$$\text{pulp property} = f(\text{specific energy}) = f(R_c, R_w)$$

To indicate the functional relationship between the motor load $L(t)$ and the production rate $R_c(t)$ (as well as the inlet consistency $C_o(t)$ for the secondary refiner), and the dilution water flow rate $R_w(t)$, the following models were proposed in [11].

For the primary refiner, the model would be:

$$L(t) = k(t) \frac{[R_c(t)]^{\alpha(t)}}{[R_w(t)]^{\beta(t)}} + \varepsilon(t), \quad (3-1)$$

and for the secondary refiner, the model would be:

$$L(t) = k(t) \frac{[R_c(t)C_o(t)]^{\alpha(t)}}{[R_w(t)]^{\beta(t)}} + \varepsilon(t). \quad (3-2)$$

Here, $\varepsilon(t)$ represents the equation error, t represents sampling instants, $t = 1, 2, \dots$, the sampling time was 10 seconds. The terms $k(t)$, $\alpha(t)$ and $\beta(t)$ represent unknown parameters. In off-line identification, they are considered time-invariant. The term $C_o(t)$ represents the outlet consistency of the primary refiner, or the inlet consistency of the secondary refiner, and is

defined as the ratio of the mass of wood chips in the pulp to the total mass of the pulp. It is given by the following equation

$$C_0(t) = \frac{R_c(t)}{R_c(t) + R_w(t) - R_s(t)} \quad (3 - 3)$$

where $R_s(t)$ is the steam production rate in the primary refiner.

The effect on the motor load of secondary refiner is different from that of primary refiner. Unlike the primary refiner, for the secondary refiner, the input material is the pulp refined by the primary refiner. The pulp contains not only the wood chip (the chip feed rate determines the production rate), but also the dilution water added in the primary refiner. Therefore, the model for the secondary refiner has to consider the outlet consistency of the primary refiner. In models (3-1) and (3-2), $L(t)$, $R_c(t)$, $R_w(t)$, and $C_0(t)$ can be collected from DCS. The terms $k(t)$, $\alpha(t)$ and $\beta(t)$ are the parameters to be identified.

Obviously, $L(t)$ is a nonlinear function of the control variables $R_c(t)$ and $R_w(t)$ (and $C_0(t)$ as well in the secondary refiner). To obtain a model with the linear regression structure, logarithmic transformations are implemented, then the following linear regression form is obtained (assuming $\varepsilon(t) = 0$):

$$y(t) = \alpha(t)u_1(t) - \beta(t)u_2(t) + K(t) \quad (3 - 4)$$

where

$$y(t) = \log L(t), \quad (3 - 5)$$

$$u_1(t) = \log R_c(t), \quad (3 - 6)$$

$$u_2(t) = \log R_w(t), \text{ and} \quad (3 - 7)$$

$$K(t) = \log k(t). \quad (3 - 8)$$

For the secondary refiner, the term

$$u_1(t) = \log[R_c(t)C_0(t)] \quad (3 - 9)$$

replaces (3-6).

Define the regression vector as

$$\varphi(t) = [u_1(t) \quad -u_2(t) \quad 1]^T, \quad (3 - 10)$$

and the parameter vector as

$$\theta(t) = [\alpha(t) \quad \beta(t) \quad K(t)]^T. \quad (3 - 11)$$

The system can then be written in linear regression form as

$$y(t) = \varphi^T(t)\theta(t). \quad (3 - 12)$$

Assume that the data collected from DCS was actually generated by

$$y(t) = \varphi^T(t)\theta(t) + e(t) \quad (3 - 13)$$

where $e(t)$ presents the measurement noise. Equation (3-12) is used as a predictor of (3 -13).

For the model described by equations (3-4), (3-10), (3-11) and (3-12), the off-line identification of the system parameters can be implemented by the least-squares algorithm

$$\hat{\theta}_N^{LS} = \left[\frac{1}{N} \sum_{t=1}^N \varphi(t)\varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t)y(t). \quad (3 - 14)$$

Assume that there are N measurements in the data, that is, $t=1, 2, \dots, N$. Refer to Section 2.3, equation (3-12) can be rewritten in the matrix form

$$Y = \Phi \theta \quad (3 - 15)$$

where, Y is defined as a column vector by

$$Y = [y(1) \quad y(2) \quad \dots \quad y(N)]^T \quad (3 - 16)$$

and Φ is defined as an $N \times 3$ matrix by

$$\Phi = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(N) \end{bmatrix} = \begin{bmatrix} u_1(1) & -u_2(1) & 1 \\ u_1(2) & -u_2(2) & 1 \\ \vdots & \vdots & \vdots \\ u_1(N) & -u_2(N) & 1 \end{bmatrix}. \quad (3 - 17)$$

The least-squares solution to the overdetermined linear equation (3-15) is given by[23]

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T Y. \quad (3 - 18)$$

As analyzed in Section 2.3, the identifiability of the parameter vector θ is determined by the singularity of $\Phi^T \Phi$. $\Phi^T \Phi$ is nonsingular if and only if Φ has full rank. From the structure of Φ , it is easy to understand that since all elements in the third column in the matrix Φ are equal to 1, to meet the requirement of full rank, the necessary and sufficient condition is that all inputs must change at least once. There are two input signals, the production rate $R_c(t)$ and the infeed dilution water flow rate $R_w(t)$, or their logarithms, say, $u_1(t)$ and $u_2(t)$ in matrix Φ . If an input signal does not change, the corresponding parameter in the model (3-1) or model (3-2) cannot be identified.

3.3 Experimental Data Sets

The experimental data sets were collected from the Avenor Paper Mill, Gatineau, Quebec. The data sets were measured and collected from the operating refiners without affecting the pulping process and with no modifications to the facilities. There are three data sets collected on June 21, 1996 and other three data sets collected on June 9, 1997. One calls them 1996 data sets a, b, c and 1997 data sets a, b, c, respectively. Each data set contains the sampling data of the production rate $R_c(t)$, the infeed dilution water flow rate $R_w(t)$, and the corresponding motor load $L(t)$ for both the primary refiner and the secondary refiner. The data sets also contain the sampling data of the inlet consistency $C_a(t)$ for the secondary refiner.

Three data sets, namely, 1996 data sets a and c, and 1997 data set b are used to identify the system parameters of the primary refiner. Two data sets, namely, 1997 data sets a and c are used to identify the parameters of the secondary refinery. These five data sets are shown in Figure 3-1, 3-2, 3-3, 3-4, and 3-5, respectively.

Figure 3-1. 1996 data set a of primary refiner.

Scale of X-axis: second $\times 10$.

Top: Production rate $R_c(t)$.

Middle: Infeed dilution water flow rate $R_w(t)$.

Bottom: Motor load $L(t)$.

Figure 3-2. 1996 data set c of primary refiner

Scale of X-axis: second $\times 10$.

Top: Production rate $R_c(t)$.

Middle: Infeed dilution water flow rate $R_w(t)$.

Bottom: Motor load $L(t)$.

Figure 3-3. 1997 data set b of primary refiner

Scale of X-axis: second $\times 10$.

Top: Production rate $R_c(t)$.

Middle: Infeed dilution water flow rate $R_w(t)$.

Bottom: Motor load $L(t)$.

Figure 3-4. 1997 data set a of secondary refiner

Scale of X-axis: second $\times 10$.

From top to bottom:

Production rate $R_c(t)$.

Infeed dilution water flow rate $R_w(t)$.

Inlet consistency $C_o(t)$.

Motor load $L(t)$.

Figure 3-5. 1997 data set c of secondary refiner

Scale of X-axis: second $\times$ 10.

From top to bottom:

Production rate $R_c(t)$.

Infeed dilution water flow rate $R_w(t)$.

Inlet consistency $C_d(t)$.

Motor load $L(t)$.

3.4 Identification Results and Discussions

The Primary Refiner

In 1996 data set c shown in figure 3-2, only the production rate $R_c(t)$ changed from 112 rpm to 120 rpm, the dilution water flow rate $R_w(t)$ was kept constant (there was noise that effected it). In this case, as analyzed in Section 3.2, only the parameter corresponding to the production rate $R_c(t)$, namely, the parameter α is identifiable. The parameter corresponding to the dilution water flow rate $R_w(t)$, namely, the parameter β and the gain k cannot be identified separately. Therefore, the model (3-1) is simplified by putting k and R_w together to get following form of the model

$$L(t) = kR_c^\alpha(t) \quad (3 - 19)$$

Logarithmic transformation of (3-19) gives the following linear regression structure

$$y(t) = \alpha u_1(t) + K \quad (3 - 20)$$

Where, $y(t) = \log L(t)$, $u_1(t) = \log R_c(t)$, $K = \log k$. The corresponding regression vector and the parameter vector become

$$\varphi(t) = [u_1(t) \quad 1]^T \quad (3 - 21)$$

$$\theta = [\alpha \quad K]^T \quad (3 - 22)$$

The number of data points is $N = 100$. The least-squares algorithm (3-14) was applied to model (3-20). The results of identification are shown in Figure 3-6.

In Figure 3-6, the top graph shows the production rate $R_c(t)$. The middle graph shows the measured motor load $L(t)$ (solid) and its least squares estimate(dotted). For the system parameters, the identified α is 0.2847 with variance 5.198×10^{-4} and the identified K (namely, $\log k$) is 1.6153 with variance 1.175×10^{-2} .

To reflect the transient behavior of the process, the following 1st-order dynamic model was proposed in [11]

$$L(t) + aL(t-1) = bR_c^\alpha(t-1) \quad (3-23)$$

The corresponding regression vector and the parameter vector are

$$\varphi(t) = [-L(t-1) \quad R_c^\alpha(t-1)]^T \quad (3-24)$$

$$\theta = [a \quad b]^T \quad (3-25)$$

Holding $\alpha = 0.2847$ constant as the results of identification on model (3-20), the results of least-squares identification applied on model (3-23) are also depicted in figure 3-6, in the bottom graph.

The bottom graph in Figure 3-6 shows the measured motor load $L(t)$ (solid) and its least squares estimate(dotted) based on the model (3-23), where the improvement on the transient behavior can be seen. For the system parameters, the identified a is -0.6785 with variance 4.96×10^{-3} , the identified b is 1.6176 with variance 1.255×10^{-1} .

In 1996 data set a shown in Figure 3-1, and 1997 data set b shown in Figure 3-3, the production rate $R_c(t)$ was kept constant, only the dilution water flow rate $R_w(t)$ changes. In this case, only the parameter corresponding to the dilution water flow rate $R_w(t)$, namely, the parameter β is identifiable. The parameter corresponding to the production rate $R_c(t)$, namely the parameter α and the gain k cannot be identified separately. Therefore, by putting k and R_c together, the model structure of model (3-1) becomes

$$L(t) = \frac{k}{R_w^\beta(t)} \quad (3-26)$$

Now, the linear regression structure of model (3-26) and the corresponding regression vector as well as the parameter vector are

$$y(t) = -\beta u_2(t) + K \quad (3-27)$$

$$\varphi(t) = [-u_2(t) \quad 1]^T \quad (3-28)$$

$$\theta = [\beta \quad K]^T, \quad (3-29)$$

where, $y(t) = \log L(t)$, $u_2(t) = \log R_w(t)$, $K = \log k$. In 1996 data set a. The number of data points is $N = 171$. In 1997 data set b, the number of data points is $N = 169$.

Figure 3-7 illustrates the results of least-squares identification based on 1996 data set a. The identified β is 0.4265 with variance 5.856×10^{-4} , and the identified K is 5.2820 with variance 1.708×10^{-2} . Figure 3-8 illustrates the results of least-squares identification based on 1997 data set b. The identified β is 1.1339 with variance 2.034×10^{-3} , and the identified K is 9.3466 with variance 6.401×10^{-2} . The difference in identified values of the system parameters based on different data sets indicates that the parameters are time-varying.

In Figure 3-7 and Figure 3-8, the top graphs show the infeed dilution water flow rate $R_w(t)$. The bottom graphs show the measured motor load $L(t)$ (solid) and its least-squares estimate(dotted).

From Figure 3-6, Figure 3-7 and Figure 3-8 one can see that the estimated motor loads approximate the mean value of the measured ones. Therefore, it is reasonable to use model (3-1) to predict the motor load.

Figure 3-6. Least-squares identification for primary refiner
based on 1996 data set c. Scale of X-axis: second $\times 10$.

Top: Production rate $R_c(t)$.

Middle: Measured motor load $L(t)$ (solid) and its least-squares
estimate(dotted) from model (3-20).

Bottom: Measured motor load $L(t)$ (solid) and its least-squares
estimate(dotted) from model (3-23).

Figure 3-7. Least-squares identification for primary refiner
based on 1996 data set a. Scale of X-axis: second $\times 10$.

Top: Infeed dilution water flow rate $R_w(t)$.

Bottom: Measured motor load $L(t)$ (solid) and its least-squares
estimate(dotted) from model (3-27).

**Figure 3-8. Least-squares identification for primary refiner
based on 1997 data set b. Scale of X-axis: second $\times 10$.**

Top: Infeed dilution water flow rate $R_w(t)$.

**Bottom: Measured motor load $L(t)$ (solid) and its least-squares
estimate(dotted) from model (3-27).**

The Secondary Refiner

Two data sets were used to identify the system parameters of secondary refiner. Both of the data sets were collected on June 9, 1997. Model (3-2) would be suitable for such a situation. In 1997 data set a shown in Figure 3-4, the infeed dilution water flow rate $R_w(t)$ was kept constant while production rate $R_c(t)$ and consistency $C_0(t)$ changed. In this case, the simplified form of model (3-2) becomes

$$L(t) = k[R_c(t)C_0(t)]^\alpha \quad (3 - 30)$$

The corresponding linear regression structure of model (3-30) and the regression vector as well as the parameter vector are

$$y(t) = \alpha u_1(t) + K \quad (3 - 31)$$

$$\varphi(t) = [u_1(t) \quad 1]^T \quad (3 - 32)$$

$$\theta = [\alpha \quad K]^T \quad (3 - 33)$$

Where, $y(t) = \log L(t)$, $u_1(t) = \log [R_c(t)C_0(t)]$, $K = \log k$.

The number of data points is $N = 278$. The least-squares algorithm (3-14) was applied to model (3-31). The results of identification based on 1997 data set a are shown in Figure 3-9. As the results of identification, the identified α is 1.8986 with variance 1.0089×10^{-2} , the identified K is -13.8161 with variance 7.6059×10^{-1} .

In order to improve the transient behavior, the following dynamic model was also proposed in [11]

$$L(t) + aL(t-1) = b[R_c(t-1)C_0(t-1)]^\alpha \quad (3 - 34)$$

The corresponding regression vector and the parameter vector are

$$\varphi(t) = [-L(t-1) \quad (R_c(t-1)C_0(t-1))^\alpha]^T \quad (3 - 35)$$

$$\theta = [a \quad b]^T \quad (3 - 36)$$

Holding $\alpha = 1.8986$ constant as the result of least-squares identification applied on model (3-31), the results of least-squares identification on model (3-34) are also shown in Figure 3-9 from which the improvement on the transient behavior can be seen. As the results of identification, the identified a is -0.4451 with variance 1.6972×10^{-3} , the identified b is 5.5591×10^{-7} with variance 1.7049×10^{-15} .

In 1997 data set c shown in figure 3-5, all the input variables including $R_c(t)$, $R_w(t)$, as well as $C_0(t)$ changed during the process, the original equation of model (3-2) would be suitable for such a situation. The linear regression structure of the model, and the corresponding regression vector as well as the parameter vector are

$$y(t) = \alpha u_1(t) - \beta u_2(t) + K \quad (3-37)$$

$$\varphi(t) = [u_1(t) \quad -u_2(t) \quad 1]^T \quad (3-38)$$

$$\theta = [\alpha \quad \beta \quad K]^T \quad (3-39)$$

Where, $y(t) = \log L(t)$, $u_1(t) = \log [R_c(t)C_0(t)]$, $u_2(t) = \log [R_w(t)]$, $K = \log k$. The number of data points in this data set is $N = 245$. The least-squares algorithm (3-14) was applied to model (3-37), and the results of identification based on 1997 data set c are shown in Figure 3-10.

As the results of identification, the identified α is 2.3701 with variance 2.1786×10^{-2} , the identified β is 0.4442 with variance 3.9185×10^{-3} , and the identified K is -16.1050 with variance 1.3061 .

From Figure 3-9 and Figure 3-10 one can see that the motor load $L(t)$ and consistency $C_0(t)$ are very similar in their shapes. This phenomenon indicates that the motor load of the secondary refiner is strongly affected by the inlet consistency. For the data set shown in Figure 3-4, the correlation coefficient between $R_c(t)$ and $L(t)$ is 0.693 , between $C_0(t)$ and $L(t)$ is 0.607 . For the data set shown in Figure 3-5, the correlation coefficient between $R_c(t)$ and $L(t)$ is 0.378 , between $C_0(t)$ and $L(t)$ is 0.694 , between $R_w(t)$ and $L(t)$ is 0.125 . For both data sets, there is stronger correlation between the inlet consistency $C_0(t)$ and motor load $L(t)$.

Figure 3-9. Least-squares identification for secondary refiner
based on 1997 data set a. Scale of X-axis: second $\times 10$.

- (a) Production rate $R_c(t)$.
- (b) Consistency $C_0(t)$.
- (c) Measured motor load $L(t)$ (solid) and its least-squares estimate(dotted) from model (3-31).
- (d) Measured motor load $L(t)$ (solid) and its least-squares estimate(dotted) from model (3-34).

Figure 3-10. Least-squares identification for secondary refiner
based on 1997 data set c. Scale of X-axis: second $\times 10$.

- (a) Production rate $R_c(t)$.
- (b) Infeed dilution water flow rate $R_w(t)$.
- (c) Consistency $C_0(t)$.
- (d) Measured motor load $L(t)$ (solid) and its least-squares estimate(dotted) from model (3-37).

Chapter 4

EXPONENTIAL FORGETTING ALGORITHM

4.1 Introduction

In many cases it is necessary to have a model of the system available on-line while the system is in operation. The model should be based on observed data up to the current time. The need for such an on-line model construction arises when a model is required in order to take some decision about the system. Methods coping with such problems using an on-line adjusted model are usually called adaptive. Thereafter, the terms adaptive control, adaptive filtering, adaptive signal processing, and adaptive prediction are widely used. The on-line computation of the model must be done in such a way that the processing of the measurement from one sample can, with certainty, be completed during one sampling interval. Otherwise the model building cannot keep up with the information flow. Identification techniques complying with this requirement are called recursive identification methods, since the measured input-output data are processed recursively(sequentially) as they become available. One can also call such techniques on-line or real-time identification, adaptive parameter estimation, or sequential parameter estimation.

4.2 The Recursive Least-Squares Algorithm

Recursive Computation Based on Least-Squares Estimation Equation

In the recursive estimation process the measurements are obtained sequentially in real time. Computation of the least-squares estimation can be arranged in such a way that the results obtained at time $t-1$ can be used to get the estimates at time t . Following such an idea, from the least-squares estimation equation

$$\begin{aligned}\hat{\theta}(t) &= \left[\sum_{k=1}^t \varphi(k) \varphi^T(k) \right]^{-1} \sum_{k=1}^t \varphi(k) y(k) \\ &= R^{-1}(t) f(t)\end{aligned}\tag{4 - 1}$$

the information matrix $R(t)$ can be written as

$$\begin{aligned}R(t) &= \sum_{k=1}^t \varphi(k) \varphi^T(k) \\ &= \sum_{k=1}^{t-1} \varphi(k) \varphi^T(k) + \varphi(t) \varphi^T(t) \\ &= R(t-1) + \varphi(t) \varphi^T(t).\end{aligned}\tag{4 - 2}$$

Similarly, the vector $f(t)$ can be written as

$$\begin{aligned}f(t) &= \sum_{k=1}^t \varphi(k) y(k) \\ &= \sum_{k=1}^{t-1} \varphi(k) y(k) + \varphi(t) y(t) \\ &= f(t-1) + \varphi(t) y(t).\end{aligned}\tag{4 - 3}$$

Naturally, from (4-1)

$$f(t) = R(t) \hat{\theta}(t), \quad f(t-1) = R(t-1) \hat{\theta}(t-1)\tag{4 - 4}$$

As the result, the least-squares estimation equation can be rewritten in the following recursive form

$$\begin{aligned}\hat{\theta}(t) &= R^{-1}(t) f(t) \\ &= R^{-1}(t) [f(t-1) + \varphi(t) y(t)] \\ &= R^{-1}(t) [R(t-1) \hat{\theta}(t-1) + \varphi(t) y(t)] \\ &= R^{-1}(t) [(R(t) - \varphi(t) \varphi^T(t)) \hat{\theta}(t-1) + \varphi(t) y(t)] \\ &= \hat{\theta}(t-1) + R^{-1}(t) \varphi(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)] \\ &= \hat{\theta}(t-1) + P(t) \varphi(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)]\end{aligned}\tag{4 - 5}$$

where

$$P(t) = R^{-1}(t) \quad (4 - 6)$$

is called the covariance matrix.

Equation (4-5) has clearly intuitive appeal. The present estimate $\hat{\theta}(t)$ is obtained by adding a correction to the previous estimate $\hat{\theta}(t-1)$. The correction term is proportional to the difference between the measured value of $y(t)$ and the prediction of $y(t)$ based on the previous parameter estimate. The term $\hat{y}(t) = \varphi^T(t)\hat{\theta}(t-1)$ is the prediction of $y(t)$. The components of the vector $P(t)\varphi(t)$ are weighting factors that tell how the correction should be implemented. The term $P(t)\varphi(t)$ is called estimator gain.

However, the above algorithm is unable to track time-varying parameters due to continuous accumulation of information in $R(t)$, which makes the estimator gain $P(t)\varphi(t)$ tend to zero. To solve the problem, a number of methods have been proposed in the literature. Among them, the most popular method is the so-called exponential forgetting algorithm. The basic idea of this method is to introduce a time-varying weighting parameter λ called the forgetting factor that discounts the old data during the estimation process. In the estimation process, the most recent data is given unit weight, the data that is n time units old is weighted by λ^n . The method is therefore called exponential forgetting.

4.3 Exponential Forgetting Algorithm

Weighted LS Criterion

Recall the equation (2-84). To minimize the weighted least-squares criterion

$$V_i(\theta, Z') = \sum_{k=1}^i \beta(t, k) [y(k) - \varphi^T(k)\theta]^2 \quad (4 - 7)$$

one needs to compute

$$\begin{aligned}
\hat{\theta}(t) &= \arg \min_{\theta} \sum_{k=1}^t \beta(t, k) [y(k) - \varphi^T(k) \theta]^2 \\
&= \left[\sum_{k=1}^t \beta(t, k) \varphi(k) \varphi^T(k) \right]^{-1} \sum_{k=1}^t \beta(t, k) \varphi(k) y(k) \\
&= R^{-1}(t) f(t)
\end{aligned} \tag{4 - 8}$$

where

$$R(t) = \sum_{k=1}^t \beta(t, k) \varphi(k) \varphi^T(k), \text{ and} \tag{4 - 9}$$

$$f(t) = \sum_{k=1}^t \beta(t, k) \varphi(k) y(k). \tag{4 - 10}$$

Recursive Algorithm

Suppose the weighting sequence has following property:

$$\begin{aligned}
\beta(t, k) &= \lambda(t) \beta(t-1, k) \quad 1 \leq k \leq t-1 \\
\beta(t, t) &= 1.
\end{aligned} \tag{4 - 11}$$

It means that

$$\begin{aligned}
\beta(t, k) &= \lambda(t) \beta(t-1, k) \\
&= \lambda(t) \lambda(t-1) \beta(t-2, k) \\
&= \lambda(t) \lambda(t-1) \dots \lambda(t-(k+1)) \beta(k, k) \\
&= \lambda(t) \lambda(t-1) \dots \lambda(t-(k+1)) \\
&= \prod_{j=k+1}^t \lambda(j)
\end{aligned} \tag{4 - 12}$$

so that one can get

$$\begin{aligned}
R(t) &= \sum_{k=1}^t \beta(t, k) \varphi(k) \varphi^T(k) \\
&= \sum_{k=1}^{t-1} \beta(t, k) \varphi(k) \varphi^T(k) + \beta(t, t) \varphi(t) \varphi^T(t)
\end{aligned}$$

$$\begin{aligned}
&= \sum_{k=1}^{t-1} \lambda(t) \beta(t-1, k) \varphi(k) \varphi^T(k) + \varphi(t) \varphi^T(t) \\
&= \lambda(t) R(t-1) + \varphi(t) \varphi^T(t).
\end{aligned} \tag{4 - 13}$$

Similarly,

$$f(t) = \lambda(t) f(t-1) + \varphi(t) y(t). \tag{4 - 14}$$

Consequently,

$$\begin{aligned}
\hat{\theta}(t) &= R^{-1}(t) f(t) \\
&= R^{-1}(t) [\lambda(t) f(t-1) + \varphi(t) y(t)] \\
&= R^{-1}(t) [\lambda(t) R(t-1) \hat{\theta}(t-1) + \varphi(t) y(t)] \\
&= R^{-1}(t) \{ [\lambda(t) R(t-1) + \varphi(t) \varphi^T(t) - \varphi(t) \varphi^T(t)] \hat{\theta}(t-1) + \varphi(t) y(t) \} \\
&= R^{-1}(t) \{ [R(t) - \varphi(t) \varphi^T(t)] \hat{\theta}(t-1) + \varphi(t) y(t) \} \\
&= \hat{\theta}(t-1) + R^{-1}(t) [\varphi(t) y(t) - \varphi(t) \varphi^T(t) \hat{\theta}(t-1)] \\
&= \hat{\theta}(t-1) + R^{-1}(t) \varphi(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)].
\end{aligned} \tag{4 - 15}$$

The recursive algorithm is implemented as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + R^{-1}(t) \varphi(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)], \tag{4 - 16}$$

$$R(t) = \lambda(t) R(t-1) + \varphi(t) \varphi^T(t). \tag{4 - 17}$$

To avoid inverting $R(t)$ at each step, it is convenient to introduce

$$P(t) = R^{-1}(t) \tag{4 - 18}$$

provided $R(t)$ is invertible. Then one can apply the matrix inversion lemma

$$[A + BCD]^{-1} = A^{-1} - A^{-1} B [DA^{-1} B + C^{-1}]^{-1} DA^{-1}. \tag{4 - 19}$$

with

$$A = \lambda(t)R(t-1), \quad B = D^T = \varphi(t), \quad \text{and} \quad C = 1. \quad (4-20)$$

One gets

$$\begin{aligned} P(t) &= R^{-1}(t) = [\lambda(t)R(t-1) + \varphi(t)\varphi^T(t)]^{-1} = [A + BCD]^{-1} \\ &= A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1} \\ &= \frac{P(t-1)}{\lambda(t)} - \frac{P(t-1)\varphi(t)}{\lambda(t)} \left[\frac{\varphi^T(t)P(t-1)\varphi(t)}{\lambda(t)} + 1 \right]^{-1} \frac{\varphi^T(t)P(t-1)}{\lambda(t)} \\ &= \frac{P(t-1)}{\lambda(t)} - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{[\lambda(t)]^2} \frac{\lambda(t)}{\varphi^T(t)P(t-1)\varphi(t) + \lambda(t)} \\ &= \frac{P(t-1)}{\lambda(t)} - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t)[\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)]} \\ &= \frac{P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}}{\lambda(t)}. \end{aligned} \quad (4-21)$$

To simplify formula (4-21), note that

$$\begin{aligned} R^{-1}(t)\varphi(t) &= P(t)\varphi(t) \\ &= \frac{P(t-1)\varphi(t)}{\lambda(t)} - \frac{1}{\lambda(t)} \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \\ &= \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}. \end{aligned} \quad (4-22)$$

The Recursive Least-Squares Algorithms can be summarized as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \quad (4-23)$$

$$K(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \quad (4-24)$$

$$\begin{aligned} P(t) &= \left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \right] / \lambda(t) \\ &= [P(t-1) - K(t)\varphi^T(t)P(t-1)] / \lambda(t) \end{aligned}$$

$$= \frac{1}{\lambda(t)} [I - K(t)\varphi^T(t)]P(t-1), \quad (4-25)$$

where I represents the identity matrix.

Initial Conditions

To implement the recursive algorithms (4-23), (4-24) and (4-25), initial values are required. According to the definition of $R(t)$, $R(0)=0$. Since $R(0)$ is not invertible, this value cannot be used. The initialization can be only at time instant t_0 when $R(t_0)$ has become invertible then use

$$P^{-1}(t_0) = R(t_0) = \sum_{k=1}^{t_0} \beta(t_0, k) \varphi(k) \varphi^T(k) \quad (4-26)$$

$$\hat{\theta}(t_0) = P(t_0) \sum_{k=1}^{t_0} \beta(t_0, k) \varphi(k) y(k) \quad (4-27)$$

to start the algorithm. In practice, it is more common to start the recursive algorithm at $t=0$ with an invertible $P(0) = P_0$, and an arbitrary $\theta(0) = \theta_0$. Then the estimated results will be

$$\hat{\theta}(t) = \left[P_0^{-1} + \sum_{k=1}^t \beta(t, k) \varphi(k) \varphi^T(k) \right]^{-1} \left[P_0^{-1} \theta_0 + \sum_{k=1}^t \beta(t, k) \varphi(k) y(k) \right]. \quad (4-28)$$

The relative importance of the initial values decays with time, as the magnitudes of the sums increase. As $P_0^{-1} \rightarrow 0$, the recursive estimates approach the off-line estimate. A common choice of initial values is to take $P(0) = CI$ and $\theta(0) = 0$, where C is a large constant. The term $\theta(0)$ is what one guesses the parameter vector to be before seeing the data. $P(0)$ reflects one's confidence in this guess.

Exponential Forgetting and Forgetting Factor $\lambda(t)$

An important reason for using recursive identification in practice is that the properties of the system may be time varying and that one wants the identification algorithm to track the parameter variations. A natural way to handle the problem is to assign less weight to older measurements that are no longer representative for the system. This means, in terms of (4-12),

one chooses $\lambda(j) < 1$. Usually, for a system that changes gradually and in a “stationary manner”, the most common choice is to take a constant $\lambda(j) = \lambda$, then

$$\beta(t, k) = \lambda^{t-k}. \quad (4 - 29)$$

The old measurements in the criterion are exponentially discounted. In this case, the algorithm is called Exponential Forgetting, and λ is called the Forgetting Factor. The forgetting factor λ is always chosen slightly less than 1 so that

$$\beta(t, k) = \lambda^{t-k} = e^{(t-k)\log \lambda} \cong e^{-(t-k)(1-\lambda)} \quad (4 - 30)$$

Where $\log \lambda \cong -(1-\lambda)$ by using Taylor's series expansion. The term $T_0 = 1/(1-\lambda)$ is called the memory time constant. When a measurement is older than T_0 samples, its weight in the criterion will be less than $e^{-1} \cong 36\%$. If the system remains approximately constant over T_0 samples, to avoid losing useful information too early in the information matrix $R(t)$, a suitable choice of λ can be determined from this relationship.

Estimator Windup

Exponential forgetting works well only if the process is properly excited all the time. There are problems with exponential forgetting when the excitation is poor. To understand this, first consider the extreme case in which there is no excitation at all. That is, $\varphi(t) = 0$. In this case, the equations of exponential forgetting algorithm become

$$\theta(t) = \theta(t-1) \quad (4 - 31)$$

$$P(t) = \frac{1}{\lambda} P(t-1) \quad (4 - 32)$$

The equation for the estimate is thus unstable since all eigenvalues equal to 1, and the equation for the P -matrix is unstable with all eigenvalues equal to $1/\lambda$ ($\lambda < 1$). In this case, the estimate will remain constant and the P -matrix will grow exponentially. Since the estimator gain is $P(t)\varphi(t)$, it will also grow exponentially. This means the estimates may change very drastically whenever $\varphi(t)$ becomes different from zero. This phenomenon is called Estimator Windup.

Persistent Excitation

Recall equations (2-16) and (2-17). Suppose a system is described by

$$y(t) = b_1 u(t-1) + b_2 u(t-2) + \dots + b_n u(t-n) \quad (4-33)$$

or

$$y(t) = \varphi^T(t) \theta \quad (4-34)$$

$$Y = \Phi \theta \quad (4-35)$$

where

$$\varphi(t) = [u(t-1) \quad u(t-2) \quad \dots \quad u(t-n)]^T \quad (4-36)$$

$$\theta = [b_1 \quad b_2 \quad \dots \quad b_n]^T \quad (4-37)$$

$$Y = [y(1) \quad y(2) \quad \dots \quad y(t)]^T \quad (4-38)$$

$$\Phi = [\varphi^T(1) \quad \varphi^T(2) \quad \dots \quad \varphi^T(t)]^T. \quad (4-39)$$

The least-squares algorithm cannot be processed unless some conditions are imposed on the input signal. The information matrix

$$\begin{aligned} R(t) &= \sum_{k=1}^t \varphi(k) \varphi^T(k) = \Phi^T \Phi \\ &= \begin{bmatrix} \sum_{k=1}^t u^2(k-1) & \sum_{k=1}^t u(k-1)u(k-2) & \dots & \sum_{k=1}^t u(k-1)u(k-n) \\ \sum_{k=1}^t u(k-1)u(k-2) & \sum_{k=1}^t u^2(k-2) & \dots & \sum_{k=1}^t u(k-2)u(k-n) \\ \vdots & & & \\ \sum_{k=1}^t u(k-1)u(k-n) & \dots & \sum_{k=1}^t u^2(k-n) \end{bmatrix} \end{aligned} \quad (4-40)$$

must be nonsingular, namely, it must have full rank. For long data sets, define C_n as

$$C_n = \lim_{t \rightarrow \infty} \frac{1}{t} \Phi^T \Phi = \begin{bmatrix} c(0) & c(1) & \dots & c(n-1) \\ c(1) & c(0) & \dots & c(n-2) \\ \vdots & & & \\ c(n-1) & c(n-2) & \dots & c(0) \end{bmatrix} \quad (4-41)$$

where

$$c(i) = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=1}^t u(k)u(k-i). \quad (4 - 42)$$

For long data sets the condition can be expressed as that the matrix in equation (4-41) must be positive definite. This leads to the following definition.

Definition for persistent excitation

A signal is called persistently exciting (PE) of order n if the limits (4-42) exist and if the matrix C_n given by Equation (4-41) is positive definite [25] [26].

An alternative definition of persistently exciting (PE)

A signal u is said to be persistently exciting (PE) of order n if for all t there exists an integer m such that [25] [26]

$$\rho_1 I > \sum_{k=t}^{t+m} \varphi(k) \varphi^T(k) > \rho_2 I \quad (4 - 43)$$

where $\rho_1 > 0$, $\rho_2 > 0$. This means that for all t the information matrix $R(t)$ is bounded and therefore the covariance matrix $P(t)$ is also bounded.

4.4 Examples of Applying Exponential Forgetting Algorithm

Example 4-1. Constant regression vector causes windup.

To illustrate the phenomenon of estimator windup, recall the model (3-19) of the primary refiner in the TMP process:

$$L(t) = k(t)R_c^\alpha(t). \quad (4 - 44)$$

Its linear regression model, the corresponding regression vector and the parameter vector are

$$y(t) = \varphi^T(t)\theta(t) \quad (4 - 45)$$

$$\varphi(t) = [u_1(t) \quad 1]^T \quad (4 - 46)$$

$$\theta(t) = [\alpha(t) \quad K(t)]^T \quad (4 - 47)$$

where $u_1(t) = \log R_c(t)$, $K(t) = \log k(t)$.

Data simulation

Take 1996 data set a and 1996 data set c for the primary refiner. Put the data set a and c together. The last data point of data set a is followed by the first data point of data set c. The total number of data points is $N = 400$. The off-line least-squares estimation algorithm is then applied to the model (4-44) based on the combined data set. The identified system parameters are $\alpha = 0.1098$ and $K = 2.4523$. The real data of production rate $R_c(t)$ of these data sets is used to generate the motor load by the model (4-44) and white noise is added as follows:

$$L(t) = k(t)R_c^\alpha(t) = e^K [R_c(t)]^\alpha = 2.7183^{2.4523} \times R_c^{0.1098} + \varepsilon(t) \quad (4 - 48)$$

where $\varepsilon(t)$ is white noise with mean zero and variance 1. In this way, one simulates a new data set in which the system parameters are determined in advanced so that one can check the properties of the exponential forgetting algorithm by simulation. The parameters $\alpha = 0.1098$ and $K = 2.4523$ need to be identified in the simulation. During the simulation process, as an input signal, the production rate R_c is kept constant for a long period of time, it changes only once in the whole process, from 112 rpm to 120 rpm, which is depicted in graph (a) of Figure 4-1.

Simulation procedure

Set the forgetting factor $\lambda = 0.98$. According to equation (4-46), take $u_1(t) = \log R_c(t)$ and the constant 1 as the elements of the regression vector $\varphi(t)$. Set the initial values of the parameter vector $\theta(t)$ as $\alpha(0) = 0$ and $K(0) = 0$. Set the initial value of covariance matrix as $P(0) = 1000I$. Implement recursive estimation by the exponential forgetting algorithm. The results of simulation are also depicted in Figure 4-1.

Figure 4-1. Constant regression vector causes windup

Scale of X-axis: second $\times 10$.

- (a) Real data of production rate $R_c(t)$.
- (b) Estimator gain G_1 .
- (c) Estimator gain G_2 .
- (d) The trace of the covariance matrix $P(t)$.
- (e) The "true value" of parameter α (dashed) and its estimate by exponential forgetting algorithm (solid).
- (f) The "true value" of parameter K (dashed) and its estimate by exponential forgetting algorithm (solid).

In example 4-1, the parameter vector is a 2×1 vector and hence the system parameters are in a two-dimensional space R^2 . To have persistent excitation, the regression vector should also span R^2 persistently. However, during the period in which $R_c(t)$ is kept constant, $u_1(t) = \log R_c(t)$ is kept constant too. In this case, the regression vector $\varphi(t)$ lies in a one-dimensional subspace of R^2 and is thus not persistently exciting. The simulation results shown in Figure 4-1 illustrate the behavior of the system. The graph (d) in Figure 4-1 shows that the trace of covariance matrix $P(t)$ grows exponentially during the period when the input signal $R_c(t)$ is kept constant. The term $G_1(t)$ is one component of estimator gain $P(t)\varphi(t)$ that is a 2×1 vector in this simulation. The abrupt change of $G_1(t)$ associated with the exponential growing of the trace of the P -matrix can be seen in graph (b). Another component of estimator gain behaves similarly, which is shown in graph (c). The estimate errors of parameters are quite large near the end of the period when the regression vector $\varphi(t)$ has been constant. When the input signal $R_c(t)$ changes so that the regression vector $\varphi(t)$ changes too, the parameter estimates change significantly.

Example 4-2. Exponential forgetting algorithm works well with persistent excitation.

Recall the TMP process and the model (3-1) for the primary refiner

$$L(t) = k(t) \frac{R_c^a(t)}{R_w^b(t)}. \quad (4 - 49)$$

Its linear regression model, the corresponding regression vector and the parameter vector are

$$y(t) = \varphi^T(t)\theta(t) \quad (4 - 50)$$

$$\varphi(t) = [u_1(t) \quad -u_2(t) \quad 1]^T \quad (4 - 51)$$

$$\theta(t) = [\alpha(t) \quad \beta(t) \quad K(t)]^T, \quad (4 - 52)$$

where $u_1(t) = \log R_c(t)$, $u_2(t) = \log R_w(t)$, $K(t) = \log k(t)$.

Data simulation

Take 1996 data set a, 1996 data set b and 1996 data set c for the primary refiner. Put the data set a, b and c together. The last data point of data set a is followed by the first data point of data set b, the last data point of data set b is followed by the first data point of data set c. The total

number of data points is $N = 700$. The off-line least-squares estimation algorithm is applied to model (4-49) based on the combined data set a, b and c. The identified values of the system parameters are $\alpha = 0.1779$, $\beta = 0.5421$, $K = 2.4523$. Then random signals are replace the production rate $R_c(t)$ and the dilution water flow rate $R_w(t)$ in the model (4-49), the system parameters are set as $\alpha = 0.1779$, $\beta = 0.5421$ and $K = 5.0614$, simulate the motor load and add white noise as follows:

$$L(t) = k(t) \frac{R_c^\alpha(t)}{R_w^\beta(t)} = e^{\kappa} \frac{[R_c(t)]^\alpha}{[R_w(t)]^\beta} = 2.7183^{5.0614} \times \frac{R_c^{0.1779}}{R_w^{0.5421}} + \varepsilon(t) \quad (4 - 53)$$

Now, the new data set consists of random signal $R_c(t)$, random signal $R_w(t)$ and the simulated motor load $L(t)$. The random input signals $R_c(t)$ and $R_w(t)$ are persistently exciting [25][26]. The parameters $\alpha = 0.1779$, $\beta = 0.5421$ and $K = 5.0614$ need to be identified in the simulation.

Simulation procedure

Set the forgetting factor $\lambda = 0.98$. Take $u_1(t) = \log R_c(t)$, $u_2(t) = \log R_w(t)$, and the constant 1 as the elements of the regression vector $\varphi(t)$. For the parameter vector $\theta(t)$, set the initial values of its elements as $\alpha(0) = 0$, $\beta(0) = 0$ and $K(0) = 0$. Set the initial value for the covariance matrix as $P(0) = 1000I$. Implement the recursive estimation by exponential forgetting algorithm on the model (4-49) based on the simulated data set. The results of simulation are shown in Figure 4-2. In order to compare with the simulation results shown in figure 4-1, only part of the simulation results corresponding to the first 400 data points is shown.

From Figure 4-2 one can see that the windup does not occur, the estimates of system parameters are quite accurate. This means that the exponential forgetting algorithm works well with persistent excitation.

Figure 4-2. Exponential forgetting algorithm works well with persistently exciting.

Scale of X-axis: second $\times 10$.

(a) Random signal of production rate $R_c(t)$. (b) Random signal of dilution water flow rate $R_w(t)$. (c) Estimator gain $G_1(t)$. (d) Estimator gain $G_2(t)$. (e) The trace of the covariance matrix $P(t)$. (f) The "true value" of parameter α (dashed) and its estimate by exponential forgetting algorithm(solid). (g) The "true value" of parameter β (dashed) and its estimate by exponential forgetting algorithm(solid). (h) The "true value" of parameter K (dashed) and its estimate by exponential forgetting algorithm(solid).

Chapter 5

DIRECTIONAL FORGETTING ALGORITHM

5.1 Introduction

As mentioned in Chapter 4, estimator windup is undesirable. When it occurs, the covariance matrix $P(t)$ will grow exponentially and the estimator gain $P(t)\phi(t)$ will also grow exponentially. In this case, the exponential forgetting algorithm will become very sensitive to noise so that the estimation may be unreliable.

To find out the reason for estimator windup, review the exponential forgetting algorithm described by equations (4-16) and (4-17):

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\phi(t)[y(t) - \phi^T(t)\hat{\theta}(t-1)] \quad (5-1)$$

$$R(t) = \lambda(t)R(t-1) + \phi(t)\phi^T(t), \quad (5-2)$$

where $0 < \lambda < 1$. As the algorithm processes the data, the information matrix $R(t)$ is updated at each step by discounting old data $R(t-1)$ and adding the new information carried by $\phi(t)$. The exponential forgetting algorithm works well when $\{\phi(t)\}$ is persistently exciting and hence contains sufficient information about the system dynamics. In such a situation, the old data will be replaced by new data gradually. When $\{\phi(t)\}$ is not persistently exciting, it does not carry sufficient information. The old data $R(t-1)$ will be discounted continuously regardless of whether there is sufficient information or not. Thus, only a part of the old data can be replaced by the new data, the others will be discounted step by step. As a result, some eigenvalues of $R(t)$ will tend to

be zero so that the covariance matrix $P(t)$ and the estimator gain $P(t)\varphi(t)$ will continuously grow and will tend to become unbounded. This is the phenomenon called estimator windup.

Several methods have been proposed to avoid estimator windup. Considering the drawback of exponential forgetting, a more suitable forgetting philosophy should be to forget old data only when it can be replaced by new data. This is the basic idea of the Directional Forgetting Algorithms suggested by Kulhavy and Karny [12], Hagglund [13] and Kulhavy [15]. In the directional forgetting algorithms, the data is considered to have directions. The old data is forgotten only in the direction where the new data is coming from. According to such a philosophy, Kulhavy and Karny [12], then furthermore Kulhavy [15] proposed a directional forgetting algorithm based on the Bayesian estimation approach. Kulhavy's algorithm can prevent estimator windup but some of eigenvalues of $R(t)$ may become unbounded (Bittanti et al [17]). It means the algorithm may lose its tracking capability in some directions.

Parkum et al [18] proposed the Selective Forgetting Algorithm in which the eigenvalues of the covariance matrix are updated directly. That is, the updating is performed in terms of eigenvectors' directions and each eigenvalue is bounded at any time. The algorithm is a kind of directional forgetting. It is very directionally sensitive and works well regardless of the excitation condition. A drawback of the selective forgetting algorithm is that the eigenvalues and eigenvectors have to be computed at each update, which will greatly increase the computational requirement compared to the standard exponential forgetting algorithm.

Recently, a new directional forgetting algorithm is proposed by Cao and Schwartz [20], which does not need to compute eigenvalues and eigenvectors at each update to determine the forgetting direction. They replaced the forgetting factor λ by the forgetting matrix $F(t)$ which is able to discount $R(t-1)$ according to the direction of $\varphi(t)$. In this way, old data is discounted only in the direction where the new data is coming from.

5.2 Decomposition of Information Matrix

The new directional forgetting algorithm was derived based on the decomposition of the information matrix $R(t)$. The main idea and the derivation process are introduced as follows.

For the purpose of directional forgetting, change the update equation (5-2) to the following form

$$R(t) = F(t)R(t-1) + \varphi(t)\varphi^T(t), \quad (5-3)$$

where the matrix $F(t)$ is called the Forgetting Matrix, which replaces the forgetting factor λ in equation (5-2). $F(t)$ should be able to discount $R(t-1)$ according to the direction of $\varphi(t)$. There are many possible ways to determine $F(t)$. To determine $F(t)$ based on the decomposition of information matrix $R(t)$, factorize $R(t-1)$ into two parts

$$R(t-1) = R_1(t-1) + R_2(t-1). \quad (5-4)$$

The new algorithm requires that the forgetting is applied only to one part, say $R_2(t-1)$. Now, how to determine $R_1(t-1)$ and $R_2(t-1)$ will be a question that needs to be solved. To solve the question, the matrix $R_1(t-1)$ is required to satisfy the following equation

$$R_1(t-1)\varphi(t) = 0, \quad \varphi(t) \neq 0, \quad (5-5)$$

which means that the image space of $R_1(t-1)$ is orthogonal to $\varphi(t)$, that is

$$\text{Im } R_1(t-1) \subseteq \varphi^\perp(t). \quad (5-6)$$

Consequently,

$$\begin{aligned} R_2(t-1)\varphi(t) &= [R(t-1) - R_1(t-1)]\varphi(t) \\ &= R(t-1)\varphi(t). \end{aligned} \quad (5-7)$$

With equations (5-5) and (5-7), $R_1(t-1)$ and $R_2(t-1)$ cannot be determined uniquely. Consider the fact that the new data is coming in the form of $\varphi(t)\varphi^T(t)$, whose rank is 1. It is reasonable to require that $R_2(t-1)$ also has rank 1 so that $R_2(t-1)$ can be determined uniquely by [21]

$$R_2(t-1) = \frac{1}{\varphi^T(t)R(t-1)\varphi(t)} [R(t-1)\varphi(t)][R(t-1)\varphi(t)]^T. \quad (5-8)$$

The following example will show the factorizing process.

Example 5-1. Factorize the information matrix $R(t-1)$ into two parts.

Suppose the information matrix $R(t-1)$ and the regression vector $\varphi(t)$ are

$$R(t-1) = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}, \varphi(t) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

According to equations (5 - 4) and (5 - 8), $R(t-1)$ can be factorized into two parts:

$$R(t-1) = R_1(t-1) + R_2(t-1),$$

where

$$R_1(t-1) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, R_2(t-1) = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix},$$

which satisfy

$$R_1(t-1)\varphi(t) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = 0, \text{ and}$$

$$R_1(t-1) + R_2(t-1) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} = R(t-1).$$

One can see that the decomposition approach works for this example.

Let

$$\alpha(t) = \frac{1}{\varphi^T(t)R(t-1)\varphi(t)}. \quad (5 - 9)$$

Then $R_1(t-1)$ can be written as

$$\begin{aligned} R_1(t-1) &= R(t-1) - R_2(t-1) \\ &= R(t-1) - \alpha(t)[R(t-1)\varphi(t)][R(t-1)\varphi(t)]^T, \end{aligned} \quad (5 - 10)$$

which satisfies equation (5-5) and its rank is $n-1$ provided that $R(t-1)$ is positive definite [21][22].

Equation (5-9) requires that $\varphi(t)$ should be a nonzero vector. To ensure this, a dead zone ε is needed for $\varphi(t)$.

Define

$$\alpha(t) = 0 \quad \text{if } |\varphi(t)| \leq \varepsilon \quad (5 - 11)$$

where ε is determined based on the noise level in the data. When $|\varphi(t)| > \varepsilon$, $\alpha(t)$ is determined by equation (5-9).

Now return to the equation (5-3), applying exponential forgetting only to $R_2(t-1)$, the update equation (5-3) for the information matrix $R(t)$ becomes

$$R(t) = R_1(t-1) + \lambda R_2(t-1) + \varphi(t)\varphi^T(t). \quad (5-12)$$

The new information matrix $R(t)$ consists of three parts: $\varphi(t)\varphi^T(t)$ represents new data, $R_1(t-1)$ represents the old data that is orthogonal to $\varphi(t)$, and $R_2(t-1)$ represents the old data that should be discounted. It can be proved that the image space of $R_1(t-1)$ is orthogonal to the image space of the new data matrix $\varphi(t)\varphi^T(t)$ [21] [22]. Therefore, forgetting is applied only to $R_2(t-1)$.

The update equation (5-3) can also be written as

$$R(t) = \bar{R}(t-1) + \varphi(t)\varphi^T(t), \quad (5-13)$$

where

$$\bar{R}(t-1) = F(t)R(t-1). \quad (5-14)$$

The matrix $\bar{R}(t-1)$ is called the modified information matrix, it is positive definite and $R(t-1) - \bar{R}(t-1)$ is positive semidefinite. This means $R(t)$ will remain positive definite for all t so that the new directional forgetting algorithm can be implemented [22] .

Comparing equation (5-12) with (5-13), one gets

$$\begin{aligned} \bar{R}(t-1) &= R_1(t-1) + \lambda R_2(t-1) \\ &= R(t-1) - R_2(t-1) + \lambda R_2(t-1) \\ &= R(t-1) - (1-\lambda)R_2(t-1) \\ &= R(t-1) - (1-\lambda)\alpha(t)[R(t-1)\varphi(t)][R(t-1)\varphi(t)]^T \\ &= R(t-1) - (1-\lambda)\alpha(t)R(t-1)\varphi(t)\varphi^T(t)R(t-1) \\ &= [I - (1-\lambda)\alpha(t)R(t-1)\varphi(t)\varphi^T(t)]R(t-1) \\ &= [I - M(t)]R(t-1), \end{aligned} \quad (5-15)$$

where $R^T(t-1) = R(t-1)$ because $R(t-1)$ is symmetrical, and

$$M(t) = (1-\lambda)\alpha(t)R(t-1)\varphi(t)\varphi^T(t). \quad (5-16)$$

Comparing equation (5-14) with equation (5-15) one can find that the forgetting matrix $F(t)$ is defined by

$$F(t) = I - M(t) \quad (5 - 17)$$

where I denotes the identity matrix.

Now, one gets the update equations as

$$\begin{aligned} R(t) &= F(t)R(t-1) + \varphi(t)\varphi^T(t) \\ &= [I - M(t)]R(t-1) + \varphi(t)\varphi^T(t) \end{aligned} \quad (5 - 18)$$

$$\begin{aligned} M(t) &= (1 - \lambda)\alpha(t)R(t-1)\varphi(t)\varphi^T(t) \\ &= (1 - \lambda)\frac{R(t-1)\varphi(t)\varphi^T(t)}{\varphi^T(t)R(t-1)\varphi(t)}, |\varphi(t)| > \varepsilon. \end{aligned} \quad (5 - 19)$$

$$M(t) = 0, |\varphi(t)| \leq \varepsilon. \quad (5 - 20)$$

One needs $P(t) = R^{-1}(t)$ to implement the algorithm. Apply the matrix inversion lemma

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1} \quad (5 - 21)$$

to (5-13), letting

$$\bar{R}(t-1) = A, \varphi(t) = B, 1 = C, \varphi^T(t) = D. \quad (5 - 22)$$

Then the inverse of the information matrix $R(t)$, namely, the covariance matrix $P(t)$ can be obtained as

$$P(t) = R^{-1}(t) = \bar{P}(t-1) - \frac{\bar{P}(t-1)\varphi(t)\varphi^T(t)\bar{P}(t-1)}{1 + \varphi^T(t)\bar{P}(t-1)\varphi(t)}. \quad (5 - 23)$$

The modified covariance matrix $\bar{P}(t-1)$ is defined by

$$\bar{P}(t-1) = \bar{R}^{-1}(t-1) = R^{-1}(t-1)F^{-1}(t) = P(t-1)F^{-1}(t) \quad (5 - 24)$$

which can be obtained by applying the matrix inverse lemma to $F(t)$ and using equation (5-16).

Let

$$I = A, (\lambda - 1)\alpha(t)R(t-1)\varphi(t) = B, 1 = C, \varphi^T(t) = D, \quad (5 - 25)$$

then one can get the following equation when $|\varphi(t)| > \varepsilon$

$$\bar{P}(t-1) = P(t-1) + \frac{1-\lambda}{\lambda} \frac{\varphi(t)\varphi^T(t)}{\varphi^T(t)R(t-1)\varphi(t)}. \quad (5 - 26)$$

In the case $|\varphi(t)| \leq \varepsilon$, $\bar{P}(t-1)$ is defined by

$$\bar{P}(t-1) = P(t-1) \quad (5 - 27)$$

Finally, as a summary, the new directional forgetting algorithm is summarized with the following equations

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \quad (5 - 28)$$

$$K(t) = P(t)\varphi(t) = \frac{\bar{P}(t-1)\varphi(t)}{1 + \varphi^T(t)\bar{P}(t-1)\varphi(t)} \quad (5 - 29)$$

$$P(t) = \bar{P}(t-1) - \frac{\bar{P}(t-1)\varphi(t)\varphi^T(t)\bar{P}(t-1)}{1 + \varphi^T(t)\bar{P}(t-1)\varphi(t)} \quad (5 - 30)$$

$$\bar{P}(t-1) = P(t-1) + \frac{1-\lambda}{\lambda} \frac{\varphi(t)\varphi^T(t)}{\varphi^T(t)R(t-1)\varphi(t)} \quad |\varphi(t)| > \varepsilon \quad (5 - 31)$$

$$\bar{P}(t-1) = P(t-1) \quad |\varphi(t)| \leq \varepsilon \quad (5 - 32)$$

$$R(t) = [I - M(t)]R(t-1) + \varphi(t)\varphi^T(t) \quad (5 - 33)$$

$$M(t) = (1-\lambda) \frac{R(t-1)\varphi(t)\varphi^T(t)}{\varphi^T(t)R(t-1)\varphi(t)} \quad |\varphi(t)| > \varepsilon \quad (5 - 34)$$

$$M(t) = 0 \quad |\varphi(t)| \leq \varepsilon. \quad (5 - 35)$$

5.3 Simulation Example

Example 5-2. Practical properties of the new algorithm applied to TMP process.

The theoretical properties of the new directional forgetting algorithm were described in the previous sections. The practical properties of this new algorithm are examined via a simulation

example of the TMP process. To compare the new algorithm with the standard exponential forgetting algorithm, the same model and the same data are used.

Recall example 4-1. The model of the primary refiner in the TMP process is

$$L(t) = k(t)R_c^\alpha(t). \quad (5 - 36)$$

Its linear regression form and the corresponding regression vector as well as the parameter vector are

$$y(t) = \varphi^T(t)\theta(t) \quad (5 - 37)$$

$$\varphi(t) = [u_1(t) \quad 1]^T \quad (5 - 38)$$

$$\theta(t) = [\alpha(t) \quad K(t)]^T \quad (5 - 39)$$

Where $u_1(t) = \log R_c(t)$, $K(t) = \log k(t)$. The simulation process is as follows.

Getting data

In this example one uses exactly the same data as those used in example 4-1, which is shown again in Figure 5-1. That is, take the real data of the TMP process collected on June 21, 1996 for the primary refiner. Put the 1996 data set a and 1996 data set c together. The last data point of data set a is followed by the first data point of data set c. The number of data points of the combined data set is $N = 400$. Implement the off-line least-squares identification on the model (5-36) based on the combined data set. The identified system parameters are $\alpha = 0.1098$, $K = 2.4523$. Then use the real data of production rate $R_c(t)$ of the combined data set to generate the motor load using the model (5-36) and add white noise as follows:

$$L(t) = k(t)R_c^\alpha(t) = e^K [R_c(t)]^\alpha = 2.7183^{2.4523} \times R_c^{0.1098} + \varepsilon(t), \quad (5 - 40)$$

where $\varepsilon(t)$ is white noise with mean zero and variance 1. In this way, the system parameters are determined in advanced so that one can check the practical properties of this new directional forgetting algorithm by simulation. The parameters $\alpha = 0.1098$ and $K = 2.4523$ need to be identified in the simulation. During the simulation process, as an input signal, the production rate $R_c(t)$ is kept constant for a long period of time, it changes only once in the whole process, from 112 rpm to 120 rpm, which is depicted in graph (a) of Figure 5 -1.

Simulation procedure, results and discussion

Set the forgetting factor as $\lambda = 0.98$. The initial values of parameter vector are set as $\alpha(0) = 0$, $K(0) = 0$. Set the initial value of covariance matrix as $P(0) = 1000I$. Implement recursive estimation by the new directional forgetting algorithm applied to the model (5-37) based on the above simulated data. The results of simulation are shown in Figure 5-1.

Compare Figure 5-1 with Figure 4-1. One can see the obvious differences and improvement. The exponential growth of the covariance matrix $P(t)$ is avoided. As a result, the associated exponential growth of the estimator gain G_1 (and G_2) is also avoided. The reason is, while implementing the new algorithm, although the excitation condition is poor, the old data in the information matrix $R(t)$ is not forgotten unless it can be replaced by new information. In other words, if no new information is available, the old data will not be forgotten. The old data is forgotten only in the directions where the new data is coming from. Therefore, none of eigenvalues in $R(t)$ would tend to zero, the exponential growth of $P(t)$ and G_1 (and G_2) does not occur. When the input signal $R_c(t)$ changes, the new data replaces the old data in $R(t)$, then the estimates of the parameters are improved based on the new data.

This is a very important property of the new algorithm for the identification of the TMP process. From the example 4-1 one can see, the standard exponential forgetting algorithm cannot be used when the excitation condition is poor, which often happens in the TMP process. Example 5-1 demonstrates that the novel directional forgetting algorithm can deal with such a situation successfully. While implementing the new algorithm, old data in $R(t)$ will be held until new data comes. When new data comes, the estimates are improved. There may not be sufficient information about the system during short periods of the data set, and the estimates may not be able to converge to their "true values" immediately. But they can be improved step by step whenever new data arrives. With such consideration, one has some reason to believe that if the data set is long enough, the estimates of the system parameters would be close enough to their "true values". With such a practical property, the new directional forgetting algorithm will be a very suitable technique for on-line identification of the TMP process. The on-line identification based on the real data of the TMP process will be presented in chapter 6. The simulation of on-line identification based on the simulated data of the TMP process will be present in Chapter 7.

Figure 5-1. Novel directional forgetting algorithm can prevent windup.

Scale of X-axis: second $\times 10$.

- (a) Real data of production rate $R_c(t)$.
- (b) Estimator gain G_1 .
- (c) Estimator gain G_2 .
- (d) The trace of the covariance matrix $P(t)$.
- (e) The "true value" of parameter $\alpha(t)$ (dashed) and its estimate by the novel directional forgetting algorithm (solid).
- (f) The "true value" of parameter $K(t)$ (dashed) and its estimate by the novel directional forgetting algorithm (solid).

Chapter 6

On-LINE IDENTIFICATION OF TMP PROCESS

6.1 Introduction

In this chapter, the novel directional forgetting algorithm (DF), which was introduced in Chapter 5, is implemented to identify the dynamics of a two-stage thermo-mechanical pulping (TMP) process. One can see from Chapter 3 that the system parameters of the TMP process are time-varying. In such a case, to control the complex TMP process safely and efficiently, and to produce pulp of desirable quality, it is necessary to have a model of the system available on-line while the system is in operation. The model should be based on observed data up to the current time so that the necessary decisions about the system can be made.

From Chapter 4, one can see that estimator windup is undesirable because in such a case the estimation may be unreliable. Also, one can see from Chapter 4 that the exponential forgetting algorithm works well only if the process is properly excited all the time. However, the TMP process does not meet this requirement since the input control variables of the TMP process are often kept constant for quite a long time while the system is in operation. As a result, the exponential forgetting algorithm is not suitable for on-line identification of the TMP process.

The directional forgetting algorithm (DF) is a very suitable technique to solve the problem. As discussed in Chapter 5, the algorithm has the same dynamic tracking ability as that of standard exponential forgetting algorithm under the condition of persistent excitation. When the excitation

condition is poor, like the situation in the TMP process, the windup phenomenon can be avoided, the old data in the information matrix $R(t)$ will not be forgotten unless it can be replaced by new data. In this way, the estimates of the system parameters can be improved step by step whenever the new data contains new information. With such a practical property, the directional forgetting algorithm (DF) is a suitable algorithm for the on-line identification of the TMP process.

6.2 Model Structures, Data and Algorithm Formulae

The models for the primary refiner and the secondary refiner are the same as those used in Chapter 3. The data sets are also the same as those used in Chapter 3. As used in Chapter 3, the static model for the primary refinery is

$$L(t) = k(t) \frac{R_c^\alpha(t)}{R_w^\beta(t)}. \quad (6 - 1)$$

The dynamic model for the primary refiner is

$$L(t) + aL(t-1) = b \frac{R_c^\alpha(t-1)}{R_w^\beta(t-1)}. \quad (6 - 2)$$

For the secondary refiner, the static model is

$$L(t) = k(t) \frac{[R_c(t)C_o(t)]^\alpha}{R_w^\beta(t)}. \quad (6 - 3)$$

The dynamic model for the secondary refiner is

$$L(t) + aL(t-1) = b \frac{[R_c(t-1)C_o(t-1)]^\alpha}{R_w^\beta(t-1)}. \quad (6 - 4)$$

Where, t represents sampling instants, $t = 1, 2, \dots$. The sampling time was 10 seconds. The terms $L(t)$, $R_c(t)$, $R_w(t)$ represent the motor load, the production rate and the dilution water flow rate, respectively. The term $C_o(t)$ represents the outlet consistency of the primary refiner, or the inlet consistency of the secondary refiner. The terms $\alpha(t)$, $\beta(t)$ and $k(t)$ represent unknown parameters, which are considered to be time-varying in on-line identification.

As is introduced in Chapter 5, the novel directional forgetting algorithm is represented with the following equations

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \quad (6-5)$$

$$P(t) = \bar{P}(t-1) - \frac{\bar{P}(t-1)\varphi(t)\varphi^T(t)\bar{P}(t-1)}{1 + \varphi^T(t)\bar{P}(t-1)\varphi(t)} \quad (6-6)$$

$$\bar{P}(t-1) = P(t-1) + \frac{1-\lambda}{\lambda} \frac{\varphi(t)\varphi^T(t)}{\varphi^T(t)R(t-1)\varphi(t)} \quad |\varphi(t)| > \varepsilon \quad (6-7)$$

$$\bar{P}(t-1) = P(t-1) \quad |\varphi(t)| \leq \varepsilon \quad (6-8)$$

$$R(t) = [I - M(t)]R(t-1) + \varphi(t)\varphi^T(t) \quad (6-9)$$

$$M(t) = (1-\lambda) \frac{R(t-1)\varphi(t)\varphi^T(t)}{\varphi^T(t)R(t-1)\varphi(t)} \quad |\varphi(t)| > \varepsilon \quad (6-10)$$

$$M(t) = 0 \quad |\varphi(t)| \leq \varepsilon \quad (6-11)$$

6.3 Identification of parameters for the Primary Refiner

6.3.1 Identification based on 1996 data set c

Three data sets are used to recursively identify the parameters of the TMP process. The first data set is 1996 data set c that is shown in Figure 6-1. The number of data points is $N = 100$. In this data set, as mentioned in previous chapters, only the production rate $R_c(t)$ changes once, from 112 rpm to 120 rpm, the dilution water flow rate $R_w(t)$ is kept constant (there was noise that effected the data). In this case, the parameter connected to $R_w(t)$, namely the parameter $\beta(t)$ cannot be identified individually. Therefore, one puts the term $R_w(t)$ and the gain $k(t)$ together, the static model (6-1) becomes

$$L(t) = k(t)R_c^\alpha(t). \quad (6-12)$$

The linear regression structure of model (6-12), and the corresponding regression vector as well as the parameter vector are

$$y(t) = \alpha(t)u_1(t) + K(t), \quad (6-13)$$

$$\varphi(t) = [u_1(t) \ 1]^T, \text{ and} \quad (6-14)$$

$$\theta(t) = [\alpha(t) \ K(t)]^T, \quad (6-15)$$

where, $y(t) = \log L(t)$, $u_1(t) = \log R_c(t)$, $K(t) = \log k(t)$.

Set the forgetting factor as $\lambda = 0.98$. Set the initial values of the parameters as $\alpha(0) = 0$, $K(0) = 0$, and the initial value of covariance matrix as $P(0) = 1000I$ (I denotes the identity matrix). Then the results of on-line identification using the model structure defined by equation (6-13) are shown in Figure 6-1.

As analyzed in Chapter 4, the production rate $R_c(t)$ changes only once in the whole process thus the input signal is not the persistently exciting. The estimator windup will occur if one implements the standard exponential forgetting algorithm with such excitation conditions. The directional forgetting algorithm can prevent the windup. Graph (d) in Figure 6-1 illustrates that the windup phenomenon does not occur. Before the sampling instant 40, $R_c(t)$ is kept constant, there is no new information arriving, the old data in the information matrix $R(t)$ is unchanged, neither replaced by the new data, nor discounted step by step. When the production rate $R_c(t)$ changes from 112 rpm to 120 rpm, the information matrix $R(t)$ receives new information about the system dynamics, then the recursive estimator identifies the system parameters based on the new information. From graphs (e) and (f) one can see, corresponding to the first data point and the last data point, the on-line identified $\alpha(t)$ varies from 0 to 0.3565, the on-line identified $K(t)$ varies from 0 to 1.2731. In Figure 6-1, the off-line identified values of parameters α and K are also depicted, in graphs (e) and (f), respectively. As analyzed in Chapter 3, the off-line identified values of the parameters are close to their “true values”. Therefore, they can be the reference values for on-line identification. The graphs (e) and (f) indicate that the on-line estimated parameter values tend to converge towards the off-line identified values of these parameters. Therefore, one has some reason to believe, if the number of data points is large enough, the on-line estimated parameter values will be close enough to their “true values”. With a real data set, like the real data sets from the TMP process, one can never know the “true values” of the system parameters. One can only estimate these “true values”. However, With a simulated data set, the “true values” of the system parameters can be known in advance. In this way, one can compare

the estimated parameter values with their “true values”. This will be demonstrated in the next chapter.

If one sets the initial values of the parameters to $\alpha(0) = 0.2847$ and $K(0) = 1.6153$, which are the identified values of these parameter by the off-line least-squares algorithm, the on-line identification results will be improved. This means that $\alpha(0) = 0.2847$ and $K(0) = 1.6153$ are what one guesses the parameters to be before the on-line identification algorithm is implemented. As analyzed in Chapter 3, these values are close to their “true values”. With such initial conditions, the on-line identified values of the parameters will have much less deviations from their reference values, thus have much less deviation from their “true values” too. The on-line identification results with such nonzero initial conditions are shown in Figure 6-2.

Figure 6-2 illustrates, corresponding to the first data point and the last data point, the on-line identified $\alpha(t)$ varies from 0.2847 to 0.2789, the on-line identified $K(t)$ varies from 1.6153 to 1.6432. Comparing Figure 6-2 with Figure 6-1 one can see, with nonzero initial conditions, the on-line identification results are significantly improved. Also, refer to the model (6-12), $L(t) = k(t)R_c^\alpha(t)$, with zero initial conditions, the estimated motor load at the started point will be zero. See Figure 6-1, graph (c). However, from Figure 6-2, graph (c), one can see, with nonzero initial conditions, the zero starting motor load is avoided, the obvious jump of estimated motor load is also avoided. In Figure 6-2, graph (e) and (f) illustrate, starting from their reference values, that the estimates of the parameters are significantly improved.

The on-line identification using the dynamic model defined by equation (6-2) is also implemented. In 1996 data set c, the infeed dilution water flow rate $R_w(t)$ does not change, only the production rate $R_c(t)$ changes once. One then combines the terms $R_w^\beta(t)$ and the parameter $b(t)$, model (6-2) then becomes

$$L(t) + a(t)L(t-1) = b(t)R_c^\alpha(t-1). \quad (6-16)$$

The corresponding regression vector and the parameter vector are

$$\varphi(t) = [-L(t-1) \quad R_c^\alpha(t-1)]^T, \text{ and} \quad (6-17)$$

$$\theta(t) = [a(t) \quad b(t)]^T. \quad (6-18)$$

Based on the results of off-line identification on model (6-12), one sets the parameter $\alpha = 0.2847$. Hold this value constant during the identification process. Set the forgetting factor as $\lambda = 0.98$, the initial values of parameters as $a(0) = 0$, $b(0) = 0$, and the initial value of covariance matrix as $P(0) = 1000I$. Then the results of on-line identification using the model (6-16) are shown in figure 6-3. Figure 6-3 illustrates, with zero initial conditions, corresponding to the first data point and the last data point, the estimated value of parameter $a(t)$ varies from 0 to -0.6782 , the estimated value of parameter $b(t)$ varies from 0 to 1.6187.

One can set the initial parameter values as $a(0) = -0.6785$ and $b(0) = 1.6176$ from the results of off-line identification, then the results of on-line identification are shown in figure 6-4. Figure 6-4 illustrates, with these nonzero initial conditions, the identified value of parameter $a(t)$ varies from -0.6785 to -0.6747 and the identified value of parameter $b(t)$ varies from 1.6176 to 1.6364.

Note the different estimated results between figure 6-1 and figure 6-3. Based on the same data set, the estimated parameter values using the dynamic model (6-16) have much less deviations from their reference values. To find the reason, one notes the different structures of the regression vectors (6-17) and (6-14),

$$\varphi(t) = [u_1(t) \quad 1]^T \quad (6-14)$$

$$\varphi(t) = [-L(t-1) \quad R_c^\alpha(t-1)]^T \quad (6-17)$$

which decide the different structures of information matrix $R(t)$ between the dynamic model (6-16) and the static model (6-12). For both models, the information matrix is defined by

$$R(t) = \frac{1}{N} \sum_{i=1}^N \varphi(i) \varphi^T(i). \quad (6-19)$$

During the period when the production $R_c(t)$ does not change, there is not any new information carried by the regression vector (6-14), but there is still new information carried by the regression vector (6-17), which is caused by changing of motor load. This is the reason why the identified results using dynamic model (6-16) is different from that of using static model (6-12).

Figure 6-1. On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set c, zero initial conditions, scale of X-axis: second $\times 10$.

(a) Production rate $R_c(t)$. (b) Measured motor load and its off-line estimate $L(t)$. (c) Measured motor load and its on-line estimate $L(t)$. (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\alpha(t)$ (solid) and off-line identified α (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Figure 6-2. On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set c, nonzero initial conditions, scale of X-axis: second $\times 10$.

(a) Production rate $R_c(t)$. (b) Measured motor load and its off-line estimate $L(t)$. (c) Measured motor load and its on-line estimate $L(t)$. (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\alpha(t)$ (solid) and off-line identified α (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Figure 6-3. On-line Identification of TMP process, the primary refiner, dynamic model, based on 1996 data set c, zero initial conditions, scale of X-axis: second $\times$ 10.

(a) Production rate $R_c(t)$. (b) Measured motor load and its off-line estimate $L(t)$. (c) Measured motor load and its on-line estimate $L(t)$. (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\alpha(t)$ (solid) and off-line identified α (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Figure 6-4. On-line Identification of TMP process, the primary refiner, dynamic model, based on 1996 data set c, nonzero initial conditions, scale of X-axis: second $\times 10$.

(a) Production rate $R_c(t)$. (b) Measured motor load and its off-line estimate $L(t)$. (c) Measured motor load and its on-line estimate $L(t)$. (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\alpha(t)$ (solid) and off-line identified α (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

6.3.2 Identification based on 1996 data set a and 1997 data set b

In 1996 data set a (shown in Figure 6-5) and 1997 data set b (shown in Figure 6-8), the production rate $R_c(t)$ is kept constant, only the infeed dilution water flow rate $R_w(t)$ changes. In this case, the parameters $\alpha(t)$ cannot be identified individually. One puts the term $R_c^\alpha(t)$ and the term $k(t)$ together, then the static model (6-1) becomes

$$L(t) = \frac{k(t)}{R_w^\beta(t)} \quad (6 - 20)$$

The linear regression structure of model (6-20) and the corresponding regression vector as well as the parameter vector are

$$y(t) = -\beta(t)u_2(t) + K(t), \quad (6 - 21)$$

$$\varphi(t) = [-u_2(t) \quad 1]^T, \text{ and} \quad (6 - 22)$$

$$\theta(t) = [\beta(t) \quad K(t)]^T. \quad (6 - 23)$$

The number of data points in 1996 data set a is $N = 171$. The number of data points in 1997 data set b is $N = 269$. One sets the forgetting factor as $\lambda = 0.98$, the initial value of P -matrix as $P(0) = 1000I$, and the initial parameter values as $\beta(0) = 0$, $K(0) = 0$ for both cases. The results of on-line identification are shown in Figure 6-5 and Figure 6-8, respectively. In Figure 6-5, corresponding to the data point 1 and 171, the on-line identified $\beta(t)$ varies from 0 to 0.0795, the on-line identified $K(t)$ varies from 0 to 3.4053. In Figure 6-8, corresponding to the data point 1 and 269, the on-line identified $\beta(t)$ varies from 0 to 0.7285, the on-line identified $K(t)$ varies from 0 to 7.0690.

For the data set shown in Figure 6-5, one can also set nonzero initial values for the parameters as $\beta(0) = 0.4265$ and $K(0) = 5.2820$, namely, take the initial values of these parameters as those of the identified values by off-line identification. In this way, the on-line identification results would be improved. The on-line identification results with such nonzero initial conditions are shown in Figure 6-6.

In Figure 6-6, corresponding to the data point 1 and 171, the on-line identified $\beta(t)$ varies from 0.4265 to 0.4481, the on-line identified $K(t)$ varies from 5.2820 to 5.3994. In Figure 6-6, graphs (e) and (f) illustrate, for the identified $\beta(t)$ and $K(t)$, there are much less deviations from their reference values, obvious improvement can be seen. With improved identified parameters, the estimated motor load is also improved. Comparing Figure 6-6 with Figure 6-5, the estimated motor load is much closer to the mean value of the measured one. Figure 6-7 illustrates the prediction errors with zero and nonzero initial conditions.

With the same reason, for 1997 data set b shown in Figure 6-8, one can set the initial values of parameters as $\beta(0) = 1.1339$ and $K(0) = 9.3466$, which were the results of off-line identification. Then Figure 6-9 illustrates that the on-line identified $\beta(t)$ varies from 1.1339 to 1.1907. The on-line identified $K(t)$ varies from 9.3466 to 9.6578. Obvious improvement of parameter identification as well as the estimated motor load can also be seen.

Figure 6-5. On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set a, zero initial conditions, scale of X-axis: second $\times 10$.

(a) Infeed dilution water flow rate $R_w(t)$. (b) Measured motor load(solid) and its off-line estimate $L(t)$ (dotted). (c) Measured motor load(solid) and its on-line estimate $L(t)$ (dotted). (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\beta(t)$ (solid) and off-line identified β (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Figure 6-6. On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set a, nonzero initial conditions, scale of X-axis: second $\times 10$.

(a) Infeed dilution water flow rate $R_w(t)$. (b) Measured motor load(solid) and its off-line estimate $L(t)$ (dotted). (c) Measured motor load(solid) and its on-line estimate $L(t)$ (dotted). (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\beta(t)$ (solid) and off-line identified β (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Figure 6-7. On-line Identification of TMP process, the primary refiner, static model, based on 1996 data set a, comparison of prediction error of motor load, scale of X-axis: second $\times$ 10.

Prediction error of motor load with zero initial conditions (dotted).

Prediction error of motor load with nonzero initial conditions (solid).

Figure 6-8. On-line Identification of TMP process, the primary refiner, static model, based on 1997 data set b, zero initial conditions, scale of X-axis: second $\times 10$.

(a) Infeed dilution water flow rate $R_w(t)$. (b) Measured motor load(solid) and its off-line estimate $L(t)$ (dotted). (c) Measured motor load(solid) and its on-line estimate $L(t)$ (dotted). (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\beta(t)$ (solid) and off-line identified β (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Figure 6-9. On-line Identification of TMP process, the primary refiner, static model, based on 1997 data set b, nonzero initial conditions, scale of X-axis: second $\times 10$.

(a) Infeed dilution water flow rate $R_w(t)$. (b) Measured motor load(solid) and its off-line estimate $L(t)$ (dotted). (c) Measured motor load(solid) and its on-line estimate $L(t)$ (dotted). (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\beta(t)$ (solid) and off-line identified $\beta(t)$ (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified $K(t)$ (dashed).

6.4 Identification of parameters for the Secondary Refiner

6.4.1 Identification based on 1997 data set a

Two data sets are used to identify the system parameters of the secondary refiner. They are 1997 data set a and 1997 data set c, both were collected on June 9, 1997. For the secondary refiner, the static model structure is expressed by equation (6-3). Figure 6-10 shows the 1997 data set a where the infeed dilution water flow rate $R_w(t)$ is kept constant, only the production rate $R_c(t)$ as well as the consistency $C_o(t)$ change. In this case, the parameter connecting to $R_w(t)$, namely, the parameter $\beta(t)$ cannot be identified individually. Combining $\beta(t)$ with $K(t)$, the modified structure of the model (6-3) becomes

$$L(t) = k(t)[R_c(t)C_o(t)]^\alpha. \quad (6 - 24)$$

Now, the linear regression structure of model (6-24), and the corresponding regression vector as well as the parameter vector are

$$y(t) = \alpha(t)u_1(t) + K(t), \quad (6 - 25)$$

$$\varphi(t) = [u_1(t) \quad 1]^T, \text{ and} \quad (6 - 26)$$

$$\theta(t) = [\alpha(t) \quad K(t)]^T, \quad (6 - 27)$$

Where, $y(t) = \log L(t)$, $u_1(t) = \log [R_c(t)C_o(t)]$, $K(t) = \log k(t)$.

In 1997 data set a, the number of data points is $N = 278$. Set the forgetting factor as $\lambda = 0.98$, and the initial value of covariance matrix as $P(0) = 1000I$. The initial values of the parameters are set to $\alpha(0) = 0$, and $K(0) = 0$. Then the results of recursive identification by the directional forgetting algorithm on the model (6-24) are shown in figure 6-10. Figure 6-10 illustrates, with zero initial conditions, corresponding to the data point 1 and data point 278, the on-line identified value of parameter $\alpha(t)$ varies from 0 to 1.3731, the on-line identified value of parameter $K(t)$ varies from 0 to -9.2375.

Referring to the identified values of these parameters from the off-line identification, the nonzero initial parameter values can be set as $\alpha(0) = 1.8986$, and $K(0) = -13.8161$. Under such nonzero

initial conditions, the results of on-line identification are shown in figure 6-11. Figure 6-11 illustrates, corresponding to the data point 1 and data point 278, the on-line identified value of parameter $\alpha(t)$ varies from 1.8986 to 1.8016, the on-line identified value of parameter $K(t)$ varies from -13.8161 to -12.9587. From graphs (e) and (f) in figure 6-11 one can see, with such nonzero initial conditions, the on-line identified parameter values are significantly improved.

The on-line identification using the dynamic model defined by equation (6-4) is also implemented. Based on the same data set shown in the figure 6-10, the parameter connecting with unchanged infeed dilution water flow rate $R_w(t)$, namely, the parameter $\beta(t)$ cannot be identified individually. Combining the term $R_w^\beta(t)$ with $b(t)$, the structure of dynamic model (6-4) becomes

$$L(t) + a(t)L(t-1) = b(t)[R_c(t-1)C_0(t-1)]^\alpha. \quad (6-28)$$

Now, the corresponding regression vector and the parameter vector are

$$\varphi(t) = \begin{bmatrix} -L(t-1) & [R_c(t-1)C_0(t-1)]^\alpha \end{bmatrix}^T, \text{ and} \quad (6-29)$$

$$\theta(t) = \begin{bmatrix} a(t) & b(t) \end{bmatrix}^T. \quad (6-30)$$

The parameter α is set to 1.8986, which was identified by the least-squares method in the off-line identification. Set the forgetting factor as $\lambda = 0.98$, and the initial values of the parameters as $a(0) = 0$, $b(0) = 0$, then the results of on-line identification using model (6-28) are shown in Figure 6-12. Figure 6-12 illustrates, with zero initial conditions, corresponding to the data point 1 and data point 278, the on-line identified value of parameter $a(t)$ varies from 0 to -0.4290, the on-line identified value of parameter $b(t)$ varies from 0 to 0.58×10^{-6} .

Take the initial values $a(0) = -0.4451$, and $b(0) = 0.5559 \times 10^{-6}$ as those from the results of off-line identification, the results of on-line identification are shown in Figure 6-13. Figure 6-13 illustrates, corresponding to the data point 1 and data point 278, the on-line identified value of parameter $a(t)$ varies from -0.4451 to -0.4188, the on-line identified value of parameter $b(t)$ varies from 0.5559×10^{-6} to 0.5902×10^{-6} .

Figure 6-10. On-line Identification of TMP process, the secondary refiner, static model, based on 1997 data set a, zero initial conditions, scale of X-axis: second $\times 10$.

(a) Production rate $R_c(t)$. (b) Consistency $C_0(t)$. (c) Measured motor load(solid) and its on-line estimate $L(t)$ (dotted). (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\alpha(t)$ (solid) and off-line identified α (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Figure 6-11. On-line Identification of TMP process, the secondary refiner, static model, based on 1997 data set a, nonzero initial conditions, scale of X-axis: second $\times 10$.

(a) Production rate $R_c(t)$. (b) Consistency $C_0(t)$. (c) Measured motor load(solid) and its on-line estimate $L(t)$ (dotted). (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $\alpha(t)$ (solid) and off-line identified α (dashed). (f) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Figure 6-12. On-line Identification of TMP process, the secondary refiner, dynamic model, based on 1997 data set a, zero initial conditions, scale of X-axis: second $\times 10$.

(a) Production rate $R_c(t)$. (b) Consistency $C_0(t)$. (c) Measured motor load(solid) and its on-line estimate $L(t)$ (dotted). (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $a(t)$ (solid) and off-line identified a (dashed). (f) On-line identified $b(t)$ (solid) and off-line identified b (dashed).

Figure 6-13. On-line Identification of TMP process, the secondary refiner, dynamic model, based on 1997 data set a, nonzero initial conditions, scale of X-axis: second $\times 10$.

(a) Production rate $R_c(t)$. (b) Consistency $C_0(t)$. (c) Measured motor load (solid) and its on-line estimate $L(t)$ (dotted). (d) The trace of the covariance matrix $P(t)$. (e) On-line identified $a(t)$ (solid) and off-line identified a (dashed). (f) On-line identified $b(t)$ (solid) and off-line identified b (dashed).

6.4.2 Identification based on 1997 data set c

In 1997 data set c shown in Figure 6–14, all input variables $R_c(t)$, and $R_w(t)$, as well as the consistency $C_d(t)$ change, the model (6-3) can be used without any modifications. The linear regression structure of the model (6 –3) is

$$y(t) = \alpha(t)u_1(t) - \beta(t)u_2(t) + K(t), \quad (6 - 31)$$

where

$$y(t) = \log L(t), \quad u_1(t) = \log [R_c(t)C_d(t)], \quad u_2(t) = \log R_w(t), \quad K(t) = \log k(t).$$

The corresponding regression vector and the parameter vector are

$$\varphi(t) = [u_1(t) \quad -u_2(t) \quad 1]^T, \text{ and} \quad (6 - 32)$$

$$\theta(t) = [\alpha(t) \quad \beta(t) \quad K(t)]^T. \quad (6 - 33)$$

In 1997 data set c, the number of data points is $N = 245$. One sets the forgetting factor as $\lambda = 0.98$, and the initial value of covariance matrix as $P(0) = 1000I$. The initial parameter values are set as $\alpha(0) = 0$, $\beta(0) = 0$, and $K(0) = 0$. Then the results of on-line identification using model (6-31) are shown in Figure 6-14. Figure 6-14 illustrates, corresponding to the data point 1 to data point 245, the on-line identified value of parameter $\alpha(t)$ varies from 0 to 1.4700, the on-line identified value of parameter $\beta(t)$ varies from 0 to 0.2538, the on-line identified value of parameter $K(t)$ varies from 0 to –9.0999.

Referring to the results of off-line identification, the nonzero initial values of the parameters are set as $\alpha(0) = 2.3701$, $\beta(0) = 0.4442$, and $K(0) = -16.1050$. The initial values of the forgetting factor and the covariance matrix are set as $\lambda = 0.98$, $P(0) = 1000I$. With such nonzero initial conditions, the results of on-line identification are show in Figure 6-15. In Figure 6-15, the on-line identified value of parameter $\alpha(t)$ varies from 2.3701 to 2.3269. The on-line identified value of parameter $\beta(t)$ varies from 0.4442 to 0.3862. The on-line identified value of parameter $K(t)$ varies from –16.1050 to –15.9782. With such nonzero initial conditions, the parameter identification and the estimation of motor load are improved.

From the on-line identification results shown in all the figures of this chapter, the following conclusions can be obtained:

While implementing the new directional forgetting algorithm applied to on-line identification of the TMP process, regardless of whether the data is persistently exciting or not, the phenomenon of estimator windup does not occur.

As analyzed in Chapter 3, the identified parameter values by the off-line least-squares algorithm are close to their “true values”. One takes these values as the reference values in the process of on-line identification. Since the real data sets have not carried sufficient information about the system dynamics, the on-line identified values of the system parameters are not close enough to their reference values when the estimator starts with zero initial conditions. However, when the estimator starts with nonzero initial conditions, especially with their reference values, the on-line identified values of the system parameters are improved significantly. The estimated motor load is also improved. From the on-line identification results with real data sets, one believes, if the data sets are long enough, the on-line identified values of the system parameters will be close enough to their reference values and therefore will be close enough to their “true values”. This will be demonstrated in the next chapter.

The on-line estimated motor loads approximate the mean values of the measured ones. It means the models fix the TMP process and they are suitable for on-line identification of the TMP process.

Figure 6-14. On-line Identification of TMP process, the secondary refiner, static model, based on 1997 data set c, zero initial conditions, scale of X-axis: second $\times 10$.

(a) Production rate $R_c(t)$. (b) Consistency $C_0(t)$. (c) Infeed dilution water flow rate $R_w(t)$. (d) Measured motor load(solid) and its on-line estimate $L(t)$ (dotted). (e) The trace of the covariance matrix $P(t)$. (f) On-line identified $\alpha(t)$ (solid) and off-line identified α (dashed). (g) On-line identified $\beta(t)$ (solid) and off-line identified β (dashed). (h) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Figure 6-15. On-line Identification of TMP process, the secondary refiner, static model, based on 1997 data set c, nonzero initial conditions, scale of X-axis: second $\times 10$.

(a) Production rate $R_c(t)$. (b) Consistency $C_0(t)$. (c) Infeed dilution water flow rate $R_w(t)$. (d) Measured motor load (solid) and its on-line estimate $L(t)$ (dotted). (e) The trace of the covariance matrix $P(t)$. (f) On-line identified $\alpha(t)$ (solid) and off-line identified α (dashed). (g) On-line identified $\beta(t)$ (solid) and off-line identified β (dashed). (h) On-line identified $K(t)$ (solid) and off-line identified K (dashed).

Chapter 7

COMPARISON BETWEEN DF AND EF

7.1 Introduction

This chapter will demonstrate the effectiveness of the directional forgetting (DF) algorithm. Some results of comparisons between the novel directional forgetting (DF) algorithm and the standard exponential forgetting (EF) algorithm are presented. It will be shown that DF algorithm has obvious advantages over EF algorithm.

As discussed in previous chapters, the standard exponential forgetting algorithm is not suitable for on-line identification of the TMP process because in such a case windup will occur and the estimation may be unreliable. On the contrary, when there is not new data coming, the novel directional forgetting algorithm can avoid the windup, the information matrix $R(t)$ can be held unchanged until there is new information about the system dynamics. In this way, the estimates of the parameters can be improved step by step and be able to tend to their "true values". Therefore, one should choose DF algorithm rather than EF algorithm to implement on-line identification of the TMP process. To demonstrate this, some results of comparisons between DF algorithm and EF algorithm are presented. In order to give a fair comparison between these two algorithms, the comparing processes are implemented under the same conditions. That is, the same models and the same data are used. The initial conditions are the same, too.

In order to investigate the ability of these two algorithms in terms of parameter converging behaviors, and in terms of ability to track fast or sudden parameter changes, one needs to know the "true values" of these parameters in advance. The real data sets are not suitable for such purposes because one never knows the "true values" of identified parameters from the real data. One can only estimate the parameter values by applying various identification techniques. If the real data of the TMP process were used, one can substitute the off-line identified values of the parameters for these "true values", which are not the "true values". The off-line identified values are not accurate enough for the purpose of comparison. For the purpose of the comparison, the simulated data are used.

7.2 Comparison between DF and EF in terms of parameter converging behaviors

The purpose of this simulation is to compare the DF algorithm with the standard EF algorithm in terms of identified parameter converging behaviors. The simulated data is contaminated by stochastic noise. In the simulation, the static model of the primary refiner of the TMP process is used.

Model structure

For the primary refiner of the TMP process, the static model is

$$L(t) = k(t) \frac{R_c^\alpha(t)}{R_w^\beta(t)}. \quad (7 - 1)$$

The corresponding linear regression structure, the regression vector and the parameter vector are

$$y(t) = \varphi^T(t) \theta(t), \quad (7 - 2)$$

$$\varphi(t) = [u_1(t) \quad -u_2(t) \quad 1]^T, \text{ and} \quad (7 - 3)$$

$$\theta(t) = [\alpha(t) \quad \beta(t) \quad K(t)]^T, \quad (7 - 4)$$

where $y(t) = \log L(t)$, $u_1(t) = \log R_c(t)$, $u_2(t) = \log R_w(t)$, $K(t) = \log k(t)$.

Data simulation

In the simulated data set, the production rate $R_c(t)$ and the infeed dilution water flow rate $R_w(t)$ are the input signals and the motor load $L(t)$ is the output. The data is simulated as follows. The

production rate $R_c(t)$ is made up of two parts. The infeed dilution water flow rate $R_w(t)$ is also made up of two parts. For $R_c(t)$ and $R_w(t)$, the first part is formed by taking the production rate $R_c(t)$ and the infeed dilution water flow rate $R_w(t)$ from the real data sets, say 1996 data set a and 1996 data set c. The last data point of data set a is followed by the first data point of data set c. The number of data points in data set a is $N = 300$. The number of data points in data set c is $N = 100$. The total number of data points in the first part of the simulated data set is $N = 400$. In the second part of the simulated data, the production rate $R_c(t)$ is a square wave with period of 218 sampling points, the signal shifts between 84.888 rpm and 119.888 rpm, the infeed dilution water flow rate $R_w(t)$ is a square wave with period of 222 sampling points, the signal shifts between 182 lpm and 222 lpm. The number of data points in the second part of the simulated data set is $N = 900$. Therefore, the total number of data points in the whole simulated data is $N = 1300$. One takes the simulated $R_c(t)$ and $R_w(t)$ as input signals, sets the parameters values as $\alpha(t) = 0.1779$, $\beta(t) = 0.5421$, and $K(t) = 5.0614$, which was the off-line identified values of these parameter based on the real data of the TMP process. Then, according to the model structure described by equation (7-1), the motor load $L(t)$ was generated by the following equation

$$\begin{aligned}
 L(t) &= k(t) \frac{[R_c(t)]^{p(t)}}{[R_w(t)]^{q(t)}} + \varepsilon(t) = e^{K(t)} \frac{[R_c(t)]^{p(t)}}{[R_w(t)]^{q(t)}} + \varepsilon(t) \\
 &= (2.7183)^{5.0614} \times \frac{[R_c(t)]^{0.1779}}{[R_w(t)]^{0.5421}} + \varepsilon(t),
 \end{aligned} \tag{7 - 5}$$

where $\varepsilon(t)$ is the white noise with mean zero and variance one. The simulated data are depicted in Figure 7-1.

Simulation procedure, results and discussion

For a fair comparison in terms of noise sensitivity between DF algorithm and EF algorithm, set the forgetting factor $\lambda_D = 0.80$ for DF algorithm, $\lambda_E = 0.98$ for EF algorithm. The initial parameter values are set as $\alpha(t) = 0$, $\beta(t) = 0$, $K(t) = 0$ for both algorithms. Both algorithms are started with $P(0) = 1000I$. Implement DF and EF using model (7-1) based on the simulated data shown in Figure 7-1, respectively. Both identification results by DF and EF are shown in Figure 7-2. The "true values" of the identified parameters are also depicted in Figure 7-2, graph (f), (g) and (h), respectively. To compare DF with EF, the trace of covariance matrix $P(t)$, and the

identified values of the parameters $\alpha(t)$, $\beta(t)$ and $K(t)$ are depicted in Figure 7-3 (and 7-4 in detail), 7-5, 7-6 and 7-7, respectively. Due to the large scale, one cannot see the trace of $P(t)$ -matrix connected with the DF algorithm. One can see it in figure 7-4 where the scale is smaller. In Figure 7-3, 7-4, 7-5, 7-6, and 7-7, in order to illustrate the identification results more clearly, only the simulation results corresponding to the first 900 data points are depicted.

From graphs (a) and (b) of Figure 7-2, one can see that, as analyzed in Chapter 4, during the first part of the simulated data, the input signals are not persistently exciting. They do not carry sufficient information about the system dynamics. Corresponding to such excitation conditions, graphs (f), (g), and (h) of Figure 7-2 illustrate that the identified values of the parameters by both EF and DF algorithms cannot converge to their "true values" during this period. Furthermore, Figure 7-3 and Figure 7-4 illustrate that estimator windup occurs with EF algorithm, whereas there is no estimator windup when using DF algorithm. In the second part of simulated data, the input signals $R_c(t)$ and $R_w(t)$ are square waves. When the input signals $R_c(t)$ and $R_w(t)$ are kept unchanged, there is no new information about the system dynamics. With such excitation conditions, the EF and DF algorithms present different properties. As analyzed in the previous chapter, for the EF algorithm, the old data contained in the information matrix $R(t-1)$ is discounted continuously regardless of whether the new data contains new information. As a result, the covariance matrix $P(t)$ and the estimator gain $P(t)\phi(t)$ grow continuously. Figures 7-3 and 7-4 illustrate that windup occurs twice for the EF algorithm, not only corresponding to the first part of simulated data set, but also corresponding to the second part of simulated data set. Whereas, for the DF algorithm, the old data is forgotten only in the direction where the new data is coming from. That is, when there is no new information in the data, the old information will not be forgotten, it is kept unchanged. Whenever the new information comes, the information matrix is updated with the new data, and the estimator identifies the values of the parameters based on the new information about the system dynamics. Figures 7-5, 7-6 and 7-7 illustrate that the identified values of the parameters by the DF algorithm gradually converge to their "true values". On the contrary, the same parameters' values identified by the EF algorithm are unreliable even bursting [14]. Based on the identified values of the parameters, the estimated motor loads with these two algorithms are shown in Figure 7-8.

Figure 7-1. Comparison between DF and EF, the simulated data set.

Scale of X-axis: second $\times$ 10.

Top: Simulated production rate $R_c(t)$.

Middle: Simulated infeed dilution water flow rate $R_w(t)$.

Bottom: Simulated motor load $L(t)$.

Figure 7-2. Comparison between DF and EF, identification results. Scale of X-axis: second $\times 10$.

(a) Simulated production rate $R_c(t)$. (b) Simulated infeed dilution water flow rate $R_w(t)$. (c) Simulated motor load $L(t)$ (solid) and the estimated motor load $L(t)$ by EF (dotted). (d) Simulated motor load $L(t)$ (solid) and the estimated motor load $L(t)$ by DF (dotted). (e) Trace of $P(t)$ -matrix. (f) "True value" of parameter $\alpha(t)$ (solid), on-line identified $\alpha(t)$ by DF (solid) and by EF (dashed). (g) "True value" of parameter $\beta(t)$ (solid), on-line identified $\beta(t)$ by DF (solid) and by EF (dashed). (h) "True value" of parameter $K(t)$ (solid), On-line identified $K(t)$ by DF (solid) and by EF (dashed).

Figure 7-3. Comparison between DF and EF in terms of windup phenomenon by presenting the trace of covariance matrix $P(t)$. Scale of X-axis: second $\times 10$.

One cannot see the trace of $P(t)$ -matrix with DF algorithm because of large scale. One can see it in Figure 7-4.

Figure 7-4. Comparison between DF and EF in terms of windup phenomenon by presenting the trace of covariance matrix $P(t)$. Trace of covariance matrix $P(t)$ in detail. Scale of X-axis: second $\times 10$.

Trace of covariance matrix $P(t)$ with EF algorithm (solid).

Trace of covariance matrix $P(t)$ with DF algorithm (dashed).

Figure 7-5. Comparison between DF and EF, On-line identified value of parameter $\alpha(t)$. Scale of X-axis: second $\times 10$.

True value of parameter $\alpha(t)$ (dashed).

On-line identified value of parameter $\alpha(t)$ by EF (dashdot).

On-line identified value of parameter $\alpha(t)$ by DF (solid).

Figure 7-6. Comparison between DF and EF, On-line identified value of parameter $\beta(t)$. Scale of X-axis: second $\times 10$.

True value of parameter $\beta(t)$ (dashed).

On-line identified value of parameter $\beta(t)$ by EF (dashdot).

On-line identified value of parameter $\beta(t)$ by DF (solid).

Figure 7-7. Comparison between DF and EF, On-line identified value of parameter $K(t)$. Scale of X-axis: second $\times 10$.

True value of parameter $K(t)$ (dashed).

On-line identified value of parameter $K(t)$ by EF (dashdot).

On-line identified value of parameter $K(t)$ by DF (solid).

Figure 7-8. Comparison between DF and EF, estimated motor loads.

Scale of X-axis: second $\times 10$.

Top: Simulated motor load (solid) and the estimated motor load by EF (dotted).

Bottom: Simulated motor load (solid) and the estimated motor load by DF (dotted).

7.3 Comparison between DF and EF in terms of ability of tracking sudden parameter changes

The purpose of this simulation is to compare the ability of the DF and EF algorithms to see if the identified parameters can track sudden parameter changes.

Model structure

The model structure is the same as that used in Section 7.2. Consequently, the linear regression structure, the regression vector and the parameter vector are the same too.

Data set structure and simulation

The whole-simulated data set consists of two parts: simulated data set 96 and simulated data set 97. The simulated data set 96 has different system parameters from the simulated data set 97 so that one can examine the ability of tracking sudden parameter changes for the EF and DF algorithms. The real data sets are depicted in Figure 7-9. They are connected by the following order: 1996 data set a, 1996 data set c, 1997 data set a, 1997 data set b, and 1997 data set c. Refer to the normal range of real data sets, the whole simulated data set is simulated as follows.

Simulated data set 96

In the simulated data set 96, the input signals, namely the production rate $R_c(t)$ and the infeed dilution water flow rate $R_w(t)$ are made up of two parts. The first part is the same as the first part of simulated data set used in Section 7.2. That is, it is formed by taking the production rate $R_c(t)$ and the infeed dilution water flow rate $R_w(t)$ from the real data sets, 1996 data set a and 1996 data set c. The number of data points in this part is $N = 400$. In the second part of simulated data set 96, the production rate $R_c(t)$ is a square wave with a period of 74 sampling points, the signal shifts between 93 rpm and 128 rpm, the infeed dilution water flow rate $R_w(t)$ is a square wave with a period of 38 sampling points, the signal shifts between 182 lpm and 222 lpm. The number of data points in this part is $N = 900$. Consequently, the number of data points in simulated data set 96 is $N = 1300$. Take the simulated $R_c(t)$ and $R_w(t)$ as input signals. Set the parameters values as $\alpha(t) = 0.1779$, $\beta(t) = 0.5421$, and $K(t) = 5.0614$, which was the off-line identified values of

these parameter based on the 1996 real data sets a, b, and c of the TMP process. Like in Section 7.2, in the simulated data set 96, the motor load $L(t)$ was generated by the following equation

$$\begin{aligned} L(t) &= k(t) \frac{[R_c(t)]^{\alpha(t)}}{[R_w(t)]^{\beta(t)}} + \varepsilon(t) = e^{K(t)} \frac{[R_c(t)]^{\alpha(t)}}{[R_w(t)]^{\beta(t)}} + \varepsilon(t) \\ &= (2.7183)^{5.0614} \times \frac{[R_c(t)]^{0.1779}}{[R_w(t)]^{0.5421}} + \varepsilon(t), \end{aligned} \quad (7 - 6)$$

where $\varepsilon(t)$ is the white noise sequence with mean zero and variance one.

Simulated data set 97

In the simulated data set 97, the production rate $R_c(t)$ and the infeed dilution water flow rate $R_w(t)$ are also made up of two parts. In the first part, $R_c(t)$ and $R_w(t)$ are constant. They are set as $R_c(t) = 93$ rpm, and $R_w(t) = 184$ lpm. The number of data points in this part is $N = 500$. In the second part of simulated data set 97, the production rate $R_c(t)$ is a square wave with a period of 56 sampling points, the signal shifts between 33 rpm and 90 rpm. The infeed dilution water flow rate $R_w(t)$ is a square wave with a period of 36 sampling points, the signal shifts between 153 lpm and 242 lpm. The number of data points in this part is $N = 1200$. Consequently, the number of data points in the simulated data set 97 is $N = 1700$. Take the simulated $R_c(t)$ and $R_w(t)$ as input signals. Set the parameters as $\alpha(t) = 0.8803$, $\beta(t) = 0.9638$, and $K(t) = 4.3542$, which were the off-line identified values of these parameters based on the 1997 real data sets a, b, and c of the TMP process. In simulated data set 97, the motor load $L(t)$ is generated by the following equation

$$\begin{aligned} L(t) &= k(t) \frac{[R_c(t)]^{\alpha(t)}}{[R_w(t)]^{\beta(t)}} + \varepsilon(t) = e^{K(t)} \frac{[R_c(t)]^{\alpha(t)}}{[R_w(t)]^{\beta(t)}} + \varepsilon(t) \\ &= (2.7183)^{4.3542} \times \frac{[R_c(t)]^{0.8803}}{[R_w(t)]^{0.9638}} + \varepsilon(t), \end{aligned} \quad (7 - 7)$$

where $\varepsilon(t)$ is the white noise sequence with mean zero and variance one. Combining simulated data set 96 and simulated data set 97 forms the whole-simulated data set. The first data point of simulated data set 97 follows the last data point of simulated data set 96. The total number of data points in the whole-simulated data set is $N = 3000$. The whole-simulated data set is depicted in Figure 7-10.

Comparison procedure

Set the forgetting factor as $\lambda_E = 0.98$ for EF algorithm, and $\lambda_D = 0.8$ for DF algorithm. Set the initial parameter values as $\alpha(t) = 0$, $\beta(t) = 0$, and $K(t) = 0$ for both DF and EF. Implement DF and EF algorithms using the model structure defined by equation (7-1) based on the whole-simulated data set. Both identification results by DF and EF are shown in Figure 7-11. To compare the DF algorithm with the EF algorithm, the traces of covariance $P(t)$ are shown in Figure 7-12 and 7-13, the identified values of parameters $\alpha(t)$, $\beta(t)$, and $K(t)$ by EF and DF are shown in Figures 7-14, 7-15, and 7-16, respectively.

Comparison results and discussion

As analyzed in Section 7.2, during the period of the first part of simulated data set 96, namely, the period between the data points 1 and 400 of the whole-simulated data set, the input signals are not persistently exciting. With such excitation conditions, the values of the parameters identified by both EF and DF algorithms cannot converge to their “true values”. Corresponding to this period, the graph (e) in Figure 7-11 shows that the first estimator windup occurs with EF algorithm. Whereas windup does not happen with the DF algorithm. One can see this in Figure 7-12 and Figure 7-13 more clearly. From Figures 7-14, 7-15 and 7-16 one can see that the estimate errors of the parameters identified by the EF algorithm are significantly large between the data points 1300 and 1800. During this period, the input signals $R_c(t)$ and $R_w(t)$ are kept constant. At the data point 1300, the “true values” of the system parameters jump suddenly. They jump from $\alpha(t) = 0.1779$, $\beta(t) = 0.5421$ and $K(t) = 5.0614$ to $\alpha(t) = 0.8803$, $\beta(t) = 0.9638$ and $K(t) = 4.3542$. There is not any new information about the system dynamics during this period, the second windup phenomenon occurs with the EF algorithm. The identified values of the parameters deviate from their “true values” significantly even bursting [14]. Under the same excitation conditions, the identified values of the parameters by the DF algorithm do not exhibit the bursting phenomenon. At the beginning, based on simulated data set 96, the identified values of the parameters by the DF algorithm converge to their “true values”, namely, $\alpha(t) = 0.1779$, $\beta(t) = 0.5421$, $K(t) = 5.0614$. When the “true values” jump to $\alpha(t) = 0.8803$, $\beta(t) = 0.9638$ and $K(t) = 4.3542$, the identified values of these parameters by the DF algorithm can track the sudden change and converge to the new “true values”. During the period when the input signals are kept

constant, the identified values of the parameters by DF have much less deviation from their “true values” compared with those values identified by the EF algorithm.

The simulation results presented in Section 7.2 and Section 7.3 demonstrate that the novel directional forgetting algorithm is a suitable choice for the on-line identification of thermo-mechanical pulping process.

Figure 7-9. Real data sets of the TMP process, collected on June 21, 1996 and June 9, 1997. Scale of X-axis: second $\times 10$.

Top: Real data of production rate $R_c(t)$ (rpm).

Middle: Real data of infeed dilution water flow rate $R_w(t)$ (lpm).

Bottom: Real data of motor load $L(t)$ (MW).

Figure 7-10. Whole-simulated data set. Scale of X-axis: second $\times 10$.

Top: Simulated production rate $R_c(t)$ (rpm).

Middle: Simulated infeed dilution water flow rate $R_w(t)$ (lpm).

Bottom: Simulated motor load $L(t)$ (MW).

Figure 7-11. Identification results based on the simulated data set

Scale of X-axis: second $\times 10$.

(a) Simulated production rate $R_c(t)$. (b) Simulated infeed dilution water flow rate $R_w(t)$. (c) Simulated motor load $L(t)$ (solid) and estimated $L(t)$ by DF (dotted). (d) Simulated motor load $L(t)$ (solid) and estimated $L(t)$ by EF (dotted). (e) Trace of covariance $P(t)$. (f) "true values" of parameter $\alpha(t)$ (dashed 1996 and solid 1997), on-line identified $\alpha(t)$ by DF (solid) and by EF (dashed). (g) "true values" of parameter $\beta(t)$ (dashed 1996 and solid 1997), on-line identified $\beta(t)$ by DF (solid) and by EF (dashed). (h) "true values" of parameter $K(t)$ (dashed 1996 and solid 1997), on-line identified $K(t)$ by DF (solid) and by EF (dashed).

Figure 7-12. Comparison between DF and EF in terms of windup phenomenon by presenting the trace of covariance matrix $P(t)$. Scale of X-axis: second $\times 10$.

One cannot see the trace of $P(t)$ -matrix with DF algorithm because of large scale. One can see it in Figure 7-13.

Figure 7-13. Comparison between DF and EF in terms of Estimator Windup phenomenon. Trace of covariance matrix $P(t)$ in detail.

Scale of X-axis: second $\times 10$.

Trace of covariance matrix $P(t)$ with EF algorithm (solid).

Trace of covariance matrix $P(t)$ with DF algorithm (dashed).

Figure 7-14. Comparison between DF and EF in terms of ability of tracking sudden parameter changes.

Identified value of parameter $\alpha(t)$.

Scale of X-axis: second $\times 10$.

Real value of 1996 $\alpha(t)$ (dashed, horizontal straight line).

Real value of 1997 $\alpha(t)$ (solid, horizontal straight line).

Identified value of $\alpha(t)$ by the EF algorithm(dashdot).

Identified value of $\alpha(t)$ by the DF algorithm(solid).

Figure 7-15. Comparison between DF and EF in terms of ability of tracking sudden parameter changes.

Identified value of parameter $\beta(t)$.

Scale of X-axis: second $\times 10$.

Real value of 1996 $\beta(t)$ (dashed, horizontal straight line).

Real value of 1997 $\beta(t)$ (solid, horizontal straight line).

Identified value of $\beta(t)$ by the EF algorithm (dashdot).

Identified value of $\beta(t)$ by the DF algorithm (solid).

Figure 7-16. Comparison between DF and EF in terms of ability of tracking sudden parameter changes.

identified value of parameter $K(t)$.

Scale of X-axis: second $\times 10$.

Real value of 1996 $K(t)$ (dashed, horizontal straight line).

Real value of 1997 $K(t)$ (solid, horizontal straight line).

Identified value of $K(t)$ by the EF algorithm (dashdot).

Identified value of $K(t)$ by the DF algorithm (solid).

Chapter 8

CONCLUSION

For the purpose of off-line identification of the two-stage thermo-mechanical pulping process, the fundamental theory of system identification is introduced. The least-squares algorithm and the instrumental-variable method are introduced and their properties are analyzed. The identification results of the TMP process by the off-line least-squares algorithm based on the real data collected from Avenor Paper Mill, Gatineau, Quebec are presented and analyzed.

For the purpose of on-line identification of the two-stage thermo-mechanical pulping process, the fundamental theory of recursive identification is introduced. The standard exponential forgetting algorithm is introduced and its properties, especially the phenomenon of estimator windup is presented and analyzed. Focusing on the on-line identification of TMP process, the novel directional forgetting algorithm is introduced, its theoretical properties are analyzed, and its practical properties are demonstrated by a simulation example. After presenting and analyzing the simulation results of comparison between the novel directional algorithm and the standard exponential forgetting algorithm, the thesis demonstrates that the new directional forgetting algorithm should be a suitable choice for the on-line identification for the two-stage thermo-mechanical pulping process.

References

- [1] R.A. Leask (ed), *Pulp and Paper Manufacture, Volume 2, Mechanical Pulping*, The Joint Textbook Committee of The Paper Industry, 1987, 287p.
- [2] A. Heikkurinen, J. Vaarasalo and A. Karnis, *Effect of Initial Defiberization on the Properties of Refiner Mechanical Pulp*, Journal of Pulp and Paper Science, Volume 19 No.3, May 1993, J119-124.
- [3] A. Karnis, *The mechanism of Fibre Development in Mechanical Pulping*, Journal of Pulp and Paper Science, Vol. 20 No. 10 October 1994.
- [4] K.B. Miles, W.D. May, *The Flow of Pulp in Chip Refiners*, Journal of Pulp and Paper Science, Volume 16 No.2, March 1990, J63-72.
- [5] B.C. strand, A. Mokvist, *On-line modeling of refiner performance*, Pulp and paper Canada, 90(12)(1989) T499-T504.
- [6] K.B. Miles, W.D. May, *Predicting the Performance of a Chip Refiner, A Constitutive Approach*, Journal of Pulp and Paper Science, Volume 19 No.6, November 1993, J268-274.
- [7] Tessier, P. and Qian, X., *Modeling and Simulation of a CTMP process*, 80 th Annual Meeting, Technical Section, CPPA, A143-A149(1993).
- [8] B. J. Allison, A. J. Isaksson and A. Karlstorm, *Grey-Box Identification of a TMP Refiner* Proc. Intl. Mech. Pulping Conf., Ottawa, Canada, 1995.
- [9] Gary Chang and H. M. Schwartz, *Modeling and Control of the TMP Process*, Control Systems '96, Halifax, Nova Scotia, Canada, p117-123.

- [10] H.M. Schwartz et al, *A Method of Modeling, Predicting and Controlling TMP Pulp Properties*, Proceedings of the 1996 IEEE International Conference on Control Applications, Dearborn MI, September 15-18, 1996, IEEE Control Systems Society, 1996, p846-851.
- [11] H.M. Schwartz, Liyu Cao and Trung Phung, *A Nonlinear Black Box Model for TMP Refiner*, Control Systems '98-Information tools to match the evolving operator role, September 1-3, 1998, Porvoo, Finland, p111-118.
- [12] R. Kulhavy and M. Karny, *Tracking of Slowly Varying Parameters by Directional Forgetting*. The 9th IFAC World Congress, Budapest, Hungary, pp.687-692, 1984.
- [13] T. Hagglund, *Recursive Estimation of Slowly Time-varying Parameters*. The 7th IFAC Symposium on Identification and System Parameter Estimation, York, U.K., pp.1137-1142, 1985.
- [14] B. D. O. Anderson, *Adaptive Systems, Lack of Persistency of Excitation and Bursting Phenomena*, Automatica, Vol.21, No. 3, pp. 247-258, 1985.
- [15] R. Kulhavy, *Restricted Exponential Forgetting in Real-time Identification*, Automatica, Vol.23, No.5, pp.589-600, 1987.
- [16] M.E. Salgado, G.C. Goodwin and R.H. Middleton, *Modified Least Squares Algorithm Incorporating Exponential Resetting and Forgetting*. International Journal of Control, vol. 47, No 2, 477-491. 1988.
- [17] S. Bittanti, P. Bolzern, and M. Campi, *Convergence and Exponential Convergence of Identification Algorithms with Directional Forgetting Factor*. Automatica, vol.26, No.5, pp929-932. 1990.
- [18] J.E. Parkum, N.K. Poulsen and J. Holst, *Recursive Forgetting Algorithms*, International Journal of Control, Vol.55, No.1, pp.109-128, 1992.
- [19] R. Kulhavy and M.B. Zarrop, *On a General Concept of Forgetting*, International Journal of Control, Vol.58, No.4, pp.905-924, 1993.

- [20] L. Cao and H.M. Schwartz, *A novel Recursive Algorithm for Directional Forgetting*, 1999 American Control Conference, San Diego, California, USA, pp.1934-1938.
- [21] L. Cao and H.M. Schwartz, *A Directional Forgetting Algorithm based on the Decomposition of Information Matrix*, The 7th Mediterranean Conference on Control and Automation, Haifa, Israel, 1999.
- [22] L. Cao and H.M. Schwartz, *A Directional Forgetting Algorithm based on the decomposition of Information Matrix*, to appear at Automatica.
- [23] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, The MIT Press, 1983, 529p.
- [24] L. Ljung, *System Identification: Theory for the User*, Prentice Hall, Englewood Cliffs, 1987, 519p.
- [25] T. Soderstrom, P. Stoica, *System Identification*, Prentice Hall International Ltd., Hemel Hempstead, 1989, 612p.
- [26] K.J. Astrom and B. Wittenmark, *Adaptive Control*, second edition, Addison-Wesley, 1995, 574p.
- [27] R.A. Horn and C.R. Johnson, *Matrix Analysis*. Cambridge University Press, Cambridge. 1985.
- [28] A. Horch, *Modeling and Control of a Thermomechanical Pulp Refiner*, Master's thesis, Dept of Signals, Sensors and Systems, Royal Institute of Technology, Sweden, 1995.
- [29] Gary Chang, *System Identification and Control of a Thermo-mechanical Pulping Refiner*, Master's thesis, Dept of Systems and Computer Engineering, Carleton University, Ottawa, Canada, 1996.
- [30] F. Stenson, *Automatic Control of Motor Load on a Wood Chip Refiner*, Master's thesis, Dept of Signals, Sensors and Systems, Royal Institute of Technology, Sweden, 1997.

Appendixes

A.2.1. Matlab program for example 2-1.

```
%chap2fig1

num=[0 1 1];num1=[1 0 0];den=[1 -1.85 0.855];
u=[zeros(1,1) ones(1,200)];k=1:201;
c=filter(num,den,u);
plot(k,c,'. '),axis([0 200 0 500]),hold on,grid,...
    xlabel('N'),ylabel('Unit-Step Response');

n1=301;u1=zeros(n1,1);
for i=1:n1/10;
    u1(10*(i-1)+1:10*(i-1)+5)=1;
end
u1=u1*2-1;
ca=filter(num,den,u1);
cal=filter(num1,den,randn(size(ca)));
cae=ca+cal;
z=[ca,u1];th=arx(z,[2 2 1]);present(th);
z1=[cae,u1];th1=arx(z1,[2 2 1]);present(th1);
num1=[0 th1(3,3) th1(3,4)]; den1=[1 th1(3,1) th1(3,2)];
c1=filter(num1,den1,u);
plot(k,c1,'--'),axis([0 200 0 500]),hold on,

n2=5001;
u2=zeros(n2,1);
for i=1:n2/10;
    u2(10*(i-1)+1:10*(i-1)+5)=1;
end
u2=u2*2-1;
cb=filter(num,den,u2);
cb2=filter(num1,den,randn(size(cb)));
cbe=cb+cb2;
z2=[cbe,u2];th2=arx(z2,[2 2 1]);present(th2);
num2=[0 th2(3,3) th2(3,4)]; den2=[1 th2(3,1) th2(3,2)];
c2=filter(num2,den2,u);
plot(k,c2),axis([0 200 0 500]),hold off,...
    legend('Model(2-32)','Model(2-38)','Model(2-39)'),
```

A.2.2. Matlab program for example 2-2.

```
%chap2fig2 IV4, input: square wave. Unit-step response
%data from 2nd to 2nd order system
num=[0 1 1];numa=[1 0 0];den=[1 -0.9 0.2];
u=[zeros(1,1) ones(1,100)];t=1:101;
Y=filter(num,den,u);
plot(t,Y,'o'),hold on,axis([0 30 0 12]),grid,xlabel('N'),...
    ylabel('Unit-Step Response'),...

%Step1 Estimate Q1 by LS Method
n1=101;u1=zeros(n1,1);
for i=1:n1/10;
    u1(10*(i-1)+1:10*(i-1)+5)=1;
end

u1=u1*2-1;
ca=filter(num,den,u1);
caa=filter(numa,den,rand(size(ca)));
y=ca+caa;
z=[ca,u1];th=arx(z,[2 2 1]);present(th);
z1=[y,u1];th1=arx(z1,[2 2 1]);present(th1);
th2=iv4(z1,[2 2 1]);present(th2);

%G1(q)=[th1(3,3)(q-1)+th1(3,4)(q-2)]/[1+th1(3,1)(q-1)+th1(3,2)(q-2)];
%getting x1(t) by input u1(t) to filter G1(q):x1=G1(q)u1(t);
num1=[0 th1(3,3) th1(3,4)];den1=[1 th1(3,1) th1(3,2)];
t=1:101;x1=filter(num1,den1,u1);X1=filter(num1,den1,u);
plot(t,X1,':'),hold on,

%Step2 generates the instruments
%g1=[-x1(t-1) -x1(t-2) u(t-1) u(t-2)]' innstrumental variable;
%x=[-y(t-1) -y(t-2) u(t-1) u(t-2)]'
%transpose of regreesion vector pha(t)

B1=[0 0 0 0;0 0 0 0;0 0 0 0;0 0 0 0];D1=[0;0;0;0];
for t=1:101
    if t<2, g1(1,1)=0;g1(2,1)=0;g1(3,1)=0;g1(4,1)=0;
        x(1,1)=0;x(2,1)=0;x(3,1)=0;x(4,1)=0;
    elseif (t>1)&(t<3),
        g1(1,1)=-x1(1);g1(2,1)=0;g1(3,1)=u1(1);g1(4,1)=0;
        x(1,1)=-y(1);x(2,1)=0;x(3,1)=u1(1);x(4,1)=0;
    else
        g1(1,1)=-x1(t-1);g1(2,1)=-x1(t-2);
        g1(3,1)=u1(t-1);g1(4,1)=u1(t-2);
        x(1,1)=-y(t-1);x(2,1)=-y(t-2);x(3,1)=u1(t-1);
        x(4,1)=u1(t-2);
    end
    A1=g1*x';B1=B1+A1;C1=g1.*y(t);D1=D1+C1;
end
B1=inv(B1);Q2=B1*D1

%G2=[Q2(3,1)(q-1)+Q2(4,1)(q-2)]/[1-Q2(1,1)(q-1)+Q2(1,2)(q-2)];
%getting x2(t) by input u1(t) to filter G2(q): x2=G2(q)u1(t);
```

```

num2=[0 Q2(3,1) Q2(4,1)];den2=[1 Q2(1,1) Q2(2,1)];t=1:101;
x2=filter(num2,den2,u1);X2=filter(num2,den2,u);
plot(t,X2,'-.'),hold on,

%step3 Let L(q)=1;x2(t)=G2(q)u1(t);already got.
%g2(t)=L(q)[-x2(t-1) -x2(t-2) u(t-1) u(t-2)]'
%x=[-y(t-1) -y(t-2) u(t-1) u(t-2)]'; also already got.
%xx=L(q)x(t);yy(t)=L(q)y(t);

B2=[0 0 0 0;0 0 0 0;0 0 0 0;0 0 0 0];D2=[0;0;0;0];
for k=1:101
    if k<2, g2(1,1)=0;g2(2,1)=0;g2(3,1)=0;g2(4,1)=0;
        x(1,1)=0;x(2,1)=0;x(3,1)=0;x(4,1)=0;
    elseif (k>1)&(k<3),
        g2(1,1)=-x2(1);g2(2,1)=0;g2(3,1)=u1(1);g2(4,1)=0;
        x(1,1)=-y(1);x(2,1)=0;x(3,1)=u1(1);x(4,1)=0;
    else
        g2(1,1)=-x2(k-1);g2(2,1)=-x2(k-2);
        g2(3,1)=u1(k-1);g2(4,1)=u1(k-2);
        x(1,1)=-y(k-1);x(2,1)=-y(k-2);
        x(3,1)=u1(k-1);x(4,1)=u1(k-2);
    end
    A2=g2*x';B2=B2+A2;C2=g2.*y(k);D2=D2+C2;
end
B2=inv(B2);Q3=B2*D2

num3=[0 Q3(3,1) Q3(4,1)];den3=[1 Q3(1,1) Q3(2,1)];k=1:101;
X3=filter(num,den,u);plot(k,X3),hold off,...
legend('original system','LS estimated model','1st VI estimated model','2nd
VI estimated model');

```

A.3.1. Off-line identification of the primary refiner based on 1996 data set c

```
% chap3fig6
L2jun21c; % run file L2jun21c to get data
pr=jun21c1(:,2);ml=jun21c1(:,4);u1=log(pr(1:100));
u2=ones(100,1);y=log(ml(1:100));u=[u1 u2];
z1=[y,u];th1=arx(z1,[0 1 1] [0 0]);
present(th1);
O1=u\y;
% estimated output Y1=x'*O
for j=1:100
    Y1(j)=2.7183^O1(2)*pr(j)^O1(1);
end

% calculate parameter vector O
u11=pr(1:100).^O1(1);m11=ml(1:100);z2=[m11,u11];
th2=arx(z2,[1,1,1]);present(th2);
[par,P,lam]=th2par(th2);

ini=filtic([0 par(2)],[1 par(1)],ml(1),u11(1));% initial conditions
Y2=filter([0 par(2)],[1 par(1)],u11,ini);

subplot(3,1,1),plot(pr),axis([25 65 110 122]),...
    title('(a)'),ylabel('Rc(t) (rpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,1,2),plot(ml),axis([25 65 19 20]),...
    hold on, plot(Y1,'.'),hold off, title('(b)'),ylabel('L(t) (MW)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,1,3),plot(ml),axis([25 65 19 20]),hold on,plot(Y2,'.'),...
    hold off, title('(c)'),xlabel('N (10s)'),ylabel('L(t) (MW)'),
```

A.3.2. Off-line identification of the primary refiner based on 1996 data set a

```
% chap3fig7 :  $L=k/D^a$ 

L2jun21a; % to run the file to get data

di=jun21a2(:,2);ml=jun21a1(:,4);
ml=ml(100:270);di=di(100:270);
n=size(di);u1=log(di);u2=ones(n,1);y=log(ml);
u=[-u1 u2];
z=[y,u];
th=arx(z,[0 [1 1] [0 0]]);
present(th);
x=u\y;

for j=1:n
    mle(j)=2.7183^x(2)/di(j)^x(1);
end

subplot(2,1,1),plot(di), ylabel('Rw(t) (lpm)');
subplot(2,1,2),plot(ml),hold on, plot(mle,':'); hold off,...
    xlabel('10s'); ylabel('L(t) (MW)');
```


A.3.3. Off-line identification of the primary refiner based on 1997 data set b

```
% chap3fig8 :  $L=k/D^a$ 

jun9b; % to run the file to get data

di=bbb(:,10);ml=bbb(:,4);
n=size(di);u1=log(di);u2=ones(n,1);y=log(ml);
u=[-u1 u2];
z=[y,u];
th=arx(z,[0,[1 1],[0 0]]);
present(th);x=u\y;

for j=1:n
    mle(j)=2.7183^x(2)/di(j)^x(1);
end

subplot(2,1,1),plot(di),axis([0 280 265 290]),ylabel('Rw(t) (lpm)'),
subplot(2,1,2);plot(ml);hold on,axis([0 280 17.5 21]),...
    plot(mle,':');hold off, xlabel('10s'), ylabel('L(t) (MW)'),
```

A.3.4. Off-line identification of the secondary refiner based on 1997 data set a

```
% chap3fig9 secondary refiner. use pr*c1 as input.

jun9a;pr=aaa(:,2);c1=aaa(:,20);ml=aaa(:,22);
n=size(pr);u11=pr.*c1;u1=log(u11);u2=ones(n,1);
u=[u1 u2];y=log(ml);z=[y,u];
th=arx(z,[0,[1 1],[0 0]]);
present(th);x=u\y;

for j=1:n
    mle(j)=2.7183^x(2)*u11(j)^x(1);
end

% to identify a dynamic model
u23=u11.^x(1);z1=[ml, u23];th1=arx(z1,[1,1,1]);present(th1);
[par,P,lam]=th2par(th1);%pause
ini=filtic([0 par(2)],[1 par(1)],ml(1),u23(1));% initial conditions
mle1=filter([0 par(2)],[1 par(1)],u23,ini);

subplot(4,1,1),plot(pr),title('(a)'),ylabel('Rc(t) (rpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,1,2),plot(c1),axis([0 300 56.5 62.5]),title('(b)'),...
    ylabel('Co(t) ( % )'),
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,1,3),plot(ml),hold on,plot(mle,':');hold off,...
    axis([0 300 12 17.5]),title('(c)'),ylabel('L(t) (MW)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,1,4),plot(ml),hold on,plot(mle1,':');hold off,...
    axis([0 300 12 17.5]),title('(d)'),xlabel('10s'),...
    ylabel('L(t) (MW)'),
```

A.3.5. Off-line identification of the secondary refiner based on 1997 data set c

```
% chap3fig10 secondary, use pr and cl as inputs.

jun9c;
pr=ccc(:,2);di=ccc(:,28);cl=ccc(:,20);ml=ccc(:,22);
n=size(pr);u11=pr.*cl;u1=log(u11);u2=log(di);u3=ones(n,1);
u=[u1 -u2 u3];y=log(ml);z=[y,u];th=arx(z,[0,[1 1 1],[0 0 0]]);
present(th);x=u\y;

for j=1:n
    mle(j)=2.7183^x(3)*u11(j)^x(1)/di(j)^x(2);
end

subplot(4,1,1);plot(pr);axis([0 250 97 101]),title('(a)'),...
    ylabel('R(t) (rpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,1,2);plot(di);axis([0 250 55 75]),title('(b)'),...
    ylabel('Rw(t) (lpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,1,3);plot(cl);axis([0 250 56 63]),title('(c)'),...
    ylabel('Co(t) ( % )'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,1,4);plot(ml);axis([0 250 11 17]),hold on,plot(mle,':'),...
    hold off, title('(d)'),xlabel('10s');ylabel('L(t) (MW)'),
```

A.4.1. Matlab program for example 4-1.

```
% chap4fig1

pdata96acCombined;

n=size(pr');u1=log(pr');u2=ones(n,1);y=log(ml');
r=0.98;OO=[0;0];I=[1 0;0 1];PP=1000*I;

for i=1:n
    VV(i)=PP(1,1)+PP(2,2);
    a(i)=0.1098;AA(i)=OO(1,1);
    k(i)=2.4523;KK(i)=OO(2,1);
    xx(1,1)=u1(i);xx(2,1)=1;
    gg=PP*xx;
    GG1(i)=gg(1,1);GG2(i)=gg(2,1);
    GG12(i)=GG1(i)+GG2(i);

    ww=r+xx'*PP*xx;
    LL=(PP*xx)/ww;
    PP=(PP-LL*xx'*PP)/r;
    yy1=xx'*OO; OO=OO+LL.*(y(i)-yy1);
    YY(i)=2.7183^yy1;
end

subplot(3,2,1),plot(pr),axis([0 400 110 125]),...
    title('(a) Rc(t)'),ylabel('rpm'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,3),plot(GG1),axis([0 400 -10 50]),title('(b) G1(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,5),plot(GG2),axis([0 400 -50 10]),title('(c) G2(t)'),...
    xlabel('lamda=0.98'),

subplot(3,2,2);plot(VV);axis([0 400 0 1e+6]),title('(d) P(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,4),plot(a,'--'),hold on,plot(AA),hold off,...
    axis([0 400 -1 2]),title('(e) Alpha(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,6),plot(k,'--'),hold on,plot(KK),hold off,...
    axis([0 400 -4 8]),title('(f) K(t)'),...
    xlabel('Initial Alpha=0, K=0'),
```

A.4.2. Simulated data set for example 4-1.

```
% pdata96acCombined  
L2jun21a;L2jun21c;  
  
pr1=jun21a1(:,2); pr3=jun21c1(:,2);  
pr(1:300)=pr1(1:300);pr(301:400)=pr3(1:100);  
  
for i=1:400  
    ml(i)=2.7183^2.4523*(pr(i)^0.1098)+randn;  
end  
  
subplot(2,1,1),plot(pr),axis([0 400 110 123]);  
subplot(2,1,2),plot(ml);
```

A.4.3. Matlab program for example 4-2.

```
% chap4fig2

pdata96acrandn;

n=size(pr');u1=log(pr');u2=-log(di');u3=ones(n,1);y=log(ml');
r=0.98;OO=[0;0;0];I=[1 0 0;0 1 0;0 0 1];PP=1000*I;

for i=1:n
    VV(i)=PP(1,1)+PP(2,2)+PP(3,3);
    a(i)=0.1779;AA(i)=OO(1,1);
    b(i)=0.5421;BB(i)=OO(2,1);
    k(i)=5.0614;KK(i)=OO(3,1);
    xx(1,1)=u1(i);xx(2,1)=u2(i);xx(3,1)=1;
    gg=PP*xx;
    GG1(i)=gg(1,1);GG2(i)=gg(2,1);GG3(i)=gg(3,1);
    GG12(i)=GG1(i)+GG2(i)+GG3(i);

    ww=r+xx'*PP*xx;
    LL=(PP*xx)/ww;
    PP=(PP-LL*xx'*PP)/r;
    yy1=xx'*OO; OO=OO+LL.*(y(i)-yy1);
    YY(i)=2.7183^yy1;
end

subplot(4,2,1),plot(pr),axis([0 400 30 160]),title('(a) Rc(t)'),...
    ylabel('rpm'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,3);plot(di),axis([0 400 100 320]),...
    title('(b) Rw(t)'),ylabel('lpm'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,5);plot(GG1),axis([0 400 -10 10]),title('(c) G1(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,7),plot(GG2),axis([0 400 -10 10]),title('(d) G2(t)'),...
    xlabel('lamda=0.98'),

subplot(4,2,2);plot(VV),axis([0 400 0 3000]),title('(e) P(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,4),plot(a),hold on,plot(AA),hold off,...
    axis([0 400 -1 2]),title('(f) Alpha(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,6),plot(b),hold on,plot(BB),hold off,...
    axis([0 400 -1 2]),title('(g) Beta(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,8),plot(k),hold on,plot(KK),hold off,...
    axis([0 400 -3 10]),title('(h) K(t)'),
    xlabel('Initial Alpha=0,Beta=0,K=0'),
```

A.4.4. Simulated data set for example 4-2.

```
% pdata96acrandn

L2jun21a;L2jun21c;

pr1=jun21a1(:,2); pr3=jun21c1(:,2);
pra(1:300)=pr1(1:300);pra(301:400)=pr3(1:100);

pr=randn(size(pra));pr=pr*20;pr=pr+100;
di=randn(size(pra));di=di*30;di=di+210;

for i=1:400
    ml(i)=2.7183^5.0614*(pr(i)^0.1779)/(di(i)^0.5421)+randn;
end

subplot(3,1,1),plot(pr),axis([0 400 30 160]);
subplot(3,1,2),plot(di),axis([0 400 100 320]);
subplot(3,1,3),plot(ml);
```

A.5.1. Matlab program for example 5-1.

```
% chap5fig1

pdata96acCombined;

n=size(pr');u1=log(pr');u2=ones(n,1);y=log(ml');

r=0.98;O=[0;0];I=[1 0;0 1];P=1000*I;R=inv(P);
for i=1:n
    V(i)=P(1,1)+P(2,2);
    a(i)=0.1098;A(i)=O(1,1);
    k(i)=2.4532;K(i)=O(2,1);
    x(1,1)=u1(i);x(2,1)=1;
    g=P*x;G1(i)=g(1,1);G2(i)=g(2,1);
    T=(1-r).*(x*x')/(x'*R*x);
    if sqrt(x'*x)>2, P1=P+T/r; M=R*T;
    else P1=P; M=0;
    end
    P=P1-(P1*x*x'*P1)/(1+x'*P1*x);
    y1=x'*O;z=y(i)-y1;O=O+P*x.*z;Z(i)=z;
    Y(i)=2.7183^y1;
    F=I-M;
    R=F*R+x*x';
end

subplot(3,2,1),plot(pr),axis([0 400 110 125]),title('(a)'),...
    ylabel('Rc(t) (rpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,3),plot(G1),axis([0 400 -10 50]),title('(b)'),...
    ylabel('G1(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,5),plot(G1),axis([0 400 -50 10]),title('(c)'),...
    ylabel('G2(t)'), xlabel('lamda=0.98'),

subplot(3,2,2),plot(V); axis([0 400 0 2000]),title('(d)'),...
    ylabel('P(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,4),plot(a,'--'),hold on, plot(A),hold off,...
    axis([0 400 -2 2]),title('(e)'), ylabel('Alpha(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,6),plot(k,'--'),hold on,plot(K),hold off,...
    axis([0 400 -4 8]),...
    title('(f)'),xlabel('Initial Alpha=0,Beta=0,K=0'),ylabel('K(t)'),
```


A.5.2. Simulated data set for example 5-1.

```
% pdata96acCombined  
  
L2jun21a;L2jun21c;  
  
pr1=jun21a1(:,2); pr3=jun21c1(:,2);  
pr(1:300)=pr1(1:300);pr(301:400)=pr3(1:100);  
  
for i=1:400  
    ml(i)=2.7183^2.4523*(pr(i)^0.1098)+randn;  
end  
  
subplot(2,1,1),plot(pr),axis([0 400 110 123]);  
subplot(2,1,2),plot(ml);
```

A.6.1. On-line identification for the primary refiner based on 1996 data set c. Static model.

```
% chap6fig1
L2jun21c;pr=jun21c1(:,2);ml=jun21c1(:,4);
u1=log(pr(1:100));u2=ones(100,1);y=log(ml(1:100));
u=[u1 u2];z=[y,u];th=arx(z,[0,[1 1],[0 0]]);
present(th);x1=u\y;
for j=1:100
    mle(j)=2.7183^x1(2)*pr(j)^x1(1);
end
r=0.98;O=[0.2847;1.6153];I=[1 0;0 1];P=1000*I;R=inv(P);n=size(u1);
for i=1:100
    V(i)=P(1,1)+P(2,2);
    a(i)=0.2847;A(i)=O(1,1);
    k(i)=1.6153;K(i)=O(2,1);
    x(1,1)=u1(i);x(2,1)=1;
    T=(1-r).*(x*x')/(x'*R*x);
    if sqrt(x'*x)>2, P1=P+T/r; M=R*T;
    else P1=P; M=0;
    end
    P=P1-(P1*x*x'*P1)/(1+x'*P1*x);
    y1=x'*O; O=O+P*x.*(y(i)-y1);
    Y(i)=2.7183^y1;
    F=I-M;
    R=F*R+x*x';
end
A(n),K(n),

subplot(3,2,1),plot(pr),axis([0 100 110 125]),title('(a)'),...
    ylabel('Rc(t) (rpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,3),plot(ml),hold on,plot(mle,':'),hold off,...
    axis([0 100 19 20]),title('(b)'),ylabel('L(t) (MW)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,5),plot(ml),hold on,plot(Y,':'),hold off,...
    axis([0 100 19 20]),title('(c)'),xlabel('lamda=0.8'),...
    ylabel('L(t) (MW)'),...

subplot(3,2,2),plot(V),axis([0 100 0 2500]),title('(d)'),...
    ylabel('P(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,4),plot(a,'--'),hold on,plot(A),hold off,...
    axis([0 100 0 0.7]),title('(e)'),ylabel('Alpha(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,6),plot(k,'--'),hold on,plot(K),hold off,...
    axis([0 100 0 2.5]),title('(f)'),
    xlabel('Initial Alpha=0, K=0'),ylabel('K(t)'),
```

A.6.2. On-line identification for the primary refiner based on 1996 data set c. Dynamic model.

```
% chap6fig3

L2jun21c;
d=0.2847;
pr=jun21c1(:,2);
pr1=pr.^d;
ml=jun21c1(:,4);y=ml;y2=y(1);

% calcualte parameter vector O
u11=pr1(1:100);ml1=ml(1:100);z2=[ml1,u11];
th2=arx(z2,[1,1,1]);present(th2);
[par,P,lam]=th2par(th2);
ini=filtic([0 par(2)],[1 par(1)],ml(1),u11(1));% initial conditions
Y2=filter([0 par(2)],[1 par(1)],u11,ini);

%recursive estimation
r=0.98;O=[0;0];I=[1 0;0 1];P=1000*I;R=inv(P);n=size(ml1);
for i=1:n
    V(i)=P(1,1)+P(2,2);
    a(i)=-0.6785;A(i)=O(1,1);
    b(i)=1.6176; B(i)=O(2,1);

    if i<2 x(1,1)=-ml(1);x(2,1)=pr1(1);
    else x(1,1)=-ml(i-1);x(2,1)=pr1(i-1);
    end

    T=(1-r).*(x*x')/(x'*R*x);

    if sqrt(x'*x)>2, P1=P+T/r; M=R*T;
    else P1=P; M=0;
    end

    P=P1-(P1*x*x'*P1)/(1+x'*P1*x);
    y1=x'*O; O=O+P*x.*(y(i)-y1);
    y2=-O(1,1).*y2+O(2,1).*x(2,1);Y(i)=y2;
    F=I-M;
    R=F*R+x*x';
end

A(n), B(n),

subplot(3,2,1),plot(pr),axis([0 100 110 125]),title('(a)'),...
    ylabel('Rc(t) (rpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,3);plot(ml),hold on,plot(Y2,':'),hold off,...
    axis([0 100 19 20]), title('(b)'),ylabel('L(t) (MW)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
```

```

subplot(3,2,5),plot(m1),hold on,plot(Y,':'),hold off,
    axis([0 100 19 20]),title('(c)'),xlabel('lamda=0.98'),...
    ylabel('L(t) (MW)'),...

subplot(3,2,2);plot(V);axis([0 100 0 2500]),title('(d)'),...
    ylabel('P(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,4),plot(a,'--'),hold on,plot(A),hold off,...
    axis([0 100 -1 0]),title('(e)'),ylabel('a(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,6),plot(b,'--'),hold on,plot(B),hold off,...
    axis([0 100 0 5]),title('(f)'),...
    xlabel('Initial a=-0.6785, b=1.6176'),ylabel('b(t)'),

```

A.6.3. On-line identification for the primary refiner based on 1996 data set a. Static model.

```
% chap6fig5 : L=k/D^a  
L2jun21a;di=jun21a2(:,2);ml=jun21a1(:,4);ml=ml(100:270);  
di=di(100:270);n=size(di);ul=log(di);u2=ones(n,1);y=log(ml);  
u=[-ul u2];z=[y,u];th=arx(z,[0 1 1] [0 0]);present(th);xl=u\y;  
for j=1:n  
    mle(j)=2.7183^xl(2)/di(j)^xl(1);  
end  
r=0.98;O=[0;0];I=[1 0;0 1];P=1000*I;R=inv(P);  
for i=1:n  
    V(i)=P(1,1)+P(2,2);  
    b(i)=0.4265;B(i)=O(1,1);  
    k(i)=5.2820;K(i)=O(2,1);  
    x(1,1)=-ul(i);x(2,1)=1;  
    T=(1-r).*(x*x')/(x'*R*x);  
    if sqrt(x'*x)>2, P1=P+T/r; M=R*T;  
    else P1=P; M=0;  
    end  
    P=P1-(P1*x*x'*P1)/(1+x'*P1*x);  
    y1=x'*O;O=O+P*x.*(y(i)-y1);  
    Y(i)=2.7183^y1;E(i)=Y(i)-ml(i);  
    F=I-M; R=F*R+x*x';  
end  
  
B(n),K(n),  
  
subplot(3,2,1),plot(di),axis([0 180 210 230]),title('(a)'),...  
    ylabel('Rw(t)(lpm)'),...  
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),  
subplot(3,2,3);plot(ml),hold on,plot(mle,':'),hold off,...  
    axis([0 180 19 20.5]),title('(b)'),ylabel('L(t)(MW)'),...  
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),  
subplot(3,2,5),plot(ml),hold on,plot(Y,':'),hold off,...  
    axis([0 180 19 20.5]),...  
    title('(c)'),xlabel('lamda=0.98'),ylabel('L(t)(MW)'),...  
  
subplot(3,2,2);plot(V);axis([0 180 0 2500]),title('(d)'),...  
    ylabel('P(t)'),...  
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),  
subplot(3,2,4),plot(b,'--'),hold on,plot(B),hold off,...  
    axis([0 180 -0.8 1.5]),title('(e)'),ylabel('Beta(t)'),...  
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),  
subplot(3,2,6),plot(k,'--'),hold on,plot(K),hold off,...  
    axis([0 180 -2 10]),...  
    title('(f)'),xlabel('Initial Beta=0, K=0'),ylabel('K(t)'),
```

A.6.4. On-line identification for the primary refiner based on 1997 data set b. Static model.

[illegible]

A.6.5. On-line identification for the secondary refiner based on 1997 data set a. Static model.

```
% chap6fig9 secondary refiner. use pr*c1 as input.
jun9a;pr=aaa(:,2);c1=aaa(:,20);ml=aaa(:,22);n=size(pr);u11=pr.*c1;
u1=log(u11);u2=ones(n,1);u=[u1 u2];y=log(ml);z=[y,u];
th=arx(z,[0,1 1],[0 0]);present(th);x1=u\y;
for j=1:n
    mle(j)=2.7183^x1(2)*u11(j)^x1(1);
end
r=0.98;O=[1.8986;-13.8161];I=[1 0;0 1];P=1000*I;R=inv(P);
for i=1:n
    V(i)=P(1,1)+P(2,2);
    a(i)=1.8986;A(i)=O(1,1);
    k(i)=-13.8161;K(i)=O(2,1);
    x(1,1)=u1(i);x(2,1)=1;
    T=(1-r).*(x*x')/(x'*R*x);
    if sqrt(x'*x)>2, P1=P+T/r; M=R*T;
    else P1=P; M=0;
    end
    P=P1-(P1*x*x'*P1)/(1+x'*P1*x);
    y1=x'*O;O=O+P*x.*(y(i)-y1);y1=x'*O;
    Y(i)=2.7183^y1;
    F=I-M; R=F*R+x*x';
end
A(n), K(n),

subplot(3,2,1),plot(pr),axis([0 280 95 105]),title('(a)'),...
    ylabel('Rc(t) (rpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,3),plot(c1),axis([0 280 56.5 62.5]),title('(b)'),...
    ylabel('Co(t) ( % )'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,5),plot(ml),hold on,plot(Y,':'),hold off,...
    axis([0 280 12 18]),...
    title('(c)'),xlabel('lamda=0.98'),ylabel('L(t) (MW)'),

subplot(3,2,2),plot(V),axis([0 280 0 2500]),title('(d)'),...
    ylabel('P(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,4),plot(a,'--'),hold on,plot(A),hold off,...
    axis([0 280 0 3]),title('(e)'),ylabel('Alpha(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,6),plot(k,'--'),hold on,plot(K),hold off,...
    axis([0 280 -20 1]),...
    title('(f)'),xlabel('Initial Alpha=0, K=0'),ylabel('K(t)'),
```

A.6.6. On-line identification for the secondary refiner based on 1997 data set a. Dynamic model.

```
% chap6fig11 secondary refiner. use pr*c1 as input.
jun9a;

pr=aaa(:,2);c1=aaa(:,20);ml=aaa(:,22);n=size(pr);u1l=pr.*c1;
u1=log(u1l);u2=ones(n,1);

% identification using static model to get alpha and motor load
% estimate.
u=[u1 u2];y=log(ml);z=[y,u];
th=arx(z,[0,[1 1],[0 0]]);
present(th);x1=u\y;

for j=1:n
    mle(j)=2.7183^x1(2)*u1l(j)^x1(1);
end

%identification using dynamic model
u23=u1l.^x1(1);z1=[ml,u23];th1=arx(z1,[1,1,1]);present(th1);
[par,P,lam]=th2par(th1);
ini=filtic([0 par(2)],[1 par(1)],ml(1),u23(1));% initial conditions
mle1=filter([0 par(2)],[1 par(1)],u23,ini);

%recursive estimation
r=0.98;O=[0;0];I=[1 0;0 1];P=1000*I;R=inv(P);y2=ml(1);

for i=1:n
    V(i)=P(1,1)+P(2,2);
    a(i)=-0.4451;A(i)=O(1,1);
    b(i)=0.5559e-6; B(i)=O(2,1);
    if i<2 x(1,1)=-ml(1);x(2,1)=u23(1);
    else x(1,1)=-ml(i-1);x(2,1)=u23(i-1);
    end
    T=(1-r).*(x*x')/(x'*R*x);
    if sqrt(x'*x)>2, P1=P+T/r; M=R*T;
    else P1=P; M=0;
    end
    P=P1-(P1*x*x'*P1)/(1+x'*P1*x);
    y1=x'*O; O=O+P*x.*(ml(i)-y1);
    y2=-O(1,1).*y2+O(2,1).*x(2,1);Y(i)=y2;
    F=I-M;
    R=F*R+x*x';
end

A(n),B(n),

subplot(3,2,1),plot(pr),axis([0 280 95 105]),title('(a)'),...
    ylabel('Rc(t) (rpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
```



```

subplot(3,2,3);plot(c1),axis([0 280 56.5 62.5]),title('(b)'),...
    ylabel('Co(t) ( % )'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,5),plot(m1),hold on,plot(Y,':'),hold off,...
    axis([0 280 12 18]),...
    title('(c)'),xlabel('lamda=0.98'),ylabel('L(t) (MW)'),...

subplot(3,2,2);plot(V);axis([0 280 0 2000]),title('(d)'),...
    ylabel('P(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,4),plot(a,'--'),hold on,plot(A),hold off,...
    axis([0 280 -1.5 0.5]),title('(e)'),ylabel('a(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(3,2,6),plot(b,'--'),hold on,plot(B),hold off,...
    axis([0 280 0 1.5e-6]),title('(f)'),...
    xlabel('Initial a=-0.4451, b=0.5559e-6'),ylabel('b(t)'),

```

A.6.7. On-line identification for the secondary refiner based on 1997 data set c. Static model.

```
% chap6fig13 secondary refiner. use pr*c1 as input.
jun9c;
pr=ccc(:,2);di=ccc(:,28);cl=ccc(:,20);ml=ccc(:,22);
n=size(pr);u11=pr.*cl;u1=log(u11);u2=log(di);u3=ones(n,1);
u=[u1 -u2 u3];y=log(ml);
z=[y,u];th=arx(z,[0,[1 1 1],[0 0 0]]);
present(th);x1=u\y;
for j=1:n
    mle(j)=2.7183^x1(3)*u11(j)^x1(1)/di(j)^x1(2);
end

%recursive estimation
r=0.98;O=[2.3701;0.4442;-16.1050];
I=[1 0 0;0 1 0;0 0 1];P=1000*I;R=inv(P);

for i=1:n
    V(i)=P(1,1)+P(2,2)+P(3,3);
    a(i)=2.3701;A(i)=O(1,1);
    b(i)=0.4442;B(i)=O(2,1);
    k(i)=-16.1050;K(i)=O(3,1);
    x(1,1)=u1(i);x(2,1)=-u2(i);x(3,1)=1;
    T=(1-r).*(x*x')/(x'*R*x);
    if sqrt(x'*x)>2, P1=P+T/r; M=R*T;
    else P1=P; M=0;
    end
    P=P1-(P1*x*x'*P1)/(1+x'*P1*x);
    y1=x'*O;
    O=O+P*x.*(y(i)-y1);y1=x'*O;
    y2=2.7183.^y1;Y(i)=y2;
    F=I-M;
    R=F*R+x*x';
end

A(n),B(n),K(n),

subplot(4,2,1),plot(pr),axis([0 250 97 101]),title('(a)'),...
    ylabel('Rc(t) (rpm)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,3);plot(cl),axis([0 250 56 63]),title('(b)'),...
    ylabel('Co(t) ( % )'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,5);plot(di),axis([0 250 57 73]),title('(c)'),...
    ylabel('Rw(t) ( lpm )'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,7),plot(ml),hold on,plot(Y,':'),hold off,...
    axis([0 250 11 17]),...
    title('(d)'),xlabel('lamda=0.98'),ylabel('L(t) (MW)'),...
```

```

subplot(4,2,2);plot(V);axis([0 250 0 3000]),title('(e)'),...
    ylabel('P(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,4),plot(a,'--'),hold on,plot(A),hold off,...
    axis([0 250 0 4]),title('(f)'),ylabel('Alpha(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,6),plot(b,'--'),hold on,plot(B),hold off,...
    axis([0 250 -1 1.5]),title('(g)'),ylabel('Beta(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,8),plot(k,'--'),hold on,plot(K),hold off,...
    axis([0 250 -25 5]),...
    title('(h)'),xlabel('Initial Alpha=0,Beta=0,K=0'),ylabel('K(t)'),

```

```
% pdata96acDFEFcomp

pdata96acCombineda;
n=size(pr96');u1=log(pr96');u2=log(di96');u3=ones(n,1);y=log(ml96');

r=0.8;rr=0.98;O=[0; 0; 0];OO=[0; 0; 0];I=[1 0 0;0 1 0;0 0 1];P=1000*I;
R=inv(P);PP=1000*I;

for i=1:n
    V(i)=P(1,1)+P(2,2)+P(3,3); VV(i)=PP(1,1)+PP(2,2)+PP(3,3);
    a(i)=0.1779;A(i)=O(1,1);AA(i)=OO(1,1);
    b(i)=0.5421;B(i)=O(2,1);BB(i)=OO(2,1);
    k(i)=5.0614;K(i)=O(3,1);KK(i)=OO(3,1);
    x(1,1)=u1(i); xx(1,1)=u1(i);
    x(2,1)=-u2(i);xx(2,1)=-u2(i);
    x(3,1)=1;      xx(3,1)=1;
    T=(1-r).*(x*x')/(x'*R*x);
    if sqrt(x'*x)>2, P1=P+T/r; M=R*T;
    else P1=P; M=0;
end
P=P1-(P1*x*x'*P1)/(1+x'*P1*x);
y1=x'*O; O=O+P*x.*(y(i)-y1);
Y(i)=2.7183^y1;
F=I-M;
R=F*R+x*x';

ww=rr+xx'*PP*xx;
LL=(PP*xx)/ww;
PP=(PP-LL*xx'*PP)/rr;
yy1=xx'*OO; OO=OO+LL.*(y(i)-yy1);
YY(i)=2.7183^yy1;

end

subplot(4,2,1),plot(pr96),axis([0 1300 80 140]),...
title('(a) Rc(t)'),ylabel('rpm'),...
set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,3),plot(di96),axis([0 1300 170 240]),...
title('(b) Rw(t)'),ylabel('lpm'),...
set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,5),plot(ml96),hold on,plot(Y,':'),hold off,...
axis([0 1300 15 30]),...
title('(c) L(t)'),ylabel('MW'),...
set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,7),plot(ml96),hold on,plot(YY,':'),hold off,...
axis([0 1300 15 30]),...
title('(d) L(t)'),xlabel('lamda1=0.8,lamda2=0.98'),ylabel('MW'),...
```

[illegible]

A.7.2. Simulated data set for A.7.1.

```
% pdata96acCombineda

L2jun21a;L2jun21c;squarewavea;

pr1=jun21a1(:,2); pr3=jun21c1(:,2);
pr96(1:300)=pr1(1:300);pr96(301:400)=pr3(1:100);
pr96(401:1300)=u1(1:900);

di1=jun21a2(:,2); di3=jun21c2(:,2);
di96(1:300)=di1(1:300);di96(301:400)=di3(1:100);
di96(401:1300)=u2(1:900);

for i=1:1300
    ml96(i)=(2.7183^5.0614)*(pr96(i)^0.1779)/(di96(i)^0.5421)+randn;
end

subplot(3,1,1),plot(pr96),axis([0 1400 80 125]),ylabel('Rc(t) (rpm)');
subplot(3,1,2),plot(di96),axis([0 1400 170 240]),ylabel('Rw(t) (lpm)');
subplot(3,1,3),plot(ml96),axis([0 1400 15 25]),xlabel('N'),...
    ylabel('L(t) (MW)');

% squarewavea
N=900;
u1=zeros(N,1);
u2=zeros(N,1);
for i=1:N/218;
    u1(218*(i-1)+1:218*(i-1)+109)=1;
end
for j=1:N/222;
    u2(222*(j-1)+1:222*(j-1)+111)=1;
end

u1=u1*35+84.888;
u2=u2*40+182;

subplot(2,1,1),plot(u1),axis([0 1400 80 125]),
subplot(2,1,2),plot(u2),axis([0 1400 170 240]),
```

153

```
subplot(4,2,7);plot(ml),hold on,plot(Y,'. '),hold off,...
    axis([0 3000 0 100]),title('(d) L(t)'),...
    xlabel('lamda1=0.8, lamda2=0.98'),ylabel('MW'),...

subplot(4,2,2);plot(V);hold on,plot(VV,'--'),hold off,...
    axis([0 3000 0 1e+006]),title('(e) P(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,4);plot(a96,'--'),hold on,plot(a97),hold on,...
    plot(A),hold on,plot(AA,'-. '),hold off,...
    axis([0 3000 -1 2.5]),title('(f) Alpha(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,6);plot(b96,'--'),hold on,plot(b97),hold on,plot(B),...
    hold on,plot(BB,'-. '),hold off,axis([0 3000 -1.5 1.5]),...
    title('(g) Beta(t)'),...
    set(gca,'XTicklabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}),
subplot(4,2,8);plot(k96,'--'),hold on,plot(k97),hold on,plot(K),...
    hold on,plot(KK,'-. '),hold off,axis([0 3000 -5 12]),...
    title('(h) K(t)'),xlabel('Initial Alpha=0,Beta=0,K=0'),
```


A.7.4. Simulated data set for A.7.3

```
%pdata9697combined
pdata96acCombined;pdata97abcCombined;
pr(1:1300)=pr96(1:1300);pr(1301:2992)=pr97(1:1692);
di(1:1300)=di96(1:1300);di(1301:2992)=di97(1:1692);
ml(1:1300)=ml96(1:1300);ml(1301:2992)=ml97(1:1692);

subplot(3,1,1),plot(pr),ylabel('Rc(t) (rpm)');
subplot(3,1,2),plot(di),ylabel('Rw(t) (lpm)');
subplot(3,1,3),plot(ml),xlabel('N (10s)'),ylabel('L(t) (MW)');

% pdata96acCombined
L2jun21a;L2jun21c;squarewave;

pr1=jun21a1(:,2); pr3=jun21c1(:,2);
pr96(1:300)=pr1(1:300);pr96(301:400)=pr3(1:100);
pr96(401:1300)=u1(1:900);

di1=jun21a2(:,2); di3=jun21c2(:,2);
di96(1:300)=di1(1:300);di96(301:400)=di3(1:100);
di96(401:1300)=u2(1:900);

for i=1:1300
    ml96(i)=(2.7183^5.0614)*(pr96(i)^0.1779)/(di96(i)^0.5421)+randn;
end

subplot(3,1,1),plot(pr96),%axis([0 1300 100 125]);
subplot(3,1,2),plot(di96),%axis([0 1300 200 230]);
subplot(3,1,3),plot(ml96);%axis([0 1300 15 25]);

% squarewave
N=900;
u1=zeros(N,1);
u2=zeros(N,1);
for i=1:N/74;
    u1(74*(i-1)+1:74*(i-1)+37)=1;
end
for j=1:N/38;
    u2(38*(j-1)+1:38*(j-1)+19)=1;
end

u1=u1*35+93;
u2=u2*40+182;

subplot(2,1,1),plot(u1),%axis([0 900 0 130]),
subplot(2,1,2),plot(u2),%axis([0 900 0 230]),

%pdata97abcCombined
squarewave97;
pr97(1:500)=ones(1,500);
pr97=pr97*93;
pr97(501:1692)=u1(1:1192);
```

```

di97(1:500)=ones(1,500);
di97=di97*184;
di97(501:1692)=u2(1:1192);

for j=1:1692
    ml97(j)=(2.7183^4.3542)*(pr97(j)^0.8803)/(di97(j)^0.9638)+randn;
end

subplot(3,1,1),plot(pr97);
subplot(3,1,2),plot(di97);
subplot(3,1,3),plot(ml97);

% squarewave97
N=1692;
u1=zeros(N,1);
u2=zeros(N,1);
for i=1:N/56;
    u1(56*(i-1)+1:56*(i-1)+28)=1;
end
for j=1:N/36;
    u2(36*(j-1)+1:36*(j-1)+18)=1;
end

u1=u1*57+33;
u2=u2*89+153;

subplot(2,1,1),plot(u1),
subplot(2,1,2),plot(u2),

```