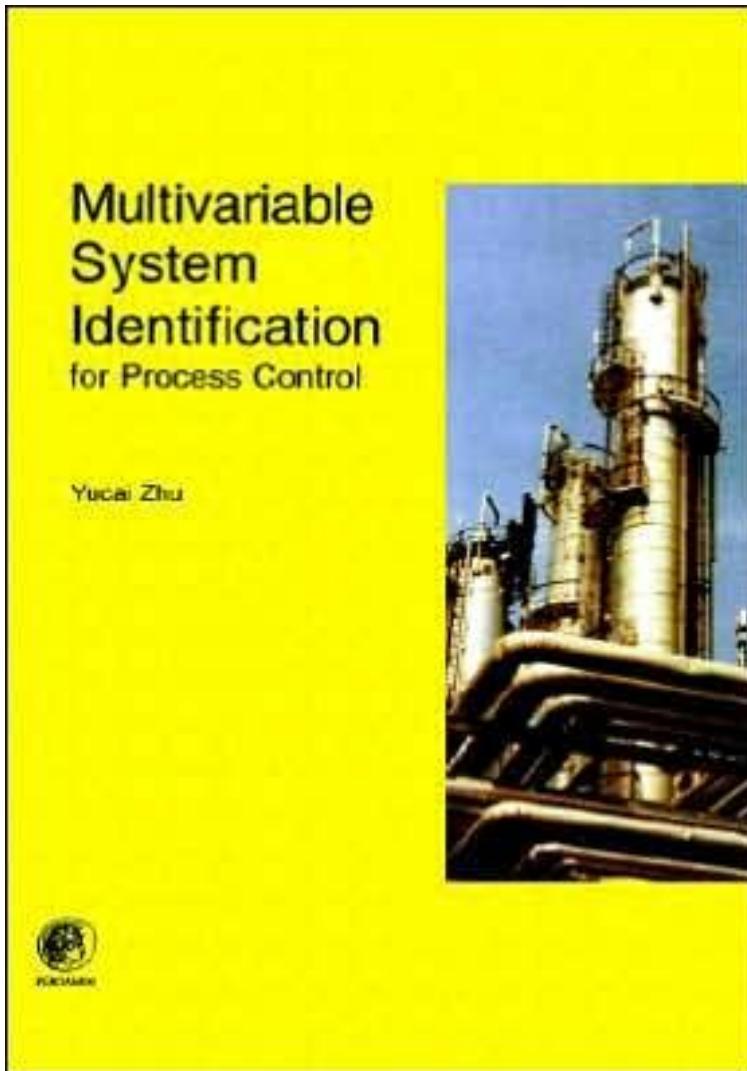


Multivariable System Identification for Process Control

by [Y. Zhu](#) (Editor)



- ISBN: 0080439853
- Pub. Date: October 2001
- Publisher: Elsevier Science & Technology Books

Foreword

Over the past decades it was highly interesting, extensive activity to watch and interact with the global scientific/engineering development of the 'new field' of *System Identification/Parameter and State Estimation/Model Predictive Control*. Along that road a clear landmark was the First IFAC (International Federation of Automatic Control) Symposium on this topic in Prague, Czechoslovakia, organized in 1967 by Professor Strejc and his coworkers under difficult circumstances. Since that time thousands of papers have been published and talks presented...

Over the past decades it was also a highly interesting, extensive activity to watch and interact with the scientific/engineering development of Dr. Yucai Zhu, the author of this book, and his talents... Much knowledge, insight and experience was derived, digested and applied in this process!

Among the problems to be solved were: the choice of the types of models that were both useful for application at hand as well as numerically tractable; the perennial struggle with a limited amount of information available; the strict limitations of additional disturbances allowed in industrial plants, hence the need for an optimal test signal design. Dr. Zhu has played an outstanding role by recognizing and focusing on the real needs in industrial environments.

This advancement has interacted with the developing of the capabilities of the computers, the efficiency of algorithms and the evolution of program packages. With his previous book, Zhu and Backx (1993), Dr. Zhu has testified already that he presents complex material in a clear and well-balanced way. Apart from the simulations and industrial examples with Matlab(R), presented in this book, it is my advice to the reader to pay serious attention to the program package on Identification that was developed by Dr. Zhu and made commercially available. This package has successfully been used in numerous industrial applications already.

Generally speaking books are not cheap. But actually they become only really expensive when the reader takes the time and effort to digest the material presented in them. Yet, for professional engineers and scientists having responsibilities or plans in this area, I do not know of any better investment than to digest, to study this book. In my opinion, it is the best way to cross the bridge from academic developments to real industrial applications...

Pieter Eijkhoff
Eindhoven, 2000

Preface

Systems and control theory has experienced a continuous development in the last few decades. The state space approach and Kalman filter are the products of the 1960s which, for the first time, made it possible to solve general linear multivariable control problems. Since 1970 adaptive control theory and techniques have been developed. In the 1980s robust control and H-infinity control of multivariable systems were developed. Fault detection and diagnosis techniques were also developed in this period. These new techniques are very promising for industrial applications and have attracted much interest from the academic researchers. The impact of these developments on process industries, however, has been very limited. When we visit plants in process industries, we find that a typical modern computer control system is a combination of the state-of-the-art computer technology and classical PID (proportional, integral and differential) control algorithms which are the restrictive single variable control techniques of the 1940s and 1950s.

Many possible reasons for this failure of technology transfer can be identified. One important reason is the lack of accurate dynamic models of industrial processes, since all the above mentioned modern techniques are model-based and need reasonably accurate process models. Another reason is the lack of good communication between the modern control community and process industries.

However, one process industry has made a distinction. In the last decade, model predictive control (MPC) technology has gained its industrial position in the refinery and petrochemical industry, and has started to attract interest from other process industries. There are over 500 control engineers from contracting and operation companies several thousands of applications have been reported. Most often, an MPC controller uses a linear dynamic model of the process that is obtained by way of black-box identification. However, due to various reasons, the cost of current MPC identification is very high and many trials and errors have to be made by the user. The test time is rather long (from several weeks to several months) and the tests are carried out manually around the clock. This, on the one hand, demands very high commitment of engineers and operators and, on the other hand, makes MPC project planning difficult. It is believed nowadays that process modeling and identification is the most difficult and time consuming part of an MPC project. Wide spread applications of MPC technology call for more effective and efficient identification technology.

Process identification is the field of mathematical modeling using test data. This branch of

automatic control has been very actively developed in the last three decades, with many books published on the topic. Most of these books have very high academic quality. However, they are too theoretically oriented for industrial users and for undergraduate students. Therefore, the purpose of this book is to fill the gap between theory and application and to provide industrial solutions to process identification that are based on sound scientific theory. We will study various identification methods; both linear and block-oriented nonlinear models will be treated. We will present, in detail, project procedures for multivariable process identification for control. Identification test design and model validation will be emphasized. The book is organized in a way that is reader friendly and easy to use for engineers and students. We will start with the simplest method, and then gradually introduce other methods. In this way, one can bring more physical insight to the reader and some mathematics can be avoided. Each method is treated in a single chapter or section, and experiment design is explained before discussing any identification algorithms. The many simulation examples and actual industrial case studies will show the power and efficiency of process identification and will make the theory more applicable. Matlab M files are included that will help the reader to learn identification in a computing environment.

Preliminary knowledge needed for reading the book consists of an introduction to linear systems theory and digital control (see, e.g., Åström and Wittenmark, 1984), and some knowledge about matrix operations, probability and random variables; see Appendixes B, C, and D of Eykhoff (1974) and Appendixes A and B of Söderström and Stoica (1989).

Examples and Matlab Files

Matlab M files for the exercises and examples and data files used in the book can be downloaded from the website of the author:

<http://www.er.ele.tue.nl/pages/people/index2.html>

For using these M files, Matlab System Identification Toolbox is needed.

There is a Matlab Toolbox called SMI Toolbox related to the subspace model identification studied in Chapter 8 that is developed by the group of Professor Michel Verhaegen. The SMI Toolbox can be downloaded from the website of his group:

<http://www.sce.tn.utwente.nl/index2.html>

Acknowledgments

The theory part of the book was obtained during my PhD research at the Measurement and Control Group, Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands, under the supervision of the late Professor Pieter Eykhoff. I thank him for his guidance and support, especially for his encouragement to pursue application orientated research

in process identification. Besides, he was so kind as to provide a review of the book and to write the Foreword. I would like to thank Professor Paul van den Bosch for offering me my current position at Eindhoven University of Technology so that I can concentrate again on research and writing. In my research I make extensive use of the asymptotic theory developed by Professor Lennart Ljung. I thank him for providing timely information about the results of his group and for the many discussions we have had. I am very grateful to Professor Michel Verhaegen for writing the chapter on subspace method and to Dr. Qinghua Zhang for writing the chapter on fault diagnosis. Their contribution has much added value to the book.

The book would have much less value without the application part. My initial contact with the industry was made possible by Professor Ton Backx. I thank him for his help and encouragement in my research and in my initial industrial career. The asymptotic method (ASYM) studied in this book was first applied in an MPC project of Setpoint Inc. I thank Mr. Mark Darby, my former Setpoint colleague, Mr. Lars H. Veland at Statoil Mongstad, for their openness to new ideas and their help. I would like to thank Nerefco B.V. and Mr. Ton Graaf for permission to publish the Crude Unit application. My thanks also to Dow Benelux N.V. and Mr. Firmin Butoyi for permission to publish the Deethanizer application.

My graduate study in Eindhoven was made possible by a scholarship from Xi'an Jiaotong University, China and the cooperation between the two universities. I am very grateful for this opportunity and I hope that, in the near future, the technology I have gathered and developed can be used by the Chinese process industries.

I also wish to express my gratitude to the late Mr. S. Liang Tan and his family for their warmth and friendship. I thank Mrs. Barbara Cornelissen-Milner for helping me to improve the language of the book. However, any remaining language errors are the responsibility of the author. Last but not least, I thank my family, Xiaohong, Xia and Ming for helping me to keep the balance between identification and life. Xiaohong retyped the first book into Latex files which were the basis of the current book.

Yucai Zhu
Eindhoven, 2001

Symbols and Abbreviations

General Mathematical Symbols

\otimes	the Kronecker product
\rightarrow	tends to
$\lim_{n \rightarrow \infty}$	the limit as n tends to infinity
$\liminf_{\delta \rightarrow 0}$	the limit as δ tends to zero
\in	is an element of
\subset	is the subset of
R^d	d dimensional real vectors
$\operatorname{argmin}_{\theta \in D} V(\theta)$	value of $\theta \in D$ that minimizes the loss function $V(\theta)$
$x \in \mathcal{N}(m, P)$	random variable x is normally distributed with mean m and covariance P
$E(x)$	mathematical expectation of the random variable x
$\operatorname{var}(x)$	variance of the random variable x
$\operatorname{cov}(x)$	covariance of the random vector x

Matrices

A_{ij}	the (i, j) element of the matrix A
A^T	the transpose of the matrix A
A^*	the conjugate transpose of the matrix A
A^{-1}	the inverse of the matrix A
A^{-T}	the inverse of the transpose of the matrix A
$\lambda(A)$	the eigenvalue of the matrix A
$\sigma(A)$	the singular value of the matrix A
$\det(A)$	the determinant of the matrix A
I, I_p	identity matrix, identity matrix with dimension $p \times p$
$\operatorname{col}(A)$	column operator stacking the columns of the matrix A on top of each other

Signals and processes models

$u(t)$	m - dimensional input vector at time t
$y(t)$	p - dimensional output vector at time t
$v(t)$	p - dimensional output disturbance vector at time t
$e(t)$	p - dimensional white noise vector

Z^N	input-output data sequence with sample number N
$R_u(\tau)$	autocorrelation function matrix of $u(t)$
$\Phi_u(\omega)$	auto-spectrum matrix of $u(t)$
$\{G_k^o\}_{k=0,1,\dots\infty}$	true impulse response of the process
$\{G_k\}_{k=0,1,\dots\infty}$	impulse response of the process model
q^{-1}	delay operator
$G^o(q)$	true transfer function matrix of the process
$G(q), \hat{G}(q)$	the model and the estimate of the process transfer function matrix
$G(q, \theta)$	the model of the process transfer function matrix parametrized in a structure with parameter vector θ
$\theta, \hat{\theta}, \theta^o$	parameter vector, its estimate and the true value
$G^o(e^{i\omega})$	the true frequency responses matrix of the process
$G(e^{i\omega}), \hat{G}(e^{i\omega})$	the model and its estimate of the process frequency responses matrix
$\Delta(e^{i\omega})$	modeling errors of the frequency response estimates
$\bar{\Delta}(e^{i\omega})$	upper bound matrix of the modeling errors

Abbreviations

w.p. 99.9%	with probability 99.9%
ARMA	Auto Regressive Moving Average
ARMAX	Auto Regressive Moving Average with eXogeneous
ARX	Auto Regressive with eXogeneous
CV	Controlled Variable (or process output)
FIR	Finite Impulse Response
FOE	Final Output Error
FPE	Final Prediction Error
IMC	Internal Model Control
IV	Instrumental Variable
LS	Least Squares
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
MV	Manipulated Variable (or process input)
OE	Output Error
SISO	Single-Input Single-Output

Table of Contents

Foreword		
Preface		
Symbols and Abbreviations		
1	Introduction	1
1.1	What is Process Identification?	1
1.2	The Hierarchy of Modern Automation Systems	4
1.3	Multivariable Model-Based Process Control	6
2	Models of Dynamic Processes and Signals	15
2.1	SISO Continuous-Time Models	15
2.2	SISO Discrete-Time Models	17
2.3	MIMO Models	21
2.4	Models of Signals	23
2.5	Linear Processes with Disturbances	26
2.6	Nonlinear Models	26
3	Identification Test Design and Data Pretreatment	31
3.1	Controller Configuration; Selection of MV's, CV's and DV's	32
3.2	Preliminary Process Tests	37
3.3	Test Signals for Final Test, Persistent Excitation	40
3.4	Test for Model Identification, Final Test	53

3.5	Sampling Frequency and Anti-Aliasing Filter	56
3.6	Pre-Treatment of Data	59
3.7	When is Excitation Allowed? Concluding Remarks	62
4	Identification by the Least-Squares Method	65
4.1	The Principle of Least-Squares	65
4.2	Estimating Models of Linear Processes	67
4.3	Industrial Case Studies	82
4.4	Properties of the Least-Squares Estimator	88
5	Extensions of the Least-Squares Method	97
5.1	Modifying the Frequency Weighting by Prefiltering	97
5.2	Output Error Method	102
5.3	Instrumental Variable (IV) Methods	110
5.4	Prediction Error Methods	113
5.5	User's Choice in Identification for Control	125
5.6	More on Order/Structure Selection	135
5.7	Model Validation	139
5.8	Identifying the Glass Tube Drawing Process	142
5.9	Recursive Parameter Estimation	151
6	Asymptotic Method; SISO Case	161
6.1	The Asymptotic Theory	162
6.2	Optimal Test Signal Spectrum for Control	167

6.3	Parameter Estimation and Order Selection	173
6.4	Model Validation using Upper Error Bound	177
6.5	Simulation Studies and Conclusion	178
7	Asymptotic Method; MIMO Case	183
7.1	MIMO Version of the Asymptotic Theory	183
7.2	Asymptotic Method	186
7.3	Identification of the Glass Tube Drawing Processes	191
8	Subspace Model Identification of MIMO Processes	199
8.1	Introduction	199
8.2	Definition of the State Space Identification Problem	200
8.3	Definition of the Data Equation	201
8.4	Analysis of Step Response Measurements	203
8.5	Subspace Identification using Generic Input Signals	208
8.6	Treatment of Additive Perturbations	211
8.7	A Simulation Study	214
8.8	Summary of Extensions	216
9	Nonlinear Process Identification	217
9.1	Identification of Hammerstein Models	218
9.2	Identification of Wiener Models	231
9.3	Identification of NLN Hammerstein-Wiener Model	241
9.4	Conclusions and Recommendations	249

10	Applications of Identification in Process Control	251
10.1	A Project Approach to Advanced Process Control	251
10.2	Identification Requirements for Process Control	256
10.3	PID Autotuning using Process Identification	258
10.4	Identification of Ill-Conditioned Processes	262
10.5	Identification of a Crude Unit for MPC	277
10.6	Closed-Loop Identification of a Deethanizer	284
10.7	Conclusions and Perspectives	290
11	Model Based Fault Detection and Isolation	293
11.1	Introduction	293
11.2	Residuals for Linear Systems with Additive Faults	295
11.3	Residuals for Non Additive Faults in Nonlinear Systems	304
11.4	Residual Evaluation	309
11.5	Industrial Applications	317
A	Refresher on Matrix Theory	329
	Bibliography	335
	Index	345

Chapter 1

Introduction

1.1 What is Process Identification?

System. A system is a collection of objects arranged in an ordered form to serve some purpose. Everything not belonging to the *system* is part of the *environment*. One may characterize the system by input-output (cause and effect) relations. What constitutes a *system* depends on the point of view of the observer. The *system* may be, for example, an amplifier consisting of electronic components, or a control loop including that amplifier as one of its parts, or a chemical processing unit having many such loops, or a plant consisting of a number of units or a number of plants operating together as a system in the environment of a global economy.

Process. A *process* is a processing plant that serves to manufacture homogeneous material or energy products. Industries that use such processing plants are called *process industries*. The common process industries are oil, chemicals, electrical power, paper, glass, mining, metals, cement, drugs, food and beverages. A common characteristic of process industries is that their products can be made to flow.

From a control point of view, different kinds of variables in a process interact and produce observable variables. The observable variables of interest to us are usually called *outputs*. The process is also affected by external variables. External variables that can be manipulated by us are *inputs* of the process. Other external variables are called *disturbances*. Disturbances can be divided into two kinds: the measured disturbances which can be directly measured, and the unmeasured disturbances which are only observed through their influence on the outputs. A process is said to be *dynamic* when the current output value depends not only on the current external stimuli but also on their earlier values. Outputs of dynamic processes whose external variables are not observed are often called *time series* (Ljung, 1987).

Model. A model is a representation of the essential aspects of a system (process) which presents knowledge of that system (process) in a usable form (Eykhoff, 1974).

There are many types of models. People are most familiar with *mental models* which do not involve any mathematical formalization. To drive a car, for example, the driver has a mental model about the relationship between the turning of the steering wheel and the turning of the car direction, and between the accelerator and the acceleration of the car. For manual control of an industrial process, the process operator needs the knowledge about how the process outputs will respond to various control actions. Sometimes it is appropriate to describe the properties of a system by means of tables or plots. Such descriptions are called graphical models. Bode plots, step responses and impulse responses of linear systems are of this type.

For the application of modern systems and control theory, it is necessary to use *mathematical models* that describe the relationships among the system variables in terms of difference or differential equations. In fact, the use of mathematical models is not limited to the control community; a major part of the engineering field deals with the use of mathematical models for simulations, forecasting and designs. The topic of this book — system identification — is about how to obtain mathematical models of systems (processes) from observations and measurements.

System Identification. System or process identification is the field of mathematical modeling of systems (processes) from test or experimental data. In technical terms, *system identification* is defined by Zadeh (1962) as: *the determination on the basis of input and output, of a system (model) within a specified class systems (models), to which the system under test is equivalent (in terms of a criterion).*

It follows from this definition that three basic entities are involved in system identification: *the data, a model set and a rule or criterion* for model estimation.

The input-output data are usually collected from an identification test or experiment that is designed to make the measured data maximally informative about the system properties that are of interest to the user.

A set of candidate models is obtained by specifying their common properties; a suitable model is searched for within this set. This is the most theoretical part of the system identification procedure. It is here that *a priori* knowledge and engineering intuition and insight have to be combined with the formal (mathematical) properties of models. In this book we will use linear/nonlinear, time-invariant and finite dimensional models of multi-input multi-output (MIMO) systems that are suitable for modeling a large class of industrial processes.

When the data are available and the model set is determined, the next step is to find the best model in this set. For model parameter estimation, an error criterion (loss function) is specified. Often the sum of the squares of some error signals (residuals) is used as the criterion. The values of the parameters are determined by minimizing the loss function.

When a model is identified, the next step is *model validation*. This step is to test whether the estimated model is sufficiently good for the intended use of the model. First of all, a check to see if the model is in agreement with the *a priori* knowledge of the system. Then a check if the model can fit the test or experimental data well, preferably using a data sequence that has not been used in model estimation. The final validation of the model is the application of the model.

To summarize, the system identification procedure has the following basic steps:

1. Identification tests or experiments
2. Model order/structure selection
3. Parameter estimation
4. Model validation

Figure 1.1.1 shows an identification procedure; see Ljung (1987).

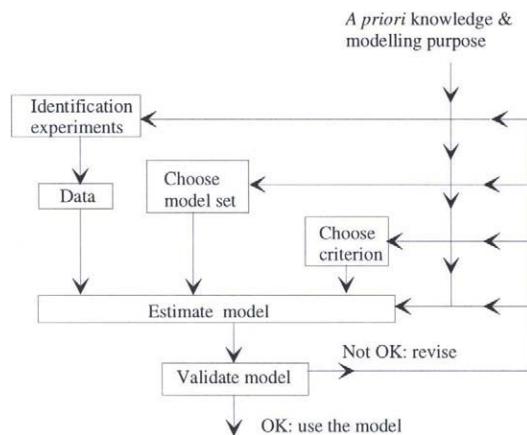


Figure 1.1.1 Idetification procedure

It is possible that a model first obtained cannot pass the model validation procedure. The causes of the model deficiency can be the following:

- The data were not informative enough due to poor test or experiment design.
- The model set did not contain a good description of the system.
- The criterion was not chosen properly.
- The numerical procedure failed to find the best model according to the criterion.

The major part of an identification application consists of addressing these problems. A good identification method, therefore, should provide systematic solutions to these problems.

1.2 The Hierarchy of Modern Automation Systems

The process industries include the following major sectors: food, textiles, paper, chemicals, petroleum, rubber and plastics, glass, metal and electricity. Due to world-wide competition, shortage of natural resources and environmental pollution, the present process industries face very dynamic and unpredictable market conditions and have to produce under very strict national and international regulations. Computer-based automation systems have been developed in process industries in order to increase the production safety, quality and flexibility, and to reduce energy and material consumption as well as environmental pollution. Some process industries, such as the refinery and petrochemical industry, recognize plant automation more and more as a cost effective way of responding to changes of market conditions and production regulations. A modern automation system in process industries can be depicted as a pyramid and consists of the following layers (see Figure 1.2.1).

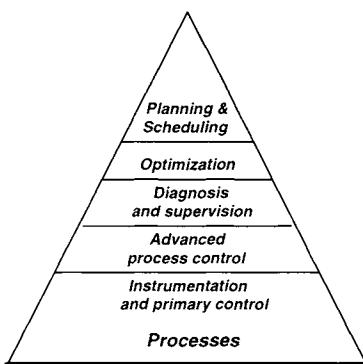


Figure 1.2.1 The hierarchy of industrial automation systems

- *Instrumentation and primary control:* This layer is usually a distributed control system (DCS) which gathers process measurements and performs simple monitoring of the measurements. The measurements include basic process variables such as temperature, flow, pressure, level and valve position. A DCS also performs PID based controls on some of the process variables. Usually one controller only takes care of a single primary variable such as flow or temperature.
- *Advanced process control (APC):* This part of the system performs multivariable model-based control that will ensure stable unit operation and push the process to its operational limits for maximum economic benefit. Here one APC controller can control a whole process such as a distillation column, a reactor. In general, identified dynamic models (most often linear) are used for APC controllers. This layer is usually present in a computer.

- *Diagnosis and supervision:* This part of the system is to improve the safety and reliability of the unit operation. A diagnosis system performs fault detection and classification and gives suggestions for maintenance and remedies. Early methods are mainly based on limit value checking of easily measurable signals and their derivatives; a recent trend is to use process models for more accurate and quicker diagnosis. The system can also evaluate the performance of the controllers at different levels. This layer is usually present in a minicomputer.
- *Optimization:* An optimization system manipulates the process degrees of freedom that are left after meeting the requirements of a safe and stable operation, to meet the unit economic objectives such as saving energy and expensive material and/or increasing throughput. The optimizer determines the best setpoints for APC controllers. Usually the optimization is carried out based on rigorous (first principle) nonlinear static models. Sometimes identified models are also used for optimization, because the cost of using and maintaining a rigorous model can be too high. Usually the optimizer is executed at a slow rate such that the APC controls are at steady state with respect to the previous setpoint change. The optimization can be performed for a single process as well as a combination of processes. An optimizer is usually located in a minicomputer.
- *Planning and scheduling:* This part may cover many units of processes and it provides decision support in production planning, allocation of raw materials and scheduling of plant operation for realizing the company's program and for maximizing profits. It is used to respond to the market changes as well as production regulation changes. This part can be located in a minicomputer or a main frame computer.

Each layer plays an unique and complementary role in the total automation system and that allows the company to react rapidly to changes.

At present, most process industries have instrumentation and primary control. Only some capital intensive sectors use higher level layers such as APC, optimization and scheduling. To our knowledge, refinery and petrochemical industries take the lead in applying computer automation systems. Due to the availability of affordable and reliable computers and to development of computing and control sciences, the time is coming that more process industries can benefit from this multi-disciplinary technology.

Process identification, which is dealt with in this book, will serve at least the following layers in a process automation system:

- *Primary control.* model-based PID tuning can increase the efficiency of control tuning and improve the performance of the controllers; see Chapter 10.
- *Advanced process control.* This layer uses model-based control technologies such as MPC. Process identification plays a crucial role here; see Chapter 10.

- *Diagnosis and monitoring.* The current trend is to use model-based diagnosis methods. This can reduce the cost of redundant equipment; see Chapter 11.
- *Optimization.* The justification for using identified models for optimization is twofold. The first is the low cost of identification compared to rigorous physical modeling. The second is that the optimizer using identified models can work at much higher sampling rate, because the identified models are simple to use and they are dynamic models. This makes the coordination between the optimizer and APC controllers easier. The ability of nonlinear model identification will make the identified models more suitable for optimization.

One sees see that process identification can and will play a very important role in modern process automation systems.

1.3 Multivariable Model-Based Process Control

Before going into identification problems, one needs to become familiar with the environment where identified models are going to be used. For this purpose we will briefly discuss the characteristics of a large class of industrial processes and also summarize the currently available model-based control technology.

1.3.1 Continuous Processes

A continuous process operates at a stationary state for a very long period of time. In each stable operating point the process is working at an equilibrium condition where the variation of material composition is small, and the process conditions such as temperatures, pressures and flow rates are kept as constant as possible or within zone limits. In general a continuous process can be effectively controlled using linear process models, at least for each mode (operating point). This is because any nonlinear function can be well approximated by a linear function around an equilibrium, which explains the success of linear model-based control technologies for continuous processes. The area where linear models are less or non-effective is during process start-up and shut-downs where nonlinear behavior of the process becomes dominant.

The following are some examples of continuous processes in process industries:

Distillation Columns. A distillation process is used to separate a given feedstock into products of higher value. The separation process is basically to vaporize the feed and draw the products from different locations (trays) of the column at different temperatures. The value of the products depends on their quality (purity). This makes quality control very important in operating a column. The operation must also be profitable and meet production goals. The role of distillation control is to meet these tightly interrelated objectives.

Continuous Reactors. Reactors are used to convert the feedstocks into more valuable products via chemical reactions. Usually reactions take place under special physical conditions, e.g., high pressure and high temperature; and catalysts are used to stimulate the reactions. The continuous reaction operation is realized by constant flows of feed and catalyst. Fluid-bed catalyst cracking (FCC) units and polyethylene reactors are two important examples of continuous reactors in petrochemical industry.

Utility Processes. This class of processes provide energy, air, steam and other components needed for production processes. These processes include boilers, air blowers, pumps and compressors. Among the operational requirements, safety and reliability are the most important here, because the malfunction of this equipment will cause shut-down of the production processes. Both control and fault diagnosis are effective ways to solve the problem of safe and reliable operation.

Rolling Mills. Rolling mills are used in the steel industry to reduce the thickness of metal strips. A basic quality measure is the flatness of the rolled strips, which demands the reduction of thickness variations. Also the tension in the strip must be constrained for safety reasons. Rolling mills have very fast dynamics as opposed to petrochemical processes. This requires a high sampling frequency for the control system; for example, a controller for a cold rolling mill needs to sample at 10 Hz.

From a control theoretical point of view, characteristics of these industrial processes can be stated as:

- mostly distributed parameter processes (physical modeling will lead to a set of partial differential equations);
- possibly nonlinear for practical ranges;
- slowly time varying dynamics;
- process and measurement time delays exist;
- continuous-time behavior.

There is no general theory available for the modeling and control of this class of process units. In this book, we try to circumvent these obstacles by lumping techniques, compensating the nonlinear relations and restricting the ranges in amplitudes and time. Therefore the following assumptions on the process are made:

- the process is operated in the neighborhood of a limited set of working points;
- the process stays near a certain working point for a long time compared to the largest process time constant;

- at each working point, the process dynamics are time-invariant or the change is extremely slow;
- at each working point, the disturbances can be considered to be stationary;
- at each working point, the process behavior can be accurately approximated by linear models or by block-oriented nonlinear models.

Though this sounds restrictive, there is a very large class of industrial processes which fulfils these conditions. Therefore, the identification methods studied in this book can be applied to them for model-based control.

1.3.2 Batch and Semibatch Processes

In batch processing the situation is quite different from that of continuous processes. The dynamics of the batch process are strongly related to the amount, composition and physical properties of the processed materials. The dominant time constants may vary by an order of magnitude from one batch to another. Also, within one batch the change in dynamic behavior may be significant, for instance as a result of changing liquid volume due to evaporation or feeding (in fed-batch applications). Each operation, from the beginning to the end, must be controlled carefully. In particular, when irreversible reactions take place, control errors in any operation may waste the complete batch product, whereas in continuous processes the effect of control errors on the product is usually washed out in the large storage tank, or can be compensated for afterwards by blending. Thus the batch control system must be designed to perform well over the entire range of the operation. Considering the nonlinearity of most batch processes, this can be non-trivial as the linearity assumption may not hold over the full range of operation. An additional complication in batch processing is the existence of different control objectives in the various unit operations. Typical examples are temperature control, reaction rate control, heat control and composition control. Changing the control objective requires a reconfiguration of the control system. Examples of batch processes are batch chemical reactors and bio-reactors. Generally speaking, controlling batch processes thoroughly needs nonlinear models and nonlinear control strategies, which is more challenging both theoretically and practically.

1.3.3 Model- Based Control Methods

Linear Time-Invariant Controllers. Linear time-invariant controllers are very popular in the electrical, mechanical and aerospace industries. This class of control strategies uses linear process models. The most successful methods that can deal with multivariable problems are the LQG (Linear Quadratic Gaussian) control and H-infinity control. The LQG control was developed in the 1960s, based on the pioneer work of Kalman (Ho and Kalman, 1960). This method uses a state space model of the process. The control objective is to minimize a quadratic loss function of

weighted states and weighted inputs. The control law is expressed in a state feedback form. The state vector $x(t)$ can be recovered from noisy input-output data using a Kalman filter. Kalman filter and LQG control have been applied successfully for estimation, prediction and control, to processes such as aircraft, ships and some power plant processes. Common characteristics of these processes are that accurate models can be developed, reliable and accurate sensors are available and actuators are powerful. However, the application of Kalman filters and LQG control is very limited in process industries. The first reason is that industrial processes are poorly modelled. LQG control does not consider the consequence of model uncertainties and is sensitive to model errors. Another reason is that an LQG controller cannot handle constraints, which is important for process industries. The need to address robustness against model uncertainties has motivated the development of so called H-infinity and robust control theory. The work was pioneered by Zames (1981), and the solutions to the problem were given in, e.g., Skogestad and Postlethwaite (1996). An H-infinity controller is obtained by minimizing the H-infinity norm of the weighted sum of the sensitivity function and control sensitivity function. The sensitivity function is the closed loop transfer function matrix from the output disturbances to the outputs. Control sensitivity is expressed by the transfer function matrix from disturbances to manipulating inputs. The weightings are used to reflect the performance requirements and model errors. The analysis and design of this class of robust controllers need not only a process model (called nominal model) but also a proper model error quantification usually expressed by some upper bounds in the frequency domain or parameter space. Because H-infinity control can cope with model errors, it is more practical for process control than LQG control. The shortcoming of H-infinity control is its inability to deal with constraints (constraint handling).

Model Predictive Control (MPC). The name model predictive control (MPC) reflects a common feature of all MPC's that an dynamic model of the process is used to predict the effect of future actions of manipulated variables to the controlled variables. Basically, an MPC controller works as follows: At present time t the behavior of the process over a time horizon is considered. Using a model, the process response to changes in manipulating inputs is predicted. The future moves of the manipulating inputs are chosen such that the predicted response meets the control objectives. Only the first computed move of the manipulating inputs is implemented. At the next sample interval the computation is repeated with the horizon moved by one. In most MPC controllers, the desirable performance is achieved by minimizing a quadratic loss function of the error between the setpoint and the predicted output, plus a penalty term for input movement. This optimization respects all required process operation constraints, such as valve high and low limits or a temperature high limit. When a manipulated variable constraint is hit, the algorithm will automatically adjust all other manipulating variables to still satisfy the control objectives. The control objectives associated with the controlled variables can be assigned with priorities and are handled accordingly. If, at a certain instance, the number of manipulated variables is insufficient to fulfil all control objectives simultaneously, the controller will satisfy the higher ranked control objectives at the cost of the lower prioritized objectives. Feedforward control is naturally included in MPC. The ability to handle constraints is the dominating factor that makes the MPC controller widely applicable. It is also much more flexible than a standard time-invariant controller. The controlled outputs can be either controlled to their setpoints or

be kept within given zone limits. The number of controlled outputs can be larger than the number of manipulating inputs. The optimization is solved on-line during each sample interval. This makes MPC controllers computationally much more involved than time-invariant control algorithms. The technology has been applied successfully to many process units in the refinery and petrochemical industry.

1.3.4 Process Control Examples

A single stand rolling mill

The schematic overview of the process is shown in Figure 1.3.1. A metal strip is rolled in order to reduce the thickness of the strip. The variation in the exit thickness is an important quality measure. The increasing demand on the quality requires the reduction of the thickness variation, and this could no longer be accomplished by the operator. To reach this goal, a computer control system needs to be developed.

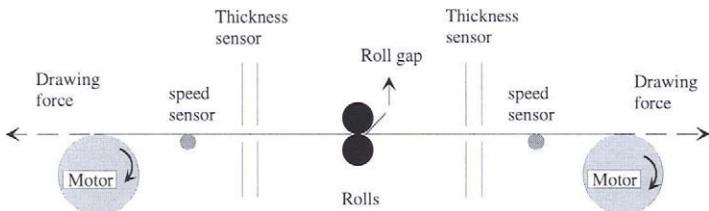


Figure 1.3.1 The single stand rolling mill

A glass tube drawing process

The outline of the process is shown in Figure 1.3.2. By indirect electric heating the glass is melted and it flows down through a ring shaped hole along a mandrel. Shaping of the tube takes place at, and just below, the end of the mandrel. The glass tube is pulled down due to gravity and a drawing machine. Two measures of dimension, wall thickness and diameter of the tube, are the most important quantities to be regulated. Many different types of tubes (differing in dimensions) are produced in the same process. The mandrel gas pressure and the drawing speed can affect the wall thickness and diameter most directly and easily. Other variables that also affect the thickness and diameter are the power supplied to melt the glass, the pressure in the melting vessel and the room temperature.

Strong interactions between different variables exist: an increase in the drawing speed causes a decrease of both the wall thickness and diameter; an increase of gas pressure will increase the diameter, but decrease the wall thickness. There are large time delays, because the tube dimensions can only be measured when the glass has considerably cooled down. Thus classical PID control could not meet the increased performance requirements for the process.

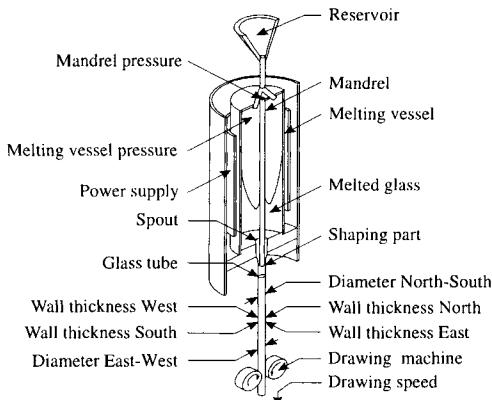


Figure 1.3.2 Glass tube drawing process

Identification and multivariable control techniques were to be developed for this process. The control objectives were: 1) a reduction of the dimension variations at different working points (increasing product quality); 2) automation of change-over between different working points, and reduction of change-over time (increasing flexibility)

A distillation column

The process is called Stabilizer which is a part of a catalyst reformer unit at a refinery. The simplified process diagram with regulatory control is shown in Figure 1.3.3. The stabilizer is used to separate from the feed the LPG (liquid petroleum gas) as the top product and reformatte as the bottom product. The feed consists of naphthas, paraffins and aromatics. The composition of the feed various but cannot be measured. Pentane (C5) and heavier components which should go to the bottom are the impurity of the top product. On the other hand, butane (C4) and lighter components which should go to the top product are the impurities of the bottom product. The pentane concentration in the top product is measured by an analyzer. Reid Vapor Pressure (RVP) is highly correlated with the butane concentration. So the butane concentration in the bottom product is measured indirectly by analyzing its RVP.

The existing regulatory control of the stabilizer is as follows:

- The mass balance of the process is realized by a bottom level controller that moves the bottom product flow and an overhead drum level controller that moves the top product (distillate) flow; the two flow controllers move the corresponding valve positions.
- The overhead pressure controller moves the vapor valve position to control the pressure.
- The feed flow is used by a level controller preceding the stabilizer, which means that the feed flow cannot be used as an input for the stabilizer control.

- Finally, the top product impurity (C5 %mol in LPG) and the bottom product quality (RVP of reformate) are controlled by the operator manually.

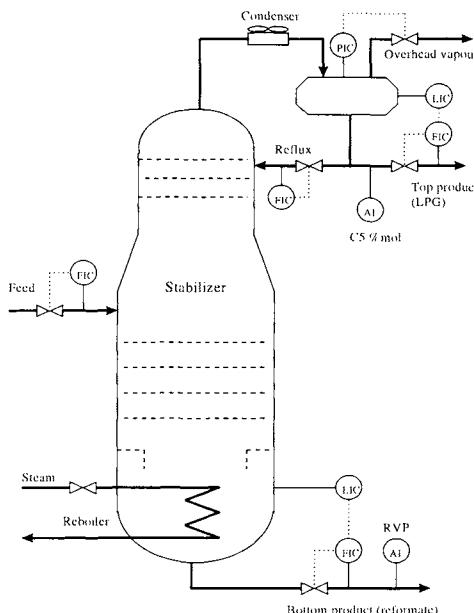


Figure 1.3.3 Stablizer flow diagram with regulatory controls. Symbols: A circle with FIC denotes a Flow Indicator and Control with the solid line points to the (flow) measurement and the dotted line points to the control output (valve position); a circle with PIC denotes a Pressure Indicator and Control with the solid line points to the measurement and dotted line points to the control output ; a circle with LIC denotes a Level Indicator and Control with the solid line points to the measurement and dotted line points to the control output; and a circle with AI denotes an Analyzer

The **control requirements** for the stabilizer are:

- *Constraint control.* The controller must respect the following process constraints: the reboiler valve between 0 and 100%, the reflux flow between 40 and 80 m³/h, and pressure controller valve between 0 and 100%, flooding (a calculated variable reflecting the load of the column) high limit 75%.
- *Product quality control.* Bottom product RVP high limit 0.5 bar and top product C5 high limit 15%mol.

- *Decoupling.* The setpoint of the bottom and top product quality can be changed independently.
- *Optimization.* Minimize bottom product RVP, maximize top product C5 up to a limit in order to maximize top product flow, and maximize pressure to its high limit in order to minimize LPG losses to overhead vapor.

1.4 Outline of the Book

In Chapter 2 several different mathematical representations of dynamic processes and signals are introduced which are necessary in process identification and also for control system design. Both linear and nonlinear models will be discussed.

Chapter 3 will discuss identification tests (experiments). This is the first and the most crucial step in an identification procedure. In this chapter, problems such as the selection of inputs and outputs, sampling frequency, test signals and pre-treatment of the data are discussed in detail. Different experiments and tests are suggested for gathering the necessary information about the process dynamics. Also attention is paid to closed-loop tests and to tests for nonlinear model identification.

When input-output data are collected and data pre-treatment finished, it is convenient for the reader to begin with the simplest least-squares method (also called equation error method or ARX model method). This method is treated in Chapter 4. After the introduction of the method, we present two industrial cases in order to illustrate the method. In the first case, the identification and control of a single stand rolling mill, very successful results are obtained. This will give our reader the first taste of the usefulness of identification. He can try the same thing on his own process. If he succeeds, he can stop reading the book and continue his other (more important) business. If he has the bad luck that the least-squares method does not work, as we experienced in the second case, the subsequent sections will explain how and why the least-squares method fails. Also, if a reader wants to obtain better results than those of the least-squares method, he will be motivated to read the following sections and chapters.

In Chapter 5 different ways to overcome the bias problem of the least-squares method are studied. Several popular techniques, such as the output error method, instrumental variable methods and the family of prediction error methods are introduced. Their properties are explained in parametric and in frequency response terms. Also practical limitations of this class of methods will be pointed out. The shortcomings of these methods call for other alternatives. Recursive estimation methods are also introduced.

In Chapter 6, the so called asymptotic method (ASYM) will be introduced. This method calculates time domain parametric models based on the frequency domain measures. In doing so, one can easily solve fundamental problems such as optimal input design and model order determination. One important feature of the method is its ability to supply an upper bound of

model errors in the frequency domain, which is needed for robust control design.

In Chapter 7, the asymptotic method (ASYM) will be extended for multivariable processes.

Chapter 8 studies the so-called subspace methods which identify MIMO state space models directly from input-output data. The advantages of subspace methods are their numerical efficiency and the ability to directly identify compact state space models.

Nonlinear process identification is treated in Chapter 9. Block-oriented models such as the Hammerstein model, Wiener model and combined Hammerstein-Wiener model will be studied. ASYM method is extended for the identification of these models. Simulations and laboratory tests will be used to verify the proposed methods.

In Chapter 10 we will discuss the applications of identification in model-based process control. First an industrial MPC project approach will be outlined. Then we will show the use of identification for tuning PID controllers. The identification of ill-conditioned processes will be discussed. Finally, two industrial case studies will be described. In this chapter, the reader can learn more on how to do identification and control, and the successful applications may encourage him to use identification to solve his own problems.

Chapter 11 studies model-based fault diagnosis and isolation (FDI). Early fault diagnosis of industrial processes can improve production safety and reliability. It will be shown that process models play a very important role here.

Appendix A contains some basic concepts of linear algebra and matrix theory.

Some new developments in identification are not treated in this book. Frequency domain identification of parametric models has reached its mature state in the last decade; see Schoukens and Pintelon (1991). When only frequency response data are available, one has to use the frequency domain method. When time domain data is used, the frequency domain method and time domain method are, in general, equivalent. In such a case, the frequency domain method will not offer obvious advantages. Most process control applications use time domain data. Also not treated in this book is the neural-network approach to modeling and identification. Compact parametric models are more economical to use than (non-parametric) neural-net models. The tests needed for parametric models are shorter than those for neural-net models and the computation demand is much lower. Moreover, the *a priori* knowledge about the process can be better utilized in a parametric model. Therefore, we would like to recommend industrial users to first try the methods in this book before considering a neural-network approach in modeling and identification.

Chapter 2

Models of Dynamic Processes and Signals

Process identification is the subject of constructing models for certain utilitarian purposes from measured data. A first step in the total identification procedure is to determine a class of models in which the most suitable model is to be sought. In this chapter we will review some well known results on how to describe dynamic processes and signals using linear models and block-oriented nonlinear models.

First linear time-invariant models will be discussed. Linear models form the most important and well-developed class of models both in practice and in theory. Although they represent idealized processes in the real world, the approximations involved are often justified for the given applications of the models. A typical example is to describe the behavior of a industrial process, possibly nonlinear, by a linear time-invariant model around a working point. Continuous-time models are discussed in Section 2.1; discrete-time models are discussed in Section 2.2. Models of multi-input multi-output (MIMO) processes are discussed in Section 2.3. Models of signals are discussed in Section 2.4. Section 2.5 combines the linear model of the process and that of the disturbance which form the basic model structure for linear process identification. Section 2.6 discusses nonlinear models where the use of block-oriented models will be motivated.

2.1 SISO Continuous-Time Models

Most of the industrial processes we study are inherently continuous-time processes; all the signals of this kind of processes are time-continuous. So it is logical to start with continuous-time models. A linear time-invariant single-input single-output process can be described by an ordinary differential equation as

$$y^{(n)}(t) + a_1y^{(n-1)}(t) + \cdots + a_ny(t) = b_0u^{(m)}(t) + b_1u^{(m-1)}(t) + \cdots + b_mu(t) \quad (2.1.1)$$

where $u(t)$ and $y(t)$ are the input and the output at time t ; superscript n denotes n -th order derivative. The coefficients $a_1, \dots, a_n, b_0, \dots, b_m$ are constant. The order of the process model is defined as n . Note that $n > m$ for a physically realizable or causal process.

Taking the Laplace transform on both sides of (2.1.1) and assuming zero initial conditions yield the transfer function of the process model:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} \quad (2.1.2)$$

where $U(s)$ and $Y(s)$ are the Laplace transforms of the input and the output. A linear time-invariant process with measurement delay d is described by a differential equation as

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \dots + a_n y(t) = b_0 u^{(m)}(t-d) + b_1 u^{(m-1)}(t-d) + \dots + b_m u(t-d) \quad (2.1.3)$$

and by the transfer function as

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} \cdot e^{-sd} \quad (2.1.4)$$

It is well known that a linear time-invariant and causal process can be described by impulse response $g(\tau)$ as follows

$$y(t) = \int_0^\infty g(\tau) u(t-\tau) d\tau \quad (2.1.5)$$

If we introduce a vector of auxiliary variables and express (2.1.1) as a set of first order differential equations, we get a state space realization of the model:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (2.1.6)$$

where $x(t) = [x_1(t) \ \dots \ x_n(t)]^T$ is called a state space vector. Assume that $m = n - 1$ in (2.1.1), then a state space realization of the process is

$$\begin{aligned} A &= \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ \vdots & & & & \\ -a_n & 0 & 0 & \dots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix} \\ C &= [1 \ 0 \ \dots \ 0], \quad D = [0 \ 0 \ \dots \ 0] \end{aligned} \quad (2.1.7)$$

This is called the observer canonical form in the literature; see, e.g., Kailath (1980).

A given differential equation can have many state space descriptions. Therefore state space realizations are not unique. A great advantage of using state space models is their convenience for the analysis and controller design for MIMO processes.

It is well known that the transformation from a state space description to the transfer function is

$$G(s) = C(sI - A)^{-1}B + D \quad (2.1.8)$$

2.2 SISO Discrete-Time Models

A schematic diagram of a computer control system is given in Figure 2.2.1. Due to the use of process computers, signals from the underlying continuous-time process are sampled and digitized. These signals have discrete amplitude values at discrete-time points. Models describing the relationship between these kind of signals are called discrete-time models.

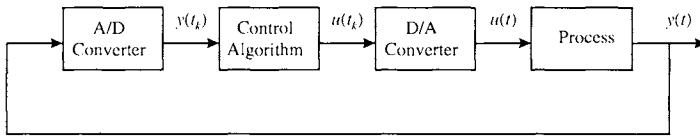


Figure 2.2.1 Schematic diagram of a computer control system

In this book, we assume synchronous sampling at computer inputs and outputs and under a single sampling rate (frequency); also, we assume that the quantization error is much smaller than the measurement noises and disturbances such that the sampled amplitudes can be considered continuous. Under these realistic assumptions, a sampled linear process can be described by a difference equation or a z-transfer function relationship.

Typically a discrete-time description of a sampled continuous-time process is obtained as follows. In most of the computer controlled systems, the process input $u(t)$ (the output of the D/A converter) is kept constant during a sampling interval using a zero order hold. This practical arrangement also greatly simplifies the theoretical analysis. Suppose the process is linear and time-invariant with impulse response $g(\tau)$, then at sampling instants $t_k = kT, k = 1, 2, \dots$, where T is the sampling interval:

$$y(kT) = \int_0^{\infty} g(\tau)u(kT - \tau)d\tau \quad (2.2.1)$$

Because the input is constant over the sampling interval (piecewise constant), then

$$u(t) = u_k, \quad kT \leq t < (k+1)T \quad (2.2.2)$$

(2.2.1) becomes

$$\begin{aligned} y(kT) &= \int_0^\infty g(\tau)u(kT - \tau)d\tau \\ &= \sum_{l=1}^{\infty} \int_{(l-1)T}^{lT} g(\tau)u(kT - \tau)d\tau \\ &= \sum_{l=1}^{\infty} \left[\int_{(l-1)T}^{lT} g(\tau)d\tau \right] u_{k-l} = \sum_{l=1}^{\infty} g_T(l)u_{k-l} \end{aligned} \quad (2.2.3)$$

where we defined

$$g_T(l) = \int_{(l-1)T}^{lT} g(\tau)d\tau, \quad l = 1, 2, \dots \quad (2.2.4)$$

which is called the discrete-time impulse response of the sampled process. Note that no approximation is involved in deriving (2.2.3)

Even if the input is not piecewise constant, the representation (2.2.3) may still be a reasonable approximation provided $u(t)$ does not change too much during a sampling interval. We will use the notation (2.2.1) to (2.2.4) when the choice of sampling interval is discussed. In most of the chapters we will study the identification of discrete-time processes. For ease of notation we shall assume that the sampling interval is one time unit and we use t to denote the sampling instants. This yields for (2.2.3)

$$y(t) = \sum_{k=1}^{\infty} g_k u(t-k), \quad t = 1, 2, \dots \quad (2.2.5)$$

Let us introduce the unit forward shift operator q

$$qu(t) = u(t+1)$$

and the unit delay operator q^{-1}

$$q^{-1}u(t) = u(t-1)$$

We can then write for the convolution sum of (2.2.5)

$$y(t) = \sum_{k=1}^{\infty} g_k [q^{-k}u(t)] = \left[\sum_{k=1}^{\infty} g_k q^{-k} \right] u(t) \quad (2.2.6)$$

Define the transfer operator of the process (2.2.5) from the convolution of (2.2.6) as:

$$G(q) = \sum_{k=1}^{\infty} g_k q^{-k} \quad (2.2.7)$$

We then obtain a convenient description of the process as:

$$y(t) = G(q)u(t) \quad (2.2.8)$$

It is known that the sampling of an n -th order linear time-invariant process using a zero-order hold results in an n -th order difference equation as follows

$$y(t) + a_1 y(t-1) + \cdots + a_n y(t-n) = b_1 u(t-1) + \cdots + b_n u(t-n) \quad (2.2.9)$$

Note that the coefficients here have different meanings to those of (2.2.1), although the symbols are the same. This will not cause confusion, because we only treat discrete-time models in the book except in Chapter 11. Note also that there is one delay in the difference equation which is caused by the zero-order hold.

The transfer operator of (2.2.9) is

$$G(q) = \frac{b_1 q^{-1} + \cdots + b_n q^{-n}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}} \quad (2.2.10)$$

If there is a delay of d samples, then the transfer operator becomes

$$G(q) = \frac{b_1 q^{-1} + \cdots + b_n q^{-n}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}} \cdot q^{-d} \quad (2.2.11)$$

One can see that in discrete-time, a finite dimensional model with a delay is still finite dimensional, while in continuous-time, a delay cannot be modelled exactly by a finite dimensional model; compare (2.1.4). This is an advantage of using discrete-time models.

Replacing q by the z -transform variable z , then $G(z)$ is called the transfer function of the process. If one evaluates the transfer function on the unit circle, $z = e^{i\omega}$, then $G(e^{i\omega})$ is called the frequency response of the process. Sometimes people do not observe the difference between q and z , which is theoretically not correct, but this will not do any harm in applications.

The leading coefficient of the denominator polynomial of the transfer operator (2.2.10) is fixed as 1 for the uniqueness of the model. A polynomial is called *monic* if its leading coefficient is unity.

A model in the form of impulse response (2.2.5) is called a non-parametric model; a model in the form of difference equation (2.2.9) or transfer function (2.2.10) is called a parametric model.

When rearrange (2.2.9) as a set of first order difference equations, one gets a state space realization of the model:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (2.2.12)$$

where $x(t) = [x_1(t) \quad \dots \quad x_n(t)]^T$ is a state space vector. The observer canonical form of (2.2.9) is given as

$$\begin{aligned} A &= \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & & 1 & & \\ & & \ddots & & \\ & & & & 1 \\ \vdots & & & & \\ -a_n & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \\ C &= [\quad 1 \quad 0 \quad 0 \quad \cdots \quad 0 \quad], \quad D = [\quad 0 \quad] \end{aligned} \quad (2.2.13)$$

The transfer operator can be obtained from a state space description by

$$G(q) = C(qI - A)^{-1}B + D \quad (2.2.14)$$

In advanced process control (APC) systems, often there are two control layers: the regulatory level with PID loops and advanced control (or, supervision control) level with model based MIMO controllers such as MPC (model predictive control). The sampling rate of the lower level is usually much faster than that of the upper level. For example, the sample interval at the lower level is 1 second and the sampling interval of MPC control is 1 minute. Normally the APC layer will send command signals to regulatory layer. However, often the two layers will interact in two directions. Under such a situation, it is sometimes more accurate to use discrete-time models for the APC control without any delay. For example, suppose the setpoint of a flow controller is a process input (or, manipulated variable, MV) of a MPC controller and the valve position (or control output) of the flow control is a process output (or, controlled variable, CV) of the MPC controller. If the flow controller settling time is in the order of few seconds, then, for the MPC controller which is sampled at 1 minute, there will be practically no delay from flow setpoint to valve position. Based on this observation, in this book we will use delay-free models as a general description and introduce delays when necessary.

The delay-free convolution model is

$$G(q) = \sum_{k=0}^{\infty} g_k q^{-k} \quad (2.2.15)$$

The transfer operator of a delay-free mode is

$$G(q) = \frac{b_0 + b_1 q^{-1} + \cdots + b_n q^{-n}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}} \quad (2.2.16)$$

The observer canonical form of (2.2.16) is given as (Kučera, 1991)

$$\begin{aligned} A &= \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & & 1 & & \\ & & \ddots & & \\ \vdots & & & & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 - a_1 b_0 \\ b_2 - a_2 b_0 \\ \vdots \\ b_n - a_n b_0 \end{bmatrix} \\ C &= [\begin{array}{ccccc} 1 & 0 & 0 & \cdots & 0 \end{array}], \quad D = [b_0] \end{aligned} \quad (2.2.17)$$

2.3 MIMO Models

Because we will mainly treat discrete-time models in this book and there are many similarities between continuous-time models and discrete-time models, we shall only study discrete-time MIMO models in this section.

Given a discrete-time linear time-invariant process with m inputs, p outputs, we can describe the relation between the inputs and the outputs by impulse responses and convolutions

$$y(t) = \sum_{k=0}^{\infty} G_k u(t-k), \quad t = 0, 1, 2, \dots \quad (2.3.1)$$

where $y(t)$ is the p -dimensional output vector and $u(t)$ is the m -dimensional input vector at sampling instant t ; G_k is a sequence of $p \times m$ matrices which form the discrete-time impulse response. Using the unit time delay operator q^{-1} we obtain

$$y(t) = G(q)u(t) \quad (2.3.2)$$

where

$$G(q) = \sum_{k=1}^{\infty} G_k q^{-k} \quad (2.3.3)$$

is called the transfer operator or transfer function of the MIMO process.

If the process is finite dimensional, then each entry of the transfer operator matrix $G(q)$ can be written as a rational function in the form of (2.2.10) or (2.2.11). This form is not convenient for identification and also not suitable for controller design.

For the purpose of identification, a suitable description of a MIMO process is a set of difference equations as follows

$$y(t) + A_1 y(t-1) + \cdots + A_n y(t-n) = B_0 u(t) + B_1 u(t-1) + \cdots + B_n u(t-n) \quad (2.3.4)$$

where $A_1(p \times p), \dots, A_n(p \times p), B_0(m \times p), B_1(m \times p), \dots, B_n(m \times p)$ are constant matrices.

Using the unit delay operator q^{-1} , we write for (2.3.4)

$$A(q)y(t) = B(q)u(t) \quad (2.3.5)$$

where $A(q)$ and $B(q)$ are polynomial matrices:

$$A(q) = I + A_1q^{-1} + \dots + A_nq^{-n}, \quad B(q) = B_0 + B_1q^{-1} + \dots + B_nq^{-n}$$

Comparing (2.3.2) and (2.3.5) we have

$$G(q) = A(q)^{-1}B(q) \quad (2.3.6)$$

provided that $A(q)$ is invertible. Expressing the transfer operator matrix by the two polynomial matrices is called *matrix fraction description* (MFD). Note that the degrees of the entries of $A(q)$ and $B(q)$ can be lower than n .

Unlike the SISO case, for a given process the representation (2.3.5) is in general not unique even when we fix the leading coefficient matrix of $A(q)$ at I . Therefore some special forms of representations (canonical forms) are needed in order to have unique models of a given process; and model uniqueness is a necessary condition for an identification algorithm to find a meaningful solution. Canonical forms for MIMO processes are often regarded as difficult. This difficulty can be avoided by using one of the simplest canonical forms, *diagonal form* MFD. The diagonal form MFD of a given $G(q)$ is

$$A(q) = \begin{bmatrix} A_{11}(q) & 0 & \cdots & 0 \\ 0 & A_{22}(q) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & A_{pp}(q) \end{bmatrix}, \quad B(q) = \begin{bmatrix} B_{11}(q) & \cdots & B_{1m}(q) \\ \vdots & \ddots & \vdots \\ B_{p1}(q) & \cdots & B_{pm}(q) \end{bmatrix} \quad (2.3.7)$$

where $A_{11}(q), \dots, A_{pp}(q)$ are all monic polynomials with relevant degrees, and the degrees of $B_{i1}(q), \dots, B_{im}(q)$ are equal to or less than that of $A_{ii}(q)$. Under this arrangement, an m -input, p -output process is decoupled into p m -input single output subprocesses

$$\begin{aligned} A_{11}(q)y_1(t) &= B_{11}(q)u_1(t) + \cdots + B_{1m}(q)u_m(t) \\ &\vdots && \vdots \\ A_{pp}(q)y_p(t) &= B_{p1}(q)u_1(t) + \cdots + B_{pm}(q)u_m(t) \end{aligned} \quad (2.3.8)$$

Any given transfer operator of a physical process can be written in the diagonal form MFD. This form has many advantages for use in identification:

- It is simple to comprehend.

- Model structure selection becomes order selection for each MISO submodel (see Chapter 5 and 7). This substantially reduces the complexity of the problem and makes the identification of large scale process identification possible.
- Using this form, (almost) all the identification algorithms developed for SISO processes (see Chapters 4, 5 and 6) can be generalized for MIMO processes.
- Delay correction (see Chapter 3) can be done for each single transfer operator.

One possible disadvantage is that the diagonal form MFD of a given process is not always minimal. This means that the order of its direct state space realization can be higher than the McMillan degree of the process. This problem, however, can be solved using some model reduction techniques; see Chapter 7. The other possible disadvantage is to use it for identifying the so-called ill-conditioned processes; see Chapter 10 for details.

2.4 Models of Signals

Time domain variables such as inputs, states, outputs and disturbances of a process, are called signals. In this section we will discuss how to describe signals using a frequency domain representation and also in a probability framework. Only the SISO discrete-time case is studied.

2.4.1 Periodograms of Signals over Finite Intervals

Consider the finite sequence of input $u(t), t = 1, 2, \dots, N$. Define the function $U_N(\omega)$ by

$$U_N(\omega) = \frac{1}{\sqrt{N}} \sum_{t=1}^N u(t) e^{-i\omega t} \quad (2.4.1)$$

The values at $\omega = 2pk/N, k = 1, \dots, N$, form the discrete Fourier transform (DFT) of the sequence. The time domain signal can be obtained by the inverse DFT as

$$u(t) = \frac{1}{\sqrt{N}} \sum_{k=1}^N U_N\left(\frac{2\pi k}{N}\right) e^{-i2\pi kt/N} \quad (2.4.2)$$

We know that $U_N(\omega)$ is periodic with period 2π .

It is well known by control engineers that DFT decomposes the time domain signal into its frequency domain components; the number $U_N(2\pi k/N)$ is the weight at $\omega = 2pk/N$. The value $|U_N(\omega)|^2$ is called *periodogram* of the signal $u(t), t = 1, \dots, N$; and $|U_N(2\pi k/N)|^2$ is a measure of the energy contribution of the signal at frequency $\omega = 2pk/N$.

The energy of the signal $u(t)$, $t = 1, \dots, N$ can be determined in both the time domain and the frequency domain, using the well known and important Parseval's formula

$$\sum_{t=1}^N \left| U_N\left(\frac{2\pi k}{N}\right) \right|^2 = \sum_{t=1}^N |u(t)|^2 \quad (2.4.3)$$

2.4.2 Signal Spectra

Loosely speaking, a stationary stochastic process is a process whose average properties, such as mean value and autocorrelation function, do not change over time.

Let $\{v(t)\}$ be a stationary stochastic process (see Eykhoff, 1974, Appendix B), its autocorrelation function is defined by

$$R_v(\tau) = E v(t)v(t - \tau) \quad (2.4.4)$$

where E denotes mathematical expectation, and its power spectrum (spectral density) is defined as

$$\Phi_v(\omega) = \sum_{t=-\infty}^{\infty} R_v(\tau) e^{-i\tau\omega} \quad (2.4.5)$$

Denote $\{w(t)\}$ as another stationary stochastic process, the cross-correlation function between $\{v(t)\}$ and $\{w(t)\}$ is defined as

$$R_{vw}(\tau) = E v(t)w(t - \tau) \quad (2.4.6)$$

and the cross-spectrum between the two is

$$\Phi_{vw}(\omega) = \sum_{t=-\infty}^{\infty} R_{vw}(\tau) e^{-i\tau\omega} \quad (2.4.7)$$

By definition of the inverse Fourier transform, we have

$$E v^2(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_v(\omega) d\omega \quad (2.4.8)$$

Let $G(q)$ be a stable transfer operator (filter) and let

$$s(t) = G(q)v(t) \quad (2.4.9)$$

Then $\{s(t)\}$ is also a stationary process and

$$\Phi_s(\omega) = |G(e^{i\omega})|^2 \Phi_v(\omega) \quad (2.4.10)$$

$$\Phi_{sv}(\omega) = G(e^{i\omega}) \Phi_v(\omega) \quad (2.4.11)$$

For a given sequence (a realization), $v(t)$, $t = 1, \dots, N$, of the process $\{v(t)\}$, the autocorrelation function can be estimated by replacing the expectation by the time averaging operator:

$$\hat{R}_v(\tau) = \frac{1}{N} \sum_{t=1}^N v(t)v(t - \tau) \quad (2.4.12)$$

Then the spectrum can be calculated as

$$\hat{\Phi}_v(\omega) = \sum_{t=-\tau_m}^{\tau_m} \hat{R}_v(\tau) e^{-i\tau\omega} \quad (2.4.13)$$

with a suitable τ_m , for example, $\tau_m = N/10$. It can be shown that when $N \rightarrow \infty$ these two estimates will converge with probability one to $R_v(\tau)$ and $\Phi_v(\omega)$, provided that the process $\{v(t)\}$ is *ergodic* (an ergodic process is a process whose time average tends to its ensemble average; see Eykhoff, 1974, Appendix B). This implies that when the number of samples N is large, these estimates will be accurate. The cross-correlation function and the cross-spectrum can be estimated similarly.

For a deterministic signal $u(t)$ we can define its autocorrelation function and spectrum by

$$R_u(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N u(t)u(t - \tau) \quad (2.4.14)$$

$$\Phi_u(\omega) = \sum_{t=-\infty}^{\infty} R_u(\tau) e^{-i\tau\omega} \quad (2.4.15)$$

provided that they exist.

If the signal $\{u(t)\}$ is partly deterministic and partly random, Ljung (1987) combined the definitions (2.4.4) and (2.4.14) to give the autocorrelation function as

$$R_u(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N E u(t)u(t - \tau) \quad (2.4.16)$$

If this function exists, $\{u(t)\}$ is called a *quasi-stationary*. It can be shown (Ljung, 1987) that the fundamental relations (2.4.10) and (2.4.11) are also valid for quasi-stationary processes. The introduction of the principle of quasi-stationary is important for the analysis in process identification. In practice, however, (2.4.12) and (2.4.13) can always be used to estimate $R_u(\tau)$ and $\Phi_u(\omega)$ without bothering about whether it is deterministic or random.

It can be shown that every stationary stochastic process can be generated by passing a white noise through a stable minimum-phase filter; see Åström and Wittenmark, 1984. This fact makes forecasting based on time series modeling possible.

Let the stationary stochastic process $\{v(t)\}$ be generated by filtering a white noise sequence

$$v(t) = H(q)e(t) \quad (2.4.17)$$

where $\{e(t)\}$ is a sequence of independent random variables with a certain (unknown) probability density function; it has zero mean and variance R ; $H(q)$ is a stable and minimum-phase filter (its inverse is stable) which is monic ($H(0) = 1$). For practical use, we parametrize $H(q)$ as a rational function of q . This description is often a reasonable characterization of random disturbances for practical purposes. According to (2.4.10) we have

$$\Phi_v(\omega) = |H(e^{i\omega})|^2 R \quad (2.4.18)$$

2.5 Linear Processes with Disturbances

We have discussed how to describe linear processes (or linear models of processes) in the previous sections in an extremely ideal situation, namely the noise-free case. In the modeling of industrial processes or other real life processes, there are always variables or signals beyond our control whose effects on the process cannot be neglected; typically they are measurement noises and unmeasured disturbance. For a linear process, the effects of these variables can be lumped into an additive term (or vector) $v(t)$ at the output(s):

$$y(t) = G(q)u(t) + v(t) \quad (2.5.1)$$

The disturbance vector $v(t)$ is typically not measurable, and is only noticeable through its effect on the outputs. The classical way of describing disturbances are in the forms of steps and pulses. In identification and stochastic control, the disturbances are modelled as realizations of stochastic processes as in the previous section. The input vector $u(t)$ can be considered as noise-free, because it is often the test signal and the control actions generated by the computer. Sometimes the process inputs are measured, but the effect of the measurement noise on the outputs is often much smaller than that of the disturbances. This implies that (2.5.1) is also practically valid when input is measured.

In the coming chapters for linear model identification, we will study how to obtain a model of an industrial process which has the form (2.5.1) using experimental data.

2.6 Nonlinear Models

Engineers know that most of the real world systems have nonlinear behavior and linear models are only approximates for a small range of process operation. In order to extend the validity of models and the applicability of model based control, some nonlinearity needs to be included in the model set. The difficulty in working with nonlinear models is the lack of a unified theoretical

representation of various nonlinear behavior. There are numerous ways to describe nonlinear systems. In this section we will discuss several nonlinear representations of dynamic processes and we will concentrate on those simple nonlinear models that are suitable for parametric identification schemes. However, let us start with a general description of nonlinear process.

2.6.1 Volterra Series

One of the nonlinear generalization of linear impulse response representation, or input-output convolution is the Volterra series; see, e.g., Eykhoff (1974). Given a nonlinear SISO process, its continuous-time Volterra series is

$$\begin{aligned} y(t) = & \int_0^\infty g_1(\tau)u(t-\tau)d\tau + \int_0^\infty \int_0^\infty g_2(\tau_1, \tau_2)u(t-\tau_1)u(t-\tau_2)d\tau_1 d\tau_2 + \dots \\ & + \int_0^\infty \dots \int_0^\infty g_n(\tau_1, \tau_2, \dots, \tau_n)u(t-\tau_1)u(t-\tau_2)\dots u(t-\tau_n)d\tau_1 d\tau_2 \dots d\tau_n + \dots \end{aligned} \quad (2.6.1)$$

It can be shown that this model can describe (approximately) a very general class of nonlinear time-invariant processes; see, e.g., Eykhoff (1974). The impulses of various degrees, $g_1(\tau)u(t-\tau)$, $g_2(\tau_1, \tau_2)$, ... are also called Volterra kernels. If the higher degree impulse responses are zeros, (2.6.1) reduces to a linear model.

Similarly, the discrete-time Volterra series can be written as

$$\begin{aligned} y(t) = & \sum_{\tau=0}^{\infty} g(\tau)_1 u(t-\tau) + \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} g_2(\tau_1, \tau_2)u(t-\tau_1)u(t-\tau_2) + \dots \\ & + \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_n=0}^{\infty} g_n(\tau_1, \tau_2, \dots, \tau_n)u(t-\tau_1)u(t-\tau_2)\dots u(t-\tau_n) + \dots \end{aligned} \quad (2.6.2)$$

Although the Volterra series is useful for theoretical and qualitative understanding of nonlinear dynamics, it is impractical to use in process identification, due to the excessive number of coefficients to be identified, even when using very low degree kernels. For example, if first degree impulse response $g_1(\tau)$ needs 50 coefficients to describe it, for the same resolution, $g_2(\tau_1, \tau_2)$ will need 2500 coefficients and $g_3(\tau_1, \tau_2, \tau_3)$ 125000 coefficients! The number of coefficients can be reduced by making higher degree kernels symmetrical (Eykhoff, 1974), but it is still too large to be practical, even for SISO processes. This should motivate us to search for more restricted but tractable nonlinear models.

2.6.2 Block-Oriented Nonlinear Models

Another way to generalize linear models is to connect static (memoryless) nonlinear function blocks to dynamical linear blocks. These are called *block-oriented models* or block structured models (Chen, 1995). In SISO case, the simplest such model is to put a nonlinear gain before the linear block; see Figure 2.6.1 (a). This structure is called a Hammerstein model (Narendra and Gallman, 1966), or N-L model (Chen, 1995) where N stands for nonlinear operator and L linear operator. When reversing the order of linear and nonlinear blocks, that is, putting a nonlinear gain after the linear block, the structure is called a Wiener model or L-N model; see Figure 2.6.1 (b). When putting a nonlinear block in between two linear blocks, the structure is called a sandwich model, or L-N-L model; see Figure 2.6.1(c). When putting a linear block in between two nonlinear blocks, the structure is called a Hammerstein-Wiener model, or N-L-N model (Zhu, 2001); see Figure 2.6.1 (d). Note that signals in between the linear and nonlinear blocks are, in general, not available. More complex block-oriented model structure can be obtained by parallel connection of N-L, L-N, L-N-L and N-L-N models; see Chen (1995).

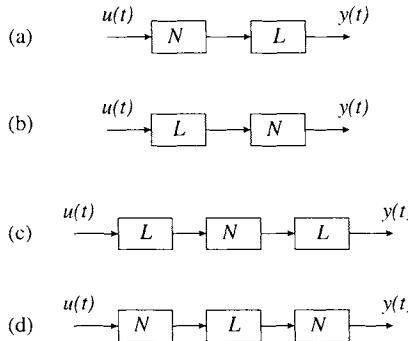


Figure 2.6.1 Block-oriented nonlinear models. (a) Hammerstein (N-L) model, (b) Wiener (L-N) model, (c) sandwich (L-N-L) model and (d) Hammerstein-Wiener (N-L-N) model.

Although this is a very restricted class of nonlinear models, the identification of block-oriented nonlinear models is more difficult than that of linear models. The difficulty starts with identification tests. In order to correctly identify the structure and parameters of a general block-oriented model, very specific test signals are necessary, for example white Gaussian noise or periodic signals; see Billings and Fakhouri (1982) and Chen (1995). These signals are often not practical to use in industrial process control. Also the test period needs to be longer than that for linear model identification. The second difficulty is to choose a good identification criterion, or, equivalently, to put the error term at a sensible location. The third difficulty is that the model

structure/order selection and parameter estimation will be more involved due to more complex models.

In this book, we will mainly treat the identification of the Hammerstein (N-L) model, the Wiener (L-N) model and the Hammerstein-Wiener (N-L-N) model in the SISO case and their simple MIMO generalizations, and we will use identification tests that are feasible in process control; see Chapters 3, and 9.

2.6.3 Nonlinear Models Motivated by Process Operation: Multiple Model Approach

Volterra models are mathematically motivated models which are impractical for use in identification. Block-oriented models are simple generalizations of linear models, hence they can only describe the process behavior in a limited operation range. From process control point of view, we can motivate other types of models that provide more global representations of nonlinear dynamics for control purposes. One way to represent a nonlinear process is to combine (or, connect) several local models each of which can adequately describe process behavior in a local range, or, operating regime. This method is called multiple model approach or operating regime-based modeling; see, e.g., Johansen (1994). We will not, in this book, treat this kind of model in detail. However, we would like to mention that linear models and block-oriented nonlinear models can be used as local models in multiple model approach.

Chapter 3

Identification Test Design and Data Pretreatment

In this chapter the problem of test design and related issues will be discussed. The basic assumption in the discussion is that the identified model will be used in control. Identification is to abstract mathematical models from data. Data are collected by observing the given process or systems. For some systems, such as economical, ecological and biological systems, the data collection is often a passive process, meaning that we can only observe the system outcomes under a given circumstance and it is too costly or technically impossible to introduce extra excitation on these systems. The data from these systems may not be informative enough. This can make the identification of these systems difficult. On the other hand, in control systems, it is most often possible to use extra excitations in order to obtain informative data for model identification. To interview a person, one needs to ask good questions. Excitations are our questions to the process to be identified. Process excitation and data collection are called identification test or experiment. The possibility of process excitation is the key factor that makes identification successful in control technology. Identification test design plays a dominant role in practical identification for control. In identification literature, much more attention has been paid to estimation algorithms than to the design of identification tests. If the tests are not performed properly then no identification algorithm, however sophisticated, can arrive at a relevant process model for control. The belief that identification is just a set of algorithms and it is simply *data-in-and-model-out* will often lead to disappointing results in identification applications.

The purpose of identification tests (for control) is to excite and to collect (control) relevant information about the process dynamics and its environment (disturbances). Often several different types of tests need to be performed, each of them for collecting certain kind of information about the process. The model is estimated from the final tests where the process inputs are perturbed by some carefully designed test signals. Usually, identification tests do not disturb normal unit operation. In this chapter we will describe a sequence of tests, where we start with very little quantitative knowledge of the process and end up with the input-output data which can be used

for deriving process models.

To carry out identification tests one needs a controller configuration with the list of input, output and feedforward variables. But controller configuration may need some preliminary process tests. In order to understand the interplay between controller configuration and identification tests, we will first discuss controller configuration before discuss various identification tests. Section 3.1 discusses the choice of process inputs and outputs are discussed. Section 3.2 will discuss several preliminary tests for collecting *a priori* information of the process. Section 3.3 introduces various type of test signals that can be used for final identification test. In Section 3.4 a concrete test procedure will be outlined that will generate data for model identification. Some related problems will also be discussed. Section 3.5 will discuss sampling frequency and anti-aliasing filtering; Section 3.6 will discuss the pre-treatment of data. Section 3.7 summarizes the chapter.

We place this chapter on test design at the beginning of the book rather than at the end. This is logical to a practitioner, because an identification work starts with tests. Because the body of identification theory has not been treated yet, often we can only tell the "hows" of the proposed methods. The reader will need to read later chapters for some of the "whys".

3.1 Controller Configuration; Selection of MV's, CV's and DV's

In process control, process inputs are also called *manipulated variables* (MV's), outputs are called *controlled variables* (CV's) and measured disturbances are called *feedforward variables* or *disturbance variables* (DV's). For a given process, these variables are determined based on control requirements and on process operation experience. Research and engineering literature can be used to find useful hints for selecting these variables for the similar processes. The following are general guidelines

3.1.1 Selection of Inputs or MV's

Some rules for the selection of the inputs for control purposes are:

1. The selected inputs (MV's) should have strong influence on (part of) the outputs (CV's) both in amplitudes and in wide frequency range.
2. Different inputs have essentially different effects on the outputs.
3. Reliable manipulation of the inputs is possible. If not, some maintenance work needs to be done before identification and controller commissioning, e.g., to replace a valve.
4. Preferably, input-output relations are almost linear or linearizable. Actuator nonlinearities such as thresholds, dead zones, hysteresis which occur in process actuators, e.g., valves and

motors, can be compensated for using primary PID controllers. The setpoints of these PID controllers will then be used as inputs (MV's) for model based control.

5. When it is not clear if a variable should be included as an input for control, due to the lack of quantitative information, then use it in the identification stage. The decision will be easier when a corresponding model is available.

The use of experience of the process operators and any available *a priori* knowledge of the process are essential for the input selection. Some pre-tests using step signals may also be needed for this purpose.

3.1.2 Selection of Feedforward Variables or DV's

Process variables that can be measured but cannot be manipulated, are the candidates for disturbance variables (DV's) or feedforward variables. During the identification stage, DV's are treated as inputs. If models between these variables and process outputs are identified, their disturbing effect can be compensated for by feedforward control. Including measured disturbance in identification will decrease the level of unmeasured disturbance, and, therefore, improve model accuracy. Some rules for selecting DV's are:

1. The speed on which the relevant CV's respond to a DV should be slower than or similar to the speed on which the CV's respond to some MV's. Feedforward is not effective if MV's cannot react promptly.
2. Delays from DV's to CV's are preferred so that the MV's can have enough time to react on them.
3. DV's should be independent from process inputs (MV's). This means that the movement of any input variable should not cause any DV to be moved. The independence can often be determined using process knowledge.
4. Slow disturbances can be used in the identification stage in order to increase the model accuracy (by decreasing the level of unmeasured disturbance). But this variable can be dropped as a feedforward variable in the controller in order to keep the controller simple. Very slow disturbance can be compensated for by feedback action.

3.1.3 Selection of Outputs or CV's

The selection of outputs or CV's are determined by control requirements and controller type. For linear time-invariant controllers, such as LQG and H-infinity, the number of inputs (MV's) should be greater than or equal to the number of outputs (CV's). For MPC control, this requirement

is not necessary due to range control, control with priorities and constraint handling. In fact, in MPC control applications, the number of inputs (MV's) is often less than that of outputs (CV's).

Suppose that all the variables to be controlled can be easily measured, then the selection of outputs is done based on the control requirements and chosen controller type, which is easy.

3.1.4 Inferential modeling

In practice, however, due to the limitations of sensors and instrumentation, some outputs which need to be controlled are difficult to measure or it is too costly to use them for feedback control. Examples are thermal couples for measuring very high temperatures (easily wearing out) and analyzers for measuring compositions (too slow). In this case, identified models can be used to predict these variables, which is called inferential modeling. The output of the inferential model is the output to be predicted; the inputs of the inferential model can be the process inputs (to be used by the controller) or some other process variables that are neither inputs nor outputs. During the identification test, the output is measured and data are collected for model identification. In control, the output is calculated using the inferential model. The measurement of the output is only used to adjust the model.

If the model is to be used for fault diagnosis, one also needs to take care that the phenomena to be detected are included in certain channels of the selected inputs and outputs.

3.1.5 Examples of Controller Configuration

The three processes introduced in Chapter 1 will be discussed.

The Single Stand Rolling Mill

As mentioned before, the exit thickness is affected by the following variables: the drawing force of the electrical motor, the roll gap, the rolling speed and the input thickness. The first three can be used as control variables, while the input thickness is a measured disturbance and its variation is an important cause of the variation of exit thickness. Other causes of thickness variation are strip hardness variation and eccentricities of various rolls. These are not measured and thus will be considered as unmeasured disturbances. It was decided to use the drawing force and the roll gap as control inputs and keep the rolling speed constant; the input thickness cannot be manipulated and will be used for feedforward control. So for the feedforward and feedback control of the process, a 3-input 1-output model needs to be identified; see Figure 3.2.1.

We will revisit this process in Chapter 4.

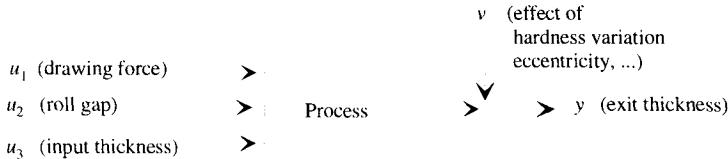


Figure 3.1.1 Control configuration for the rolling mill

The Glass Tube Drawing Process

The wall thickness and diameter of the tube, are the most important quantities to be regulated; hence they are the outputs of the process. The mandril gas pressure and the drawing speed can affect the wall thickness and diameter most directly and easily; so they are good candidates for control inputs. Other variables, such as the power supplied to melt the glass, the pressure in the melting vessel and the room temperature, will be considered as disturbances. Therefore the process can be modelled as a 2-input 2-output process with disturbances. Let us name the input-output variables as follows:

- $u_1(t)$: input 1, gas pressure setpoint;
- $u_2(t)$: input 2, drawing speed setpoint;
- $y_1(t)$: output 1, average wall thickness;
- $y_2(t)$: output 2, average diameter.

We will revisit this process in Chapter 4.

The Distillation Column

Selection of inputs (MV's). According to distillation column operation, the following three variables have strong influence on the top and bottom product qualities: reflux, reboiler steam valve position and overhead pressure. Roughly speaking, increasing the reflux will decrease top product C5, increase bottom product RVP and increase flooding; increasing the reboiler steam valve opening will bring more heat to the column and in turn will increase top product C5, decrease bottom product RVP and increase flooding; an increase in column pressure will increase both to C5 and bottom RVP and decrease flooding.

A comment should be made about the steam valve position. It is well known that the relation between the valve opening and the flow is nonlinear. Therefore, it is better to first add a steam flow controller at the regulatory control level and use the setpoint of the flow as the input for the model based control. This will reduce the nonlinearity between process inputs and outputs. An even better choice is to configure a cascade duty controller that moves the steam flow. It is the duty (heating power) that heats up the column and determines the operation condition. The

advantage of duty control is the compensation of variations in steam temperature and steam pressure.

Suppose that time and money have been spent on configuring the steam flow controller and the duty controller. Then the following will be a good choice of process inputs (MV's): 1) reflux flow, 2) reboiler duty, and 3) pressure.

Selection of feedforward variables (DV's). The main disturbances that cause product quality variations are the following:

- Feed composition variations, not measured.
- Feed flow variations, measured.
- Temperature and pressure variations of the reboiler steam.
- Atmosphere temperature and humidity variations, not measured.

Feed composition is not measured due to the high cost of analyzers. If the steam flow setpoint is used as the process input, then steam temperature and pressure can be used as feedforward variables. When the reboiler duty setpoint is used as the input, the variations of steam temperature and pressure will be compensated by the duty controller and, therefore, they can be dropped from the candidates. The measurement for atmosphere temperature and humidity are not very costly. However, their variations are, in general, very slow and their influence on the process can be compensated for by feedback control.

This will only leave feed flow as feedforward variable.

Selection of process outputs (CV's) for linear control. For linear controllers, the number of process outputs (CV's) should be less than or equal to that of the process inputs (MV's). This implies that we can only have up to three outputs. The most usual choice of the outputs to meet the control requirements are: 1) top product C5, 2) bottom product RVP and 3) flooding.

The configuration of the linear controller is summarized in the following table.

Inputs (MV's)	Constraints control requirements
Reflux flow	Amplitude limits: 40 - 80 M ³ /H
Reboiler duty	Control output limits (reboiler valve): 0 - 100%
Overhead pressure	Amplitude limits: 14 -16 bar; maximize pressure control output limits (vapor valve): 0 - 100%
Feedforward (DV)	
Feed flow	
Outputs (CV's)	Constraints and control requirements
Top product C5	High limit: 15%mol; maximize C5
Bottom product RVP	High limit: 0.5 bar; minimize RVP
Flooding	High limit: 75%

It is obvious that the given controller configuration cannot meet the control requirements as listed above. More precisely, the linear controller can only fulfill quality control by controlling the three CV's at their setpoints and decoupling. The constraint control and optimization can only be done with tricks that may not always work. Feedforward control will also cost additional work. These limitations explain partly why linear controllers are not widely applied in process industries.

Selection of process outputs (CV's) for MPC control. This time, instead of a linear controller, an MPC controller will be configured to meet the same control requirements. In general, the selection of inputs and feedforward variables are the same for linear control and MPC control. The difference lies in the selection of outputs. The following table summarizes an MPC controller configuration.

Inputs (MV's)	Constraints control requirements
Reflux flow	Amplitude limits: 40 - 80 M ³ /H
Reboiler duty	Control output is a CV
Overhead pressure	Amplitude limits: 14 - 16 bar; maximize
Feedforward (DV)	
Feed flow	
Outputs (CV's)	Constraints and control requirements
Top product C5	High limit: 15%mol; maximize C5
Bottom product RVP	High limit: 0.5 bar; minimize RVP
Flooding	High limit: 75%
Reboiler valve	Low and high limits: 0 - 100%
Vapor valve	Low and high limits: 0 - 100%

Note that the two MV control outputs, reboiler valve and overhead vapor valve which represent process operation constraints, are added as CV's and their limits will be respected by the MPC controller. As discussed before, an MPC controller can meet all the control requirements using multi-objective optimization and range control.

3.2 Preliminary Process Tests

The design of a good identification test needs some *a priori* knowledge of the process such as the dominant time constant, bandwidth of the dynamics, nonlinearity and disturbance characteristics. Some of this knowledge can be obtained from process unit operation experiences. Thus good communication between control engineers and operators is very important for acquiring process knowledge. Suppose that very little is known, then the following preliminary tests can be used to obtain relevant information for the final identification test design. Note that the second role of preliminary tests is to verify if the given controller configuration is appropriate. It is possible that the control configuration will be modified after some pre-tests and, sometimes, even after the final identification test.

3.2.1 Collecting Historical Data or Free-Run Tests

In this step the data of the given process under normal operation is collected without intentionally introducing disturbance (excitation) to it. The purpose here is to get a first look into the given process. Try to collect the data during different operation situations. One situation is the open-loop operation without the intervention of the operator. The process outputs are measured over such a long period of time that the statistical properties of the measured output signals no longer show significant change. This type of data is directed to finding the characteristics of the disturbances on the process outputs. The range of the amplitudes of these signals and their frequency spectra are important information for both the identification and control of the process. This step is technically easy and economically cheap. Often the required data already exit in a database.

The list of data collection should include at least the following signals:

- 1) all the candidate process inputs (MV's), usually they are the setpoints of the regulatory controllers;
- 2) all the candidate MV process values (measured values);
- 3) all the candidate MV controller outputs, usually they are the valve positions;
- 4) all the candidate disturbances (measured values);
- 5) all the candidate process outputs (CV's); and
- 6) variables that carry the information of process operation.

The same data collection list should be used for all pre-tests and the final test.

3.2.2 Step Test or Staircase Test

In a step test the process is operating in open-loop without the model based controller and each input is stepped separately and step responses are recorded. The operator intervention should be kept minimum during the test. Two to three steps can be made for each input. The step size can be determined according to process operation experience and most operators are able to provide this information. The step length should be longer than the settling time of the process. Steps are mostly acceptable because they are often used by process operators for regulating the processes. A step test is used for at least three purposes:

- 1) To obtain a rough estimate of dominant process time constants, and some process gains.
This knowledge is needed in the design of the final test for model identification.

- 2) To check and to tune the regulatory controllers of various inputs (MV's). It is crucial that each MV controller works properly before commissioning a MIMO controller. If an MV controller is not functioning well, one need to retune the controller or even replace the actuator (e.g., a sticking valve).
- 3) A control engineer can learn the process dynamics intuitively by watching the step responses.

When the nonlinearity of the process needs to be checked, the staircase test can be used. Staircase test signals are applied to the selected candidate process inputs; see Figure 3.2.1. Again the process is operating in open-loop. The time interval of one stair should allow the process to reach its steady state. This type of test is directed to testing the range of linearity of the process, to finding the static gains and to obtaining an estimation of the dominant time constant.

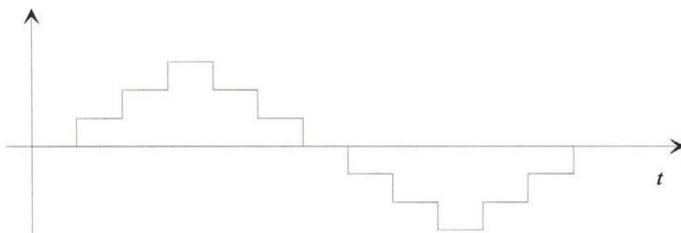


Figure 3.2.1 Staircase test signal

Note that, in general, staircase tests are much more costly to carry out than step tests. A staircase test will take much longer time than a step test for a given process; it will introduce a much larger disturbance to the process than a step test will do. It is possible that staircase is not permitted for some processes due to technical and economical reasons. For example, to carry out a step test on a large scale distillation column will take several days without disturbing normal operation; to carry out a staircase test will take several weeks and product qualities can be affected.

3.2.3 White Noise Test, Optional

White noise tests are for determining the process bandwidth and time delays. In this type of test, the inputs are supplied with mutually independent *white noise* signals. A discrete-time white noise $u(t)$ is a sequence of independent and identically distributed random variables of zero mean and variance R . Thus the autocorrelation function of the white noise signal is:

$$R^u(\tau) = E[u(t)u(t - \tau)] = \begin{cases} R & \text{for } \tau = 0 \\ 0 & \text{for } \tau \neq 0 \end{cases} \quad (3.2.1)$$

and its power spectrum density is constant over all frequencies. According to formula (2.4.11), we can estimate each frequency response by

$$\hat{G}_{ij}(e^{i\omega}) = \frac{\hat{\Phi}_{y_i u_j}(\omega)}{\hat{\Phi}_{u_j}(\omega)} \quad (3.2.2)$$

where $\hat{\Phi}_{y_i u_j}(\omega)$ and $\hat{\Phi}_{u_j}(\omega)$ are the estimated cross-spectrum between $u_j(t)$ and $y_i(t)$ and the auto-spectrum of $u_j(t)$. Because white noises have broad frequency bands and the disturbances usually concentrate in the low and middle frequency band, the estimates of the frequency responses are accurate at high frequencies. From these estimates the bandwidth, or equivalently the smallest time constant, of the process can be determined. The bandwidth of the open-loop process is used to determine the sampling frequency of the final identification experiment which is also the sampling frequency of computer controlled system. This frequency is called the *working frequency*.

It is known that when the inputs of a linear process are independent white noises, the cross-correlation functions between inputs and outputs are the impulse responses of the corresponding input-output transfer function. From this fact we can estimate the delays of the various input-output transfers from the computed cross-correlation functions.

The white noise signals can be approximated by PRBS (pseudo random binary sequence) or GBN (generalized binary noise); see Section 3.3.

The sampling frequency of the white noise experiment can be higher than the *working frequency* that is used for the modeling and control. On the other hand, if a white noise is applied to a slow process with too high a sampling frequency, there will be hardly any response from the process. So some trade-off is required.

In general, carrying out white noise tests are not expensive. The test signal is very fast so the disturbance to the operation will be small. The test period need not be too long, because only high frequency information is searched for. White noise tests can be avoided if one knows the desired speed (bandwidth) of the controlled system, or when the speed of setpoint tracking is not of primary importance.

3.3 Test Signals for Final Test, Persistent Excitation

An advantage and convenience of identification for control is that one can introduce test signals, or extra excitations during an identification test. By test signals we mean the signals generated by the user and supplied to the process during the final identification test. In open-loop tests, the process inputs are the same as the test signals, except for the offsets. In closed-loop tests, however, the inputs are the sums of test signals and the feedback control actions. Two aspects are important in selecting test signals in process control, the first is the shape or waveform of the signal and the second is its power spectrum or frequency content.

Signal waveform. In order have data with high signal-to-noise ratios, one needs to have as much signal power as possible. In practice, on the other hand, the test signal amplitudes are constrained because 1) they should not cause upset in normal process operation and 2) they should not excite (too much) nonlinearity (for linear model identification). Therefore, for a given signal power, a small amplitude is desirable. This property can be expressed in terms of crest factor C_r . Denote a signal with zero mean as $u(t)$, its crest factor is defined as (Ljung, 1999)

$$C_r = \sqrt{\frac{\max_t u^2(t)}{\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N u(t)}} \quad (3.3.1)$$

A good signal shape needs to have a small crest factor. Normally used signals are binary signals, filtered white noises and multiple sinusoids. Binary signals have the smallest crest factor which is 1.

Signal frequency content. It can be shown that the model error distribution in the frequency domain is affected by the spectrum of the test signal; see Ljung (1987) and Chapters 4, 5 and 6. Therefore, the test signal spectrum can be used as a design variable that can be optimized with respect to control performance; see Chapter 6.

3.3.1 Test Signals for Model Identification

Four types of test signals will be discussed in the sequel:

- Pseudo random binary sequence (PRBS) (see Eykhoff, 1974 or Söderström and Stoica, 1989).
- Generalized binary noise (GBN) (Tulleken, 1990)
- Filtered white noise or autoregressive moving average (ARMA) process.
- Sum of sinusoids.

In practice, most often test signals are added to the inputs (MPs) of the open-loop process and/or at the setpoints of the closed-loop system in order to move the process up and down around a working point. Therefore, the mean value of test signals should be zero or close to zero.

Pseudo-Random-Binary-Sequence (PRBS)

A pseudo-random-binary-sequence (PRBS) is a two state signal which can be generated by using a feedback shift register as shown in Figure 3.3.1 (Söderström and Stoica, 1989); where n denotes

the number of registers or of states and \oplus denotes modulo-two addition. The register variables are fed with 1 or 0; and the initial state vector is non zero. When the clock pulse is applied, the value of the k th state goes to the $(k + 1)$ th state and the first state value determined by the feedback path. The feedback coefficients, a_1, \dots, a_n , take the value of 0 or 1. The clock period is denoted as T_{clk} which is often equal to the sampling time. Thus the shift register will generate a sequence of ones and zeros. This is called pseudo-random-binary-sequence (PRBS) because it is a deterministic signal and its autocorrelation function can resemble the autocorrelation function of a white random noise.

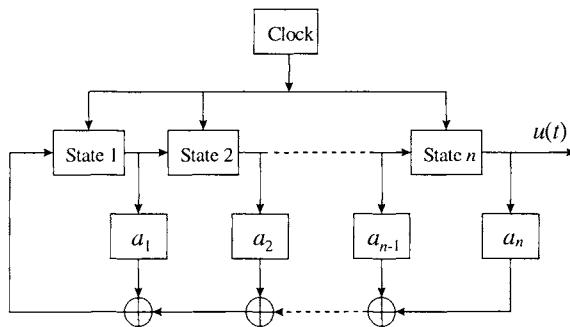


Figure 3.3.1 Shift register for generating PRBS signal

The maximum possible period of a PRBS is $M = 2^n - 1$, because the all-zero state must not be included. A PRBS with period equal to M is called a maximum length PRBS or an m -sequence. The feedback coefficients, a_1, \dots, a_n , determine whether or not a PRBS is maximum length; see Söderström and Stoica (1989).

It is the maximum length PRBS that approximates the white noise signal for use in identification tests. Let $u(t)$ be a maximum length PRBS with period $M = 2^n - 1$ where n is the number of registers. This signal has two amplitudes $-a$ (when the register output is 0) and a (when the register output is 1); see Figure 3.3.2. Assume that the sampling time equals the clock period T_{clk} , then the following can be shown (Söderström and Stoica, 1989 and Ljung, 1987):

- The mean value of $u(t)$ is

$$\frac{a}{M} \quad (3.3.2)$$

- The autocorelation function of $u(t)$ is

$$\begin{aligned} R_u(\tau) &= a^2 & \tau = 0, \pm M, \pm 2M, \dots \\ &= -\frac{a^2}{M} & \text{else} \end{aligned} \quad (3.3.3)$$

- The power spectrum of $u(t)$ is

$$\Phi_u(\omega) = \frac{2\pi a^2}{M} \sum_{k=1}^M \delta(\omega - 2\pi k/M), \quad 0 \leq \omega < 2\pi \quad (3.3.4)$$

It can be seen that when the clock period T_{clk} equals the sampling time, and when M is a large number, a PRBS signal can approximate a white noise signal with zero mean. Such signals have maximum power for a given amplitude, in other words, they have minimum crest factor. Note that generating a part of a period of a PRBS will not give a signal with the above properties. This makes the selection of test time less flexible.

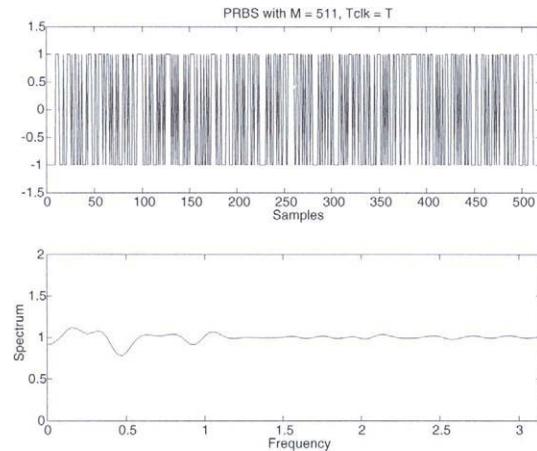
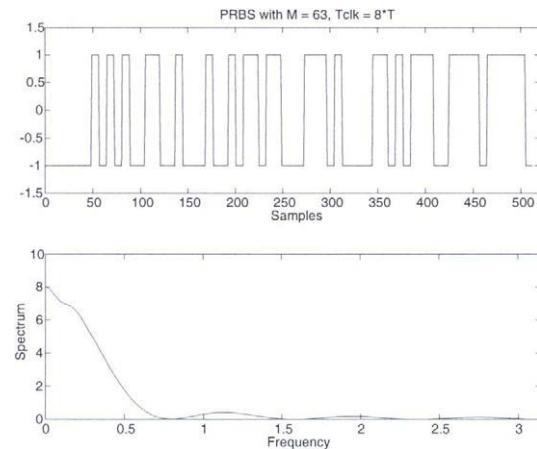
White-noise-like PRBS signals have been traditionally used for nonparametric model identification, such as frequency response estimation and correlation analysis. As will be shown later, white noise signals are not the best test signals for parametric model identification. In a process control environment, a white noise signal will over-emphasize the high frequency band at the cost of the low and middle frequency band; its frequent fluctuations can be harmful for control actuators. Very often, a good test signal for processes control has a low-pass character, meaning that its power spectrum looks like a low-pass filter. Such a signal can be obtained by filtering a white noise signal or a PRBS signal. A low-pass signal can also be obtained by increasing the clock period T_{clk} of the signal. Then the power spectrum has an envelop of the form

$$T_{clk} \left(\frac{\sin(\omega T_{clk}/2)}{\omega T_{clk}/2} \right)^2 \quad (3.3.5)$$

One disadvantage of using a PRBS signal this way is that its spectrum has dips around frequencies $2\pi/T_{clk}, 4\pi/T_{clk}, 6\pi/T_{clk}, \dots$, which will result in low signal-to-noise ratios in these frequency ranges; see Figure 3.3.3. A better way to generate binary signals with low-pass character is the so-called generalized binary sequence (GBN) which will be discussed next.

Exercise 3.3.1 Assuming the clock period equals the sample time, create a PRBS signal with $M = 511$ using Matlab's System Identification Toolbox, calculate its power spectrum and plot the results. Assume $T_{clk} = 4T$ where T is the sampling time, calculate the spectrum of a PRBS with $M = 117$ and plot it.

Answer. The PRBS signal and its spectrum are shown in Figure 3.3.2 and Figure 3.3.3. Note that the spectra are not exactly the same as given in the Formulas (3.3.4) and (3.3.5) due to finite data effect. The Matlab M file for the exercise is as follows.

Figure 3.3.2 PRBS signal with $M = 511, T_{clk} = T$ Figure 3.3.3 PRBS signal with $M = 63, T_{clk} = 8T$

```
%%% prbssignal.m; Matlab M file for studying PRBS signals %%%
% Create PRBS with M = 511 and Tclk = T
PRBS1 = idinput(512,'PRBS',[9,1]);
% Create PRBS with M = 117 and Tclk = 8*T
PRBS2 = idinput(512,'PRBS',[6,1/8]);
```

```
% Calculate the spectra
w=[0:100]'/100*pi;
FI1 = spa(PRBS1,50,w);
FI2 = spa(PRBS2,50,w);
% Plot PRBS1 and its spectrum
figure;
subplot(211),plot(PRBS1);
title('PRBS with M = 511, Tclk = T');
xlabel('Samples');
axis([0 520 -1.5 1.5]);
subplot(212),plot(w,FI1(2:102,2));
ylabel('Spectrum');
xlabel('Frequency');
axis([0 pi 0 2]);
% Plot PRBS2 and its spectrum
figure;
subplot(211),plot(PRBS2);
title('PRBS with M = 63, Tclk = 8*T');
xlabel('Samples');
axis([0 520 -1.5 1.5]);
subplot(212),plot(w,FI2(2:102,2));
ylabel('Spectrum');
xlabel('Frequency');
axis([0 pi 0 10]);
```

Generalized Binary Noise (GBN)

This signal was proposed by Tullenken (1990). The motivation was to generate a test signal that is suitable for control-relevant identification of industrial processes. A GBN signal $u(t)$ takes two values $-a$ and a . At each candidate switching time t , it switches according to the following rule:

$$\begin{aligned} P[u(t) = -u(t-1)] &= p_{sw} \\ P[u(t) = u(t-1)] &= 1 - p_{sw} \end{aligned} \quad (3.3.6)$$

where p_{sw} is switching probability. The distribution for the event at each switching time is an independent alternative distribution with parameter p_{sw} . Because of this, the GBN has zero mean. Define minimum switching time T_{min} as the time (in samples) to keep the signal constant and switching time T_{sw} as the elapsed time (in samples) between two switches, then the mean switching time is determined by

$$ET_{sw} = \frac{T_{min}}{p_{sw}} \quad (3.3.7)$$

and GBN power spectrum is given as

$$\Phi_u(\omega) = \frac{(1 - q^2)T_{\min}}{1 - 2q \cos T_{\min}\omega + q^2}; \quad q = 1 - 2p_{sw} \quad (3.3.8)$$

see Tullenken (1990).

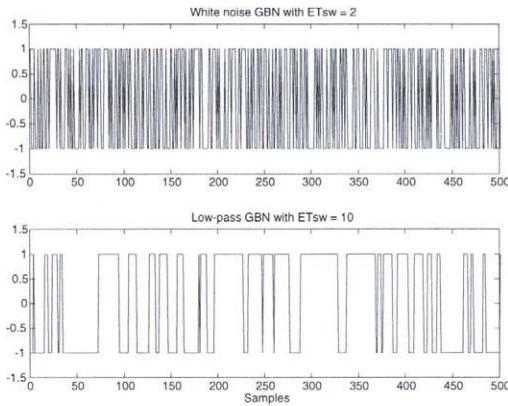


Figure 3.3.4 GBN signals

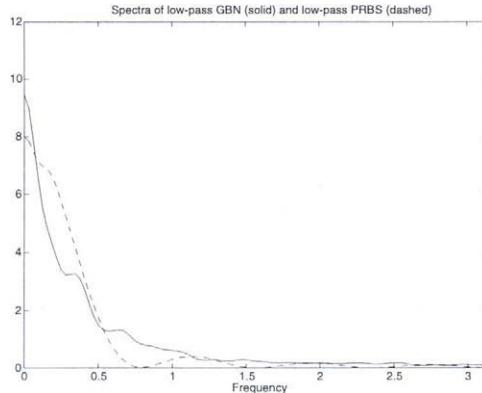


Figure 3.3.5 Compare the power spectra of low-pass GBN and low-pass PRBS

One can see that a white noise GBN is obtained when $p_{sw} = 1/2$. In this case $ET_{sw} = 2T_{\min}$. As mentioned before, white noise signals are not a good choice of test signals. Low-pass GBN signals

are generated by reducing the switching probability p_{sw} , or, equivalently, increasing the mean switching time ET_{sw} . Note that unlike the low-pass PRBS signal, the spectrum of a low-pass GBN signal does not have dips at frequencies $2\pi/T_{clk}$, $4\pi/T_{clk}$, ..., which is advantageous; see Figure 3.3.5. Another advantage with GBN is that the signal length is flexible. The GBN also has a minimum crest factor.

GBN signals can be generated using realizations of independent uniformly distributed random variables in the set [0, 1]. The following is a Matlab M file for generating GBN signals with $T_{min} = 1$.

```
%%% gbnen.m; Matlab M file for generating GBN signal %%%
function U = gbnen(N,Tsw,Seeds)
%
% U = prbs(N,Tsw,Seeds)
%
% This function generates GBN signal U in [-1 1] with Tmin = 1
%
% Input arguments:
% N : Number of samples
% Tsw : Average switching time in samples
% Seeds : Seeds for random generator
%
if nargin < 2
error('Not enough input arguments, try again')
end
% Determine switching probability
psw = 1/Tsw;
if nargin > 2
rand('seed',Seeds);
end;
R=rand(N,1);
%Determinetheinitialvalue
if R(1)>0.5,
P_M=1;
else
P_M=-1;
end;
U=zeros(N,1);
for k=1:N,
if (R(k)<psw)
P_M=-P_M;
end;
U(k)=P_M;
```

```
end
```

By visual inspection, one notes the similarity between a PRBS signal and a GBN signal in the time domain.

Exercise 3.3.2 Create a white noise GBN signal with 1000 samples and a low-pass GBN with $ET_{sw} = 10$ samples using file `gbngen.m`. Plot the signals. Also compare the power spectrum of the low-pass GBN with that of low-pass PRBS of the previous exercise.

Answer. The GBN signals are shown in Figure 3.3.4. The spectra of the low-pass GBN and of the low-pass PRBS are shown in Figure 3.3.5. The Matlab M file for the exercise is as follows.

```
%%% gbnexercise.m; M file for GBN signal exercise %%%
% Create white noise GBN with N = 1000 and ETsw = 2 (psw = 1/2)
GBN1 = gbnengen(1000,2);
% Create low-pass GBN with N = 1000 and ETsw = 10 (psw = 1/10)
GBN2 = gbnengen(1000,10);
% Plot GBN1 and GBN2
figure;
subplot(211),plot(GBN1);
title('White noise GBN with ETsw = 2');
axis([0 500 -1.5 1.5]);
subplot(212),plot(GBN2);
title('Low-pass GBN with ETsw = 10');
axis([0 500 -1.5 1.5]);
xlabel('Samples')
% Create PRBS with M = 117 and Tclk = 8*T
PRBS2 = idinput(512,'PRBS',[6,1/8]);
% Calculate the spectra of GBN2 and PRBS2
w=[0:100]'/100*pi;
FIgbn2 = spa(GBN2,50,w);
FIprbs2 = spa(PRBS2,50,w);
% Plot the spectra of GBN2 and PRBS2
figure;
plot(w,FIgbn2(2:102,2),'-',w,FIprbs2(2:102,2),'--k');
xlabel('Frequency');
title('Spectra of low-pass GBN (solid) and low-pass PRBS (dashed)');
axis([0 pi 0 12]);
```

Filtered White Noise (Colored Noise)

Because of the limited flexibility of power spectrum of GBN and PRBS signals, it is not always possible to realize a desired distribution of frequency contents using these signals. This problem

can be solved using filtered white noise. From Chapter 2 we know that when a signal is generated by filtering a zero mean white noise sequence $e(t)$ with variance λ

$$u(t) = F(q)e(t) \quad (3.3.9)$$

where $F(q)$ is a stable filter, its power spectrum is determined by

$$\Phi_u(\omega) = |F(e^{i\omega})|^2 \lambda \quad (3.3.10)$$

Therefore, any distribution of frequency contents can be approximated arbitrarily well by a proper design of the filter $F(q)$. The amplitude distribution of white noise is not crucial for generating colored noise; see Figure 3.3.6.

The amplitude distribution of a filtered white noise is continuous which is better for approximating a nonlinear process. The crest factors of this kind of signals are larger than that of binary signals.

Exercise 3.3.3 Create a colored noise using a 1st order filter with the pole at 0.9 and Gaussian white noise with unit variance. Do the same using a white noise GBN. Then normalize the signals by making them zero mean and dividing by their standard deviations. Finally, calculate their crest factors.

Answer. The two signals are shown in Figure 3.3.6. The Matlab M file for the exercise is as follows.

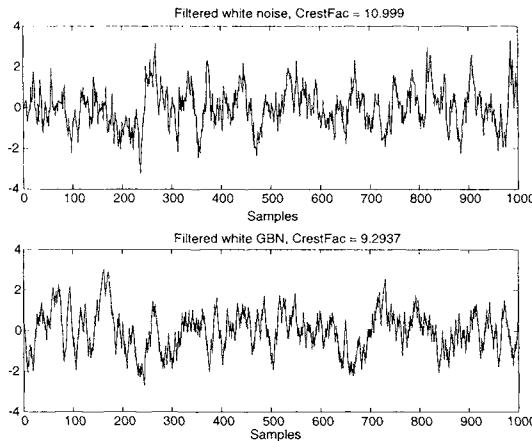


Figure 3.3.6 Colored noises created by filtering white noises

```
%%% colorednoise.m; M file for generating colored noise %%%
```

```
% Create white Gaussian noise, N = 1000
WIT = randn(1000,1);
% Create white noise GBN with N = 1000 and ETsw = 2 (psw = 1/2)
GBN1 = gbn1(1000,2);
% Filter the signals and normalize them
WITf = filter(1,[1 -0.9],WIT);
WITf = (WITf - mean(WITf))/std(WITf);
GBN1f = filter(1,[1 -0.9],GBN1);
GBN1f = (GBN1f - mean(GBN1f))/std(GBN1f);
% Calculate crest factors
FacWITf = max((WITf).^2)/cov(WITf);
FacGBN1f = max((GBN1f).^2)/cov(GBN1f);
% Plot the signals
figure;
subplot(211),plot(WITf);
title(['Filtered white noise, CrestFac = ',num2str(FacWITf)]);
axis([0 1000 -4 4]);
xlabel('Samples')
subplot(212),plot(GBN1f);
title(['Filtered white GBN, CrestFac = ',num2str(FacGBN1f)]);
axis([0 1000 -4 4]);
xlabel('Samples')
```

Multiple Sinusoids

Consider the sum of m sinusoids

$$u(t) = \sum_{j=1}^m a_j \sin(\omega_j t + \varphi_j); \quad 0 < \omega_1 < \omega_2 < \dots < \omega_m < \pi \quad (3.3.11)$$

Its spectrum is given as

$$\Phi_u(\omega) = 2\pi \sum_{j=1}^m \frac{a_j^2}{4} [\delta(\omega - \omega_j) + \delta(\omega + \omega_j)] \quad (3.3.12)$$

Assume that the frequencies ω_j are uniformly distributed in $[0, \pi]$ and that the number m is large enough. Then any desired distribution of frequency contents can be realized by setting the amplitudes a_j properly.

The crest factor of a multi-sines can be large. Assuming that all amplitudes a_j are equal, the power of the signal is $ma^2/2$. If all sinusoids are in phase, the squared amplitude will be $(ma)^2$, and the crest factor is thus $\sqrt{2m}$ which will be large when m is large. The crest factor can be reduced by setting the phases φ_j so that the sines are out of phase as much as possible. The

crest factor of a multi-sines can be minimized using some optimization schemes; see e.g., van der Ouderaa *et. al.* (1998). A simple solution has been developed by Schroeder (1970). The so-called *Schroeder phase signal* is created when the amplitudes are equal and the phases are spread as follows:

$$\begin{aligned}\varphi_1 &= \text{arbitrary} \\ \varphi_j &= \varphi_1 - \frac{j(j-1)}{m}\pi; \quad 2 \leq j \leq m\end{aligned}\tag{3.3.13}$$

Another simple way is to choose phases randomly.

Comments on test signals for nonlinear model identification. Not all the above mentioned signals are good for nonlinear model identification. Suppose that the process has a Hammerstein model with static nonlinearity at the input. If binary input is used, it will not be possible to detect the nonlinearity. Therefore, multi-amplitude or continuous amplitude signals should be used to excite the nonlinearity. To this end, filtered white noises, multi-sines and staircase signals can be used.

3.3.2 Persistent Excitation

In order to guarantee that the estimation algorithms have unique solutions, some minimum requirement should be imposed on the test signals. This is called a *persistent excitation* condition.

As it will be shown in Chapter 4, the formula for estimating a finite impulse response model of order n involves the inversion of the $n \times n$ matrix

$$\frac{1}{N} \sum_{t=1}^N \varphi^T(t) \varphi(t)\tag{3.3.14}$$

where

$$\varphi(t) = [u(t-1) \quad u(t-2) \quad \cdots \quad u(t-n)]$$

The estimation problem will have a unique solution or the process is *identifiable* if and only if the matrix (3.3.14) is invertible or non singular. This motivates the following definition of persistent excitation:

A discrete-time signal $u(t)$ is said to be *persistently exciting* of order n if the following limit exists:

$$R_u(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N u(t)u(t-\tau)\tag{3.3.15}$$

and the matrix

$$\begin{bmatrix} R_u(0) & R_u(1) & \cdots & R_u(n-1) \\ R_u(-1) & R_u(0) & & \vdots \\ \vdots & & \ddots & \\ R_u(1-n) & \cdots & & R_u(0) \end{bmatrix} \quad (3.3.16)$$

is nonsingular. Note that the matrix (3.3.14) tends to matrix (3.3.16) as $N \rightarrow \infty$.

The concept can also be extended to other model forms. It can be shown that the identifiability of an n th order linear process

$$G(q) = \frac{b_1 q^{-1} + \cdots + b_n q^{-n}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}}$$

is that the test signal is persistently exciting of order $2n$ (Ljung, 1987, Söderström and Stoica, 1989). Note that this statement holds for both open-loop tests and closed-loop tests. In an open-loop test, the test signal equals process input; in a closed-loop test, the test signal is added at the setpoint or at the input and it is not equal to the process input. For a closed-loop test, it is important to remember that it is the persistent excitation of the test signal that guarantees the identifiability, not the persistent excitation of the process input.

The frequency domain interpretation of persistent excitation of order n is that the spectrum of the signal is nonzero in at least n frequencies in the interval $(-\pi, \pi)$ (Ljung, 1987, Söderström and Stoica, 1989). Based on this property we give some comments on commonly used test signals.

Pseudo random binary sequence (PRBS). Consider a PRBS of period M . It can be shown that the order of persistent excitation of the PRBS is M . When the clock time of the PRBS signal is small, which is the case when we want to approximate a white noise, M is quite large which is enough for most identification methods.

Generalized binary noise (GBN). Because a GBN has a continuous spectrum over $[0, \pi]$, it is persistently exciting with any finite order.

Filtered white noise. This type of signal has a continuous spectrum over the whole frequency range. Hence a filtered white noise is persistently exciting for any finite order.

Sum of sinusoids. It is well known that its spectrum has exactly $2m$ lines in $(-\pi, \pi)$. Thus for the identification of an n th order process, at least n sinusoids need to be injected to the process. For good noise reduction the number of sinusoids should be much greater than the process order.

Therefore, the four types of test signals discussed above are all persistent exciting signals.

3.4 Test for Model Identification, Final Test

Assume that the process is stable and the following *a priori* process knowledge has been collected by interviewing the operators and/or by some pre-tests:

1. the range of normal operation, or the high/low limits of process outputs (CV's);
2. rough estimate of (longest) process settling time;
3. the allowed amplitudes of process inputs (MV's) that will cause obvious CV movement, but will not disturb normal operation.

Based on the information, a final test design for linear model identification, that is guided by identification theory and based on considerable industrial project experience, can be done as follows (Zhu, 1998). Theoretically optimal design will be studied in Chapters 6 and 7.

3.4.1 An Open-Loop Test Design for Control

Duration of one test. The test time is set in the range of 6 to 18 times the process settling time. The minimum length of test time is necessary for at least two reasons: 1) to average out the effect of unmeasured disturbance and 2) to make use of available theoretical results of identification which are often asymptotic in the number of data N (assuming $N \rightarrow \infty$). The test time can be short (5 to 8 times the settling time) if the number of inputs is small and if the signal-to-noise ratio is high. The test time should be long (14 to 18 times the settling time) if there are many MV's and the signal-to-noise ratio is low. Of course, if the process is linear and almost noise free, the test can be much shorter (1 to 2 times the settling time). Roughly speaking, quadrupling the test time will also halve the model error; see Chapter 6.

Signal shape and amplitudes. GBN signals are used. An additional advantage in practice is that operators are familiar with binary test signals. The amplitudes of the test signals should be chosen such that the signal-to-noise ratio is reasonably high and, at the same time, the process operation is not disturbed too much and the process is not driven out of the linear range. It can be shown that doubling the amplitude will halve the model error; see Chapter 6. Note that, if necessary, signal amplitudes can be traded with test time.

GBN mean switching time (signal spectrum). When the type and the amplitudes of the test signals are determined, it is then important to chose the proper spectral distributions of the test signals. Often the so-called *optimal input design* refers to the design of the test input spectrum (Ljung, 1987). Logically, the signal spectra should be designed in such a way that the identified model is *best for the intended use of the model*. Therefore, the test signal design will be influenced by model application. If only the error caused by disturbance (variance error)

is considered, then it is possible to derive an optimal spectrum $\Phi_u^{opt}(\omega)$ for a single variable open-loop test

$$\Phi_u^{opt}(\omega) \approx \mu \sqrt{\Phi_u^c(\omega)\Phi_v(\omega)} \quad (3.4.1)$$

where $\Phi_v(\omega)$ is the spectrum of disturbance, $\Phi_u^c(\omega)$ is the spectrum of the control signal and μ is a constant adjusted so that the input power is constrained. In plain words, the formula tells us that the power of the test signal should be high at frequencies where the power of the control signal is high, in order to mimic the situation where the model is used; the power should also be high when the disturbance power is high, in order to counteract on it. Guided by this formula, we have used the following design rule for the mean switching time of the GBN signal

$$ET_{sw} = \frac{98\% \text{ settling time}}{3} \quad (3.4.2)$$

The reasoning behind this formula is that the spectra of both disturbance and the control signal are determined by the bandwidth of the process. The factor 1/3 is obtained from simulation exercises and project experience. Using this formula the mean switching time is a simple function of process settling time which is assumed to be known from pre-test or from operation experience. Note that the GBN signal designed using this formula will be much slower than a white noise GBN. For example, assume that the settling time of a distillation column is 120 minutes, then the mean switching time of the GBN will be 40 minutes which is 20 times slower than white noise GBN!

Number of inputs (MV's) per test. Under the above test conditions, up to 10 MV's can be moved in one test. This is a rule of the thumb that is based on many industrial projects. The number MV's per test can be larger if one uses a longer test time.

Correlations between test signals. Normally, one can use independent GBN signals for MV's to be tested. However, the independence between MV's is neither necessary for the identifiability, nor is it optimal. A necessary identifiability condition is that neither of the two MV's are linearly dependent (100% correlated). For identifying an ill conditioned process such as a high purity distillation column, some correlation between certain MV's, not 100% of course, will be beneficial; see Chapter 10.

Current industrial practice of identification often uses the open-loop single variable step test approach. The biggest problem of this test method is its high cost. It will cost about 20 days to test a refinery crude unit. Table 3.4.1 shows our multivariable test designs for some typical refinery/petrochemical processes using GBN signals. Compared to the conventional single variable step test, our design can save about 70% test time! Now, it will cost 4 to 6 days to test a crude unit. Over 50 processes have been identified using the multivariable test method with success; see, e.g., Zhu (1998).

Processes	Mean switching time	Duration of one test
Main fractionator	40 - 60 minutes	40 - 50 hours
Atmospheric tower	40 - 60 minutes	40 - 50 hours
FCC react/regen	15 - 20 minutes	15 - 20 hours
Stabilizer, depropanizer, ...	30 - 50 minutes	30 - 40 hours

Table 3.1 Test designs for some processes using GBN signals

Remark. This test approach is for use with parametric model identification. For the identification of the nonparametric FIR model, a longer test time is necessary; see Chapters 6 and 7.

This design method has been derived partly from control engineering common sense and partly based on the optimal design formula (3.4.1). Test for ill-conditioned processes will be treated in Chapter 10.

3.4.2 Comments on Closed-Loop Test

In practice, it is often desirable or even necessary to carry out identification tests in closed-loop operation. For some processes, when some controllers are already in place, it will cause production loss or even safety problems if the loops are open. In this case, one has to carry out identification tests in closed-loop, or, not test at all. Even when open-loop testing is permitted, there are many advantages to closed-loop testing:

- **Reduce disturbance to process operation.** When a multivariable open-loop test is used, some of the CV's may drift away and the operator needs to intervene in order to prevent product qualities from off-specification. In a closed-loop test, however, one can specify the amplitude of the setpoint movement and the controller will help to keep the CV's within their operation limits.
- **Easy to carry out.** This is related to disturbance reduction. In an open-loop test, operator intervention is often necessary in order to keep the process CV's in range. Manual control can be difficult when many MV's are excited. Operator control will be reduced or even avoided if the process is under feedback control. Therefore, closed-loop test is more acceptable to operators.
- **Better model for control.** This can be explained in several ways. Under the same CV variance constraints, the control performance degradation caused by model errors will be less if a closed-loop test is carried out; see Gevers and Ljung (1986) and Hjalmarsson *et. al.* (1996). In fact, CV variances in a closed-loop test can be actually larger than those in an open-loop test, because the process operation constraints are in CV amplitude limits, not in variance limits and the controller can remove the slow drifts of the CV's. This increases the signal-to-noise ratio in the data further. The effect of feedback will

have additional advantage if the process is ill-conditioned meaning that several CV's are strongly correlated. High purity distillation columns are often ill-conditioned where top and bottom compositions have strong correlation. For the control of ill-conditioned processes, it is important to identify the model that has a good estimate of low-gain direction. In an open-loop test where MV's are moved independently, the information of low gain direction will have very low power and cannot be estimated accurately from noisy data. In order to amplify the power of low-gain direction, a certain correlation between MV movement is needed. This correlation can be created naturally by (partial) feedback control; see Section 10.4 and Koung and MacGregor (1993) and Jacobsen (1994).

In a total closed-loop test where all loops are closed, we recommend the application of test signals at all MVs and at CV setpoints. In a partial closed-loop test, part of test signals are applied at the MV's and CV setpoints in the closed-loop part and part of test signals are applied at MV's in the open-loop part.

Test design for nonlinear process identification will be discussed in Section 9.1.

3.5 Sampling Frequency and Anti-Aliasing Filter

In a computer controlled system, sampling continuous signals leads to information losses. Therefore, it is important to select the sampling frequency so that these losses are not harming control application. Assuming uniform sampling, we will discuss the choice of sampling time T for model identification and control.

Industrial processes are typically slow with respect to the sampling speed of present day computer systems. Although it is true that a higher sampling frequency will cause less information loss, the following factors may prevent us from sampling as fast as possible:

- Building discrete-time models with a very small sampling time compared to the natural time constants is a numerically sensitive procedure, because all poles cluster around the point $(1, i0)$ on the z-plane.
- If the model is biased, the model fit may be concentrated in a high frequency band (Wahlberg and Ljung, 1986), which is not desirable for most of the model applications.
- A fast sampled model will often be non-minimum phase even if the original continuous-time process is minimum phase (Åström and Wittenmark, 1984).
- The computation load of MPC control is high. When models are used in MPC, the sampling time should be long enough to enable control computation to finish.

Although it is possible to optimize the sampling time for very simple problems (Åström and Wittenmark, 1984 and Ljung, 1999), it is not feasible to do so in practical situations. Reasons

are that the problem is more complex and that the process model is not known. There are several rules for determining the sampling time that are related to the smallest time constant, process bandwidth or process settling time.

Smallest time constant. Denote the smallest process time constant of interest to the user by τ_{\min} seconds and denote T as the sampling time. Then a reasonable relation between the two is (Åström and Wittenmark, 1984)

$$T \approx \tau_{\min}/3 \text{ (s)} \quad (3.5.1)$$

This implies that the sampling frequency is

$$f_s = \frac{3}{\tau_{\min}} \text{ (Hz)} \quad (3.5.2)$$

Process bandwidth. Denote f_o as the cut-off frequency of the process, then a good choice of sampling frequency is

$$f_s = 10f_o \text{ (Hz)} \quad (3.5.3)$$

As mentioned before, the smallest time constant and bandwidth can be estimated from the data of white noise experiment.

Process settling time. Denote T_{st} as the process settling time, then the sampling time can be chosen in the range of

$$T = \frac{T_{st}}{20} \text{ to } \frac{T_{st}}{100} \quad (3.5.4)$$

The settling time can be determined using a step test.

For a process with both slow outputs and fast outputs, some trade-off need to be made for selecting sampling time. For the MPC control of distillation columns, a proper sampling time for product qualities is in the order of minutes; a good sampling time for valve positions is in the order of seconds. The typical sampling time for a distillation column is 1 minute which, according to the rules, is slightly shorter for product qualities but much too long for the valves. However, process operation experience has shown that the saturation of valves in a short period in a distillation column is often tolerable. For other processes, such as a compressor, a saturated valve can cause equipment break down. In this situation the sampling rate should be high enough to control the valves and sampling time can be in the order of milliseconds. From a control theory point of view, better control performance can be achieved using a multi-rate sampling strategy which is much more complex to handle. Often only single-rate sampling is used for multivariable controllers, because a reasonable compromise between slow and fast output variables can be found for most of the situations.

The information loss due to sampling is best described in the frequency domain. Denote $\omega_s = 2\pi f_s$ as the sampling frequency in rad/s, then the *Nyquist frequency* is $\omega_N = \omega_s/2$. It is known

that the part of the signal spectrum that corresponds to frequencies above the Nyquist frequency ω_N will be interpreted as contributions from lower frequencies. This is the *alias* phenomenon or frequency folding (Åström and Wittenmark, 1984).

The information about the frequencies higher than the Nyquist frequency is thus lost by sampling. It is then important not to make bad worse by letting the aliasing effect distort the interesting part of the spectrum below the Nyquist frequency. This is achieved by a presampling filter or *anti-aliasing filter* $K(s)$ which is an analog filter. Normally the cut-off frequency of the filter is equal to the Nyquist frequency. Such filters should always be applied before sampling if we suspect that the signal has non-negligible energy above the Nyquist frequency.

Anti-aliasing filters can also be used for noise reduction. A typical situation in identification tests is that the signal consists of a useful part and a disturbance/noise part. For industrial processes, usually the disturbances have a low frequency content, while measurement noises are more broadband than that of the useful signal. Then the sampling time is usually chosen so that most of the spectrum of the useful part is below ω_N . The anti-aliasing filter then cuts away the high frequency noise content.

The scheme of applying an anti-aliasing filter in identification is shown in Figure 3.5.1; here $u(t)$ and $y(t)$ denote continuous-time input and output, and $u(kT)$ and $y(kT)$ denote the sampled input and output. Note that the filter is *applied on both input and output signals*.

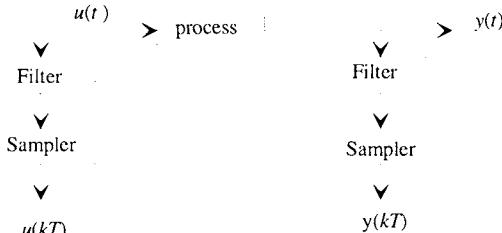


Figure 3.5.1 Anti-aliasing filter of input/output signals

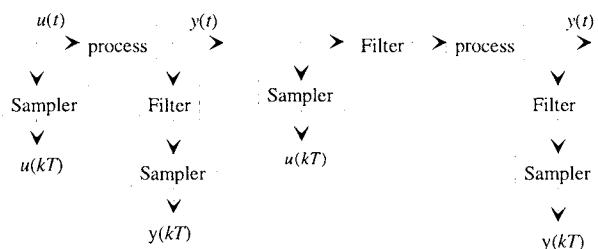


Figure 3.5.2 Using anti-aliasing filtering in the wrong places

Two schemes where the anti-aliasing filter is incorrectly placed are shown in Figure 3.5.2. On the left hand side of Figure 3.5.2 the filter will be included in the identified model; on the right hand side of Figure 3.5.2 the filter will be included in the model twice. When the anti-aliasing filter is included in the model, it will not cause significant change of the model amplitude, however, this is not true for the phase change which is usually not negligible. This causes problems in control design.

3.6 Pre-Treatment of Data

When data has been collected from the final test, it is often not suitable for immediate use in identification algorithms. There are several possible deficiencies:

1. High frequency (measurement) noise in the data.
2. Occasional spikes and outliers.
3. Sometimes considerable unmeasured disturbances occur that may drive the process out of its linear range.
4. Change of operation point during the test may cause nonlinearity in the data.
5. Drift and offset, low frequency disturbance.

If it is not expensive, it is advisable to use a sampling frequency that is a multiple of the desired working frequency. This will build redundancy into the measured data that can reduce the information loss in the pre-treatment of the data. When the data are collected, one must always first plot the data in order to inspect them for these deficiencies. In this section we shall discuss how to process the data in order to eliminate or to reduce the impact of these problems.

3.6.1 Peak Shaving

Peak shaving is required to reduce the effects of spikes. In industrial situations spikes are induced in the sensors and data acquisition system. The amplitudes of the noise spikes may be very large compared to the actual signal range. As the energy contents of the spikes may be large, their potential influence on the estimation results can be considerable. Removal of spikes from the process signals can be accomplished using the following signal processing procedure (Backx and Damen, 1989):

1. Clip the signal amplitudes to values never to be reached by the real process signals (using *a priori* knowledge).

2. Compute the standard deviation of the clipped signal.
3. Interpolate all samples of the original signal that are outside a band defined by the trend signal $\pm\alpha$ times the computed standard deviation, where α is chosen such that no signal value exceeds the permitted signal range.

All consecutive sample values outside the permitted band are replaced by values obtained from a linear interpolation starting from the last accepted sample value and stopping at the first sample value within the permitted band after the spike.

Most often, eye inspection and process knowledge are sufficient to carry out the job. A user friendly graphical user interface (GUI) can be helpful as well.

3.6.2 Signal Slicing

For linear model identification, usually slicing is used to remove the portion of a signal for two reasons: 1) severe unmeasured disturbances caused by process upset; and 2) the process is driven out of its linear range by the change of working point. In general, slicing is done by visual inspection using process knowledge. A good graphical user interface (GUI) is necessary for this job. There is a difference between peak shaving and slicing. The values removed by peak shaving are normally not valid measurements; the portion of a signal removed by slicing is relevant measurement, but the signal-to-noise ratio of this part is considerably lower than the average over the whole test period, or too much nonlinearity has been excited.

3.6.3 Trend Correction or High-Pass Filtering

Drifts or slow variations are common in industrial data. Examples of the causes are feed composition changes and changes of ambient temperature. Trends and drifts have bad influence on the estimation results. They do not average out because of their low frequency behavior and they will cause model errors at low frequency. Hence, it is natural to remove them from the data.

At least two approaches can be used for trend correction. The first one is to estimate the trends and to remove them from the signals. A good approach for the computation of the trend of a signal is to use a symmetric, noncausal, lowpass filter, as such filter has no phase shift (Backx, 1987 and Backx and Damen, 1989). This filter can be accomplished off-line by filtering the signal with a causal lowpass filter of appropriate cut-off frequency both in forward and reversed time. Note that when trend estimation is not accurate, error will be introduced to the data. Hence this approach should be used with caution.

The second method of trend correction is to perform highpass filtering on inputs and outputs. The band of the filter should cover the dynamics of the process that is important for control. In this way we can remove the trend/drifts and also some information at very low frequency. If parametric models are used, the gain can be extrapolated well from higher frequency information.

There is an additional motivation of highpass filtering. Identification theory often uses the assumption that the disturbances are stationary stochastic processes. Data with trends and drifts do not satisfy the assumption. Highpass filtering will make the data more stationary.

3.6.4 Scaling and Offset Correction (Normalization)

In industrial practice not all inputs and outputs have the same order of magnitude. The numerical values obtained are related to physical quantities which, in general, do not have the same dimensions. The signals with the largest numerical values will automatically get the highest weight in the quadratic loss function which is minimized for determining the model, if these signals are directly used. This problem can be overcome by correcting the signals for offsets and by scaling them afterwards. The steps are:

- Subtract average signal values in order to allow the use of a linear model without offset to describe the dynamic behavior of the process around its working point.
- Scale both input and output signals with respect to their power contents.

3.6.5 Delay Correction, Optional

In the preparation for the parameter estimation, delay times can be compensated by shifting input and output signals relative to each other. One of the signals is used as a reference and the remaining signals are shifted in time in order to compensate the delays. For a process with m inputs and p outputs, in general, each output can show a delay in the response to each input so that finally $m - p$ different time delays may be found. The number of delays that can be compensated by shifting the signals against each other is only $m + p - 1$, because only $m + p$ signals are available and one signal is fixed as the reference. As a result not all delays can be compensated exactly. The compensation has to be such that the total sum of delays left is minimal. Time delays remaining after the compensation have to be estimated as part of the model.

3.6.6 Lowpass Filtering and Sampling Rate Reduction, Optional

Remember that we have built redundancy into the data by sampling faster than necessary. If there is noise energy in the high frequency band beyond the interested range, we can use digital low pass filtering to cut it off. Do filter *all* the inputs and outputs. Suppose that the sampling frequency is l times the working frequency, we can reduce the frequency to the working frequency by taking one sample from each l samples.

It is well known that data prefiltering will affect the bias distribution of the model; see Ljung (1987). However, we do not recommend using this function of prefilter. This is because prefilter

and disturbance model jointly affect the bias. In order to use prefilter effectively, one needs to fix the disturbance model, which is not a good idea for closed-loop identification; see Chapter 5 for explanations.

3.6.7 Precautions for Nonlinear Model Identification

For the identification of nonlinear models, not all of the above operations are relevant.

Peak shaving. The purpose of peak-shaving is to remove the signal outliers which are caused by sensor or system failure. This should be carried out for both linear and nonlinear model identification.

Signal slicing. For nonlinear model identification, one should only use signal slicing in the case of process upset which is caused by unmeasured disturbances. The change of working point is necessary for nonlinear model identification.

High-pass and low-pass filtering. It is well known that linear filtering will not change the input-output relationship only when the underlying process is linear. When the process is nonlinear, filtering will change the input-output relationship. Therefore, both high-pass and low-pass filtering should **not** be used for nonlinear model identification.

Scaling and offset correction. Signal normalization can be done for nonlinear model identification. One should be careful that, when using the identified model, the removed signal offsets should be put back in the nonlinear model; while this is not necessary for linear models because they are not dependent on signal range.

3.7 When is Excitation Allowed? Concluding Remarks

In the discussions of test design, we have assumed that extra excitation is applied to the process in open-loop or closed-loop operation. Excitation will certainly cause disturbances to process operation. Hence, it is sensible to ask when or how often can test signals be applied. The answer is not always a technical one. In practice, the obstacles for applying test signals are the following:

1. The operation personnel are not convinced about the importance of good model for control performance, or previous identification has not been successful due to poor test design.
2. There is a continuous debate over the identification approach and the physical modeling (first principle modeling) approach. Part of the project team may prefer the physical modeling approach.
3. Open-loop test is not permitted due to safety and economical considerations.

Therefore, if there is a good communication between control engineers and operation personnel, if the cost effectiveness of identification approach is fully understood, and if a closed-loop test is used whenever possible, test signals can be applied to most industrial processes. This is a very positive view indeed. Good test design and successful control projects will increase the acceptance of process identification.

In this chapter we have recommended following tests:

- Collecting historic data, or free-run tests.
- Step or staircase tests.
- White noise test.
- Final test.

The first three type of tests are used to collect a *priori* knowledge of the process and of the disturbances. The last experiment is for model identification.

In order to guarantee the identifiability, the test signals of the final experiment should be persistently exciting with a certain order. The test signals proposed in this chapter all fulfill this requirement. The spectra of the test signals should have proper distributions over the frequencies in order to have good performance in control. The test procedure using GBN signals has been industrially proven. Thus it is highly recommended. Before using the data for model estimation, do not forget to polish them by peak shaving, trend correction, offset correction, scaling and delay compensation.

Chapter 4

Identification by the Least-Squares Method

The least-squares principle was invented by Karl Gauss at the end of the eighteenth century for determining the orbits of planets. Since then this method has become a major tool for parameter estimation using experimental data. Most existing parametric identification methods can be related to the least-squares method. The method is easy to comprehend and, due to the existence of a closed solution, it is also easy to implement. The least-squares method is also called *linear regression* (in statistical literature) and *equation error method* (in identification literature).

Section 4.1 will introduce the principle of least-squares; in Section 4.2 the method will be applied to the estimation of finite impulse response (FIR) models and the estimation of parametric models. Before making assumptions and pursuing a theoretical analysis of the method, we first test the method on two industrial processes, a single stand rolling mill and a glass tube production process (Section 4.3). The method is successful for the first process; but it fails for the second one. Why? In Section 4.4 some theoretical analysis is carried out, and reasons are given why the least-squares method can fail. Finally in Section 4.5 we will draw conclusions about the least-squares method.

4.1 The Principle of Least-Squares

The least-squares technique is a mathematical procedure by which the unknown parameters of a mathematical model are chosen (estimated) such that the sum of the squares of some chosen error is minimized. Suppose a mathematical model is given in the form

$$y(t) = x_1(t)\theta_1 + x_2(t)\theta_2 + \cdots + x_n(t)\theta_n \quad (4.1.1)$$

where $y(t)$ is the observed variable, $\{\theta_1, \theta_2, \dots, \theta_n\}$ is a set of constant parameters, $x_1(t), x_2(t), \dots, x_n(t)$ are known functions that may depend on other known variables. The variable t often denotes time.

Assume that N samples of measurements of $y(t)$ and $x_1(t), x_2(t), \dots, x_n(t)$ are made at time $1, 2, \dots, N$. Filling the data samples into equation (4.1.1) results in a set of linear equations

$$y(t) = x_1(t)\theta_1 + x_2(t)\theta_2 + \dots + x_n(t)\theta_n \quad t = 1, 2, \dots, N \quad (4.1.2)$$

This can be arranged in a simple matrix form

$$\mathbf{y} = \Phi\boldsymbol{\theta} \quad (4.1.3)$$

where

$$\mathbf{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}, \quad \Phi = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_n(1) \\ x_1(2) & x_2(2) & & x_n(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(N) & x_2(N) & \cdots & x_n(N) \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

A necessary condition that this set of equations has a solution is $N \geq n$. For $N = n$, we have a unique solution

$$\hat{\boldsymbol{\theta}} = \Phi^{-1}\mathbf{y} \quad (4.1.4)$$

provided Φ^{-1} , the inverse of the square matrix Φ , exists. $\hat{\boldsymbol{\theta}}$ denotes the estimate of $\boldsymbol{\theta}$. This is well known. However, when $N > n$, it is generally not possible to find a $\hat{\boldsymbol{\theta}}$ vector which can fit the data samples exactly, because the data may be contaminated by disturbances and measurement noise. A too low model order or a wrong model structure are other possible causes of misfit. A way to determine the parameters is to estimate them on the basis of least-squares-error.

Introduce a residual (error), $\varepsilon(t)$, and let

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \varphi(t)\hat{\boldsymbol{\theta}}$$

Now we will choose $\hat{\boldsymbol{\theta}}$ such that the criterion (loss function)

$$V_{LS} = \frac{1}{N} \sum_{t=1}^N \varepsilon(t)^2 = \frac{1}{N} \sum_{t=1}^N [y(t) - \varphi(t)\hat{\boldsymbol{\theta}}]^2 = \frac{1}{N} \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} \quad (4.1.5)$$

is minimized. Here

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon(1) \\ \varepsilon(2) \\ \vdots \\ \varepsilon(N) \end{bmatrix}$$

To carry out the minimization, we express

$$V_{LS}(\theta) = \frac{1}{N}(\mathbf{y} - \Phi\theta)^T(\mathbf{y} - \Phi\theta) = \frac{1}{N}[\mathbf{y}^T\mathbf{y} - \theta^T\Phi^T\mathbf{y} - \mathbf{y}^T\Phi\theta + \theta^T\Phi^T\Phi\theta]$$

Taking the first derivative of $V_{LS}(\theta)$ with respect to θ and equating the result to zero, we have

$$\frac{\partial V_{LS}(\theta)}{\partial \theta} \Big|_{\theta=\hat{\theta}} = \frac{1}{N}[-2\Phi^T\mathbf{y} + 2\Phi^T\Phi\hat{\theta}] = 0$$

Hence the solution is given by the equation

$$\Phi^T\Phi\hat{\theta} = \Phi^T\mathbf{y} \quad (4.1.6)$$

or

$$\hat{\theta} = [\Phi^T\Phi]^{-1}\Phi^T\mathbf{y} \quad (4.1.7)$$

This result is the well known *least-squares estimator* of θ .

From linear algebra, we know that equation (4.1.6) will have an unique solution if and only if the matrix

$$\Phi^T\Phi = \frac{1}{N}\sum_{t=1}^N \varphi^T(t)\varphi(t) \quad (4.1.8)$$

is nonsingular. This is called the parameter identifiability condition which has been discussed in Chapter 3.

The criterion (4.1.5) can be generalized by introducing a weighting matrix to allow each error term to be weighted individually. Let W be the desired weighting matrix which is symmetrical positive definite, then the weighted error criterion becomes

$$V_w(\theta) = \frac{1}{N}\boldsymbol{\varepsilon}^T W \boldsymbol{\varepsilon} = \frac{1}{N}(\mathbf{y} - \Phi\theta)^T W (\mathbf{y} - \Phi\theta)$$

Minimization of $V_w(\theta)$ with respect to θ follows the same procedure as above, the parameter estimator is given by

$$\hat{\theta}_w = [\Phi^T W \Phi]^{-1} \Phi^T W \mathbf{y} \quad (4.1.9)$$

This is called *weighted least-squares estimator*. If $W = I$, this reduces to the normal least-squares estimator.

4.2 Estimating Models of Linear Processes

The least-squares method can be used to estimate models of linear dynamic processes. The particular way to do this will depend on the character of the model and its parametrization.

4.2.1 Finite Impulse Response (FIR) Model

A linear time-invariant dynamic process is uniquely characterized by its impulse response. For stable processes, the impulse response will tend to zero for increasing time and may then be truncated. This results in so-called finite impulse response (FIR) models or Markov parameter models. For a single-input single-output process such a model is given by

$$\begin{aligned} y(t) &= g_1 u(t-1) + g_2 u(t-2) + \cdots + g_n u(t-n) \\ &= \sum_{k=1}^N g_k u(t-k) = \varphi(t)\theta \end{aligned} \quad (4.2.1)$$

where

$$\theta = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}, \quad \varphi(t) = [u(t-1) \ u(t-2) \ \cdots \ u(t-n)]$$

This model is identical to the regression model in (4.1.1) when replacing $x_k(t)$ by $u(t-k)$. Hence the least-squares estimator (4.1.7) can be applied directly.

Given the measured input-output data sequence:

$$y(1), u(1), \dots, y(N), u(N) \quad (4.2.2)$$

and introducing the residual as

$$\begin{aligned} y(t) &= g_1 u(t-1) + g_2 u(t-2) + \cdots + g_n u(t-n) + \varepsilon(t) \\ &= \varphi(t)\theta + \varepsilon(t) \end{aligned} \quad (4.2.3)$$

In matrix form

$$\mathbf{y} = \Phi\theta + \boldsymbol{\varepsilon} \quad (4.2.4)$$

where

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon(n+1) \\ \varepsilon(n+2) \\ \vdots \\ \varepsilon(N) \end{bmatrix} \\ \Phi &= \begin{bmatrix} u(n) & u(n-1) & \cdots & u(1) \\ u(n+1) & \ddots & & \vdots \\ \vdots & \vdots & & \vdots \\ u(N-1) & \cdots & \cdots & u(N-n) \end{bmatrix} \end{aligned}$$

Then we have according to least-squares formula (4.1.7)

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T \mathbf{y} \quad (4.2.5)$$

Figure 4.2.1 shows the block diagram of FIR model estimation.

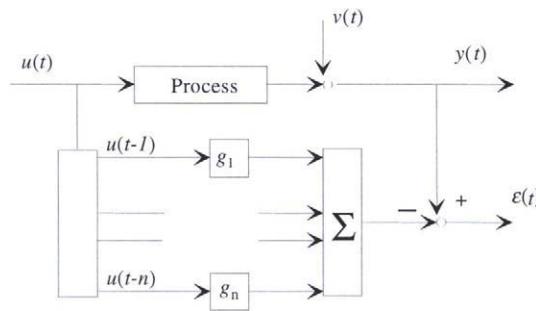


Figure 4.2.1 Error generation in FIR model estimation

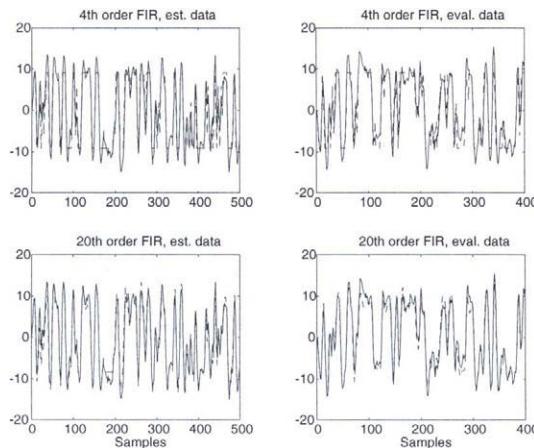


Figure 4.2.2 Simulated outputs (dashed line) of FIR models and process output (solid line)

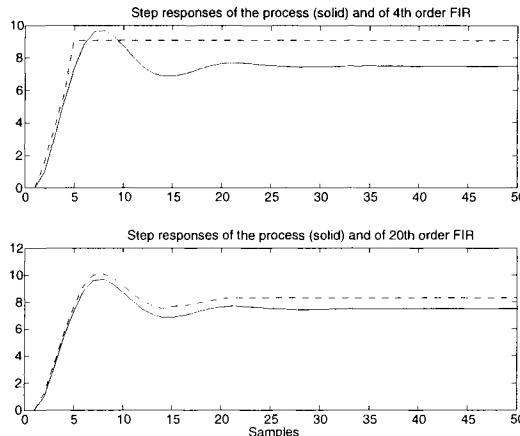


Figure 4.2.3 Compare step responses of the FIR models (dashed line) and of the process (solid line)

Exercise 4.2.1. Given the process (Åström and Eykhoff, 1971)

$$y(t) = G^o(q)u(t) + v(t) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}}u(t) + v(t)$$

Simulate the process using a GBN signal with mean switching time 8 samples as input $u(t)$. The disturbance $v(t)$ is a lowpass filtered white noise $c(t)$:

$$v(t) = \frac{1}{1 - 0.9q^{-1}}e(t)$$

Adjust the variance of $v(t)$ so that the noise-to-signal ratio at output is 5% in power:

$$\text{var}(v(t))/\text{var}(y_o(t)) = 5\%$$

Generate one data set of 500 samples for model estimation, 400 samples fresh for model evaluation (not to be used in model estimation). Then estimate two FIR models with order 4 (too short) and 20 (proper length) respectively. Compare model output with process output using both estimation data and fresh data. Also compare the step responses of the two models to the true one.

Answer. The M file is given as follows. The model fits and step responses are shown in Figures 4.2.2 and 4.2.3.

```
%%% firexercise.m; M file for FIR model estimation %%%
```

```
function [] = firexercise( )
% Create low-pass GBN with N = 500, 400 and ETsw = 8
U_est = gbnengen(500,8);
U_eval = gbnengen(400,8);
% Simulate the process
Ao = [1 -1.5 0.7];
Bo = [0 1 0.5];
Yo_est = filter(Bo,Ao,U_est);
Yo_eval = filter(Bo,Ao,U_eval);
% Add 5% output disturbance
v_est = filter(1,[1 -0.9],randn(500,1));
v_est = v_est/std(v_est)*sqrt(0.05)*std(Yo_est);
Y_est = Yo_est + v_est;
v_eval = filter(1,[1 -0.9],randn(400,1));
v_eval = v_eval/std(v_eval)*sqrt(0.05)*std(Yo_eval);
Y_eval = Yo_eval + v_eval;
% Estimate FIR models with n = 4, 20
THfir4 = arx([Y_est U_est],[0 4 1]);
THfir20 = arx([Y_est U_est],[0 20 1]);
% Simulation of the models
Yh_est4 = idsim(U_est,THfir4);
Yh_est20 = idsim(U_est,THfir20);
Yh_eval4 = idsim(U_eval,THfir4);
Yh_eval20 = idsim(U_eval,THfir20);
% Calculate step responses
STPo = filter(Bo,Ao,ones(50,1));
STP4 = idsim(ones(50,1),THfir4);
STP20 = idsim(ones(50,1),THfir20);
% Plot model fit
t_est=1:500;
t_eval=1:400;
figure;
subplot(221),plot(t_est,Y_est,'-r',t_est,Yh_est4,'--b');
title('4th order FIR, est. data');
axis([0 500 -20 20])
subplot(222),plot(t_eval,Y_eval,'-r',t_eval,Yh_eval4,'--b');
title('4th order FIR, eval. data');
axis([0 400 -20 20])
subplot(223),plot(t_est,Y_est,'-r',t_est,Yh_est20,'--b');
title('20th order FIR, est. data');
axis([0 500 -20 20])
xlabel('Samples')
subplot(224),plot(t_eval,Y_eval,'-r',t_eval,Yh_eval20,'--b');
```

```

title('20th order FIR, eval. data');
axis([0 400 -20 20])
xlabel('Samples')
% Plot step responses
figure;
subplot(211),plot(1:50,STPo,'-r',1:50,STP4,'--b');
title('Step responses of the process (solid) and of 4th order FIR');
subplot(212),plot(1:50,STPo,'-r',1:50,STP20,'--b');
title('Step responses of the process (solid) and of 20th order FIR');
xlabel('Samples')

```

The 4th order FIR model cannot catch the overshoot; the 20th order FIR model gives a good fit to the true step response. One can see that, when process settling time is not too long, the FIR model order is sufficiently high and disturbance level is moderate, a good model can be obtained. The reader is encouraged to run the M file. Note that the models from various runs are different. This is because the disturbance realizations are different, which causes randomness in identified models.

The advantages of FIR models are the following:

1. Less *a priori* knowledge of the process is required to estimate a FIR model. The question of the order or structure of the process as required when using other models do not arise.
2. FIR model estimate is statistically unbiased (the expectation of the estimate equals the true value) and consistent (the estimate tends to the true value when the number of data points tends to infinity); see Section 4.4 for the analysis.

Of course FIR models also have disadvantages:

1. This model structure often needs many parameters; the estimated process transfer function is not accurate for a finite number of data, because the variance of the estimate is proportional to the number of parameters of the FIR model (see Chapter 6).
2. The FIR model is not suitable for linear control design methods. Often a model reduction is necessary to arrive at a compact parametric model.

The extension of the FIR model estimation to multivariable processes is straightforward. We will show this by a case study in the next section.

4.2.2 Rational Transfer Function Models or ARX Models

The least squares method can be used to estimate the parameters of transfer function models or difference equation models. Let the process be described by an n th order difference equation

$$y(t) + a_1y(t-1) + \cdots + a_ny(t-n) = b_1u(t-1) + \cdots + b_nu(t-n) \quad (4.2.6)$$

From Chapter 2 we know that this process has a transfer function

$$G(q) = \frac{B(q)}{A(q)} \quad (4.2.7)$$

where

$$\begin{aligned} A(q) &= 1 + a_1q^{-1} + \cdots + a_nq^{-n} \\ B(q) &= b_1q^{-1} + \cdots + b_nq^{-n} \end{aligned}$$

Given the input-output data sequence as

$$y(1), u(1), \dots, y(N+n), u(N+n)$$

and assuming that the order n is known, then we need to estimate the parameters a_i and b_i . To do this, we introduce the error as

$$y(t) + a_1y(t-1) + \cdots + a_ny(t-n) = b_1u(t-1) + \cdots + b_nu(t-n) + \varepsilon(t) \quad (4.2.8)$$

or

$$A(q)y(t) = B(q)u(t) + \varepsilon(t) \quad (4.2.9)$$

The term $\varepsilon(t)$ is used to account for the fitting error. In literature $\varepsilon(t)$ is referred to as the *residual* or *equation error* and equation (4.2.9) is called ARX (AutoRegressive with eXogenous input) model.

Rewrite equation (4.2.8) as

$$\begin{aligned} y(t) &= -a_1y(t-1) - \cdots - a_ny(t-n) + b_1u(t-1) + \cdots + b_nu(t-n) + \varepsilon(t) \\ &= \varphi(t)\theta + \varepsilon(t) \end{aligned} \quad (4.2.10)$$

where $\varphi(t)$ is the data vector

$$\varphi(t) = [-y(t-1), \dots, -y(t-n), u(t-1), \dots, u(t-n)]$$

and θ is the parameter vector

$$\theta = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Using the data sequence we can form a system of N equations ($N >> 2n$):

$$\mathbf{y} = \Phi\theta + \boldsymbol{\varepsilon} \quad (4.2.11)$$

where

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon(n+1) \\ \varepsilon(n+2) \\ \vdots \\ \varepsilon(N) \end{bmatrix} \\ \Phi &= \begin{bmatrix} \varphi(n+1) \\ \varphi(n+2) \\ \vdots \\ \varphi(N) \end{bmatrix} = \left[\begin{array}{ccccc|ccccc} -y(n) & \cdots & -y(1) & | & u(n) & \cdots & u(1) \\ -y(n+1) & & \vdots & | & u(n+1) & & \vdots \\ \vdots & & \vdots & | & \vdots & & \vdots \\ -y(N-1) & \cdots & -y(N-n) & | & u(N-1) & \cdots & u(N-n) \end{array} \right] \end{aligned}$$

Then according to the least-squares principle of Section 4.1, the estimate which minimizes the loss function

$$V_{LS} = \sum_{t=n+1}^N \varepsilon(t)^2 = \sum_{t=n+1}^N [y(t) - \varphi(t)\theta]^2 \quad (4.2.12)$$

is

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T \mathbf{y} = \left[\sum_{t=n+1}^N \varphi^T(t) \varphi(t) \right]^{-1} \left[\sum_{t=n+1}^N \varphi^T(t) y(t) \right] \quad (4.2.13)$$

This solution exists if matrix

$$\Phi^T \Phi = \sum_{t=n+1}^N \varphi^T(t) \varphi(t)$$

is nonsingular, which can be guaranteed if the process order is n and the input signal $u(t)$ is persistently exciting with order $2n$ (see Chapter 3). Figure 4.2.4 shows the block diagram of error generation in transfer operator estimation using the least-squares method.

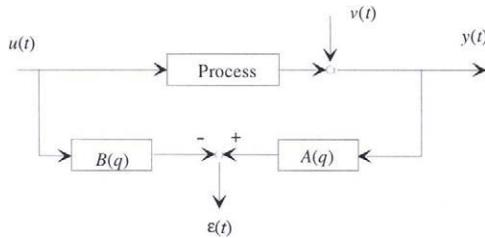


Figure 4.2.4 Error generation in transfer operator estimation

The advantage of parametric identification by the least-squares method, or *equation error method*, is its numerical simplicity. The method has a closed solution. This is due to the fact that the error (residual) is linear in parameters a_i and b_i and that a quadratic loss function (4.2.12) is minimized. It will work well when the noise level is low and model order is correct. Under practical conditions, however, the least-squares estimator of the transfer function model is biased if the model order is not sufficiently high; see Section 4.4. This will cause accuracy problems if the noise level increases. Moreover, in terms of frequency response estimate, the least-squares estimator can give a poor fit to the process transfer function at low and middle frequencies; see Section 4.4. This is of course not desired in applications such as control.

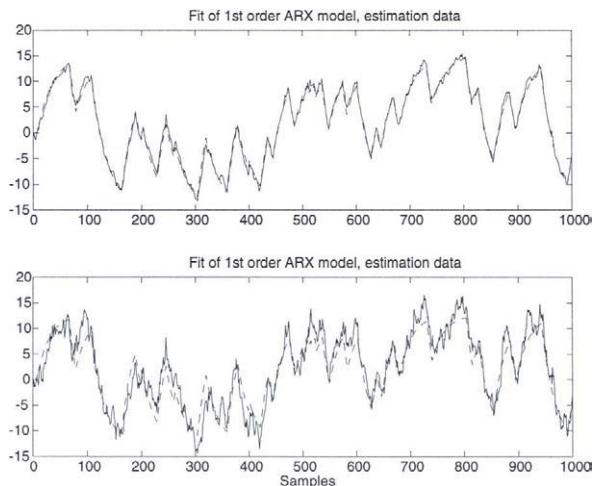


Figure 4.2.5 Model fit of the ARX models

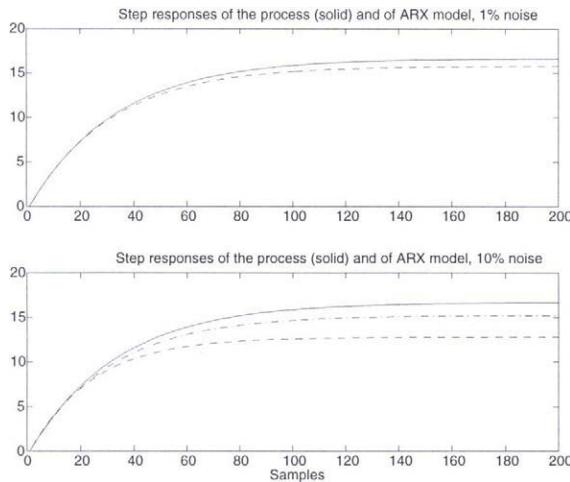


Figure 4.2.6 Step responses of the ARX models and of the process. The dashed-dot line is the step response of a 1st order output error model; see Chapter 5.

Exercise 4.2.2. Given the process

$$y(t) = \frac{0.5q^{-1}}{1 - 0.97q^{-1}} u(t) + v(t)$$

Simulate the process using a GBN signal with mean switching time 20 as the input $u(t)$. Let the number of samples be = 1000. The disturbance $v(t)$ is a lowpass filtered white noise $e(t)$:

$$v(t) = \frac{1}{1 - 0.9q^{-1}} e(t)$$

Adjust the variance of $v(t)$ so that the noise-to-signal ratios at output are 1% and 10% in power:

$$\text{var}(v(t))/\text{var}(y^o(t)) = 1\%, 10\%$$

Estimate 1st order ARX models using both data sets. Compare model outputs with process output; also compare the step responses of the two models to the true one.

Answer. The M file is as follows. The model fits and step responses are shown in Figures 4.2.5 and 4.2.6.

```
%%% arxexercise.m; M file for ARX model estimation %%%
function [] = arxexercise()
% Create low-pass GBN with ETsw = 20
```

```

N = 1000;
U = gbnugen(N,20);
% Simulate the process
Ao = [1 -0.97];
Bo = [0 0.5];
Yo = filter(Bo,Ao,U);
% Add 1% and 10% output disturbances
v = filter(1,[1 -0.9],randn(N,1));
v01 = v/std(v)*sqrt(0.01)*std(Yo);
Y01 = Yo + v01;
v10 = v/std(v)*sqrt(0.1)*std(Yo);
Y10 = Yo + v10;
% Estimate 1st order ARX models
THarx01 = arx([Y01 U],[1 1 1]);
THarx10 = arx([Y10 U],[1 1 1]);
% Estimate 1st order output error model for 10% noise data
THoe10 = oe([Y10 U],[1 1 1]);
% Simulation of the models
Yh01 = idsim(U,THarx01);
Yh10 = idsim(U,THarx10);
% Calculate step responses
Nstp = 200;
STPo = filter(Bo,Ao,ones(Nstp,1));
STP01 = idsim(ones(Nstp,1),THarx01);
STP10 = idsim(ones(Nstp,1),THarx10);
STPoe10 = idsim(ones(Nstp,1),THoe10);
% Plot model fit
t = 1:N;
figure;
subplot(211),plot(t ,Y01,'-r',t ,Yh01,'--b');
title('Fit of 1st order ARX model, estimation data');
%axis([0 N -20 20])
subplot(212),plot(t ,Y10,'-r',t ,Yh10,'--b');
title('Fit of 1st order ARX model, estimation data');
%axis([0 N -20 20])
xlabel('Samples')
% Plot step responses
figure;
subplot(211),plot(1:Nstp,STPo,'-r',1:Nstp,STP01,'--b');
title('Step responses of the process (solid) and of ARX model, 1% noise');
subplot(212),
plot(1:Nstp,STPo,'-r',1:Nstp,STP10,'--b',1:Nstp,STPoe10,'-.k');
title('Step responses of the process (solid) and of ARX model, 10% noise');

```

```
xlabel('Samples')
```

One sees that the model is quite accurate in the 1% noise case. The error in model gain is large in the 10% noise case. Of course, this increased error is partly due to higher level of disturbance. However, the other part of the cause is due to ARX model structure. To show this, we have also estimated a so-called *output error* model using the 10% noise data (see Chapter 5 for the details of output error method). The error of the output error model is indeed much smaller. This exercise shows that if the level of disturbance is low, a least-squares (ARX) model will be accurate. When the disturbance level increases, the model quality will deteriorate. When the M file is run, again randomness of model results can be observed.

4.2.3 Order Selection, A Simulation Approach

The identification method discussed so far assumed that the order of the model is known, and that only the parameters are to be estimated. This is only part of the story because, in practice, the order of the process is seldom exactly known. Model order or structure determination is an important topic in process identification. The literature on order or structure selection is enormous. At this stage of development, we present a simple and practical technique. The order of FIR model is related to the time to steady state of the process or settling time. It can be often determined using *a priori* process knowledge. Here we will only discuss the order selection of the rational transfer function model, or, ARX model.

Order selection should not be confused with model validation. Order selection is to find a model order in a given model structure (class) that best explains the given data. The task of model validation means to check whether a model is good enough for the intended use of the model. A model with certain order can give the best fit to the data, but the best model derived from the data may not be good enough for its use (in control). We will revisit these problems in the following chapters when we know more about identification.

For model applications such as control and simulation, a simple and effective method of model validation is to simulate the estimated model using some test input, and compare the model output with the measured output; see Figure 4.2.7.

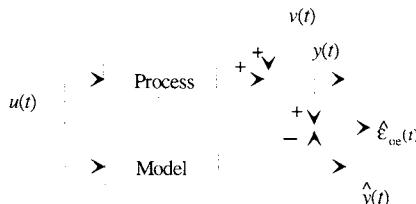


Figure 4.2.7 Order selection by simulation

The goodness of fit is measured by the sum of the squares of the simulation error or *output error*:

$$V_{OE} = \frac{1}{N} \sum_{t=1}^N \hat{\varepsilon}_{OE}(t)^2 \quad (4.2.14)$$

where

$$\hat{\varepsilon}_{OE}(t) = y(t) - \frac{\hat{B}(q^{-1})}{\hat{A}(q^{-1})} u(t) = y(t) - \hat{y}(t) \quad (4.2.15)$$

Note that this error is *not* the same as the equation error introduced in least-squares method; compare Figure 4.2.2. The relation between the two errors is

$$\hat{\varepsilon}(t) = \hat{A}(q) \hat{\varepsilon}_{OE}(t) \quad (4.2.16)$$

When using a finite impulse response (FIR) model, then $A(q) = 1$ and the equation error is equal to the output error.

For order selection we use the output error instead of the equation error because the output error is more directly related to the error of the transfer function. From Figure 4.2.7 and equation (4.2.16) we have

$$\hat{\varepsilon}_{OE}(t) = [G^o(q) - \hat{G}(q)]u(t) + v(t) \quad (4.2.17)$$

Thus this error consists of the model misfit and the output disturbance.

In general, the loss function V_{OE} decreases as the order n increases. The reduction of V_{OE} ceases to be significant when the order of the model is high enough for simulation purposes. Based on this principle, a procedure for order selection is simply to compute the least-squares estimates and corresponding output error loss function for a sequence of model orders $n = 1, 2, 3, \dots$. The appropriate model order can be chosen as the one for which V_{OE} stops decreasing significantly.

An important remark is that order selection should be performed on a data set which has not been used for model estimation. This is sometimes called *cross-validation*. The reason is that a good fit to the data which were used to estimate the parameters is not sufficient to prove the model quality. In the extreme case, when the model order equals the number of equations $n = N$, then the fitting error is zero; the model quality, however, can be very poor due to large variances.

Exercise 4.2.3. Given the process as the same as in Exercise 4.2.1:

$$y(t) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} u(t) + v(t)$$

Simulate the process using a GBN signal with mean switching time of 8 samples. The disturbance $v(t)$ is a lowpass filtered white noise $e(t)$:

$$v(t) = \frac{1}{1 - 0.9q^{-1}} e(t)$$

Adjust the variance of $v(t)$ so that the noise-to-signal ratios are:

$$\text{var}(v(t))/\text{var}(y^o(t)) = 0.1, 1, 10, 50\%$$

Generate 500 samples for model estimation and the second 500 samples of fresh data for order selection. Plot the loss function V_{OE} as a function of model order for different noise-to-signal ratios and use both estimation data and fresh data. Check if the true order 2 can be found from this kind of test.

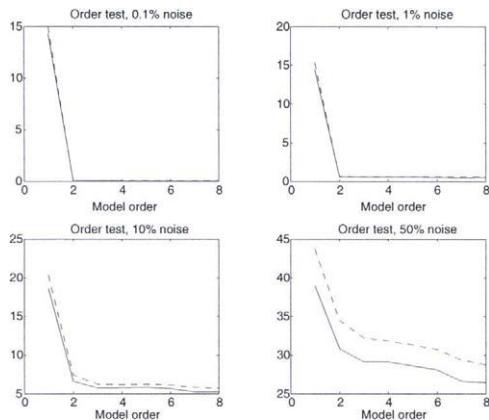


Figure 4.2.8 Order selection for ARX models using output error criterion. The solid lines are the loss functions evaluated using the fresh data, dashed lines are the loss functions using estimation data.

Answer. The M file for the exercise is as follows. The plots of output error loss functions are in Figure 4.2.6. We find that if the noise level is very low (0.1% noise), the true order can be found. As the noise level increases, higher order will be selected.

```
%%% arxordersel.m; M file for ARX model order selection %%%
function [] = arxordersel( )
% Create low-pass GBN with ETsw = 8
N = 1000;
N1 = 500;
U = gbneng(N,8);
% Simulate the process
Ao = [1 -1.5 0.7];
Bo = [0 1 0.5];
Yo = filter(Bo,Ao,U);
```

```
% Add output disturbances
v = filter(1,[1 -0.9],randn(N,1));
v001 = v/std(v)*sqrt(0.001)*std(Yo);
Y001 = Yo + v001;
v01 = v/std(v)*sqrt(0.01)*std(Yo);
Y01 = Yo + v01;
v10 = v/std(v)*sqrt(0.1)*std(Yo);
Y10 = Yo + v10;
v50 = v/std(v)*sqrt(0.5)*std(Yo);
Y50 = Yo + v50;
% Estimate ARX models between orders 1 and 8
% and calculate output error loss functions
for k = 1:8
    THarx001 = arx([Y001(1:N1) U(1:N1)], [k k 1]);
    Error001 = Y001 - idsim(U, THarx001);
    V001_est(k) = cov(Error001(1:N1));
    V001_eval(k) = cov(Error001(N1+1:N));
    THarx01 = arx([Y01(1:N1) U(1:N1)], [k k 1]);
    Error01 = Y01 - idsim(U, THarx01);
    V01_est(k) = cov(Error01(1:N1));
    V01_eval(k) = cov(Error01(N1+1:N));
    THarx10 = arx([Y10(1:N1) U(1:N1)], [k k 1]);
    Error10 = Y10 - idsim(U, THarx10);
    V10_est(k) = cov(Error10(1:N1));
    V10_eval(k) = cov(Error10(N1+1:N));
    THarx50 = arx([Y50(1:N1) U(1:N1)], [k k 1]);
    Error50 = Y50 - idsim(U, THarx50);
    V50_est(k) = cov(Error50(1:N1));
    V50_eval(k) = cov(Error50(N1+1:N));
end
% Plot output error loss functions
figure;
subplot(221), plot(1:8, V001_eval, '-r', 1:8, V001_est, '--b');
title('Order test, 0.1% noise');
xlabel('Model order')
subplot(222), plot(1:8, V01_eval, '-r', 1:8, V01_est, '--b');
title('Order test, 1% noise');
xlabel('Model order')
subplot(223), plot(1:8, V10_eval, '-r', 1:8, V10_est, '--b');
title('Order test, 10% noise');
xlabel('Model order')
subplot(224), plot(1:8, V50_eval, '-r', 1:8, V50_est, '--b');
title('Order test, 50% noise');
```

```
xlabel('Model order')
```

For a finite dimensional process, when the level of the disturbance is low, the true process order may be found using this procedure. When the level of disturbance increases, the order determined by this procedure can be higher than the true order; the higher order model is needed to model the disturbance in order to overcome the bias of the least-squares estimator (see Section 4.4).

Often it is not desirable to use a model order which is higher than the true order, because a high order model will need more computing power and it may cause numerical problems in controller design.

At this moment a critical reader might ask why not *estimate the parameters by minimizing the output error loss function V_{oe} directly*, which is closer to model application? The answer is that the least-squares method is often used because its solution is numerically simple and reliable. In fact the least-squares (equation error) loss function (4.2.8) is only a way of mathematical approximation; it is *not* a physically sensible criterion. The output error loss function is a better criterion if one wishes to identify the process transfer function. However, the output error is nonlinear in the parameters of the $A(q)$ polynomial, which means that no analytical solution to the minimization problem exists. Thus a nonlinear search algorithm needs to be used for finding a minimum. This is much more time-consuming than the least-squares method; numerical problems such as local minimum and non-convergence often occur. "We use it because it is simple" is often a way of engineering life. Indeed, it is always advisable to first try the simplest least-squares method. In model order selection, however, one should not forget the purpose of modeling and use a criterion (loss function) that is as close as possible to model application. The output error criterion is a good candidate to be used for control.

4.3 Industrial Case Studies

Now we have learned the least-squares principle and its applications in linear dynamic process identification. How does it perform in real world applications? To answer the question, typically, a theorist will start making assumptions and then analyze the properties of the method; an engineer, on the other hand, may try this tool on his process and give his judgment based on test results. Obviously the two ways of life are complementary. We will first take the engineers approach. In this section, two industrial processes are used to illustrate the least-squares method. The first process is the rolling mill; and the second one the glass tube drawing process. The extension of the least-squares method to multi-input multi-output (MIMO) process will be discussed when treating the problems.

4.3.1 Identification and Control of the Rolling Mill

Let us revisit the rolling mill introduced in Chapters 1 and 3. It was decided to use drawing force and roll gap as control inputs and keep the rolling speed constant; the input thickness cannot be manipulated and will be used for feedforward control. So, for the feedforward and feedback control of the process, a 3-input 1-output model needs to be identified; see Fig. 4.3.1.

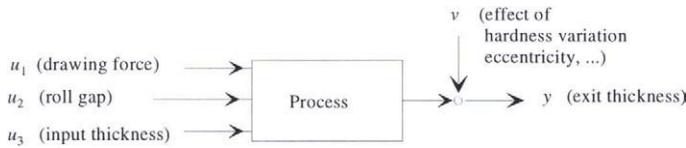


Figure 4.3.1 A 3-input 1-output rolling mill model

Thus the process model has the following form

$$y(t) = G_1(q)u_1(t) + G_2(q)u_2(t) + G_3(q)u_3(t) + v(t) \quad (4.3.1)$$

Based on a staircase experiment, we learned that the process is reasonably linear around working points and process step responses are very short. Hence an FIR model is used in identification:

$$y(t) = \sum_{k=1}^{n_1} g_{1,k}^o u_1(t-k) + \sum_{k=1}^{n_2} g_{2,k}^o u_2(t-k) + \sum_{k=1}^{n_3} g_{3,k}^o u_3(t-k) + v(t) \quad (4.3.2)$$

Introduce the residuals

$$\varepsilon(t) = y(t) - \left[\sum_{k=1}^{n_1} g_{1,k} u_1(t-k) + \sum_{k=1}^{n_2} g_{2,k} u_2(t-k) + \sum_{k=1}^{n_3} g_{3,k} u_3(t-k) \right] \quad (4.3.3)$$

Then we are ready to apply the least-squares method.

Remark. There is a principal difference between the output disturbance $v(t)$ and the residual $\varepsilon(t)$, although they appear at the same place in equations (4.3.2) and (4.3.3). The term $v(t)$ accounts for the effect of all the unmeasured disturbances acting at the process output; the term $\varepsilon(t)$ is used to account for model misfit which is a function of model parameters, $\varepsilon(t) = \varepsilon(t, \theta)$. Note that in equation (4.3.2) the parameters are fixed (unknown) true values; in equation (4.3.3) the parameters are the variables to be estimated.

For the model estimation input-output data are collected, where u_1 (drawing force) and u_2 (roll gap) are driven by two PRBS test signals, and u_3 (input thickness) is measured disturbance. Denote the data sequence as

$$Z^N := y(1) \ u_1(1) \ u_2(1) \ u_3(1) \ \dots \ y(N) \ u_1(N) \ u_2(N) \ u_3(N)$$

Then it is straightforward to extend the least-squares formula (4.2.2) for a 3-input 1-output FIR model as follows

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T \mathbf{y} \quad (4.3.4)$$

where

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix}, \quad \theta = \begin{bmatrix} g_{1,1} \\ \vdots \\ g_{1,n_1} \\ g_{2,1} \\ \vdots \\ g_{2,n_2} \\ g_{3,1} \\ \vdots \\ g_{3,n_3} \end{bmatrix} \\ \Phi &= \begin{bmatrix} u_1(n_1) & \cdots & u_1(1) & u_2(n_2) & \cdots & \cdots & u_3(n_3) & \cdots & u_3(1) \\ \vdots & & \vdots & \vdots & & & \vdots & & \vdots \\ u_1(N-1) & \cdots & u_1(N-n_1) & u_2(N-1) & \cdots & \cdots & u_3(N-1) & \cdots & u_3(N-n_3) \end{bmatrix} \end{aligned}$$

After the pretreatment of the data, a FIR model is estimated. Then the model fit is checked using a data sequence from another PRBS experiment. Figure 4.3.2 shows part of the measured exit thickness and the simulated one. The power of the simulation error is about 5% of that of the output. For control purposes this is an accurate model.

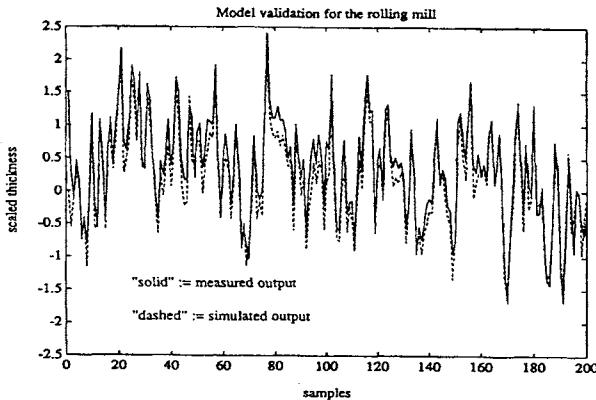


Figure 4.3.2 Model fit of the rolling mill

Feedback and feedforward controllers have been designed based on the identified model, and on extensive simulations of the control system. Figure 4.3.3 shows the control scheme. There is a large measurement delay at the output due to the placement of the thickness meter. In such a case the feedback controller is only effective at low frequencies, which means that only slow disturbances can be compensated for by the feedback loop. Thus, if possible, a feedforward controller should be used to compensate for fast disturbances.

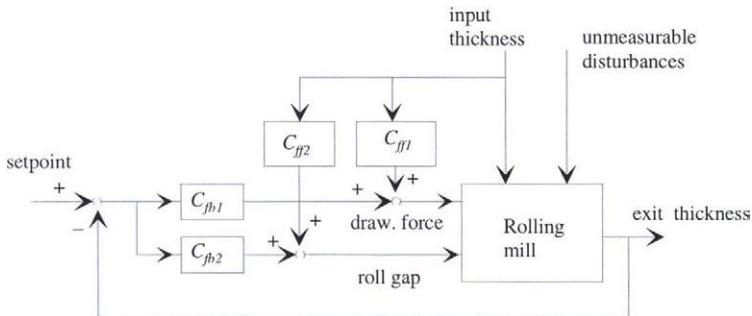


Figure 4.3.3 The control scheme for the rolling mill, C_{fb1} and C_{fb2} are the two feedback controllers, C_{ff1} and C_{ff2} are the two feedforward controllers.

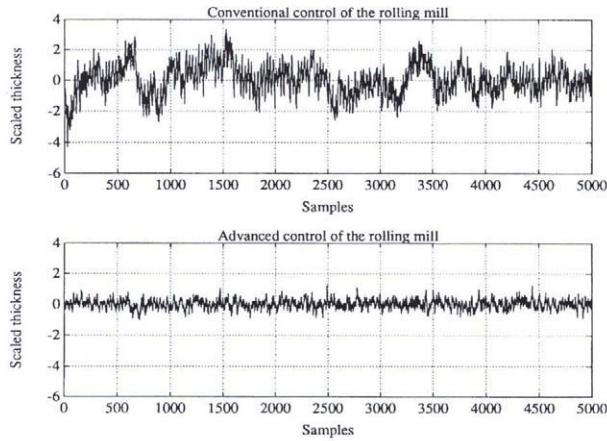


Figure 4.3.4 Manual control versus computer control of the rolling mill

According to simulation of the controlled system, a considerable reduction of thickness variation can be achieved by the designed controllers. The industrial tests have confirmed this. Figure

4.3.4 shows the measured strip thickness during manual control and during computer control respectively. A 70% reduction of standard deviation has been realized! This is considered to be a big achievement.

4.3.2 Identification of the Glass Tube Process

Let us recall the inputs and outputs selected for the glass drawing process:

Inputs (MV's):

$u_1(t)$: gas pressure

$u_2(t)$: drawing speed

Outputs (CV's):

$y_1(t)$: wall thickness

$y_2(t)$: diameter.

A general linear relation between the inputs and outputs can be described by a transfer function matrix

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} G_{11}^o(q) & G_{12}^o(q) \\ G_{21}^o(q) & G_{22}^o(q) \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} + \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} \quad (4.3.5)$$

or

$$\begin{cases} y_1(t) = G_{11}^o(q)u_1(t) + G_{12}^o(q)u_2(t) + v_1(t) \\ y_2(t) = G_{21}^o(q)u_1(t) + G_{22}^o(q)u_2(t) + v_2(t) \end{cases}$$

where $G_{ij}^o(q)$ is a rational of polynomials in the delay operator q^{-1} .

First we must parametrize the model. The generalization of the difference equation model leads to a so called matrix fraction description (MFD). For a 2-input 2-output process model a left MFD is defined as

$$G(q) = \begin{bmatrix} A_{11}(q) & A_{12}(q) \\ A_{21}(q) & A_{22}(q) \end{bmatrix}^{-1} \begin{bmatrix} B_{11}(q) & B_{12}(q) \\ B_{21}(q) & B_{22}(q) \end{bmatrix} \quad (4.3.6)$$

where $A_{ij}(q)$, $B_{ij}(q)$ are polynomials in the delay operator q^{-1} .

In general, parametrization of MIMO MFD models for identification is not an easy task which will not be discussed in this book. The diagonal form MFD is, however, simple and physically appealing. For the 2-input 2-output process, the diagonal form MFD is given as

$$G(q) = \begin{bmatrix} A_1(q) & 0 \\ 0 & A_2(q) \end{bmatrix}^{-1} \begin{bmatrix} B_{11}(q) & B_{12}(q) \\ B_{21}(q) & B_{22}(q) \end{bmatrix} \quad (4.3.7)$$

Applying this description to the general model (4.3.6) yields

$$\begin{cases} A_1^o(q)y_1(t) = B_{11}^o(q)u_1(t) + B_{12}^o(q)u_2(t) + A_1^o(q)v_1(t) \\ A_2^o(q)y_2(t) = B_{21}^o(q)u_1(t) + B_{22}^o(q)u_2(t) + A_2^o(q)v_2(t) \end{cases} \quad (4.3.8)$$

We note that in a diagonal form the model is decoupled into two two-input single-output sub-models; for each sub-model there is a common denominator polynomial. Thus the two sub-models can be estimated separately.

Let the degrees of the all the polynomials in a sub-model be equal and call this degree the order of the sub-model. Denote the orders of the two sub-models n_1 and n_2 respectively; then $[n_1, n_2]$ defines the model structure. Denote the input-output data sequence as

$$Z^N := y_1(1) \ y_2(1) \ u_1(1) \ u_2(1) \cdots \cdots \ y_1(N) \ y_2(N) \ u_1(N) \ u_2(N)$$

where $n = \max\{n_1, n_2\}$. Then the least-squares estimate of the parameters of the first sub-model which minimizes the loss function

$$V_1 = \sum_{t=n+1}^N (A_1(q)y_1(t) - [B_{11}(q)u_1(t) + B_{12}(q)u_2(t)])^2$$

is

$$\hat{\theta} = [\Phi_1^T \Phi_1]^{-1} \Phi_1^T \mathbf{y}_1 \quad (4.3.9)$$

where

$$\mathbf{y} = \begin{bmatrix} y_1(n_1 + 1) \\ y_1(n_1 + 2) \\ \vdots \\ y_1(N) \end{bmatrix}, \quad \theta_1 = \begin{bmatrix} a_{1,1} \\ \vdots \\ a_{1,n_1} \\ b_{11,1} \\ \vdots \\ b_{11,n_2} \\ b_{12,1} \\ \vdots \\ b_{12,n_3} \end{bmatrix}$$

$$\Phi_1 = \begin{bmatrix} y_1(n_1) & \cdots & y_1(1) & u_1(n_1) & \cdots & u_1(1) & \cdots & u_2(1) \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ y_1(N-1) & \cdots & y_1(N-n_1) & u_1(N-1) & \cdots & u_1(N-n_1) & \cdots & u_2(N-n_1) \end{bmatrix}$$

The same can be done for the second sub-model.

For the estimation of the model of this glass tube process, a PRBS experiment was performed. After the pretreatment of the data, 1269 samples are available. We shall use the first 600 samples for model estimation and the remaining 669 samples for model validation. The model structure [4, 4] is used. Figure 4.3.5 shows the result of model validation; the relative simulation errors are 39.8% and 41.7%. We find that this model is rather inaccurate. With this poor quality we feel insecure about the applicability of the model.

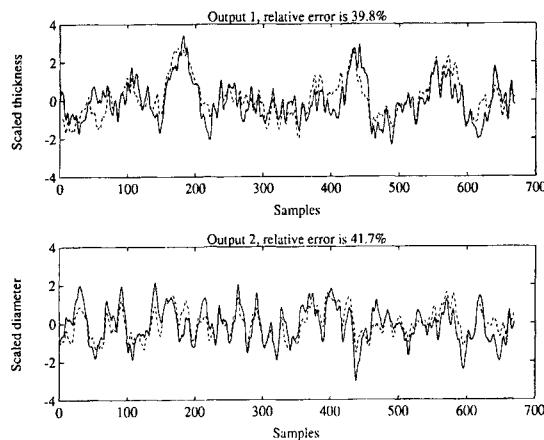


Figure 4.3.5 Model fit for the glass tube process "solid" := measured output; "dashed" := model output

Why does least-squares function well for the rolling mill process but perform poorly for the glass tube process? In the next section we are going to analyze the least-squares method and answer this question. Also, based on the analysis, ways to improve the least-squares method are highlighted.

At this point, we will congratulate you if the least-squares method does solve your problem in process modeling. You may stop reading and spend your time on other more important business. However, those readers who cannot obtain satisfactory results with the least-squares method, we encourage you to read on; this also holds for readers who would like to know more about recent progress in process identification.

4.4 Properties of the Least-Squares Estimator

In the previous section we have tested the least-squares method by using data from industrial processes. The FIR model of the rolling mill process performs very well; the transfer function

model of the glass process, however, is not accurate according to our model simulation. So this is a good moment to carry out some theoretical analysis on the least-squares method in order to find explanations of the behavior of the method and to search for better methods.

First, look back at Section 4.1. Assume that the data are generated by the process:

$$y(t) = \varphi(t)\theta^o + e(t) \quad (4.4.1)$$

where $e(t)$ is the noise, $\varphi(t)$ is the data vector

$$\varphi(t) = [x_1(t) \ x_2(t) \ \cdots \ x_n(t)]$$

and θ^o is the true parameter vector

$$\theta^o = \begin{bmatrix} \theta_1^o \\ \theta_2^o \\ \vdots \\ \theta_n^o \end{bmatrix}$$

For the analysis let us make two other assumptions:

- A1** The error term $e(t)$ is a stationary stochastic process with zero mean value ($Ee(t) = 0$)
- A2** The error $e(t)$ is uncorrelated with the signals of the data vector ($E\varphi^T(t)e(t) = 0$)

Under these assumptions, the least-squares estimate $\hat{\theta}$ from (4.1.7) and the weighted least-squares estimate $\hat{\theta}_w$ from (4.1.9) are random variables. Therefore, their accuracies can be measured by a number of statistical properties such as bias, error covariance, efficiency and consistency.

Common desirable properties are defined for the estimators:

Unbiased estimator: if for each sample number N

$$E\hat{\theta} = \theta^o$$

Consistent estimator:

if $N \rightarrow \infty$ then $\hat{\theta} \rightarrow \theta^o$ with probability 1

Efficient or minimum variance estimator: if for all unbiased estimators $\hat{\theta}^*$

$$\text{cov}(\hat{\theta}) \leq \text{cov}(\hat{\theta}^*), \text{ or, } \det[\text{cov}(\hat{\theta})] - \det[\text{cov}(\hat{\theta}^*)] \leq 0$$

where the covariance matrix is defined as

$$\text{cov}(\hat{\theta}) = E\{[\hat{\theta} - E(\hat{\theta})][\hat{\theta} - E(\hat{\theta})]^T\}$$

If the first and third properties hold only for $N \rightarrow \infty$, then they are called asymptotic unbiasedness and asymptotic efficiency.

Substituting (4.4.1) into (4.1.7) yields

$$\hat{\theta} = \theta^o + [\Phi^T \Phi]^{-1} \Phi^T \mathbf{e} \quad (4.4.2)$$

Taking the expectation on both sides of this equation, noting that $E\mathbf{e}(t) = 0$ and $E\mathbf{e}(t)\varphi(t-\tau) = 0$, we obtain

$$E\hat{\theta} = E\theta^o + E[(\Phi^T \Phi)^{-1} \Phi^T]E(\mathbf{e}) = \theta^o \quad (4.4.3)$$

Therefore *under assumption A1 and A2* the least-squares estimator is unbiased. A similar proof can be given for the weighted least-squares estimator:

$$E(\hat{\theta}_w) = \theta^o$$

The covariance matrix of the least-squares estimator is

$$\begin{aligned} \text{cov}(\hat{\theta}) &= E\{[\hat{\theta} - E(\hat{\theta})][\hat{\theta} - E(\hat{\theta})]^T\} \\ &= E\{[(\Phi^T \Phi)^{-1} \Phi^T \mathbf{e}] [(\Phi^T \Phi)^{-1} \Phi^T \mathbf{e}]^T\} \\ &= (\Phi^T \Phi)^{-1} \Phi^T E(\mathbf{e} \mathbf{e}^T) \Phi (\Phi^T \Phi)^{-1} \\ &= (\Phi^T \Phi)^{-1} \Phi^T R \Phi (\Phi^T \Phi)^{-1} \end{aligned} \quad (4.4.4)$$

where R is the $N \times N$ dimensional covariance matrix of the error \mathbf{e}

$$R = E(\mathbf{e} \mathbf{e}^T) \quad (4.4.5)$$

Similarly for the weighted least-squares estimators, we can show:

$$\text{cov}(\hat{\theta}_w) = (\Phi^T W \Phi)^{-1} \Phi^T W R W^T \Phi (\Phi^T W \Phi)^{-1} \quad (4.4.6)$$

If we know the error covariance R defined by (4.4.5), then it is instructive to use the weighting

$$W = R^{-1}$$

This means that we will give a data point a higher weighting when it is more reliable and a lower weighting otherwise. This choice of weighting matrix simplifies the expression of the error covariance matrix:

$$\text{cov}(\hat{\theta}_w)|_{W=R^{-1}} = (\Phi^T R^{-1} \Phi)^{-1} \quad (4.4.7)$$

We shall denote the corresponding estimate as $\hat{\theta}_{mv}$

$$\hat{\theta}_{mv} = (\Phi^T R^{-1} \Phi)^{-1} \Phi^T R^{-1} \mathbf{y} \quad (4.4.8)$$

This estimator is called the *minimum variance* estimator (or Markov estimator) which is efficient because for any other choice of weighting matrix W

$$\text{cov}(\hat{\theta}_{mv}) \leq \text{cov}(\hat{\theta}_w)$$

We refer to Chapter 4 of Söderström and Stoica (1989), for proofs of this relation.

The principle of minimum variance or efficient estimator is a good guideline for developing estimation methods. It can, however, hardly be applied in its present form to practical situations, because the error variance matrix is not available.

When the error $e(t)$ is white noise with zero mean and variance R , its covariance matrix is RI . In this case the least-squares estimator is a minimum variance estimator (efficient estimator). However, the assumption that the error is white noise is very restricted and not realistic.

Now we will show the consistency of the least-squares method. It is not difficult to verify that

$$\Phi^T \Phi = \sum_{t=n+1}^N \varphi^T(t) \varphi(t), \quad \Phi^T \mathbf{y} = \sum_{t=n+1}^N \varphi^T(t) y(t)$$

Using these equations we can rewrite the least-squares estimator (4.1.7) as

$$\hat{\theta} = \left[\frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t) \varphi(t) \right]^{-1} \frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t) y(t) \quad (4.4.9)$$

Combining (4.4.1) and (4.4.9) gives an expression of the parameter error

$$\hat{\theta} - \theta^\circ = \left[\frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t) \varphi(t) \right]^{-1} \frac{1}{N} \sum_{t=n+1}^N \varphi^T(t) e(t) \quad (4.4.10)$$

Under weak conditions (see Chapter 2 and Ljung, 1987) the sums in (4.4.10) tend to the corresponding expected values as $N \rightarrow \infty$

$$\lim_{N \rightarrow \infty} \frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t) \varphi(t) = E \varphi^T(t) \varphi(t), \quad \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=n+1}^N \varphi^T(t) e(t) = E \varphi^T(t) e(t) \quad (4.4.11)$$

Hence the parameter error tends to zero, i.e., the estimator is consistent, if

$E\varphi^T(t)\varphi(t)$ is nonsingular

$$E\varphi^T(t)e(t) = 0$$

The first condition is persistent excitation which is often satisfied and the second condition is assumption **A2**. Hence the consistency is proved. From (4.4.10) one can interpret consistency as the function of "averaging out" the effect of disturbance when taking more measurements.

The Properties of FIR Models

In finite impulse response (FIR) model estimation, we assume that the data is generated by

$$y(t) = \varphi(t)\theta^o + v(t) \quad (4.4.12)$$

where $v(t)$ is the output disturbance which is a zero mean stationary stochastic process,

$$\varphi(t) = [u(t-1) \cdots u(t-n)], \quad \theta^o = \begin{bmatrix} g_1^o \\ g_2^o \\ \vdots \\ g_n^o \end{bmatrix}$$

Assume that the length of the true impulse response is n and that the input is persistently exciting of order n , then from the above analysis we can say that *the least-squares estimate of the FIR model is unbiased and consistent if the disturbance $v(t)$ is independent of the input $u(t)$* .

Remark. In some literature the disturbance $v(t)$ is assumed to be white noise. This assumption is too restrictive and also not necessary for the properties of unbiasedness and consistency of the FIR model.

In practice, the independence between the input and disturbance is the consequence of open-loop identification tests, i.e., there is no feedback from $y(t)$ to $u(t)$. In turn this implies that *when the data is collected from a closed loop experiment, the FIR estimate will be biased*.

The Bias of the Least-Squares Estimator of Transfer Function Models

Assume that a SISO process is described as a n th order transfer function

$$\begin{aligned} y(t) &= \frac{B^o(q)}{A^o(q)}u(t) + v(t) \\ &= \frac{b_1^o q^{-1} + \cdots + b_n^o q^{-n}}{1 + a_1^o q^{-1} + \cdots + a_n^o q^{-n}} u(t) + v(t) \end{aligned} \quad (4.4.13)$$

where $v(t)$ is the output disturbance which is a realization of stationary stochastic process with zero mean. Assume that the data are collected from an open-loop test so that the disturbance $v(t)$ and the input $u(t)$ are independent.

In the least-squares method one rewrites this relation in a n th order equation error model as

$$y(t) = \varphi(t)\theta^o + e(t) \quad (4.4.14)$$

where

$$\begin{aligned} \varphi(t) &= [-y(t-1) \cdots -y(t-n) u(t-1) \cdots u(t-n)] \\ \theta^o &= \begin{bmatrix} a_1^o \\ \vdots \\ a_n^o \\ b_1^o \\ \vdots \\ b_n^o \end{bmatrix} \end{aligned}$$

We note that, unfortunately, the data vector $\varphi(t)$ is now correlated with the equation error $e(t)$, because the output $y(t)$ contains the disturbance $v(t)$ and $e(t)$ is correlated with $v(t)$ via

$$e(t) = A^o(q)v(t)$$

This implies that condition **A2** no longer holds for a transfer function model; therefore, *the least-squares estimator of the transfer function model (ARX model) is biased and not consistent.*

One may ask under what conditions the least-squares estimator of transfer function model will be unbiased and consistent. To answer this question let us do some more analysis. The least-squares estimator is given by

$$\hat{\theta} = \left[\frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t)\varphi(t) \right]^{-1} \frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t)y(t) \quad (4.4.15)$$

Combining (4.4.14) and (4.4.15) we obtain

$$\hat{\theta} - \theta^o = \left[\frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t)\varphi(t) \right]^{-1} \frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t)e(t)$$

$$\begin{aligned}
&= \left[\frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t) \varphi(t) \right]^{-1} \frac{1}{N-n} \sum_{t=n+1}^N \begin{bmatrix} -y^o(t-1) \\ \vdots \\ -y^o(t-n) \\ u(t-1) \\ \vdots \\ u(t-n) \end{bmatrix} e(t) \\
&\quad + \left[\frac{1}{N-n} \sum_{t=n+1}^N \varphi^T(t) \varphi(t) \right]^{-1} \frac{1}{N-n} \sum_{t=n+1}^N \begin{bmatrix} v(t-1) \\ \vdots \\ v(t-n) \\ 0 \\ \vdots \\ 0 \end{bmatrix} e(t)
\end{aligned}$$

where $y^o(t)$ is the noise-free output. When $N \rightarrow \infty$, the first term of the right hand side will tend to zero due to open-loop experiment; the second term will tend to

$$\left[E \varphi^T(t) \varphi(t) \right]^{-1} \begin{bmatrix} Ev(t-1)e(t) \\ \vdots \\ Ev(t-n)e(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

This is the asymptotic bias of the least-squares estimator, which will be zero only if the equation error $e(t)$ is white noise. This is an *extremely* restrictive condition and it cannot be satisfied in practical situations (think of the two industrial processes described in the previous section). Even if the output disturbance is white noise, which might happen if the disturbance consists of only measurement noise, the equation error will still be a colored noise because $e(t)$ is filtered $v(t)$ ($e(t) = A^o(q)v(t)$).

Assuming a white equation error is not realistic; one must be extremely lucky to encounter such a situation. However, trying to obtain white noise residuals is a way to an unbiased and consistent estimator (see Chapter 5).

Bias Error Distribution in the Frequency Domain

So far we have analyzed the properties of the least-squares method in the parameter space. A control engineer, however, is more concerned about the model behavior in the frequency domain, because for model applications such as control and simulation, the parameter set of a discrete-time model is merely a vehicle to arrive at a good transfer function estimate. If the model order is lower than the true process order, which often happens in applications, it does not make sense to talk about parameter errors; on the other hand, the error of the transfer function is still well defined.

Assume that an open-loop experiment has been performed. Let us write the loss function of the least-squares estimate for transfer function models as

$$V_{LS} = \frac{1}{N-1} \sum_{t=n+1}^N \varepsilon(t)^2 = \frac{1}{N-1} \sum_{t=n+1}^N (A(q)y(t) - B(q)u(t))^2$$

When $N \rightarrow \infty$

$$V_{LS} \rightarrow E(A(q)y(t) - B(q)u(t))^2$$

$$\begin{aligned} [\text{apply (4.4.13)}] &= E(A(q)[G^o(q)u(t) + v(t)] - B(q)u(t))^2 \\ &= EA^2(q)\{(G^o(q) - \frac{B(q)}{A(q)})u(t) + v(t)\}^2 \end{aligned}$$

$$[Eu(t)v(t) = 0] = EA^2(q)(G(q) - \frac{B(q)}{A(q)})^2 u^2(t) + EA^2(q)v^2(t)$$

Applying Parseval's formula (see Chapter 2) we obtain the asymptotic least-squares loss function in the frequency domain

$$V_{LS} \rightarrow \frac{1}{2\pi} \int_{-\pi}^{\pi} (G^o(e^{i\omega}) - \frac{B(e^{i\omega})}{A(e^{i\omega})})^2 A^2(e^{i\omega}) \Phi_u(\omega) d\omega + \frac{1}{2\pi} \int_{-\pi}^{\pi} A^2(e^{i\omega}) \Phi_v(\omega) d\omega \quad (4.4.17)$$

where $G^o(e^{i\omega})$ is the true transfer function, $\Phi_u(\omega)$ and $\Phi_v(\omega)$ are the spectra of the input and the disturbance. The first term of (4.4.17) shows that in the least-squares loss function the filter $A^2(q)$ is part of the weighting in the transfer function approximation. For most of the industrial processes, $A(q)$ is a highpass filter because $1/A(q)$ is lowpass. Therefore, *the effect of $A^2(q)$ is to impose heavy weighting at high frequencies*. This is certainly not desirable in model applications such as control and simulation.

Now we are in a position to give theoretical explanations to what happened in the two case studies:

1. The model of the rolling mill is accurate because the least-squares estimate of the FIR model is unbiased and consistent
2. The quality of the glass tube process is poor because the least-squares estimate of the transfer function model is biased; and the criterion emphasizes the fit of the transfer function at high frequencies at the cost of the fit at middle and low frequencies.

Logically, ways to improve the estimate of transfer function models are to remove the bias and to change the frequency weighting. For these purposes different methods, such as the generalized least-squares method, the instrumental variable method, the output error method and the prediction error method, have been proposed and studied in the past three decades. These methods will be studied in the next chapter.

4.5 Conclusions

In this chapter we have introduced the least-squares principle and we have shown how to use it in linear process identification. Simulation exercises have been used to demonstrate the method. The method has also been tested by two industrial processes, where it succeeded in the first application but failed in the second one. Then theoretical explanations were given for the two very different performances of the method. To end this chapter the following conclusions can be drawn:

- The least-squares method is simple to understand and easy to use due to the existence of the closed solution.
- The least-squares method will deliver an unbiased and consistent estimate if the process can be properly described by a FIR model and if the data are from an open-loop experiment.
- The least-squares method can deliver a good transfer function model if the level of the disturbance is low and the model order of the process is correct. Otherwise, the least-squares estimator is biased and the model fit at high frequencies is over-emphasized.

When the level of the disturbance increases, the performance of the least-squares method can be poor even when the correct model order for process model $G(q)$ is used. To solve the bias problem, one may use higher order in ARX model (see Chapter 6), or, other methods need to be used to obtain an unbiased model (see Chapter 5).

Chapter 5

Extensions of the Least-Squares Method

In the previous chapter we studied the least-squares (LS) method. From the two case studies and the analysis we have learned that the LS estimates of the parameters of a transfer operator or difference equation model are biased and not consistent; and the model fit in the frequency domain is not always good, because the high frequencies are over-emphasized. In the last three decades, much research has been done to modify and to extend the LS method in order to arrive at an unbiased and consistent estimator. In this chapter, we will discuss various ways of modifying the LS method.

We shall start from the frequency domain considerations. In this way, we can arrive at the Steiglitz-McBride method (Section 5.1). In Section 5.2 the output error method is studied. The instrumental variable (IV) method is introduced in Section 5.3. These methods can work well for an open-loop experiment but will yield biased estimates when closed-loop data is used. This problem can be overcome by a prediction error method which is studied in Section 5.4. Section 5.5 discusses the choice among various methods. Section 5.6 will revisit the problem of order selection. Section 5.7 will treat model validation. In Section 5.8, the glass tube process will be identified using some methods studied in this chapter. Section 5.9 treats the recursive parameter estimation. In concluding this chapter we shall point out some practical limitations of these methods for control applications.

5.1 Modifying the Frequency Weighting by Prefiltering

In control system design, we are more concerned about the errors of the transfer function in the frequency domain than the errors of parameters. Let us write down again the least-squares loss

function in the frequency domain as $N \rightarrow \infty$ (4.4.17)

$$V_{LS} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(G^o(e^{i\omega}) - \frac{B(e^{i\omega})}{A(e^{i\omega})} \right)^2 A^2(e^{i\omega}) \Phi_u(\omega) d\omega + \frac{1}{2\pi} \int_{-\pi}^{\pi} A^2(e^{i\omega}) \Phi_v(\omega) d\omega \quad (5.1.1)$$

Note that the transfer function error is weighted by the frequency weighting $A^2(e^{i\omega})\Phi_u(\omega)$.

Suppose that the desired frequency weighting is given as $W(e^{i\omega})$. Then based on the expression (5.1.1) we have at least two ways to realize this: *input design and prefiltering*.

When using the idea of input design, one should choose a test input such that its spectrum $\Phi_u(w)$ has the form

$$\Phi_u(\omega) = W^2(e^{i\omega})/A^2(e^{i\omega})$$

However, this cannot be done in a precise way because $A(q)$ is not known before parameter estimation. One may ask why not perform identification test and parameter estimation repeatedly or iteratively. The answer is that doing an identification test is often a very costly business.

If the input spectrum is the desired frequency weighting

$$\Phi_u(\omega) = W^2(e^{i\omega})$$

then we can eliminate the effect of $A^2(e^{i\omega})$ in the least-squares method by prefiltering. Denote $L(q)$ as a stable filter and perform prefiltering on the input-output data:

$$\begin{cases} u_f(t) = L(q)u(t) \\ y_f(t) = L(q)y(t) \end{cases} \quad (5.1.2)$$

By applying the least-squares method on the filtered data, the following loss function is minimized

$$\begin{aligned} V_{FLS} &= \frac{1}{N-n} \sum_{t=n+1}^N \varepsilon_f(t)^2 = \frac{1}{N-n} \sum_{t=n+1}^N [A(q)y_f(t) - B(q)u_f(t)]^2 \\ &= \frac{1}{N-n} \sum_{t=n+1}^N L^2(q) [A(q)y(t) - B(q)u(t)]^2 \end{aligned} \quad (5.1.3)$$

Assuming that the true process is

$$y(t) = \frac{B^o(q)}{A^o(q)} u(t) + v(t) = \frac{b_1^o q^{-1} + \cdots + b_n^o q^{-n}}{1 + a_1^o q^{-1} + \cdots + a_n^o q^{-n}} + v(t) \quad (5.1.4)$$

and following the same derivation as for (5.1.1) (see Section 4.4), we can show that as $N \rightarrow \infty$

$$\begin{aligned} V_{FLS} &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(G^o(e^{i\omega}) - \frac{B(e^{i\omega})}{A(e^{i\omega})} \right)^2 L^2(e^{i\omega}) A^2(e^{i\omega}) \Phi_u(\omega) d\omega \\ &\quad + \frac{1}{2\pi} \int_{-\pi}^{\pi} L^2(e^{i\omega}) A^2(e^{i\omega}) \Phi_v(\omega) d\omega \end{aligned} \quad (5.1.5)$$

In order to cancel $A^2(e^{i\omega})$ in the weighting the filter should be

$$L(q) = 1/A(q)$$

Of course, $A(q)$ is unknown. But this can be solved by the following iteration procedure, assuming the process is stable:

1. To start, perform a normal least-squares estimation, and denote the estimate as $\hat{G}^1(q) = \hat{B}^1(q)/\hat{A}^1(q)$.
2. Denote $\hat{G}^{(k)}(q) = \hat{B}^k(q)/\hat{A}^k(q)$ as the model estimate at iteration k . First filter the input $u(t)$ and the output $y(t)$ by $1/\hat{A}^k(q)$. Then apply the least-squares method on the filtered data.

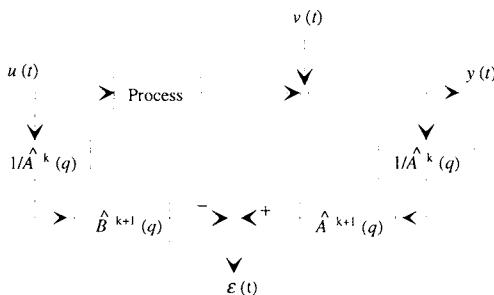


Fig. 5.1.1 Error generation for the Steiglitz-McBride method

If the iteration converges, we will get an estimate which is a local minimum of the following loss function

$$V_{oc} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(G^o(e^{i\omega}) - \frac{B(e^{i\omega})}{A(e^{i\omega})} \right)^2 \Phi_u(\omega) d\omega + \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_v(\omega) d\omega \quad (5.1.6)$$

This algorithm was proposed by Steiglitz and McBride (1965), where their motivation was to find a simple algorithm for minimizing the output error loss function (5.2.2). A block diagram of the method is shown in Figure 5.1.1. The method is numerically simple and has a straightforward frequency domain interpretation liked by engineers.

Exercise 5.1.1. Write a Matlab M file to implement the Steiglitz-McBride method.

Answer. The M file is as follows.

```
%%% smmethod.m; M file for Steiglitz-McBride method %%%
```

```

function [Ap, Bp] = smmmethod(U,Y,na,nb,Delay,iter)
%
% function [Ap, Bp] = smmmethod(U,Y,na,nb,Delay,iter)
%
% This function estimates model using Steiglitz-McBride method.
%
% Input arguments:
% U: Vector of input signal
% Y: Vector of output signal
% na: Degree of A polynomial (denominator)
% nb: Degree of B polynomial (numerator)
% Delay: Delay in samples
% iter: number of iterations
%
% Output arguments:
% Ap: Parameter vector of A polynomial (denominator)
% Bp: Parameter vector of B polynomial (numerator)
%
Yf = Y;
Uf = U;
for k = 1:iter
    THarx = arx([Yf Uf],[na nb Delay]);
    [Ap,Bp]=th2poly(THarx);
    Ap = fstab(Ap);
    Uf = filter(1,Ap,U);
    Yf = filter(1,Ap,Y);
end

```

The following exercise checks the performance of the Steiglitz-McBride method.

Exercise 5.1.2. Given the process as

$$y(t) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} u(t) + v(t)$$

Simulate the process using a GBN signal with mean switching time of 8 samples. Add three kind of noises:

1) white noise $v(t) = e(t)$

2) wide band colored noise: $v(t) = \frac{1}{1 - 0.6q^{-1}}e(t)$

3) narrow band colored noise: $v(t) = \frac{1}{1 - 0.9q^{-1}}e(t)$

and create three data sets of 500 samples. Adjust the variances of the disturbances so that the noise-to-signal ratios are 10% in power. Then calculate 2nd order Steiglitz-McBride models and 2nd order ARX models using each data set and compare their step responses with the true one. In what situations does the Steiglitz-McBride method perform better than the least-squares method?

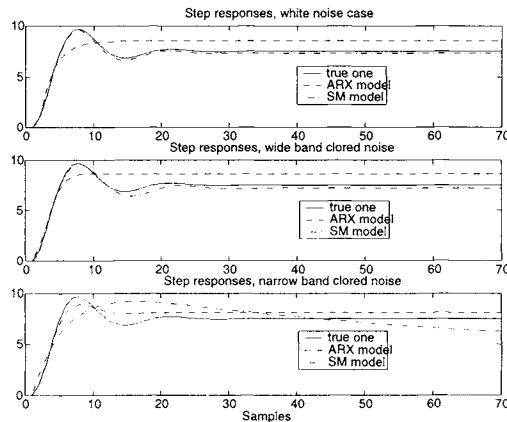


Figure 5.1.2 Compare the Steiglitz-McBride method with the least-squares (ARX) method

Answer. The M file is as follows. The step responses are shown in Figure 5.1.2. When the output disturbance is white noise or wide band colored noise, the Steiglitz-McBride model is much more accurate than the ARX model. When the output disturbance is narrow band colored noise, the Steiglitz-McBride model is not better than the ARX model or can be even less accurate. Note that the results are different for different runs, but Figure 5.1.2 shows a typical comparison.

```
%%% testsm.m; M file for testing Steiglitz-McBride Method %%%
function [ ] = testsm( )
% Simulate the process using GBN signal with Tsw = 8
Ao = [1 -1.5 0.7];
Bo = [0 1 0.5];
U = gbneng(500,8);
Yo = filter(Bo,Ao,U);
% Add disturbances
Vw = randn(500,1);
Vc1=filter(1,[1 -0.6],randn(500,1));
Vc2=filter(1,[1 -0.9],randn(500,1));
Vw=Vw/std(Vw)*std(Yo)*sqrt(0.1);
```

```

Vc1=Vc1/std(Vc1)*std(Yo)*sqrt(0.1);
Vc2=Vc2/std(Vc2)*std(Yo)*sqrt(0.1);
Yw=Yo+Vw;
Yc1=Yo+Vc1;
Yc2=Yo+Vc2;
% Estimate the models
THarxw=arx([Yw U],[2 2 1]);
[Aarxw,Barxw]=th2poly(THarxw);
[Asmw,Bsmw]=smmethod(U,Yw,2,2,1,20);
THarxc1=arx([Yc1 U],[2 2 1]);
[Aarxc1,Barxc1]=th2poly(THarxc1);
[Asmc1,Bsmc1]=smmethod(U,Yc1,2,2,1,20);
THarxc2=arx([Yc2 U],[2 2 1]);
[Aarxc2,Barxc2]=th2poly(THarxc2);
[Asmc2,Bsmc2]=smmethod(U,Yc2,2,2,1,20);
% Calculate step responses and plot them
STPo=filter(Bo,Ao,ones(70,1));
STParxw=filter(Barxw,Aarxw,ones(70,1));
STPsmw=filter(Bsmw,Asmw,ones(70,1));
STParxc1=filter(Barxc1,Aarxc1,ones(70,1));
STPsmc1=filter(Bsmc1,Asmc1,ones(70,1));
STParxc2=filter(Barxc2,Aarxc2,ones(70,1));
STPsmc2=filter(Bsmc2,Asmc2,ones(70,1));
t=1:70;
figure;
subplot(311), plot(t,STPo,'-r',t,STParxw,'--b',t,STPsmw,'-.k')
legend('true one','ARX model','SM model')
title('Step responses, white noise case')
subplot(312), plot(t,STPo,'-r',t,STParxc1,'--b',t,STPsmc1,'-.k')
legend('true one','ARX model','SM model')
title('Step responses, wide band clored noise')
subplot(313), plot(t,STPo,'-r',t,STParxc2,'--b',t,STPsmc2,'-.k')
legend('true one','ARX model','SM model')
title('Step responses, narrow band clored noise')
xlabel('Samples')

```

5.2 Output Error Method

Let us now go back to a fundamental problem — the choice of criterion (loss function) for model estimation. We have mentioned before that equation error criterion used in the least-squares method is not a natural choice; we choose it because it is easy to compute and simple

to comprehend. We have proposed the output error criterion for model order selection because it is closer to model applications as required in control and simulation. Now we will use this criterion for model estimation.

Denote the process model as

$$G(q) = \frac{B(q)}{A(q)} = \frac{b_1 q^{-1} + \cdots + b_n q^{-n}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}}$$

The output error is defined as

$$\varepsilon_{oe}(t) = y(t) - \frac{B(q)}{A(q)} u(t) = y(t) - \hat{y}(t) \quad (5.2.1)$$

Given the input-output data sequence as

$$Z^N := y(1) \ u(1) \ \cdots \ \cdots \ y(N) \ u(N)$$

and assuming that the process order n is known. Then the output error method estimates parameters by minimizing the loss function

$$V_{oe}^N = \frac{1}{N-n} \sum_{t=n+1}^N \varepsilon_{oe}(t)^2 \quad (5.2.2)$$

The block diagram for generating the output error is given in Figure 5.2.1.

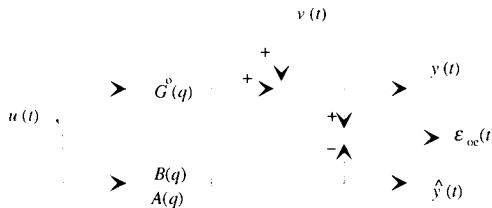


Figure 5.2.1 Output error generation

Note that the output error $\varepsilon_{oe}(t)$ is *nonlinear* in the parameters of the $A(q)$ polynomial; the consequence of this nonlinearity is that there exists no analytical solution to this minimization problem. Therefore, a numerical search algorithm is needed to find a minimum; this is much more time consuming than the least-squares method. Problems such as local minima and non-convergence can occur. Also the theoretical analysis becomes more involved.

Before showing how to calculate the estimate, let us study the properties of the output error method for open-loop data. First the consistency is shown. Assume that

1. The true process is given by (5.1.4) where $v(t)$ is a stationary stochastic process with zero mean
2. The model order n is correct
3. The input is persistently exciting with an order higher than $2n$
4. The minimization of (5.2.2) converges to the global minimum for all N .

Denote the model error as

$$\Delta G(q) = G^o(q) - \hat{G}(q)$$

then if $N \rightarrow \infty$

$$\begin{aligned} V_{\text{oe}}^N &\rightarrow EV_{\text{oe}} = E[\Delta G(q)u(t) + v(t)]^2 \\ &= [\Delta G(q)]^2 Eu^2(t) + 2\Delta G(q)E[u(t)v(t)] + Ev^2(t) \\ &= [\Delta G(q)]^2 Eu^2(t) + Ev^2(t) \end{aligned} \quad (5.2.3)$$

The last equality in (5.2.3) is due to an open-loop experiment. In such a case the test input is not correlated with the output disturbance. If the model order is correct and the minimization finds the global minimum, we have

$$[\Delta G(q)]^2 \rightarrow 0 \text{ as } N \rightarrow \infty$$

which implies that

$$\hat{G}(q) \rightarrow G^o(q) \text{ as } N \rightarrow \infty \quad (5.2.4)$$

Because model order is correct and hence model parameters are uniquely determined (identifiable), (5.2.4) is equivalent to saying that

$$\hat{\theta} \rightarrow \theta^o(q) \text{ as } N \rightarrow \infty \quad (5.2.5)$$

where

$$\theta = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Hence the output error method is consistent under relatively weak conditions. In plain words, when using the output error method the effect of the output disturbance can be averaged out,

provided that the input is not correlated with the disturbance (in the case of open-loop test). For mathematically more rigorous proof of the consistency, see Chapter 8 of Ljung (1987).

If the model order is lower than the true one (undermodeling), which can happen in applications, the model will be biased; we shall see how the model error is distributed for the output error method model. Applying Parseval's formula on (5.2.3) we obtain for $N \rightarrow \infty$,

$$V_{\text{oe}}^N \rightarrow \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(G^o(e^{i\omega}) - \frac{B(e^{i\omega})}{A(e^{i\omega})} \right)^2 \Phi_u(\omega) d\omega + \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_v(\omega) d\omega \quad (5.2.6)$$

Hence using the output error criterion, the bias error of transfer function is weighted by the input spectrum, which can be manipulated by input design. This is an advantage of the method.

Summarizing, in an open-loop operation, the output error method is consistent if the order of model $G(q)$ is correct. If the model order is lower than the true one, the error of the transfer function estimate is weighted by the input spectrum. In other words, if the order is correct, the identified model will be accurate; if the model order is too low, the model error can be effectively affected by input design. Remember, however, that a precondition for these nice properties is the global convergence of the numerical search algorithm. In practical situations this is not always guaranteed.

A Minimization Algorithm

The loss function V_{oe} is nonlinear in the parameters of $A(q)$ and it must be minimized using a numerical search routine. There exists a large variety of numerical methods for optimization, such as steepest descent, Newton-Raphson, Marquardt, and stochastic approximation, to name a few. We will not, in this book, become too involved in numerical optimization problems. There are two reasons for this. First, many methods are available as standard routines from different computer packages for identification and control design. Secondly, numerical difficulties can be reduced to a great extent by exploiting the model structure in our identification method (see Chapters 6 and 7). Here we will introduce the Gauss-Newton method which is well suited for the sum of squares criterion. The purpose is to give some insight into numerical optimization.

Denote $\hat{\theta}^k$ as the estimate at k th iteration. Assume that $\hat{\theta}^k$ is close to a local minimum. For parameter vectors close to $\hat{\theta}^k$ we can approximate the output error residual by truncating the higher order terms of its Taylor expansion so that

$$\begin{aligned} \varepsilon_{\text{oe}}(t, \theta) &\approx \varepsilon(t, \hat{\theta}^k) + \frac{\partial \varepsilon(t, \theta)}{\partial \theta^T} \Big|_{\theta=\hat{\theta}^k} (\theta - \hat{\theta}^k) \\ &= \varepsilon(t, \hat{\theta}^k) + \varphi^k(t)(\theta - \hat{\theta}^k) \\ &= -[\varphi^k(t)\hat{\theta}^k - \varepsilon(t, \hat{\theta}^k)] + \varphi^k(t)\theta \end{aligned} \quad (5.2.7)$$

Here $\varphi^k(t)$ is the gradient of the output error residual (5.2.1)

$$\begin{aligned}\varphi^k(t) &= \frac{\partial \varepsilon(t, \theta)}{\partial \theta^T} \Big|_{\theta=\hat{\theta}^k} \\ &= \left[\frac{\partial \varepsilon(t, \theta)}{\partial a_1} \Big|_{a_1=\hat{a}_1^k} \cdots \frac{\partial \varepsilon(t, \theta)}{\partial a_n} \Big|_{a_n=\hat{a}_n^k} \frac{\partial \varepsilon(t, \theta)}{\partial b_1} \Big|_{b_1=\hat{b}_1^k} \cdots \frac{\partial \varepsilon(t, \theta)}{\partial b_n} \Big|_{b_n=\hat{b}_n^k} \right] \\ &= \left[\frac{\hat{B}^k(q)}{\hat{A}^k(q)^2} u(t-1) \cdots \frac{\hat{B}^k(q)}{\hat{A}^k(q)^2} u(t-n) \frac{-1}{\hat{A}^k(q)} u(t-1) \cdots \frac{-1}{\hat{A}^k(q)} u(t-n) \right]\end{aligned}$$

Under this approximation, we find that the error is linear in the parameters; hence the least-squares method can be used to find the refined estimate. Comparing (5.2.7) to linear regression equation (4.2.10), we find that now $[\varphi^k(t)\hat{\theta}^k - \varepsilon(t, \hat{\theta}^k)]$ takes the position of $y(t)$ and $\varphi^k(t)$ takes the position of $\varphi(t)$. Therefore the new estimate is given by

$$\begin{aligned}\hat{\theta}^{k+1} &= \left[\sum_{t=1}^N [\varphi^k(t)]^T \varphi^k(t) \right]^{-1} \sum_{t=1}^N [\varphi^k(t)]^T [\varphi^k(t)\hat{\theta}^k - \varepsilon(t, \hat{\theta}^k)] \\ &= \hat{\theta}^k - \left[\sum_{t=1}^N [\varphi^k(t)]^T \varphi^k(t) \right]^{-1} \sum_{t=1}^N [\varphi^k(t)]^T \varepsilon(t, \hat{\theta}^k)\end{aligned}\quad (5.2.8)$$

Some conditions should be satisfied in order to make the algorithm work. First the models from each iteration must be stable, because otherwise the elements of $\varphi^k(t)$ will be unbounded. If an unstable intermediate model is obtained, we can approximate it by a stable one. Secondly $\hat{A}^k(q)$ and $\hat{B}^k(q)$ must have no common factor (coprime) and the input must be persistently exciting with order $2n$. This is to ensure that the matrix

$$\sum_{t=1}^N [\varphi^k(t)]^T \varphi^k(t)$$

is nonsingular. These conditions are not very restrictive for practical applications.

The least-squares estimate, or the estimate from the Steiglitz-McBride method, can be used to start the iteration. This method is also called a quasilinearization algorithm due to the approximation in (5.2.7).

Remark. There is a slight change in data arrangement: the factor $1/(N-n)$ is replaced by $1/N$ and the summation is over 1 to N . This is for the ease of notation and will be adopted in the rest of the book for the analysis. If $N \gg n$, the effect of this arrangement is negligible. When implementing the estimation algorithm, the old arrangement should be used, because it is more accurate.

Output Error Method versus Steiglitz-McBride Method

Some readers might have found similarities between the output error method and the Steiglitz-McBride method; some readers might even think that they are identical. This is not surprising because the Steiglitz-McBride method was proposed in order to approximate the output error method. There exists, however, a conceptual difference between the two methods. The output error method aims at minimizing the output error loss function (5.2.2); the Steiglitz-McBride method is an *ad hoc* iteration scheme which is not a minimization procedure, in other words, the output loss function will not necessarily decrease at each iteration. Therefore, if the latter converges, one will find a local minimum of the output error loss function. Söderström and Stoica (1981) have shown that if the disturbance $v(t)$ is white noise and if the model order is correct, then when the Steiglitz-McBride iteration converges, it will converge to the global minimum of the loss function (5.2.2). The requirement for a white noise disturbance is a restrictive condition. Söderström and Stoica (1982) have shown that if the test input $u(t)$ (not disturbance) is white noise, then the output error method will converge to the global minimum of the loss function; otherwise, local minima may exist. In general, under the same conditions, an output error model of a process will be more accurate than a Steiglitz-McBride model.

The Steiglitz-McBride method is simple to implement and readily understood. But its performance is not good when the disturbance has a narrow band lowpass spectrum. The following exercise will compare the performance of output error method with Steiglitz-McBride method.

Exercise 5.2.1. Given the same process and disturbances as in Exercise 5.1.2, calculate 2nd order Steiglitz-McBride models and 2nd order output error models using each data set and compare their step responses with the true one. Compare the accuracy of the two models for each situation.

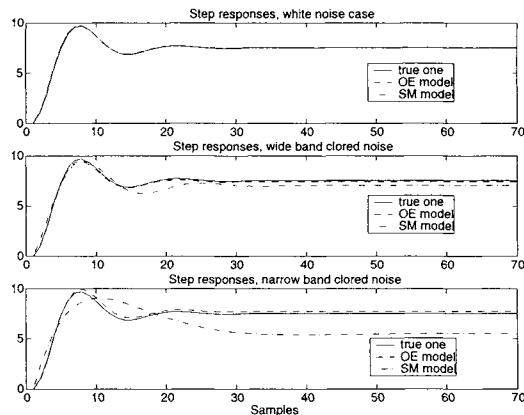


Figure 5.2.2 Compare output error method with Steiglitz-McBride method

Answer. The M file name is given below. The step responses are shown in Figure 5.2.2. When

the disturbance is white noise, the accuracy of the two methods are similar; when the disturbance is colored, output error models are more accurate than Steiglitz-McBride models.

```
%%% testoe.m; M file for testing output error method %%%
function [ ] = testoe( )
% Simulate the process using GBN signal with Tsw = 8
Ao = [1 -1.5 0.7];
Bo = [0 1 0.5];
U = gbneng(500,8);
Yo = filter(Bo,Ao,U);
% Add disturbances
Vw = randn(500,1);
Vc1=filter(1,[1 -0.6],randn(500,1));
Vc2=filter(1,[1 -0.9],randn(500,1));
Vw=Vw/std(Vw)*std(Yo)*sqrt(0.1);
Vc1=Vc1/std(Vc1)*std(Yo)*sqrt(0.1);
Vc2=Vc2/std(Vc2)*std(Yo)*sqrt(0.1);
Yw=Yo+Vw;
Yc1=Yo+Vc1;
Yc2=Yo+Vc2;
% Estimate the models
THoew=oe([Yw U],[2 2 1]);
[a,Boew,c,d,Aoew]=th2poly(THoew);
[Asmw,Bsmw]=smmethod(U,Yw,2,2,1,20);
THoec1=oe([Yc1 U],[2 2 1]);
[a,Boec1,c,d,Aoec1]=th2poly(THoec1);
[Asmc1,Bsmc1]=smmethod(U,Yc1,2,2,1,20);
THoec2=oe([Yc2 U],[2 2 1]);
[a,Boec2,c,d,Aoec2]=th2poly(THoec2);
[Asmc2,Bsmc2]=smmethod(U,Yc2,2,2,1,20);
% Calculate step responses and plot them
% Omitted here ... %
```

In Section 4.2.3 we have suggested that simulation or output error criterion be used as a tool for model order selection. Now that the output error criterion is the same as the criterion of model validation, we should expect a result different from that of the least-squares method. The following exercise studies order selection.

Exercise 5.2.2. Given the same process as in Exercise 4.2.3:

$$\begin{aligned}y(t) &= \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} u(t) + v(t) \\v(t) &= \frac{1}{1 - 0.9q^{-1}} e(t)\end{aligned}$$

Simulate the process using a GBN signal with mean switching time of 8 samples. Adjust the variance of $v(t)$ so that the noise-to-signal ratios are:

$$\text{var}(v(t))/\text{var}(y_o(t)) = 0.1, 1, 10, 50\%$$

Generate 500 samples for model estimation and the second 500 samples of fresh data for order selection. Estimate the output error models with various order. Plot the loss function V_{oe} as a function of model order for different noise-to-signal ratios and use both estimation data and fresh data. Check if the true order 2 can be found from this kind of test. What is the difference between the loss function using fresh data and that using estimation data?

Answer. The M file name is `oeordersel.m` that is very similar to `arxordersel.m` and hence it is not shown here. The loss functions are shown in Figure 5.2.3. The true order can be found in all cases when the fresh data sets are used. After the true order has passed, the loss function using fresh data increases slowly with the order. But the loss function using estimation data will continue to decrease.

Note the difference between the two data sets in order selection. When the fresh data set is used, the true order can be found; when the order is higher than the true one, the criterion will increase, which means that the model error will increase. This is due to the fact that the variance of the model (caused by the disturbance) increases with model order; see Chapter 6 for explanations. However, when the estimation data is reused in order selection, the criterion will continuously decrease with the order. This is because the model is over fitting the disturbance. This comparison shows that when estimation data set is used in order selection, it may lead to the selection of a too high order.

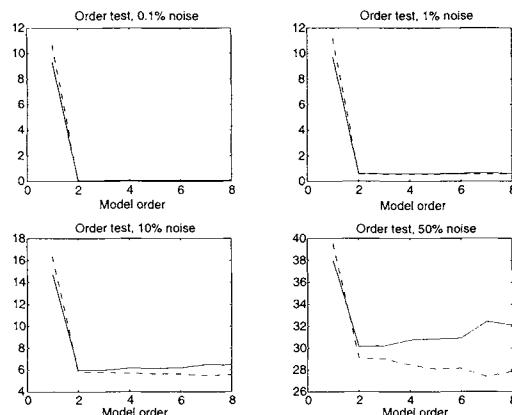


Figure 5.2.3 Order selection for output error method. The solid lines are the output error loss function using fresh data sets, the dashed lines are those using estimation data sets

5.3 Instrumental Variable (IV) Methods

Another way of modifying the least-squares method to overcome the bias problem is to use *instrumental variable* (IV) methods. Write down the equation error model used in the LS method:

$$A(q)y(t) = B(q)u(t) + \varepsilon(t) \quad (5.3.1)$$

We have shown that this model can be written as the linear regression form

$$y(t) = \varphi(t)\theta + \varepsilon(t) \quad (5.3.2)$$

where $\varphi(t)$ is the data vector

$$\varphi(t) = [-y(t-1), \dots -y(t-n) \quad u(t-1) \dots u(t-n)]$$

and θ is the parameter vector

$$\theta = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Then the LS estimate which minimizes the loss function is

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

Assume that the true process is given by

$$y(t) = \varphi(t)\theta^o + e(t) \quad (5.3.3)$$

see (4.4.13) and (4.4.14). The parameter errors can be determined by

$$\hat{\theta} - \theta^o = \left[\frac{1}{N} \sum_{t=1}^N \varphi^T(t) \varphi(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi^T(t) e(t)$$

When N tends to infinity it becomes

$$\hat{\theta} - \theta^o = [E\varphi^T(t)\varphi(t)]^{-1} E\varphi^T(t)e(t)$$

This is the asymptotic bias which will not be zero unless $e(t)$ is white noise.

Now assume that $Z(t)$ is a signal vector with dimension $1 \times 2n$ (the same dimension as $\varphi(t)$), the entries of which are signals correlated with the input $u(t)$ but *uncorrelated* with the equation disturbance $e(t)$. Then we may try to estimate the parameters by exploiting this property. Correlating the both sides of equation (5.3.2) with $Z(t)^T$ using the data for $t = 1, 2, \dots, N+n$ we obtain

$$\frac{1}{N} \sum_{t=1}^N Z(t)^T y(t) = \frac{1}{N} \sum_{t=1}^N Z(t)^T \varphi(t) \theta^o + \frac{1}{N} \sum_{t=1}^N Z(t)^T e(t) \quad (5.3.4)$$

Since $Z(t)$ is assumed uncorrelated with equation disturbance $e(t)$, for large N we have:

$$\frac{1}{N} \sum_{t=1}^N Z(t)^T e(t) \approx 0$$

Then equation (5.3.4) gives rise to an estimate

$$\hat{\theta} = \left[\frac{1}{N} \sum_{t=1}^N Z(t)^T \varphi(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N Z(t)^T y(t) \quad (5.3.5)$$

This is called the *basic instrumental variable* (IV) estimate of θ . The elements of the vector $Z(t)$ are called the *instrument variables*. If $Z(t)$ is replaced by $\varphi(t)$ we can rederive the LS estimate.

It is easy to verify that the parameter errors are given by

$$\hat{\theta} - \theta^o = \left[\frac{1}{N} \sum_{t=1}^N Z(t)^T \varphi(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N Z(t)^T e(t)$$

Due to the definition of $Z(t)$, the errors tend to zero when N tends to infinity, and the instrumental variable estimator is therefore consistent.

There are many ways to choose the instrument variable vector $Z(t)$. All the choices will give consistent estimates; the differences lie in the corresponding covariances. One possibility for choosing the instruments is the following:

$$Z(t) = [-x(t-1) \ \dots \ -x(t-n)u(t-1) \ \dots \ u(t-n)] \quad (5.3.6)$$

where the signal $x(t)$ is obtained by filtering the input

$$x(t) = \frac{D(q)}{C(q)} u(t) \quad (5.3.7)$$

Again, the polynomials $C(q)$ and $D(q)$ can be chosen in many ways. One special choice is to let $C(q)$ and $D(q)$ be the LS estimates of $A(q)$ and $B(q)$ respectively. Another special choice is to use delayed inputs

$$Z(t) = [u(t-1) \ \dots \ u(t-n) \ \dots \ u(t-2n)] \quad (5.3.8)$$

The block diagram of the IV method is shown in Figure 5.3.1

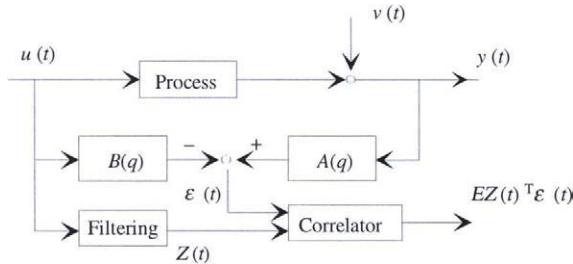


Figure 5.3.1 Instrumental variable methods

Logically, one will ask what the optimal choice of instrumental variables is. It can be shown (Söderström and Stoica, 1989, Chapter 8) that the optimal $Z(t)$ which gives the smallest covariance of $\hat{\theta}$ can be generated by the true transfer function $G^o(q)$ and the true disturbance filter $H^o(q)$. Because these true models are unknown, some iterations are inevitable for approximating an optimal $Z(t)$. We say *an* optimal $Z(t)$ instead of *the* optimal $Z(t)$ because a nonsingular linear transformation of $Z(t)$ has no influence on the estimate. It can easily be seen from (5.3.5) that a change of instruments from $Z(t)$ to $TZ(t)$, where T is a nonsingular matrix, will not change $\hat{\theta}$.

The advantages of the IV-methods are: numerical simplicity and the consistent estimate. So, in general, this method is superior to the least-squares method. However, for the implementation of an optimal IV, iterations are needed; this means that the advantage of numerical simplicity will be lost. For the higher model accuracy, one may use a prediction error method which will be studied in the following section.

If the model order is lower than the process order, it is not clear in what sense an IV estimate approximates the process transfer function. In this case, we would prefer to use an output error method, because we can easily influence the model misfit in the frequency domain.

Exercise 5.3.1. Given the process and disturbances the same as in Exercises 5.1.2 and 5.2.1, calculate 2nd order IV models and 2nd order output error models using each data set and compare their step responses with the true one. Compare the accuracy of the two models for each situation.

Answer. The command `iv4` of Matlab Identification Toolbox is used for the IV method. The M file name is `testiv.m` which is very similar to `testoe.m` in Exercise 5.2.1. Hence it will not be shown here. The step responses are shown in Figure 5.3.2. For all the situations, the accuracy of the IV models are similar to that of the output error models.

So far we have assumed that the identification test is carried out in an open-loop operation. In a closed loop-test, all the above three methods will be biased, hence not consistent, because the process input $u(t)$ is *correlated* with the disturbance through feedback. It is possible, however, to modify the IV method by using an IV vector $Z(t)$ in which the process input is replaced by

an external input (input from outside the loop) which is not correlated with the disturbance; see Söderström and Stoica (1989).

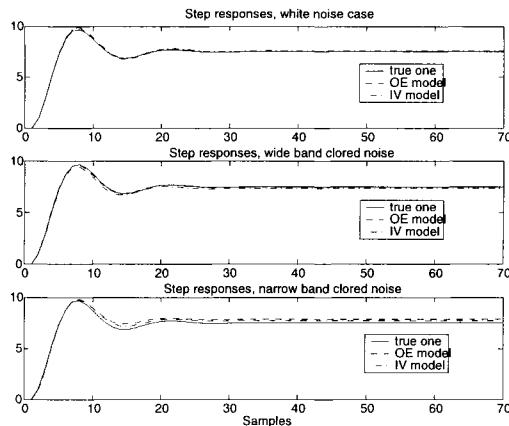


Figure 5.3.2 Comparison of the IV method and the output error method

5.4 Prediction Error Methods

In the development of the Steiglitz-McBride method, the output error method and the instrumental variable methods, one has aimed at obtaining a good estimate of the process model. Under realistic conditions, these methods can give consistent estimates of the process transfer function, which is a big step forward from the biased least-squares method. However, at least three areas can be improved in parameter estimation: (1) to maintain the consistency in the case of closed-loop tests, (2) to obtain minimum variance or efficient estimate and (3) to estimate a disturbance model. The first two are needed again for a good process model; and the third is for optimal disturbance reduction in control. Prediction error methods can meet these requirements.

5.4.1 Viewing the Least-Squares Method as a Prediction Error Method

Rewrite the equation error or ARX model (5.3.1) as

$$\begin{aligned} y(t) &= -a_1 y(t-1) - \cdots - a_n y(t-n) + b_1 u(t-1) + \cdots + b_n u(t-n) + \varepsilon(t) \\ &= [1 - A(q)]y(t) + B(q)u(t) + \varepsilon(t) \end{aligned} \quad (5.4.1)$$

Neglecting the equation error $\varepsilon(t)$ in this model, one can predict the output at time t using *both* the input u and the output y at $t - 1, t - 2, \dots$ as

$$\begin{aligned}\hat{y}(t) &= \varphi(t)\theta \\ &= [1 - A(q)]y(t) + B(q)u(t)\end{aligned}\quad (5.4.2)$$

Hence the equation error

$$\varepsilon(t) = y(t) - \hat{y}(t) \quad (5.4.3)$$

can be interpreted as a prediction error. See Figure 5.4.1 for the block diagram of this interpretation. Therefore, the LS method determines the parameters such that the sum of the squared prediction errors is minimized. We know from Chapter 4 that the LS estimate will be unbiased and efficient (minimum variance) if the equation disturbance $e(t)$ in (5.3.3) is zero mean white noise. In practice, however, the equation error $e(t)$ with a moderate order is not white noise, which causes the bias.

The idea behind different prediction error methods is to model the equation (or output) error using shaping filters in order to arrive at a consistent and efficient estimate. Different forms of disturbance filters lead to different names of methods in the identification literature.

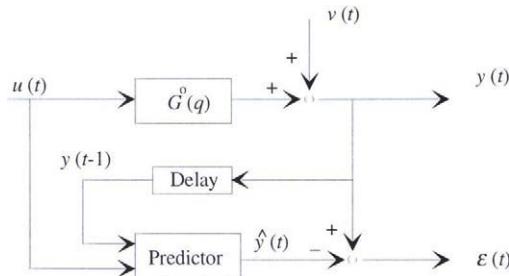


Fig. 5.4.1 Viewing the LS method as a prediction error method

5.4.2 Generalized Least-Squares (GLS) Method

This method was proposed by Clarke (1967), where he extended the equation error model and assumed that the true process is given by

$$A^o(q)y(t) = B^o(q)u(t) + \frac{1}{D^o(q)}e(t) \quad (5.4.4)$$

or

$$y(t) = \frac{B^o(q)}{A^o(q)}u(t) + \frac{1}{A^o(q)D^o(q)}e(t)$$

where

$$\begin{aligned} A^o(q) &= 1 + a_1^o q^{-1} + \cdots + a_n^o q^{-n} \\ B^o(q) &= b_1^o q^{-1} + \cdots + b_n^o q^{-n} \\ D^o(q) &= 1 + d_1^o q^{-1} + \cdots + d_{n_d}^o q^{-n_d} \end{aligned}$$

and $e(t)$ is white noise with zero mean and variance R . So the equation disturbance is assumed to be an AR (autoregressive) process. Rewrite equation (5.4.4) as

$$D^o(q)A^o(q)y(t) = D^o(q)B^o(q)u(t) + e(t) \quad (5.4.5)$$

This enlarged equation has a white noise disturbance $e(t)$. From the study of the least-squares method, we know that consistent and efficient estimates of a_i^o , b_i^o , d_i^o can be obtained by minimizing the loss function

$$V_{GLS} = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t) = \frac{1}{N} \sum_{t=1}^N (D(q)[A(q)y(t) - B(q)u(t)])^2 \quad (5.4.6)$$

This implies that, in the identification a model should be used which has the same structure as the true process

$$D(q)A(q)y(t) = D(q)B(q)u(t) + \varepsilon(t) \quad (5.4.7)$$

where $\varepsilon(t)$ is the residual; see Figure 5.4.2.

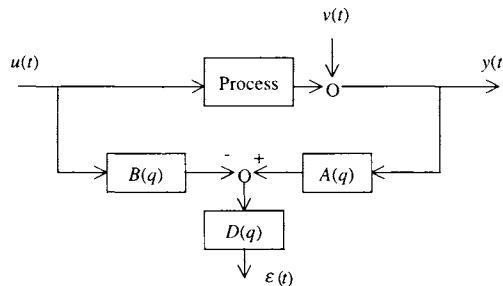


Figure 5.4.2 Error generation of the generalized least-squares method

The minimization of the loss function (5.4.6) has no analytical solution because the error $\varepsilon(t)$ is nonlinear in the parameters. We note, however, that the error $\varepsilon(t)$ of (5.4.7) has a bilinear feature. For given $D(q)$ it is linear in $A(q)$ and $B(q)$, and vice versa. The bilinear property can be exploited to obtain a simple algorithm for minimizing the loss function (5.4.6). Specifically, the algorithm consists of repeating the following two steps until convergence (Clarke, 1967).

At iteration $k + 1$:

Step 1 For given $\hat{D}^k(q)$ define the residual

$$\varepsilon_1^{k+1}(t) = A(q)\hat{D}^k(q)y(t) - B(q)\hat{D}^k u(t)$$

The error $\varepsilon_1^{k+1}(t)$ is linear in $A(q)$ and $B(q)$, hence we can determine $\hat{A}^{k+1}(q)$ and $\hat{B}^{k+1}(q)$ by solving an LS problem where the loss function

$$V_1 = \frac{1}{N} \sum_{t=1}^N \varepsilon_1^{k+1}(t)^2 = \frac{1}{N} \sum_{t=1}^N [A(q)\hat{D}^k(q)y(t) - B(q)\hat{D}^k u(t)]^2$$

is minimized.

Step 2 For given $\hat{A}^{k+1}(q)$ and $\hat{B}^{k+1}(q)$, define the residual

$$\varepsilon_2^{k+1}(t) = D(q)[\hat{A}^{k+1}(q)y(t) - \hat{B}^{k+1}(q)u(t)]$$

Then determine $\hat{D}^{k+1}(q)$ by minimizing

$$V_2 = \frac{1}{N} \sum_{t=1}^N \varepsilon_2^{k+1}(t)^2 = \frac{1}{N} \sum_{t=1}^N \left(D(q)[\hat{A}^{k+1}(q)y(t) - \hat{B}^{k+1}(q)u(t)] \right)^2$$

This is again an LS problem.

Thus each step of the algorithm solves an LS problem. This is perhaps why the name generalized least-squares (GLS) is given to the algorithm. The iteration can be started with a normal LS estimation. Figure 5.4.3 shows the block diagram of error generation for the GLS algorithm.

A question to be answered is whether the alternative minimization of V_1 and V_2 will minimize the original loss function V_{GLS} in (5.4.6). The intuitive answer of the reader may be positive. This is indeed true. The iteration procedure is a special case of the so called *separable least-squares* problem studied in Golub and Pereyra (1973). Under the persistent excitation condition of the test signal, they can show that, if the iteration converges, it will reach a local minimum of the original loss function V_{GLS} in (5.4.6). Thus the iteration is a minimization procedure. Note, however, that the convergence to the global minimum is not guaranteed here.

There are several ways to modify the GLS algorithm in order to simplify the computation or to speed up the convergence rate. The main idea of these modifications is first to apply the LS method on the model (5.4.7) with order $n + n_d$ in order to obtain consistent estimates of polynomials $D(q)A(q)$ and $D(q)B(q)$, then to perform some kind of model reduction to retrieve $A(q)$, $B(q)$ and $D(q)$; see , e.g., Söderström and Stoica (1989), and Hsia (1977).

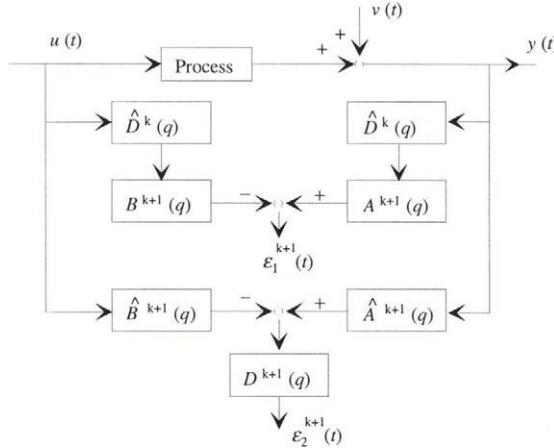


Figure 5.4.3 Error generation of the GLS algorithm

For the problem of model order selection, one can still use the output error criterion. Now, however, there is another possibility. Because the GLS method aims at obtaining white noise residuals, a natural way to order selection is to check the whiteness of residuals for increasing orders. The sample autocorrelation function of the residuals can be used for this test. Note that we have to select both the process order and the order of the disturbance filter. To simplify the procedure, we can let them be equal, i.e., $n = n_d$. More discussions on order selection will be given in a later section.

To see why the GLS method can be called a prediction error method, rewrite the true process (5.4.4) as

$$\begin{aligned} y(t) &= \frac{B^o(q)}{A^o(q)} u(t) + \frac{1}{A^o(q)D^o(q)} e(t) \\ &= \frac{B^o(q)}{A^o(q)} u(t) + \left[\frac{1}{A^o(q)D^o(q)} - 1 \right] e(t) + e(t) \end{aligned} \quad (5.4.8)$$

Because the coefficients of the highest degree terms of $A^o(q)$ and $D^o(q)$ are 1 (monic polynomials), their product will also have this property:

$$F^o(q) = A^o(q)D^o(q) = 1 + f_1 q^{-1} + \cdots + f_{2n} q^{-1}$$

Thus the filter

$$\left[\frac{1}{A^o(q)D^o(q)} - 1 \right] = \frac{-f_1 q^{-1} - \cdots - f_{2n} q^{-1}}{A^o(q)D^o(q)}$$

has one unit delay. This means that the second term in (5.4.8) is a signal that only depends on the past data up to time $t - 1$. When expressing this signal in terms of $u(t)$ and $y(t)$ we have

$$\begin{aligned} y(t) &= \frac{B^o(q)}{A^o(q)} u(t) + \left[\frac{1}{A^o(q)D^o(q)} - 1 \right] A^o(q)D^o(q) \left[y(t) - \frac{B^o(q)}{A^o(q)} u(t) \right] + e(t) \\ &= D^o(q)B^o(q)u(t) + [1 - D^o(q)A^o(q)]y(t) + e(t) \\ &= z(t) + e(t) \end{aligned} \quad (5.4.9)$$

where

$$z(t) = D^o(q)B^o(q)u(t) + [1 - D^o(q)A^o(q)]y(t)$$

Note that $z(t)$ and $e(t)$ are uncorrelated. If $z(t)$ is used as the one step ahead prediction of the output $y(t)$, the prediction error $e(t)$ is white noise. One would expect that this predictor is the best one in some sense, because when the prediction error is white noise, it contains no useful information at all. Indeed, this can be shown more formally. Let $y^*(t)$ be an arbitrary predictor of $y(t)$. Then the variance of the prediction error is

$$\begin{aligned} E[y(t) - y^*(t)]^2 &= E[z(t) + e(t) - y^*(t)]^2 \\ &= E[z(t) - y^*(t)]^2 + Ee(t)^2 \geq Ee(t)^2 = E[y(t) - z(t)]^2 \end{aligned}$$

Thus $z(t)$ is the optimal predictor in the sense of minimum variance.

In identification, we will write down the optimal filter in terms of unknown polynomials as

$$\hat{y}(t|\theta) = D(q)B(q)u(t) + [1 - D(q)A(q)]y(t) \quad (5.4.10)$$

and determine the parameters by minimizing the sum of the squares of the prediction errors

$$V = \frac{1}{N} \sum_{t=1}^N [y(t) - \hat{y}(t|\theta)]^2 = \frac{1}{N} \sum_{t=1}^N [D(q)(A(q)y(t)) - [B(q)[u(t)]]]^2 \quad (5.4.11)$$

Again this is the loss function of the GLS method (5.4.6).

The model structure (5.4.4) is one way to model the equation error noise. Other model structures can also be used.

5.4.3 ARMAX Model

The true process model is assumed to be

$$A^o(q)y(t) = B^o(q)u(t) + C^o(q)e(t) \quad (5.4.12)$$

or

$$y(t) = \frac{B^o(q)}{A^o(q)} u(t) + \frac{C^o(q)}{A^o(q)} e(t)$$

where

$$A^o(q) = 1 + a_1^o q^{-1} + \cdots + a_n^o q^{-n}$$

$$B^o(q) = b_1^o q^{-1} + \cdots + b_n^o q^{-n}$$

$$C^o(q) = 1 + c_1^o q^{-1} + \cdots + c_n^o q^{-n}$$

and $e(t)$ is white noise with zero mean and variance R . So the equation disturbance is assumed to be an MA (moving-average) process; see Figure 5.4.4 for the block diagram of ARMAX model. This model was proposed by Åström and Bohlin (1965) and has become a well known model.

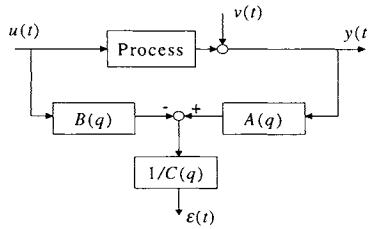


Figure 5.4.4 Error generation of ARMAX model

Similarly as for the GLS method, we can show that the predictor for this model structure has the form

$$\hat{y}(t|\theta) = \frac{B(q)}{C(q)} u(t) + \left[1 - \frac{A(q)}{C(q)} \right] y(t) \quad (5.4.13)$$

The parameters of this model are determined by minimizing

$$V_{ARMAX} = \frac{1}{N} \sum_{t=1}^N \left(\frac{1}{C(q)} [A(q)y(t) - B(q)u(t)] \right)^2 \quad (5.4.14)$$

An algorithm for this problem can be derived using the quasilinearization technique used for the output error method in Section 5.2.

When comparing the GLS method and the ARMAX model method, the advantage of the GLS method, as we have seen, is that simple minimization algorithms can be derived for the model structure; the advantage of the ARMAX model is that it is very suitable for some controller design techniques; see Åström and Wittenmark (1984).

Note that the common idea of both the GLS method and the ARMAX model method is to model the equation disturbance; a consequence of this is that the process transfer function and output disturbance share the same $A(q)$ polynomial.

5.4.4 Box-Jenkins Method

A natural development of the output error method is to further model the output disturbance $v(t)$. This can be done by assuming that the true process is

$$y(t) = \frac{B^o(q)}{A^o(q)}u(t) + \frac{C^o(q)}{D^o(q)}e(t) \quad (5.4.15)$$

where

$$\begin{aligned} A^o(q) &= 1 + a_1^o q^{-1} + \cdots + a_n^o q^{-n} \\ B^o(q) &= b_1^o q^{-1} + \cdots + b_n^o q^{-n} \\ C^o(q) &= 1 + c_1^o q^{-1} + \cdots + c_n^o q^{-n} \\ D^o(q) &= 1 + d_1^o q^{-1} + \cdots + d_n^o q^{-n} \end{aligned}$$

This model structure was suggested by Box and Jenkins (1970). The predictor for this model is

$$\hat{y}(t|\theta) = \frac{D(q)B(q)}{C(q)A(q)}u(t) + [1 - \frac{D(q)}{C(q)}]y(t) \quad (5.4.16)$$

The parameters of this model are determined by minimizing

$$V_{BJ} = \frac{1}{N} \sum_{t=1}^N \left(\frac{D(q)}{C(q)} \left[y(t) - \frac{B(q)}{A(q)}u(t) \right] \right)^2 \quad (5.4.17)$$

Figure 5.4.5 shows the block diagram of the Box-Jenkins model

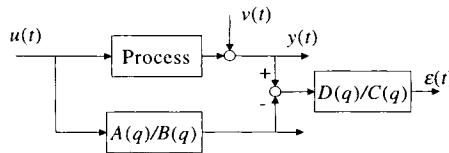


Figure 5.4.5 Error generation of Box-Jenkins model

The Box-Jenkins model has several advantages over the output error method. Firstly, it will supply both a process model and a disturbance model. Soon it will be shown that the Box-Jenkins model will be consistent also in closed loop operation; this implies that a Box-Jenkins method will give a more accurate process model than an output error method for a given process under closed-loop conditions. However, the Box-Jenkins model has a more complex structure, which implies that numerical optimization will be more complicated.

5.4.5 A General Model

The following model is used by Ljung (1987) to cover all the existing model structures. The true process is assumed to be

$$F^o(q)y(t) = \frac{B^o(q)}{A^o(q)}u(t) + \frac{C^o(q)}{D^o(q)}e(t) \quad (5.4.18)$$

or

$$y(t) = \frac{B^o(q)}{F^o(q)A^o(q)}u(t) + \frac{C^o(q)}{F^o(q)D^o(q)}e(t)$$

where $A^o(q)$, $B^o(q)$, $C^o(q)$, $D^o(q)$ and $F^o(q)$ are polynomials of q^{-1} , all being monic except $B(q)$, $e(t)$ is white noise with zero mean and variance R .

This structure is too general for practical applications. One would fix one or more polynomials to unity for a specific process. However, by developing algorithms and results for the general model, one can cover all the special cases. The predictor for (5.4.18) is

$$\hat{y}(t|\theta) = \frac{D(q)B(q)}{C(q)A(q)}u(t) + \left[1 - \frac{D(q)F(q)}{C(q)}\right]y(t) \quad (5.4.19)$$

5.4.6 Properties of Prediction Error Methods

Consistency

Assume that the true process satisfies

$$y(t) = G^o(q)u(t) + H^o(q)e(t) \quad (5.4.20)$$

where $G^o(q)$ is the stable process transfer function, $H^o(q)$ is the disturbance filter which is stable and minimum phase (its inverse is stable), and $e(t)$ is white noise with zero mean and variance R . Further assume that

$$\begin{cases} G^o(0) = 0 & (G^o(q) \text{ has at least one delay}) \\ H^o(0) = 1 & H^o(q) \text{ is monic} \end{cases} \quad (5.4.21)$$

Assume that the correct model structure is used. The predictor for the process is

$$\hat{y}(t|\theta) = \frac{G(q, \theta)}{H(q, \theta)}u(t) + \left[1 - \frac{1}{H(q, \theta)}\right]y(t) \quad (5.4.22)$$

where θ is the parameter vector containing all the parameters of the process model and the disturbance model. The prediction error is

$$\varepsilon(t|\theta) = \frac{1}{H(q, \theta)}[y(t) - G(q, \theta)u(t)] \quad (5.4.23)$$

The parameter vector θ is determined by minimizing

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N \varepsilon(t|\theta)^2 = \frac{1}{N} \sum_{t=1}^N \left(\frac{1}{H(q, \theta)} [y(t) - G(q, \theta)u(t)] \right)^2 \quad (5.4.24)$$

When $N \rightarrow \infty$, we have

$$V_N(\theta) \rightarrow E\varepsilon(t|\theta)^2 = E \left(\frac{1}{H(q, \theta)} [y(t) - G(q, \theta)u(t)] \right)^2 \quad (5.4.25)$$

It follows from (5.4.20) and (5.4.23) that

$$\begin{aligned} \varepsilon(t|\theta) &= \frac{1}{H(q, \theta)} [G^o(q)u(t) + H^o(q)e(t) - G(q, \theta)u(t)] \\ &= \frac{1}{H(q, \theta)} [G^o(q) - G(q, \theta)] u(t) + \frac{H^o(q)}{H(q, \theta)} e(t) \\ &= e(t) + \text{a term independent of } e(t) \end{aligned} \quad (5.4.26)$$

The last equality is due to the conditions in (5.4.21). Hence one gets

$$V_\infty(\theta) = E\varepsilon(t|\theta)^2 \geq Ee(t)^2 = R \quad (5.4.27)$$

and

$$E\varepsilon(t|\theta)^2 = \frac{1}{H(q, \theta)^2} \Delta G(q, \theta)^2 E u(t)^2 + \frac{H^o(q)^2}{H(q, \theta)^2} R \quad (5.4.28)$$

If the global minimum is obtained for all N and if the input is persistently exciting with a sufficiently high order, then one can see from (5.4.28) that the lower bound of the asymptotic loss function (5.4.27) is attained when

$$\Delta G(q, \theta)^2 = 0, \quad \frac{H^o(q)^2}{H(q, \theta)^2} = 1$$

This implies that when $N \rightarrow \infty$

$$G(q, \theta) = G^o(q), \quad H(q, \theta) = H^o(q)$$

Therefore, the prediction error model estimate is consistent. Note that one does not need to assume an open-loop operation. It is essential that one chooses the error at the place where white noise (assumedly) enters the true process. For a mathematically more rigorous proof of the consistency, see Ljung (1987).

The reader is reminded that the consistency of the prediction error methods is established on the assumption that the global minimum of the loss function is reached. In practice, however, this can not be guaranteed.

When the process is operating in closed-loop, both the order of the process model and the order of the disturbance model should be correct in order to get a consistent estimate. For open-loop data, the above condition can be somewhat relaxed. If the process model and the disturbance model are parametrized independently, i.e., $F(q) = 1$ in (5.4.18), we can obtain a consistent estimate of the process model for open-loop data even when the order of the disturbance model is too low. We have shown that the output error model is consistent for an open-loop experiment. The same can be done for the Box-Jenkins model. This property can be desirable in practice, because the accuracy of the process model is more important than that of the disturbance model.

Asymptotic Distribution of Parameter Estimates

Consider the parameter estimate $\hat{\theta}_N$ determined by minimizing $V_N(\theta)$ in (5.4.24). Assume that the linear process is stable, the model structure is correct, the test signal is persistently exciting with sufficient order and the minimization converged to its global minimum. Denote θ^o as the true parameter vector. Then we have

$$\hat{\theta}_N \rightarrow \theta^o, \quad \text{w.p. 1 as } N \rightarrow \infty \quad (5.4.29)$$

and

$$\sqrt{N}(\hat{\theta}_N - \theta^o) \in As N(0, P_\theta) \quad (5.4.30)$$

where (5.4.29) says that the parameter estimate will tend to the true values with probability 1 as N tends to infinity; (5.4.30) states that the error of parameter estimate multiplied by \sqrt{N} will have asymptotically in N a Gaussian (normal) distribution with zero mean and covariance P_θ . For the proof of the result and the expression of P_θ , see Chapter 9 of Ljung (1987).

When we are interested in the covariance, we get from (5.4.30)

$$\text{cov } \hat{\theta}_N \approx \frac{1}{N} P_\theta \quad (5.4.31)$$

So the covariance matrix is inversely proportional to the number of data or test time. This result shows clearly the averaging effect of identification: the estimate will become more accurate when more data points are used.

Relationship to the Maximum likelihood (ML) method

The *maximum likelihood* (ML) estimate of θ is obtained by maximizing the likelihood function, i.e., the probability density function of observations conditioned on the parameter vector θ . Suppose that the observations are represented by the random variable $\mathbf{y}^T = [y(1), y(2), \dots, y(N)]$. Denote the probability density function of \mathbf{y} as

$$f(\theta; y_1, y_2, \dots, y_N) = f_y(\theta; \mathbf{y}) \quad (5.4.32)$$

Here the unknown parameter vector θ that describes properties of the observed variable will be estimated using the observations in \mathbf{y} . If the observed value of \mathbf{y} is \mathbf{y}^* , then the maximum

likelihood (ML) estimator is to select an estimate so that the observed event becomes "as likely as possible." That is, we determine θ such that

$$L(\theta) = f_y(\theta; \mathbf{y}^*) \quad (5.4.33)$$

is maximized. Here $L(\theta)$ is called the *likelihood function* which is a deterministic function of the parameter vector θ once the numerical value \mathbf{y}^* is inserted. (In contrast, the probability density function (5.4.32) is a function of the random variable \mathbf{y} for the fixed parameter vector θ .)

The principle of maximum likelihood was introduced by Fisher in 1912. The argument of the ML method is simple and intuitively attractive. It says that if the given event (observation or data) has happened, its probability must be higher than other events which have not yet taken place. Therefore, we would expect that the ML method has some nice properties. Indeed, it can be shown that ML method gives asymptotically (when $N \rightarrow \infty$) unbiased and efficient (minimum variance) estimates. In this sense the ML method gives the best possible estimator.

In order to apply the ML method to the identification of a linear process, one needs to further assume that the white noise in the true process (5.4.20) $e(t)$ has a Gaussian (normal) distribution. It is equally valid to use the probability density function of the disturbance because there is a one-to-one transformation between $\{y(t)\}$ and $\{e(t)\}$ as given by (5.4.20) if the effect of the initial conditions is neglected. Using the expression for the multivariable Gaussian distribution function, we have

$$L(\theta) = f_e(\theta; \varepsilon) = \frac{1}{(2\pi)^{N/2} R^{1/2}} \exp \left[-\frac{1}{2} \frac{1}{N} \sum_{t=1}^N \varepsilon(t, \theta)^2 / R \right] \quad (5.4.34)$$

Maximizing this likelihood function is equivalent to minimizing the loss function

$$-\ln L(\theta) = \frac{1}{2} \frac{1}{N} \sum_{t=1}^N \varepsilon(t, \theta)^2 / R + \frac{1}{2} \ln R + C \text{ (constant)} \quad (5.4.35)$$

This is equivalent to the loss function of the prediction error methods (5.4.24). Therefore, the prediction error methods are the maximum likelihood method if the white noise $e(t)$ has a Gaussian distribution, hence they give efficient estimates. This conclusion gives the prediction error methods a sound theoretical basis. For more discussion on the maximum likelihood method, see Ljung (1987).

Bias Distribution in the Frequency Domain

When the model structure is incorrect or orders are too low, a prediction error model is biased. One can derive an asymptotic frequency domain expression for the loss function (5.4.25) for an open-loop experiment (using Parseval's formula)

$$V_\infty(\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(G^o(e^{i\omega}) - \frac{B(e^{i\omega})}{A(e^{i\omega})} \right)^2 \frac{\Phi_u(e^{i\omega})}{H(e^{i\omega})^2} d\omega + \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\Phi_v(e^{i\omega})}{H(e^{i\omega})^2} d\omega \quad (5.4.36)$$

Hence the disturbance model is part of the frequency weighting for the process transfer function approximation. The expression for closed loop data is rather complex and will not been shown here.

5.5 User's Choice in Identification for Control

Having learned so many methods, the reader may wonder which is the best one. The answer is not simple and is related to the use of the model and to the application environment. It is clear from the above discussion, that various methods are related to the choice of disturbance model, or equivalently, the error criterion. In the following, several situations will be discussed in the choice of various identification methods. This section is based on Zhu (2000a).

Given the true linear time-invariant discrete-time process

$$\begin{aligned} y(t) &= G^o(q)u(t) + v(t) \\ v(t) &= H^o(q)e(t) \end{aligned} \quad (5.5.1)$$

Because the output disturbance $v(t)$ is assumed to be a realization of a stochastic process, the estimated models $\hat{G}(e^{i\omega})$ is a random variable; see Ljung (1987). Then one can analyze the model error in a probability framework. Denote the bias error of the model as

$$\tilde{B}(e^{i\omega}) = G^o(e^{i\omega}) - E[\hat{G}(e^{i\omega})] \quad (5.5.2)$$

and variance error as

$$\tilde{V}^2(e^{i\omega}) = E\{G^o(e^{i\omega}) - E[\hat{G}(e^{i\omega})]\}^2 \quad (5.5.3)$$

Then the total mean square error is

$$\tilde{M}^2(e^{i\omega}) = E[G^o(e^{i\omega}) - \hat{G}(e^{i\omega})]^2 = \tilde{B}^2(e^{i\omega}) + \tilde{V}^2(e^{i\omega}) \quad (5.5.4)$$

5.5.1 Open-Loop Identification of Stable Processes

When the process is stable, it is often possible to carry out an identification test in an open-loop operation. Assume that the process $G^o(q)$ is stable, the disturbance filter $H^o(q)$ is stable and minimum phase. When the identified model is used for control, often a model that minimizes the output error (simulation error) criterion for a given class of input is desirable. This is because an accurate input-output model $G(q)$ is important for control.

Suppose that the control requirement for model identification has been translated into the minimization of the following loss function

$$V_{OE}^o = \frac{1}{2\pi} \int_{-\pi}^{\pi} |G^o(e^{i\omega}) - \hat{G}(e^{i\omega})|^2 W(\omega) d\omega \quad (5.5.5)$$

where $G^o(e^{i\omega})$ and $\hat{G}(e^{i\omega})$ are the frequency responses of the process and the model respectively, $W(\omega)$ is a weighting function that is continuous over frequency ω . Assume that an open-loop identification test is performed and the test input signal $u(t)$ is chosen such that its power spectrum $\Phi_u(\omega)$ is equal to $W(\omega)$, namely,

$$\Phi_u(\omega) = W(\omega)$$

Then the criterion (5.5.5) is asymptotically equivalent to the noise-free output error criterion

$$V_{OE}^o = \sum_{t=1}^N ([G^o(q) - G(q)]u(t))^2 \rightarrow \frac{1}{2\pi} \int_{-\pi}^{\pi} |G^o(e^{i\omega}) - \hat{G}(e^{i\omega})|^2 W(\omega) d\omega \quad (5.5.6)$$

Note that this criterion is not the same as the output error criterion for noisy data (5.2.2). The relation between the noise-free criterion and the noisy criterion is

$$V_{OE} = V_{OE}^o + \sum_{t=1}^N v^2(t) \quad (5.5.7)$$

Due to this close relation, some researchers and users believe that a model estimated using the noisy output error criterion will also minimize the noise-free output error criterion and is therefore optimal for control.

The discussion below compares the model quality between an output-error model and a prediction error model which does not depend on specific model parametrization. However, for the readability for control engineers, we will use the following specific model structures. Parametrize the output-error model as

$$\begin{aligned} y(t) &= G(q, \theta)u(t) + e(t) \\ &= \frac{b_0 + b_1q^{-1} + \dots + b_nq^{-n}}{1 + a_1q^{-1} + \dots + a_nq^{-n}} + e(t) \end{aligned} \quad (5.5.8)$$

where $\theta = [b_0, b_1, \dots, b_n, a_1, \dots, a_n]^T$ is the vector of process model parameters. Parametrize the prediction error model as a Box-Jenkins model

$$\begin{aligned} y(t) &= G(q, \theta)u(t) + H(q, \alpha)e(t) \\ &= \frac{b_0 + b_1q^{-1} + \dots + b_nq^{-n}}{1 + a_1q^{-1} + \dots + a_nq^{-n}} u(t) + \frac{1 + c_1q^{-1} + \dots + c_nq^{-n}}{1 + d_1q^{-1} + \dots + d_nq^{-n}} e(t) \end{aligned} \quad (5.5.9)$$

where

$$\begin{aligned} \theta &= [b_0, b_1, \dots, b_n, a_1, \dots, a_n]^T \\ \alpha &= [c_1, \dots, c_n, d_1, \dots, d_n]^T \end{aligned}$$

are the parameter vector of the model and that of the disturbance model respectively. Note that the key difference between a prediction error model and an output error model is on the use of the disturbance model $H(q)$.

Suppose that one will use the identified model to simulate the process output using an input $u^*(t)$ with spectrum $\Phi_u^*(\omega) = W(\omega)$. Perform an open-loop identification test using input $u(t)$ that has the same spectrum $\Phi_u(\omega) = W(\omega)$. Denote $\hat{G}(q, \hat{\theta}^{BJ})$ as the Box-Jenkins model estimated using the test data and $\hat{G}(q, \hat{\theta}^{OE})$ as that of the output-error model. Note that $u^*(t)$ has not been used in identification. Denote

$$y^o(t) = G^o(q)u^*(t)$$

as the true noise-free output. When simulating the process using $u^*(t)$, the error of the Box-Jenkins model is

$$\tilde{y}(t, \hat{\theta}^{BJ}) = [\hat{G}(q, \hat{\theta}^{BJ}) - G^o(q)]u^*(t) \quad (5.5.10)$$

with the variance, or, the output-error criterion when $N \rightarrow \infty$ as

$$V_{OE}(\hat{\theta}^{BJ}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |G^o(e^{i\omega}) - \hat{G}(e^{i\omega}, \hat{\theta}^{BJ})|^2 \Phi_u^*(\omega) d\omega \quad (5.5.11)$$

similarly, the output-error criterion for the output error model is when $N \rightarrow \infty$

$$V_{OE}^V(\hat{\theta}^{OE}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |G^o(e^{i\omega}) - \hat{G}(e^{i\omega}, \hat{\theta}^{OE})|^2 \Phi_u^*(\omega) d\omega \quad (5.5.12)$$

Remember that the estimated models $\hat{G}(e^{i\omega}, \hat{\theta}^{BJ})$ and $\hat{G}(e^{i\omega}, \hat{\theta}^{OE})$ can be treated as random variables. Then, taking expectations with respect to $\hat{G}(e^{i\omega}, \hat{\theta}^{BJ})$, we obtain the average output-error criterion of the Box-Jenkins model

$$EV_{OE}^V(\hat{\theta}^{BJ}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \text{var}[\hat{G}(e^{i\omega}, \hat{\theta}^{BJ})] \Phi_u^*(\omega) d\omega \quad (5.5.13)$$

and the average output-error criterion of the output-error model

$$EV_{OE}^V(\hat{\theta}^{OE}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \text{var}[\hat{G}(e^{i\omega}, \hat{\theta}^{OE})] \Phi_u^*(\omega) d\omega \quad (5.5.14)$$

Denote $\theta^o = [b_0^o, b_1^o, \dots, b_n^o, a_1^o, \dots, a_n^o]^T$ as the vector of true process model parameters. Under the above assumptions, according to (5.4.29) and (5.4.30) one has

$$\hat{\theta}^{BJ} \rightarrow \theta^o, \quad w.p. 1 \text{ as } N \rightarrow \infty \quad (5.5.15)$$

$$\hat{\theta}^{OE} \rightarrow \theta^o, \text{ w.p. 1 as } N \rightarrow \infty \quad (5.5.16)$$

and

$$\sqrt{N}(\hat{\theta}^{BJ} - \theta^o) \in AsN(0, \text{cov}(\hat{\theta}^{BJ})) \quad (5.5.17)$$

$$\sqrt{N}(\hat{\theta}^{OE} - \theta^o) \in AsN(0, \text{cov}(\hat{\theta}^{OE})) \quad (5.5.18)$$

where $\text{cov}[\hat{\theta}]$ is the covariance matrix of the parameter vector; see Ljung (1987) for the expression.

By Taylor's expansion

$$\hat{G}(e^{i\omega}, \hat{\theta}^{BJ}) - G^o(e^{i\omega}, \theta^o) = [G'_\theta(e^{i\omega}, \theta^o)]^T (\hat{\theta}^{BJ} - \theta^o) + O(|\hat{\theta}^{BJ} - \theta^o|) \quad (5.5.19)$$

where $G'_\theta(e^{i\omega}, \theta^o)$ is a column vector, the derivative of $G(e^{i\omega}, \theta)$ with respect to θ at θ^o , then, according to (5.5.16) one obtains the asymptotic distribution in the frequency domain

$$\sqrt{N}(\hat{G}(e^{i\omega}, \hat{\theta}^{BJ}) - G^o(e^{i\omega}, \theta^o)) \in AsN(0, \text{var}[\hat{G}(e^{-i\omega}, \hat{\theta}^{BJ})]) \text{ as } N \rightarrow \infty \quad (5.5.20)$$

with

$$\text{var}[\hat{G}(e^{-i\omega}, \hat{\theta}^{BJ})] = [G'_\theta(e^{i\omega}, \theta^o)]^T \text{cov}(\hat{\theta}^{BJ}) G'_\theta(e^{i\omega}, \theta^o) \quad (5.5.21)$$

Similarly, one has for the output error model

$$\sqrt{N}(\hat{G}(e^{i\omega}, \hat{\theta}^{OE}) - G^o(e^{i\omega}, \theta^o)) \in AsN(0, \text{var}[\hat{G}(e^{-i\omega}, \hat{\theta}^{OE})]) \text{ as } N \rightarrow \infty \quad (5.5.22)$$

with

$$\text{var}[\hat{G}(e^{-i\omega}, \hat{\theta}^{OE})] = [G'_\theta(e^{i\omega}, \theta^o)]^T \text{cov}(\hat{\theta}^{OE}) G'_\theta(e^{i\omega}, \theta^o) \quad (5.5.23)$$

Now

$$\begin{aligned} & EV_{OE}^V(\hat{\theta}^{BJ}) - EV_{OE}^V(\hat{\theta}^{OE}) \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \{ \text{var}[\hat{G}(e^{-i\omega}, \hat{\theta}^{BJ})] - \text{var}[\hat{G}(e^{-i\omega}, \hat{\theta}^{OE})] \} \Phi_u^*(\omega) d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \{ [G'_\theta(e^{i\omega}, \theta^o)]^T [\text{cov}(\hat{\theta}^{BJ}) - \text{cov}(\hat{\theta}^{OE})] G'_\theta(e^{i\omega}, \theta^o) \} \Phi_u^*(\omega) d\omega \end{aligned} \quad (5.5.24)$$

Here one relates the variance of the frequency response to the covariance matrix of parameter vector. It is left to show which model has a smaller covariance matrix.

From the discussion of prediction error method, one has learned that the prediction error method is a maximum likelihood estimator which is a minimum variance estimator; the output error

method is not a minimum variance estimator if the disturbance $v(t)$ is not white noise. Therefore, one expects the following relation

$$\text{cov}(\hat{\theta}^{BJ}) < \text{cov}(\hat{\theta}^{OE}) \quad (5.5.25)$$

equivalently, for any non zero vector X ,

$$X^T[\text{cov}(\hat{\theta}^{BJ}) - \text{cov}(\hat{\theta}^{OE})]X < 0 \quad (5.5.26)$$

In fact, this can be proved explicitly without the argument of maximum likelihood; see Söderström and Stoica (1982).

It is now the moment to summarize the discussion by stating the following

Result on variance error. *Given a stable linear process as in (5.5.1), assume that: 1) Output disturbance $v(t)$ is colored noise; 2) an open-loop test has been performed using a input signal with a continuous power spectrum; 3) the true process model order is n which is used for estimating a Box-Jenkins model $\hat{G}(q, \hat{\theta}^{BJ})$ and an output-error model $\hat{G}(q, \hat{\theta}^{OE})$; 4) the numerical search routines have converged to their global minima. Then*

$$EV_{OE}^V(\hat{\theta}^{BJ}) < EV_{OE}^V(\hat{\theta}^{OE}) \text{ as } N \rightarrow \infty \quad (5.5.27)$$

This is easily obtained by inserting (5.5.26) into (5.5.24) with $X = G'_\theta(e^{i\omega}, \theta^o)$.

This result goes somewhat against intuition. Some researchers believe that an output-error model is optimal for the output-error criterion. This is true when data is noise-free and bias is the only error source, also true when the identification data are used to evaluate the error criterion. However, when the process $G^o(q)$ is in the model set $\{G(q)\}$ and when the data are noisy, the Box-Jenkins or a prediction error model will have a smaller noise-free output error criterion than the output-error model. Equivalently, the Box-Jenkins model will also have a smaller noisy output error criterion when evaluated on a fresh data set. Note that in (5.5.27) the result is given in expectations, meaning that exceptions may exist.

Exercise 5.5.1. Check the result (5.5.27) for finite data sequence. Given the process

$$y(t) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} u(t) + v(t)$$

with

$$v(t) = \frac{\alpha}{1 - 0.95q^{-1}} e(t)$$

The input signal is a GBN with average switch time of 8 samples. The test time is 500 samples. The disturbance power is 10% of that of the noise-free output. Run 100 open-loop simulations. Use each data set to estimate a 2nd order Box-Jenkins model and a 2nd order output-error model. Compare the following noise-free output-error criterion of the two models

$$V_{OE} = \frac{1}{500} \sum_{t=1}^{500} [G^o(q)u(t) - \hat{G}(q)u(t)]^2$$

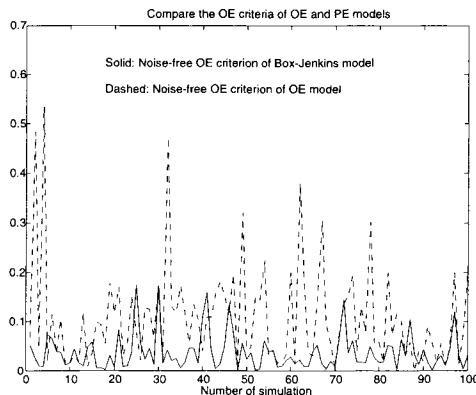


Figure 5.5.1 Compare Box-Jenkins model with output error model

Answer. The M file is as follows. The results are shown in Figure 5.5.1. One can see that Box-Jenkins mode indeed has much smaller average output-error criterion. On some occasions, the output-error model has slightly smaller error criteria.

```
%%% bj_oe.m; M file for comparing the OE criteria of BJ and OE models %%%
function [THoe,ErrOE,THbj,ErrBJ] = bj_oe(Ao,Bo,N,n,sn,n_sim,Delay)
%
% [THoe,ErrOE,THbj,ErrBJ] = bj_oe(Ao,Bo,N,n,sn,n_sim,Delay)
%
% This function compares the OE criterion of BJ model and of OE model
%
% Input arguments:
% [Ao, Bo]: Process polynomials
% N: Number of samples to be simulated
% n: Order of identified model
% sn: noise-to-signal ratio
% n_sim: number of simulations
% Delay: Delay, optional
%
% Output arguments:
% THoe: Parameter vector of OE model
% ErrOE: OE Error of OE model
% THbj: Parameter vector of bj model
% ErrBJ: OE Error of BJ model
%
THoe = [];
```

```

THbj = [];
ErrOE = [];
ErrBJ = [];
if exist('Delay') == 0
    Delay = 0;
end
for k = 1:n_sim
    U = gbngen(N,8,1);
    Yo = filter(Bo,Ao,U);
    V = filter(1,[1 -.95],randn(N,1));
    V = V/std(V)*sqrt(sn)*std(Yo);
    Y = Yo + V;
    thOE = oe([Y U],[n n Delay]);
    [a,boe,c,d,foe]=th2poly(thOE);
    yoe = filter(boe,foe,U);
    ErrOE = [ErrOE;cov(Yo-yoe)];
    THoe = [THoe; foe boe];
    thBJ = bj([Y U],[n 1 1 n Delay]);
    [a,bbj,c,d,fbj]=th2poly(thBJ);
    ybj = filter(bbj,fbj,U);
    ErrBJ = [ErrBJ;cov(Yo-ybj)];
    THbj = [THbj; fbj bbj];
end

```

From numerical point of view, it is sometimes advantageous to first estimate a high order model and then perform a model reduction on the high order model; see, e.g., Wahlberg (1989), Bayard (1994), Zhu and Backx (1993) and Zhu (1998). Recently results show that this approach can reduce the variance when compared with direct estimation of a low order model; see Tjärnström and Ljung (1999). Basically, two types of models are used for the high order model: the ARX model and the output error model. According to (5.5.24), an ARX model will be more accurate than an output error model with the same order, because a high order ARX model is practically unbiased and it is a prediction error model. The following example demonstrates this.

Example 5.5.2. The process is

$$y(t) = \frac{1 + .5q^{-1} + .4q^{-2} + 2q^{-3}}{1 + .5q^{-1} - .4q^{-2} - .26q^{-3} - .03q^{-4}} u(t) + v(t)$$

where

$$v(t) = \frac{\alpha}{1 - 0.9q^{-1}} e(t)$$

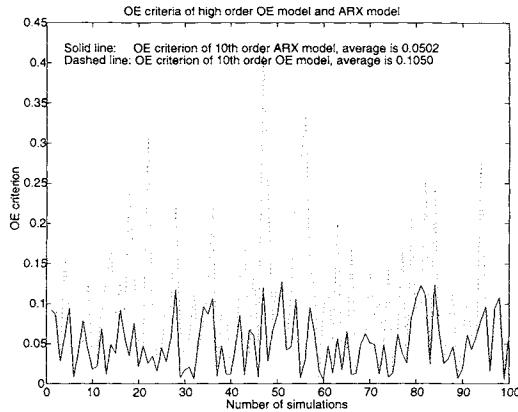


Figure 5.5.2 Comparison of a high order OE model and a high order ARX model

$e(t)$ is a white Gaussian noise and α is a constant that is used to obtain a signal-to-noise ratio of 20% in power at the output. The input signal is a GBN signal with average switch time of 10 samples; the number of samples is 1000. In total 100 open-loop simulations are carried out. For each data set, a 10th order ARX model and a 10th order OE model are estimated using the Identification Toolbox of Matlab (R). Noise-free output-error criterion is used to compare the quality of the two models. Figure 5.5.2 shows the results. Note that the ARX models are much simpler numerically and are more accurate than the output error models. If one uses an ARX model as the initial value for high order output error model estimation, the iteration will in fact increase the true output error criterion. This is a situation where one obtains more accuracy for less effort.

Bias error. Assume that the disturbance is zero and the model order is lower than the true one. Using the same argument as for the noisy case, it can be shown that the limiting estimate of an OE model is

$$\theta^{OE} = \arg \min_{\theta} \int_{-\pi}^{\pi} |G^o(e^{i\omega}) - G(e^{i\omega}, \theta)|^2 \Phi_u(\omega) d\omega \quad (5.5.28)$$

and the limiting estimate of a Box-Jenkins model is

$$\theta^{BJ} = \arg \min_{\theta} \int_{-\pi}^{\pi} |G^o(e^{i\omega}) - G(e^{i\omega}, \theta)|^2 \Phi_u(\omega) \frac{D(e^{i\omega})}{C(e^{i\omega})} d\omega \quad (5.5.29)$$

These two expressions show that when only bias error exists, an output error is optimal for output error criterion. This is because the noise-free criterion is minimized directly.

Total error. In practice, when a low order model is estimated directly, both variance error and bias error exist. The question is what argument should be used for selecting the estimation

criterion, variance or bias. An answer to the question is based on the following result (Guo and Ljung, 1994).

Denote the bias error of the model as $\tilde{B}(e^{i\omega})$ and variance error as $\tilde{V}^2(e^{i\omega})$, and the total mean square error as

$$\tilde{M}^2(e^{i\omega}) = \tilde{B}^2(e^{i\omega}) + \tilde{V}^2(e^{i\omega}) \quad (5.5.30)$$

Assume that, when model order increases, the decrease of bias error $\tilde{B}^2(e^{i\omega})$ is faster than the increase of variance error $\tilde{V}^2(e^{i\omega})$, then for the optimal model (order) that minimizes the total error $\tilde{M}^2(e^{i\omega})$, the following relation holds

$$\tilde{B}^2(e^{i\omega}) \leq \tilde{V}^2(e^{-i\omega}) \quad (5.5.31)$$

that is, the bias error is dominated by the variance error.

This result indicates that in low order model estimation, it is sensible to consider variance error for criterion selection when the total error is to be minimized. Therefore, in general, a prediction error model is preferred to an output error model for use in control. More specifically, in direct low order model estimation, a Box-Jenkins model or an ARMAX model is recommended; in high order model estimation, an ARX model is recommended.

Remark. The discussion of this section is theoretically motivated and it is assumed that the numerical optimization routines converge to their global minima. In practice, the numerical problem cannot be ignored. A more complex model structure will, in general, have more numerical problems. Therefore, it is advisable to identify models with several structures and chose the best one using order/structure selection.

5.5.2 Closed-Loop Identification

As discussed in Chapter 3, closed-loop tests have many advantages. However, identification using closed-loop data is more difficult than that using open-loop data. When data are collected from a closed-loop test, the input $u(t)$ is correlated with the disturbance $v(t)$ due to feedback. Then the output error method and instrumental variable methods will produce a biased estimate, because the effect of disturbance cannot be averaged out when $u(t)$ and $v(t)$ are correlated. For the output method, one now gets

$$\begin{aligned} V_{OE}^N &\rightarrow EV_{OE} = E[\Delta G(q)u(t) + v(t)]^2 \\ &= \Delta G(q)^2 Eu^2(t) + 2\Delta G(q)E[u(t)v(t)] + Ev^2(t) \\ &\neq \Delta G^2(q)Eu^2(t) + Ev^2(t) \end{aligned} \quad (5.5.32)$$

The inequality here destroys the consistency of the output error model. So the output error model is biased for closed-loop data when the disturbance is not white noise. Hence it is not recommended for closed-loop identification with noisy data.

On the other hand, prediction error models will remain consistency using closed-loop data, provided that orders of both $G(q)$ and $H(q)$ are correct and only the variance error exists. When bias exists and the total error is minimized, it is still recommended to use prediction error methods because the error is dominated by the variance as just mentioned.

When the process is unstable, however, not all the prediction error methods are equally applicable. All prediction error methods need some minimization routines to search for the minima of their loss functions, except the ARX method whose solution is linear least squares. At each iteration, the gradient needs to be calculated. As can be shown (Ljung, 1987), this gradient is related to the predictor model. The gradient will only be bounded if the predictor model remains stable for all the iterations. If the process is unstable, $1/A(q)$ will be unstable. This means that when a predictor has a factor $1/A(q)$ in the model, it will be generally unstable and hence the calculated gradient will be unbounded and numerical problems will occur. The following will examine the prediction error methods discussed in this chapter.

Least-squares method or ARX model. The predictor is

$$\hat{y}(t|\theta) = B(q)u(t) + [1 - A(q)]y(t)$$

Generalized least-squares method. The predictor is

$$\hat{y}(t|\theta) = D(q)B(q)u(t) + [1 - D(q)A(q)]y(t)$$

ARMAX model. The predictor is

$$\hat{y}(t|\theta) = \frac{B(q)}{C(q)}u(t) + \left[1 - \frac{A(q)}{C(q)}\right]y(t)$$

Box-Jenkins model. The predictor is

$$\hat{y}(t|\theta) = \frac{D(q)B(q)}{C(q)A(q)}u(t) + [1 - \frac{D(q)}{C(q)}]y(t)$$

Hence the Box-Jenkins model will have, in general, an unstable predictor if the process is unstable, and other models will still have stable predictors. This means that the Box-Jenkins model cannot be used to identify an unstable process. The ARMAX model is recommended to do the job.

However, the problem of the Box-Jenkins model can be fixed; see Forssell and Ljung (2000). The idea is to introduce a "fake" noise model as

$$y(t) = \frac{B(q)}{A(q)}u(t) + \frac{A_a^*(q)}{A_a(q)}\frac{C(q)}{D(q)}e(t) \quad (5.5.33)$$

where $A_a(q)$ is a monic polynomial that has all the unstable zeros of $A(q)$ and $A_a^*(q)$ is a monic polynomial that has all the (stable) zeros of $A_a(q)$ reflected into the unit disc. The predictor for this model is then

$$\hat{y}(t|\theta) = \frac{A_a(q)}{A_a^*(q)} \frac{D(q)B(q)}{C(q)A(q)} u(t) + [1 - \frac{A_a(q)}{A_a^*(q)} \frac{D(q)}{C(q)}] y(t) \quad (5.5.34)$$

and the unstable part of $A(q)$ will cancel out. Hence the predictor (5.5.34) is stable and the gradient can be calculated without problems. It can be shown (see Forssell and Ljung, 2000) that the modified Box-Jenkins model is asymptotically equal to the original one, but now without numerical problems.

5.6 More on Order/Structure Selection

The purpose of order/structure selection is to find the model order or structure that is most accurate for its use (in control). There are some order selection methods that do not need complete parameter estimation of the model. These include checking the rank of covariance matrices, plotting the singular value of the Hankel matrix (Appendix A); see Ljung (1987). In general these methods are not accurate for noisy data. In this work, it is recommended to first estimate models with increasing orders, and then to choose the best order using an error criterion.

Basically two choices need to be made: 1) error criterion, and 2) data set on which the criterion is evaluated. So far we suggest using output error criterion for order selection and evaluating the criterion on fresh data, the so-called cross-validation. The use of a fresh data set for order selection is a very natural one. The disadvantage of cross-validation is its high cost, because an additional test is required to collect fresh data. It is more economical to use estimation data for order selection. However, when an estimation data is used, an error criterion will, in general, monotonically decrease as a function of order, even when the order passes the best value, since a higher order model will have more freedom to fit the given estimation data. This is the so-called over-fit problem. In technical terms, the error criterion using the estimation data is a biased estimate of the error criterion using the validation data.

A number of criteria exist that use estimation data for order selection. In identification literature, often the prediction error criterion is used in model order selection and model validation. Denote prediction error criterion using a validation data set as

$$V_{PE}^V = \frac{1}{N} \sum_{t=1}^N \hat{H}^{-1}(q)[y^v(t) - \hat{G}(q)u^v(t)]^2 \quad (5.6.1)$$

where $u^v(t)$ and $y^v(t)$ are the input and output from the validation data set. Denote prediction

error criterion using the estimation data set as

$$V_{PE}^E = \frac{1}{N} \sum_{t=1}^N \hat{H}^{-1}(q)[y(t) - \hat{G}(q)u(t)]^2 \quad (5.6.2)$$

Assume that the white noise $e(t)$ is Gaussian and the model order is correct, then Akaike (1974, 1981) has derived an asymptotically unbiased estimate of V_{PE}^V using V_{PE}^E :

$$EV_{PE}^V = \frac{N+d}{N-d} V_{PE}^E \quad (5.6.3)$$

where N is the number of estimation data samples and d is the number of model parameters. This is the famous *Akaike's Final Prediction Error* criterion (FPE). Here the factor $\frac{N+d}{N-d}$ is used to correct for the over-fit effect or to punish the higher order.

For the purpose of control, the most important requirement is to obtain a model between process input and output. This requires that the simulation error, or, output error of the model should be small. Many practical examples have shown that a small prediction error does not always lead to a small output error and sometimes the result based on prediction error can be totally misleading; see Söderström (1974) for an example. Therefore, output error criterion is much more suitable for order selection for control. Based on this observation and motivated by saving test time, we will propose the following *final output error criterion* (FOE) (Zhu, 2000a and Zhu 2000c):

$$\begin{aligned} FOE &= \frac{N+d}{N-d} V_{OE}^E \\ &= \frac{N+d}{N-d} \frac{1}{N} \sum_{t=1}^N [y(t) - \hat{G}(q)u(t)]^2 \end{aligned} \quad (5.6.4)$$

where V_{OE} is the output-error (or simulation error) criterion on the estimation data set. Here, the factor $\frac{N+d}{N-d}$ is used to correct the overfit and V_{OE} is used to reflect the model purpose, namely, control. It is still an open problem to prove that FOE is an unbiased estimate of the output criterion on validation data.

Remember that in the last section we recommended the prediction error criterion for parameter estimation which is a different criterion than the FOE for order selection. We may call this way of using different criteria for model estimation and order selection the *cross-criterion approach*. The following exercise will compare FOE and FPE in two different situations.

Exercise 5.6.1. (1) Unbiased case. Simulate the process:

$$y(t) = \frac{B^o(q)}{A^o(q)} u(t) + v(t)$$

where

$$\begin{aligned} A^o(z^{-1}) &= 1.0 - 1.15q^{-1} - 0.295q^{-2} + 0.7525q^{-3} - 0.02575q^{-4} - 0.14555q^{-5} \\ &\quad - 0.00005q^{-6} + 0.002465q^{-7} - 0.000105q^{-8} \\ B^o(z^{-1}) &= 0 + 1.0q^{-1} + 1.0q^{-2} + 0.65q^{-3} + 2.20q^{-4} + 1.0q^{-5} \\ v(t) &= \frac{\alpha}{1 - 0.9q^{-1}} e(t) \end{aligned}$$

$e(t)$ is a white Gaussian noise and α is a constant that is used to obtain a signal-to-noise ratio of 10% in power at the output. Use a GBN with an average switch time of 8 samples as the input signal; the number of samples is 1000. Run 100 open-loop simulations. For each simulation, estimate the Box-Jenkins models of orders between 2 and 12. Then calculate the the FOE, the Akaike's FPE and the noise-free output error criteria for each model. Finally plot the average of the three criteria over the 100 simulations.

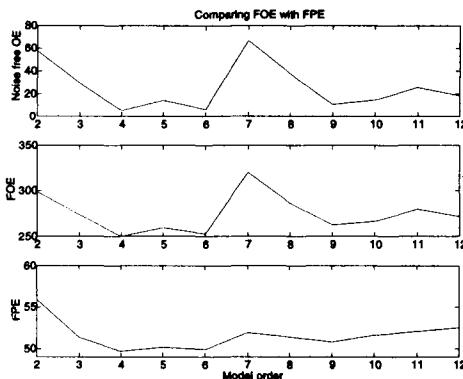


Figure 5.6.1 The averages of the FOE, FPE and the true OE criteria over the 100 simulations.

(2) **Biased case.** Do the same for the following process:

$$y(t) = \frac{1 + .5q^{-1} + .4q^{-2} + 2q^{-3}}{1 + .5q^{-1} - .4q^{-2} - .262q^{-3} - .03q^{-4}} u(t) + v(t)$$

but this time estimate the model with 1 delay. Note that the delay will cause all models be biased.

Answer. The M file is as follows; only the calculation part is given. The results of the first process are plotted in Figure 5.6.1 and the results of second process in Figure 5.6.2. We see that the FOE performs better than the FPE, especially in the biased case. This means that the FOE is more robust in practical situations. On a few occasions, unstable models are identified which are caused by the delay. The criteria of unstable models are not included in computing the averages.

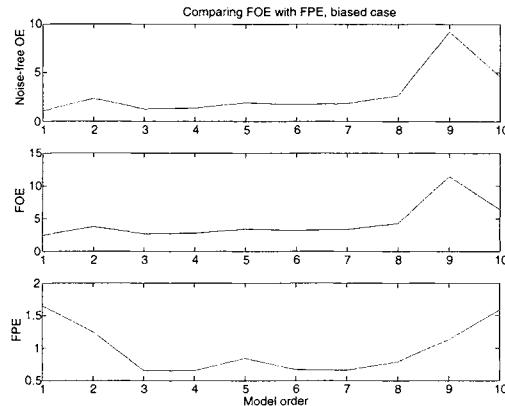


Figure 5.6.2 Compare FOE with FPE

```
%%% foe_fpe.m; M file for comparing FOE and FPE %%%
function [FOEbj,FPEbj,TOEbj] = foe_fpe(Ao,Bo,N,sn,n_sim,Delay)
%
% [FOEbj,FPEbj,TOEbj] = foe_fpe(Ao,Bo,N,sn,n_sim,Delay)
%
% This function compares the FOE and FPE criterion of BJ model
%
% Input arguments:
% [Ao, Bo]: Process model polynomials
% N: Number of samples to be simulated
% sn: noise-to-signal ratio
% n_sim: number of simulations
% Delay: Delay, optional
%
% Output arguments:
% FOEbj: FOE Error of the BJ models
% FPEbj: FPE of the BJ models
% TOEbj: True (noise-free) OE of BJ models
%
FOEbj = [];
FPEbj = [];
TOEbj = [];
if exist('Delay') == 0
    Delay = 0;
```

```

end
for k = 1:n_sim
    U = gbngen(N,10);
    Yo = filter(Bo,Ao,U);
    V = filter([1],[1 -.9],randn(N,1));
    V = V/std(V)*sqrt(sn)*std(Yo);
    Y = Yo + V;
    FOEtem = zeros(1,12);
    FPEtem = zeros(1,12);
    TOEtem = zeros(1,12);
    for n = 1:12
        thBJ = bj([Y U],[n n n n Delay]);
        ybj = idsim(U,thBJ);
        FOEtem(n) = (N+4*n)/(N-4*n)*cov(Y-ybj);
        FPEtem(n) = thBJ(2,1);
        TOEtem(n) = cov(Yo-ybj);
    end
    FOEBj = [FOEbj;FOEtem];
    FPEbj = [FPEbj;FPEtem];
    TOEBj = [TOEBj;TOEtem];
end

```

5.7 Model Validation

As mentioned in Chapter 1, the goal of model validation is to check if the identified model is good enough for use in control. If the answer is negative, preferably, the validation step can provide the necessary information for a possible remedy. At this stage of the development, we shall discuss the standard methods in the literature.

Compare the Model Property against the Process *a Priori* Knowledge

In practice, some *a priori* knowledge is often available that is obtained from process operation experience and process pretest. This often includes a rough estimate of the process gain and sometimes the dominant time constant. Therefore, model gains and settling time can be compared with their rough estimates. This can give an initial indication of model quality. A convenient way to do this is to plot the model step response.

Check the Agreement of Different Methods

When the test data is informative and signal-to-noise ratio is high, different models or methods will give similar results. Therefore, one can estimate FIR, ARX, ARMAX and Box-Jenkins models with proper orders and compare their step responses and frequency responses.

Simulation

The simulated process output using the identified model $\hat{G}(q)$ is given as

$$\hat{y}(t) = \hat{G}(q)u(t) \quad (5.7.1)$$

and the simulation error or output error is

$$\begin{aligned}\hat{e}(t) &= y(t) - \hat{y}(t) \\ &= [G^o(q) - \hat{G}(q)]u(t) + v(t)\end{aligned} \quad (5.7.2)$$

Hence the discrepancies are due to the model error and the disturbance. If the signal-to-noise ratio is high, a good fit of $\hat{y}(t)$ to $y(t)$ will imply a good process model. Plotting $\hat{y}(t)$ and $y(t)$ together will, in general, give a very good intuitive judgment of the model quality for control when the input is properly designed. Naturally, for model simulation, fresh data will be more reliable due to the over-fit problem when using estimation data. However, when the number of data N is large (> 300) and model order is low (< 6), the overfit problem is not serious. Therefore, the estimation data can be used for the simulation if additional test is expensive.

For a single variable process a small simulation error generally implies a good model, provided that the test is properly designed. For a multi-input process, the situation becomes complex; a small simulation error only means that models of some inputs are good. On the other hand, a poor fit of simulation does not necessarily mean a poor model. The simulation error in (5.7.2) contains the disturbance. If the disturbance level is high, the simulation error will be large even if the model is perfect. Parametric model identification has the power of averaging which reduces the model error caused by disturbance; see (5.4.31). For example, when the signal-to-noise ratio is very low but the test time is very long, a good model can still be obtained. Unfortunately, this can not be seen using a simulation check.

In literature, sometimes prediction is also used to check the quality of the model. As mentioned before, a small prediction error may not mean a small simulation error. Experience shows that it is quite easy to create a good one-step-ahead prediction with a poor process model. A well known example is the following trivial predictor

$$\hat{y}(t+1|\theta) = y(t) \quad (5.7.3)$$

It will predict that the next output equals the current one. If the process output does not vary very fast, which is often the case in practice, this predictor will have a very small prediction error. But $G(q) = 0$, and hence it is not useful at all for control and simulation.

Residual Analysis

In the framework of the prediction error method, it is assumed that the disturbance $v(t)$ is created by filtering a white noise source $e(t)$. When the model is estimated, the prediction error can be estimated as

$$\varepsilon(t) = \varepsilon(t, \hat{\theta}) = y(t) - \hat{y}(t|\theta) \quad (5.7.4)$$

with the predictor

$$\hat{y}(t|\theta) = \frac{\hat{G}(q, \hat{\theta})}{\hat{H}(q, \hat{\theta})} u(t) + \left[1 - \frac{1}{\hat{H}(q, \hat{\theta})} \right] y(t) \quad (5.7.5)$$

Here $\varepsilon(t, \hat{\theta})$ is called residual which can be treated as the leftovers of the modeling. If the identified model is accurate then $\varepsilon(t, \hat{\theta})$ will be a good estimate of the noise source $e(t)$. Remember that two assumptions have been made about $e(t)$:

1. $e(t)$ is white noise with zero mean
2. $e(t)$ is uncorrelated with past input $u(t)$ (it can be correlated to future input in closed-loop test)

These two conditions can be checked using the residuals. Calculate the estimate of autocorrelation function of the residuals

$$\hat{R}_\varepsilon(\tau) = \frac{1}{N} \sum_{t=1}^N \varepsilon(t) \varepsilon(t - \tau) \quad (5.7.6)$$

When $\varepsilon(t)$ is white noise, $\hat{R}_\varepsilon(\tau)$ should look like an impulse (has a peak at $\tau = 0$ and zero else where). Calculate the estimate of the cross correlation function between residuals and past input

$$\hat{R}_{\varepsilon u}(\tau) = \frac{1}{N} \sum_{t=1}^N \varepsilon(t) u(t - \tau), \quad \tau = 1, 2, \dots \quad (5.7.7)$$

Then $\hat{R}_{\varepsilon u}(\tau)$ should be nearly zero for $\tau > 0$. If this correlation is not zero, then there is part of the output originating from the past input left in the residuals and has not yet been picked up by the model. In linear process identification, this usually means that the model order is too low.

Note that $\hat{R}_\varepsilon(\tau)$ and $\hat{R}_{\varepsilon u}(\tau)$ are finite sample estimates, their values will deviate from theoretical ones. Therefore, some thresholds need to be established in order to test if these correlation estimates are zero; see Ljung (1987) and Söderström and Stoica (1989) for details.

Summarizing, the first three methods are more or less qualitative tests, the residual analysis is to check if the model agrees sufficiently well with the test data. None of the methods can tell if the model is suitable for control. Assume that a model has passed the residual analysis. This only means that one has done a good job for the given data. But one still does not know if the data is good enough (for control). Therefore these methods are not control relevant.

In identification for control, an effective way to model validation is to quantify the error in the frequency domain. This information can then be used for analyzing the robust stability and performance of the controlled system using the model. This direction received considerable attention in the 1990's; see, e.g., Goodwin and Salgado (1989), Goodwin *et. al.* (1992), Hakvoort and van den Hof (1997), Zhu (1989a) and Zhu *et. al.* (1991). In Chapter 6, a method of estimating the model error bound in the frequency domain will be given.

5.8 Identifying the Glass Tube Drawing Process

Now it is a good moment to come back to practice. Here the glass tube process is identified. The purpose of the exercise is two fold: 1) to outline an identification procedure using prediction error methods and 2) to compare various methods in a practical situation.

Let us recall the definitions of input-output variables

Inputs: gas pressure and drawing speed

Outputs: wall thickness and diameter

The same data sets used in Chapter 4 will be used here. Remember that one has an estimation data set of 600 samples and validation data set of 669 samples.

Model Structure for Parameter Estimation

For the 2-input 2-output process, a diagonal form ARMAX model can be written as

$$\begin{cases} A_1(q)y_1(t) = B_{11}(q)u_1(t) + B_{12}(q)u_2(t) + C_1(q)\varepsilon_1(t) \\ A_2(q)y_2(t) = B_{21}(q)u_1(t) + B_{22}(q)u_2(t) + C_2(q)\varepsilon_2(t) \end{cases} \quad (5.8.1)$$

where $A_i(q)$, $B_{ij}(q)$ and $C_i(q)$ are polynomials of the delay operator q^{-1} . Polynomial $A_i(q)$ and $C_i(q)$ are monic. Hence, we get two MISO sub-models; each of which can be estimated separately. Note that polynomial $A_i(q)$ is the common denominator of the process transfer functions and $C_i(q)/A_i(q)$ is the disturbance model of $y_i(t)$. An output error model can be written as

$$\begin{cases} y_1(t) = \frac{B_{11}(q)}{A_{11}(q)}u_1(t) + \frac{B_{12}(q)}{A_{12}(q)}u_2(t) + \varepsilon_1(t) \\ y_2(t) = \frac{B_{21}(q)}{A_{21}(q)}u_1(t) + \frac{B_{22}(q)}{A_{22}(q)}u_2(t) + \varepsilon_2(t) \end{cases} \quad (5.8.2)$$

where $A_{ij}(q)$ and $B_{ij}(q)$ are polynomials of the delay operator q^{-1} , and $A_{ij}(q)$ are monic. A Box-Jenkins model can be written as

$$\begin{cases} y_1(t) = \frac{B_{11}(q)}{A_{11}(q)}u_1(t) + \frac{B_{12}(q)}{A_{12}(q)}u_2(t) + \frac{C_1(q)}{D_1(q)}\varepsilon_1(t) \\ y_2(t) = \frac{B_{21}(q)}{A_{21}(q)}u_1(t) + \frac{B_{22}(q)}{A_{22}(q)}u_2(t) + \frac{C_2(q)}{D_2(q)}\varepsilon_2(t) \end{cases} \quad (5.8.3)$$

where $A_{ij}(q)$ and $B_{ij}(q)$ are polynomials of the delay operator q^{-1} , and $A_{ij}(q)$, $C_i(q)$ and $D_i(q)$ are monic. These structures are used in Matlab System Identification Toolbox.

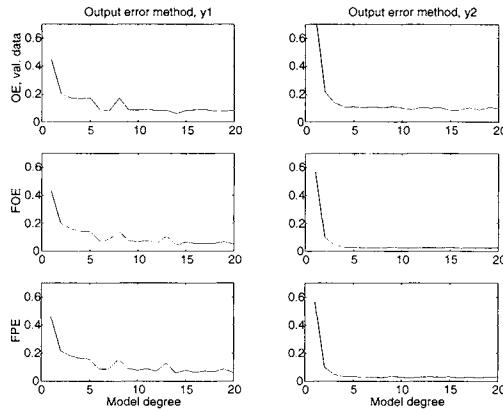


Figure 5.8.1 Order selection using FPE, FOE and true OE criterion, output error models

Order and Structure Selection

The above three model structures or methods are used here. For each model structure, models are estimated with their polynomial degrees increased from 1 to 20. For all models, Akaike's FPE and FOE criteria are calculated using the estimation data set. Then output error criterion is calculated using the validation data set (cross-validation) which will be called the true OE criterion. The M file that performs the order selection is given in the following. The three criteria as a function of model degree are plotted in Figures 5.8.1, 5.8.2 and 5.8.3 for the three model structures. One sees that the shapes of the FOE is very similar to that of true OE, which mean that it is more reliable for order selection in practice. Note that the FPE and the FOE are theoretically the same for output error models, the small difference in Figure 8.1 is due to numerical implementation. Note also that the criteria do not decrease monotonically; this can be caused by numerical problems. According to the true OE criterion, the best degrees are [14, 14, 15, 15] for the output error model with the true OE criteria as 0.0605 and 0.0815; the best degrees are [14, 14, 19, 19] for the Box-Jenkins model with true the OE criteria being 0.0687 and 0.1024. These orders may be much higher than one expects. If one only estimates a second order model for each transfer function, the error will be considerably larger, especially for Box-Jenkins model.

However, much lower orders can be obtained if one is willing to give up a little bit accuracy. From Figures 5.8.1 and 5.8.2 one can see that a "good" output error model has the degrees [6, 6, 4, 4] with the true OE [0.0896, 0.1096], and a "good" Box-Jenkins model has the degrees [7, 7, 4, 4] with the true OE [0.0844, 0.1338]. These models will be validated in the next step. Note that the best OE model is better than the best Box-Jenkins model for the glass tube process, but the difference is marginal.

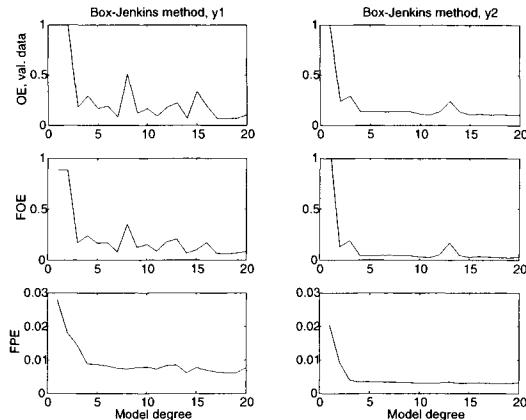


Figure 5.8.2 Order selection using FPE, FOE and true OE criterion, Box-Jenkins models

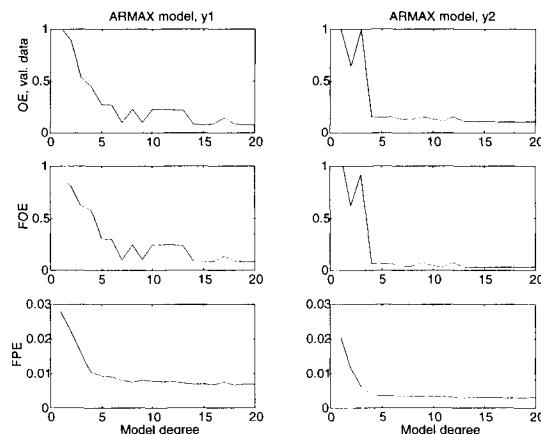


Figure 5.8.3 Order selection using FPE, FOE and true OE criterion, ARMAX models

```
%%% glassorder.m; order selection for the glass tube drawing process %%%
function [ERRvaloe,FPEoe,FOEoe,ERRvalbj,FPEbj,...  
FOEbj,ERRvalabc,FPEabc,FOEabc] = glassorder();  
  
% Load data  
load glassdata2  
ERRvaloe = [] ; FPEoe = [] ; FOEoe = [] ;  
ERRvalbj = [] ; FPEbj = [] ; FOEbj = [] ;
```

```

ERRvalabc = [] ;FPEabc = [] ;FOEabc = [] ;
N = length(Uest);
ords = 1:20;
iter = 30;
% Loop with increasing order
for n = ords
    % Model estimation
    TH1oe = oe([Yest(:,1) Uest],[n n n n 1 1],iter);
    TH2oe = oe([Yest(:,2) Uest],[n n n n 1 1],iter);
    TH1bj = bj([Yest(:,1) Uest],[n n n n n 1 1],iter);
    TH2bj = bj([Yest(:,2) Uest],[n n n n n 1 1],iter);
    TH1abc = armax([Yest(:,1) Uest],[n n n n 1 1],iter);
    TH2abc = armax([Yest(:,2) Uest],[n n n n 1 1],iter);
    % Get FPE
    FPEbj = [FPEbj;TH1bj(2,1) TH2bj(2,1)];
    FPEoe = [FPEoe;TH1oe(2,1) TH2oe(2,1)];
    FPEabc = [FPEabc;TH1abc(2,1) TH2abc(2,1)];
    % Calculate FOE
    fact = (N+3*n)/(N-3*n);
    Ye1oe = idsim(Uest,TH1oe);
    Ye2oe = idsim(Uest,TH2oe);
    FOEoe = [FOEoe; fact*cov(Yest(:,1)-Ye1oe) fact*cov(Yest(:,2)-Ye2oe)];
    Ye1bj = idsim(Uest,TH1bj);
    Ye2bj = idsim(Uest,TH2bj);
    FOEBj = [FOEBj; fact*cov(Yest(:,1)-Ye1bj) fact*cov(Yest(:,2)-Ye2bj)];
    Ye1abc = idsim(Uest,TH1abc);
    Ye2abc = idsim(Uest,TH2abc);
    FOEabc = [FOEabc; fact*cov(Yest(:,1)-Ye1abc) fact*cov(Yest(:,2)-Ye2abc)];
    % Calculate OE criterion using validation data
    Y1oe = idsim(Uval,TH1oe);
    Y2oe = idsim(Uval,TH2oe);
    Y1bj = idsim(Uval,TH1bj);
    Y2bj = idsim(Uval,TH2bj);
    Y1abc = idsim(Uval,TH1abc);
    Y2abc = idsim(Uval,TH2abc);
    Erroe = [cov(Y1oe-Yval(:,1))/cov(Yval(:,1)) cov(Y2oe-Yval(:,2))/cov(Yval(:,2))];
    ERRvaloe = [ERRvaloe;Erroe];
    Errbj = [cov(Y1bj-Yval(:,1))/cov(Yval(:,1)) cov(Y2bj-Yval(:,2))/cov(Yval(:,2))];
    ERRvalbj = [ERRvalbj;Errbj];
    Errabc=[cov(Y1abc-Yval(:,1))/cov(Yval(:,1)) cov(Y2abc-Yval(:,2))/cov(Yval(:,2))];
    ERRvalabc = [ERRvalabc;Errabc];
    disp(['n = ',num2str(n)])
end

```

```
% Plot error criteria
% Omitted .....
```

Model Validation

The output error models with degrees [14, 15] and [6, 4] and the Box-Jenkins models with degrees [14, 19] and [7, 4] will be validated using the methods introduced in Section 5.7. First the residual analysis is carried out. The results of output 1 (thickness) are plotted in Figures 5.8.4 to 5.8.7. The results for output 2 (diameter) are very similar. Note that residual whiteness test is not applicable for output error models when the disturbance $v(t)$ is not white noise. Hence only cross-correlation functions need to be checked. We can say that all models pass the test. The differences between the low and high order models are marginal.

The four models are simulated using the validation data set. The measured outputs and the simulated outputs are plotted in Figures 5.8.8 and 5.8.9. From eye inspection these models are very good. Compare the model fit here with that of low order least-squares model in Section 4.3.2, considerable improvement has been obtained.

The M file for model validation is as follows.

```
%%% glassvalid.m; model validation for the glass tube process %%%
% Load dada
load glassdata2
N = length(Uval);
iter = 30;
% Model estimation and residual analysis
TH1oe6 = oe([Yest(:,1) Uest],[6 6 6 6 1 1],iter);
TH2oe4 = oe([Yest(:,2) Uest],[4 4 4 4 1 1],iter);
TH1oe14 = oe([Yest(:,1) Uest],[14 14 14 14 1 1],iter);
TH2oe15 = oe([Yest(:,2) Uest],[15 15 15 15 1 1],iter);
TH1bj7 = bj([Yest(:,1) Uest],[7 7 7 7 7 1],iter);
TH2bj4 = bj([Yest(:,2) Uest],[4 4 4 4 4 1],iter);
TH1bj14 = bj([Yest(:,1) Uest],[14 14 4 4 14 14 1 1],40);
TH2bj19 = bj([Yest(:,2) Uest],[19 19 4 4 19 19 1 1],40);
disp('Press ''Enter''')
[E1oe6,R1oe6]=resid([Yval(:,1) Uval],TH1oe6);
disp('Press ''Enter''')
[E2oe4,R2oe4]=resid([Yval(:,2) Uval],TH2oe4);
disp('Press ''Enter''')
[E1oe14,R1oe14]=resid([Yval(:,1) Uval],TH1oe14);
disp('Press ''Enter''')
[E2oe15,R2oe15]=resid([Yval(:,2) Uval],TH2oe15);
disp('Press ''Enter''')
[E1bj7,R1bj7]=resid([Yval(:,1),Uval], TH1bj7);
```

```

disp('Press ''Enter'''')
[E2bj4,R2bj4]=resid([Yval(:,2),Uval], TH2bj4);
disp('Press ''Enter'''')
[E1bj14,R1bj14]=resid([Yval(:,1),Uval], TH1bj14);
disp('Press ''Enter'''')
[E2bj19,R2bj19]=resid([Yval(:,2),Uval], TH2bj19);
% Simulations and plots; omitted...

```

Remark. The prediction error residuals of the models are not used here. The reason is that it is easy to obtain a prediction error model (ARMAX or Box-Jenkins) with a very small prediction error while the simulation or output error is large. For example, using the validation data set, the prediction error residual of the 7th degree Box-Jenkins model for output 1 is 0.79% of the output in power and 4th degree Box-Jenkins model for output 2 is 0.52% of the output. These values are misleadingly small. For the same data, the trivial predictor $\hat{y}(t+1|\theta) = y(t)$ will have errors of 6.25% and 5.42% of the two outputs which are smaller than the output errors of the best model.

The step responses of the four models are plotted in Figure 5.8.10. It is surprising that the gains of the models are so different, especially those of transfer between input 1 and output 1. If we had not checked these plots, we may have thought that all four models have the same behavior, because residual analysis and simulation results for the models are nearly the same.

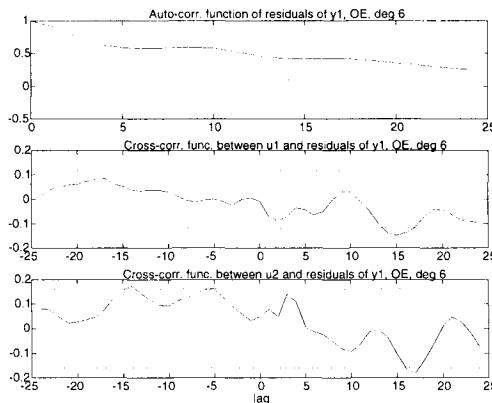


Figure 5.8.4 Residual analysis for output 1, output error model with degree 6. Dotted lines define the 99% confidence interval

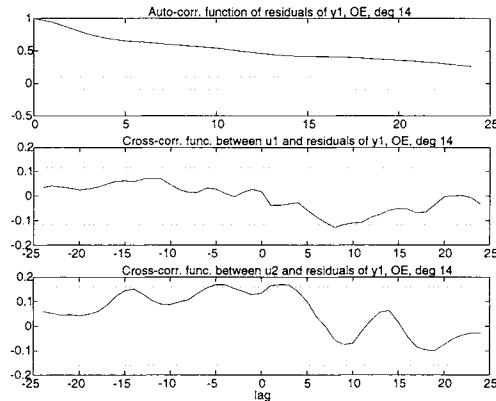


Figure 5.8.5 Residual analysis for output 1, output error model with degree 14. Dotted lines define the 99% confidence interval

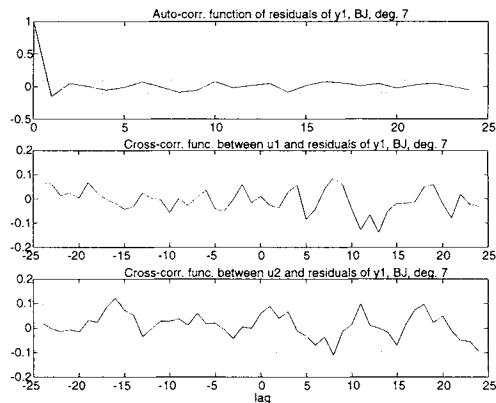


Figure 5.8.6 Residual analysis for output 1, Box-Jenkins model with degree 7. Dotted lines define the 99% confidence interval

Discussion about the Results

First of all, with the methods learned from this chapter, one can obtain much better results with respect to simulation. This is achieved by using more complex model structures which require nonlinear minimization routines which are numerically not always reliable. The numerical difficulties are apparent in the plots of output error criteria of the two model structures in Figures 8.1 and 8.2. Often the criteria do not decrease with the order, especially for output 1.

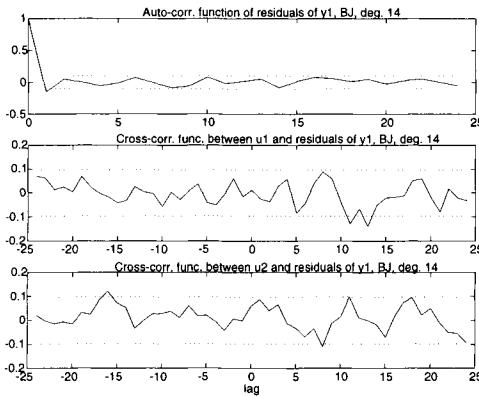


Figure 5.8.7 Residual analysis for output 1, Box-Jenkins model with degree 14. Dotted lines define the 99% confidence interval

In model validation, the residual analysis and simulation can only tell how well the model can fit the given data, but they cannot tell if the model or data set is good enough for control. One may be very pleased that all four models pass residual analysis and simulation errors are very small. But the step responses show that there is big uncertainty in model gains. This uncertainty is caused by the lack of information of the data set at low frequencies. Remember that the test inputs are white-noise-like PRBS signals which have flat power spectra.

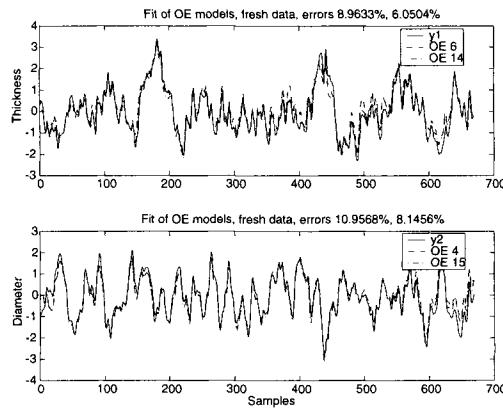


Figure 5.8.8 Model simulations of the output error models, validation data. The relative errors in power are also given.

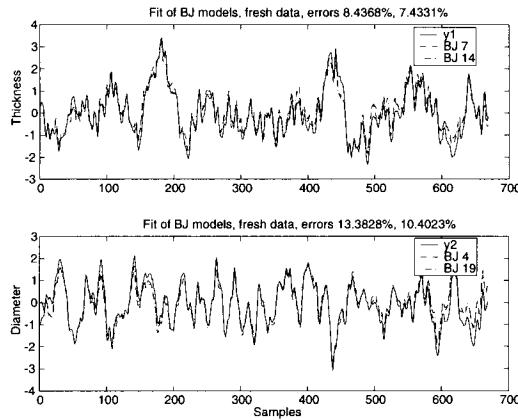


Figure 5.8.9 Model simulations of the Box-Jenkins models, validation data. The relative errors in power are also given.

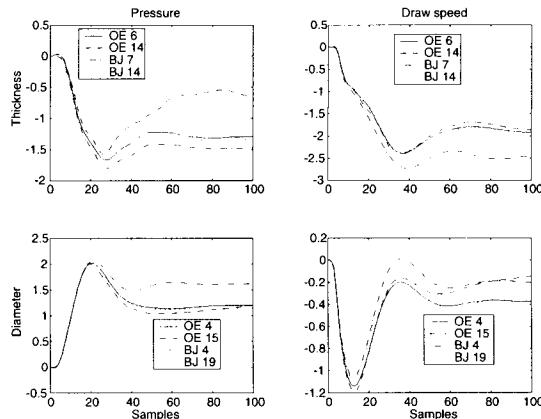


Figure 5.8.10 Step responses of the four selected models.

In order selection, one may need to use a very high order to have a criterion minimized. But if one does not care about the last percent of the error criterion, a much lower order model can be selected. Model uncertainty in the low frequency range are obvious in the step response plots. Take the transfer function between input 1 and output 1 as an example. The output error model with degree 14 has the smallest error criterion, but its gain is about half of the gains of other three models. Because other three models are very close to each other, they may have

better gain estimates. If we have not plotted various step responses together, we may not notice the gain uncertainty. Therefore, some form of model uncertainty description in the frequency domain is desired.

We shall revisit this process in Chapter 7.

5.9 Recursive Parameter Estimation

In some cases it may be necessary to estimate a model on-line while the process is in operation. A typical example is adaptive control. The model will be updated when the new observations are available. Hence for computing efficiency, it is desirable to arrange the algorithms in such a way that the results obtained previously can be used for on-line updating. This way of computing the estimates is called recursive estimation or adaptive estimation which will be studied in this section. First recursive LS algorithm will be derived and then a recursive output error algorithm will be outlined.

5.9.1 Recursive Least-Squares Method

Write transfer function model in the linear regression

$$y(t) = \varphi(t)\theta + \varepsilon(t) \quad (5.9.1)$$

where

$$\begin{aligned} \varphi(t) &= [-y(t-1) \cdots -y(t-n) \ u(t-1) \cdots u(t-n)] \\ \theta &= [a_1 \cdots a_n \ b_1 \cdots b_n]^T \end{aligned}$$

and $\varepsilon(t)$ is the equation error. The LS estimate which minimizes the loss function

$$V_{LS}(t) = \frac{1}{t} \sum_{k=1}^t \varepsilon(k)^2 = \frac{1}{t} \sum_{k=1}^t [y(k) - \varphi(k)\theta]^2$$

is

$$\hat{\theta}(t) = \left[\sum_{k=1}^t \varphi(k)^T \varphi(k) \right]^{-1} \left[\sum_{k=1}^t \varphi(k)^T y(k) \right] \quad (5.9.2)$$

The argument t is used here to indicate the dependence of $\hat{\theta}$ on time. Introduce the matrix

$$P(t) = \left[\sum_{k=1}^t \varphi(k)^T \varphi(k) \right]^{-1} \quad (5.9.3)$$

We have

$$\hat{\theta}(t) = P(t) \left[\sum_{k=1}^t \varphi(k)^T y(k) \right] \quad (5.9.4)$$

It is easy to see that

$$\begin{aligned} P(t+1)^{-1} &= \left[\sum_{k=1}^{t+1} \varphi(k)^T \varphi(k) \right] = \left[\sum_{k=1}^t \varphi(k)^T \varphi(k) \right] + \varphi(t+1)^T \varphi(t+1) \\ &= P(t)^{-1} + \varphi(t+1)^T \varphi(t+1) \end{aligned} \quad (5.9.5)$$

Then at time $t+1$

$$\begin{aligned} \hat{\theta}(t+1) &= P(t+1) \left[\sum_{k=1}^t \varphi(k)^T y(k) + \varphi(t+1)^T y(t+1) \right] \\ &= P(t+1) \left[[P(t)^{-1} P(t)] \sum_{k=1}^t \varphi(k)^T y(k) + \varphi(t+1)^T y(t+1) \right] \\ &\quad [\text{use (5.9.4)}] \\ &= P(t+1) \left[P(t)^{-1} \hat{\theta}(t) + \varphi(t+1)^T y(t+1) \right] \\ &\quad [\text{use (5.9.5)}] \\ &= P(t+1) \left\{ [P(t+1)^{-1} - \varphi(t+1)^T \varphi(t+1)] \hat{\theta}(t) + \varphi(t+1)^T y(t+1) \right\} \\ &= \hat{\theta}(t) + P(t+1) \varphi(t+1)^T [y(t+1) - \varphi(t+1) \hat{\theta}(t)] \end{aligned} \quad (5.9.6)$$

This leads to the first version of the recursive formulas

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t+1) \varepsilon(t+1) \quad (5.9.7a)$$

$$\varepsilon(t+1) = y(t+1) - \varphi(t+1) \hat{\theta}(t) \quad (5.9.7b)$$

$$K(t+1) = P(t+1) \varphi(t+1)^T \quad (5.9.7c)$$

Remember that the equation error $\varepsilon(t)$ can be interpreted as a prediction error. The interpretation of the recursive formula (5.9.7a) is, if $\varepsilon(t)$ is small, then estimate $\hat{\theta}(t)$ is good and should not be modified very much. The vector $K(t+1)$ is a weighting showing how much each parameter will be modified.

In this version (5.9.5) must be used to calculate $P(t+1)$. This needs a matrix inversion at each time step, which is very time-consuming. Using the matrix inversion lemma (Appendix A):

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1}$$

one obtains

$$\begin{aligned} P(t+1) &= \left[\sum_{s=1}^t \varphi(s)^T \varphi(s) + \varphi(t+1)^T \varphi(t+1) \right]^{-1} \\ &= P(t) - P(t) \varphi(t+1)^T \varphi(t+1) P(t) [1 + \varphi(t+1)^T P(t) \varphi(t+1)]^{-1} \end{aligned} \quad (5.9.8)$$

Applying this result yields

$$\begin{aligned} K(t+1) &= P(t+1) \varphi(t+1)^T \\ &= P(t) \varphi(t+1)^T [1 + \varphi(t+1)^T P(t) \varphi(t+1)]^{-1} \end{aligned} \quad (5.9.9)$$

Now one obtains the second version of the recursive LS method

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t+1) \varepsilon(t+1) \quad (5.9.10a)$$

$$\varepsilon(t+1) = y(t+1) - \varphi(t+1) \hat{\theta}(t) \quad (5.9.10b)$$

$$\begin{aligned} K(t+1) &= P(t+1) \varphi(t+1)^T \\ &= P(t) \varphi(t+1)^T [1 + \varphi(t+1)^T P(t) \varphi(t+1)]^{-1} \end{aligned} \quad (5.9.10c)$$

$$P(t+1) = P(t) - P(t) \varphi(t+1)^T \varphi(t+1) P(t) [1 + \varphi(t+1)^T P(t) \varphi(t+1)]^{-1} \quad (5.9.10d)$$

Here no matrix inversion is needed.

Note that in the derivation of the recursive LS algorithm, no approximation has been made. Therefore, the recursive LS estimate and the off-line estimate are theoretically identical. This is another advantage of the LS method.

The updating (5.9.10d) for $P(t+1)$ is not always numerically robust. Rounding errors may accumulate and make the computed $P(t+1)$ indefinite, even though it is theoretically always positive definite. When $P(t+1)$ becomes indefinite the parameter estimates tend to diverge. A way to overcome this difficulty is to use a square root algorithm. Define $S(t+1)$ through

$$P(t+1) = S(t+1) S(t+1)^T \quad (5.9.11)$$

and update $S(t+1)$ instead of $P(t+1)$. For details of square root algorithms we refer the interested reader to the book of Ljung and Söderström (1983).

The initial values $P(0)$ and $\hat{\theta}(0)$ can be obtained by the off-line LS method, or by simply taking

$$\hat{\theta}(0) = 0, \quad P(0) = \rho I \quad (5.9.12)$$

where ρ is a "large" number.

Forgetting factor

When the process is slowly time varying, the measurements obtained a long time ago contain less information about the process than the recent measurements. In order to let the estimator follow the change of the process, it is desirable to discount the old measurements in the estimation algorithm. This can be realized by introducing a forgetting factor λ and modifying the LS loss function as follows

$$V_{LS}(t) = \frac{1}{t} \sum_{s=1}^t \lambda^{t-s} \varepsilon(s)^2 \quad (5.9.13)$$

Here λ is a number just less than one (for example 0.99). This means that with increasing t the measurements obtained previously are discounted. The recursive LS method with a forgetting factor is the same as in (5.9.10) except that $P(t+1)$ is modified to

$$P(t+1) = \{P(t) - P(t)\varphi(t+1)^T \varphi(t+1)P(t)[\lambda + \varphi(t+1)P(t)\varphi(t+1)^T]^{-1}\}/\lambda \quad (5.9.14)$$

5.9.2 A Recursive Output Error Method

In Section 5.2 it has been shown that the Gauss-Newton algorithm for the output error method is

$$\hat{\theta}^{k+1} = \hat{\theta}^k - \left[\sum_{t=1}^N [\varphi^k(t)]^T \varphi^k(t) \right]^{-1} \sum_{t=1}^N [\varphi^k(t)]^T \varepsilon(t, \hat{\theta}^k) \quad (5.9.15)$$

where $\varepsilon(t, \hat{\theta}^k)$ is the output error residual

$$\varepsilon(t, \hat{\theta}^k) = y(t) - \hat{G}^k(q)u(t) \quad (5.9.16)$$

$\varphi^k(t)$ is the gradient of the output error residual

$$\varphi^k(t) = \left[\frac{\hat{B}^k(q)}{\hat{A}^k(q)^2} u(t-1) \cdots \frac{\hat{B}^k(q)}{\hat{A}^k(q)^2} u(t-n) \frac{-1}{\hat{A}^k(q)} u(t-1) \cdots \frac{-1}{\hat{A}^k(q)} u(t-n) \right] \quad (5.9.17)$$

At time $t+1$, we can approximate (7.2.1) by replacing $\hat{\theta}^k$ by $\hat{\theta}(t)$, $\hat{\theta}^{k+1}$ by $\hat{\theta}(t+1)$ and $-\varphi^k(t+1)$ by

$$\varphi(t+1) = \left[\frac{-\hat{B}_t(q)}{\hat{A}_t(q)^2} u(t) \cdots \frac{-\hat{B}_t(q)}{\hat{A}_t(q)^2} u(t-n+1) \frac{1}{\hat{A}_t(q)} u(t) \cdots \frac{1}{\hat{A}_t(q)} u(t-n+1) \right] \quad (5.9.18)$$

This yields

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \left[\sum_{k=1}^{t+1} \varphi(k)^T \varphi(k) \right]^{-1} \sum_{k=1}^{t+1} \varphi(k)^T \varepsilon_{OE}(s, \hat{\theta}(t))$$

Introducing

$$P(t+1) = \left[\sum_{k=1}^{t+1} \varphi(k)^T \varphi(k) \right]^{-1} \quad (5.9.19)$$

one obtains

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1) \left[\sum_{k=1}^t \varphi(k)^T \varepsilon_{OE}(k, \hat{\theta}(t)) + \varphi(t+1)^T \varepsilon_{OE}(t+1, \hat{\theta}(t)) \right] \quad (5.9.20)$$

Observe that the off-line estimate $\hat{\theta}(t)$ is obtained such that

$$\frac{\partial}{\partial \theta} \left[\sum_{k=1}^t \varepsilon_{OE}(k, \hat{\theta}(t))^2 \right] = 2 \sum_{k=1}^t \varphi(k)^T \varepsilon_{OE}(k, \hat{\theta}(t)) = 0$$

Using this equation in (5.9.20) with some approximation, we obtain

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1) \varphi(t+1)^T \varepsilon_{OE}(t+1, \hat{\theta}(t)) \quad (5.9.21)$$

This equation has the same algebraic structure as (5.9.7a). Therefore, one immediately obtains the recursive version for the output error method (5.9.15)

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t+1) \varepsilon_{OE}(t+1, \hat{\theta}(t)) \quad (5.9.22a)$$

$$\varepsilon(t+1, \theta(t)) = y(t+1) - \hat{G}_t(q) u(t+1) \quad (5.9.22b)$$

$$\begin{aligned} K(t+1) &= P(t+1) \varphi(t+1)^T \\ &= P(t) \varphi(t+1)^T [1 + \varphi(t+1) P(t) \varphi(t+1)^T]^{-1} \end{aligned} \quad (5.9.22c)$$

$$P(t+1) = P(t) - P(t) \varphi(t+1)^T \varphi(t+1) P(t) [1 + \varphi(t+1) P(t) \varphi(t+1)^T]^{-1} \quad (5.9.22d)$$

where $\varphi(t)$ is given in (5.9.18).

Similarly as for the recursive LS method, we can use a forgetting factor to track a time-varying process.

Exercise 5.9.1. Compare the recursive output error method with the off-line output error method. Given the same process as in Exercise 4.2.3:

$$y(t) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} u(t) + v(t)$$

Simulate the process using a GBN signal with mean switching time of 8 samples. The disturbance $v(t)$ is a lowpass filtered white noise $e(t)$:

$$v(t) = \frac{1}{1 - 0.9q^{-1}} e(t)$$

Adjust the variance of $v(t)$ so that the noise-to-signal ratio $\text{var}(v(t))/\text{var}(y_o(t)) = 10\%$. Generate 300 samples for model estimation. Estimate output error models with the true order using the output error method (`oe.m`) and the recursive output error method (`roe.m`). Check the means and the standard deviations of the parameters of the two models. Which model is more accurate and why? Suppose that this comparison is typical in practical situations, what advise can you give to identification users?

Table 5.9.1. Parameter mean values and standard deviations (in parentheses) of recursive OE models and that of off-line OE models.

True parameters	Recursive OE	Off-line OE
$a_1 = -1.5$	-1.3980 (0.3818)	-1.4956 (0.0189)
$a_2 = 0.7$	0.6405 (0.1990)	0.6953 (0.0165)
$b_1 = 1.0$	1.0863 (0.2895)	1.0305 (0.1231)
$b_2 = 0.5$	0.7918 (1.2823)	0.4892 (0.1320)

Answer. The .m file is as follows.

```
function [THoe,THroe] = roe_oe(Ao,Bo,N,n,sn,n_sim,Delay)
%%% roe_oe.m; M file for comparing ROE and OE models %%%
%
% [THoe,THroe] = roe_oe(Ao,Bo,N,n,sn,n_sim,Delay)
%
% This function compares the accuracies of recursive OE model and
% of off-line OE model
%
% Input arguments:
% [Ao, Bo]: Process polynomials
% N: Number of samples to be simulated
% n: Order of identified model
% sn: noise-to-signal ratio
% n_sim: number of simulations
%
% Output arguments:
% THoe: Parameter vector of OE model
% THroe: Parameter vector of recursive OE model
%
THoe = [];
THroe = [];
ErrOE = [];
ErrROE = [];
```

```

if exist('Delay') == 0
    Delay = 0;
end
for k = 1:n_sim
    U = gbneng(N,8,1);
    Yo = filter(Bo,Ao,U);
    V = filter(1,[1 -.9],randn(N,1));
    V = V/std(V)*sqrt(sn)*std(Yo);
    Y = Yo + V;
    thOE = oe([Y U],[n n Delay]);
    [a,boe,c,d,foe]=th2poly(thOE);
    yoe = filter(boe,foe,U);
    ErrOE = [ErrOE;cov(Yo-yoe)];
    THoe = [THoe;foe boe];
    thROE = roe([Y U],[n n Delay],'ff',1);
    broe = [zeros(1,Delay) thROE(N,1:n)];
    froe = [1 thROE(N,n+1:2*n)];
    yroe = filter(broe,froe,U);
    ErrROE = [ErrROE;cov(Yo-yroe)];
    THroe = [THroe; froe broe];
end

```

The mean values and standard deviations of 20 simulations are given in Table 5.9.1. From the comparison one sees that the accuracy of the off-line output error model is much higher than that of the recursive output error method. This is because the recursive method is derived from the off-line method with certain approximations which cause errors. The practical implication of this exercise is that one should use the off-line version of an estimation method whenever possible; the recursive version is only recommended when computation time and/or computer memory are limited. Note that, when necessary, the forgetting factor λ can be easily implemented for off-line methods.

For more extensive treatment of recursive identification, see the book of Ljung and Söderström (1984).

5.10 Conclusions and Discussion

In this chapter, different ways of modifying the least-squares method have been studied in order to overcome the bias problem and to obtain a model with optimal accuracy. Looking at the model misfit in the frequency domain, we have rederived the Steiglitz-McBride method. The output error method has been motivated by choosing the identification criterion in a natural way. Using the correlation technique, we have obtained the instrumental variable (IV) methods. The family of prediction error methods have been motivated in order to obtain white noise

residuals and to model the disturbance. More order selection methods are introduced and the FOE criterion is recommended. Traditional model validation methods are briefly discussed.

Most methods studied will give consistent estimates under weak conditions. The prediction error methods can give consistent estimates also for closed-loop tests. The accuracies of the prediction models are generally higher than those of the output error model and of the IV models. The price of these advantages of the prediction error methods is more extensive computation. When the model order is incorrect (undermodeling), transfer function error distributions in the frequency domain are given.

In our treatment, we have focused on the essence of the ideas that motivated the different methods. This process reflects, to some degree, the real development of identification techniques and theory in the 70's and 80's. Hence the study of this chapter can be helpful in understanding the vast literature of identification. For theoretically more rigorous treatments of these identification methods, we recommend the books of Ljung (1987) and Söderström and Stoica (1989).

For the choice among the different methods, if the user is only interested in process transfer function and if the process is operating in open-loop, we would suggest to use the output error method, or even Steiglitz-McBride method if output disturbance is nearly white noise. If very high accuracy is required, or also a disturbance model is needed, or if the process is operating in a closed-loop, the Box-Jenkins model and the ARMAX model are good choice. If some physical a priori knowledge about the model structure is available, one should of course try to make use of it in choosing the identification method (or equivalently the model structure). Experience has shown that, when bias error is dominant, for example, when low order is imposed, then the output error model is more accurate.

In our treatment, the problem of closed-loop identification is not treated separately as a special issue, because the recommended methods can treat the closed-loop data without problem. The only precondition is to apply persistently exciting test signal(s), which is the same as that in open-loop identification. For recent research on closed-loop identification for control, see van den Hof (1997).

The theory of the prediction error methods is considered to be a mature part in the field of identification. However, when applying the theory to industrial processes, following difficulties exist:

- Numerical difficulties. As mentioned before, in general, there exists no algorithm that guarantees the convergence to the global minimum. In practice, all the nice properties of prediction error methods will be lost if the numerical search fails.
- It is difficult to perform model order/structure selection for MIMO processes. It is extremely tedious to try all the different combinations of model order/structure in order to search for the correct one. This can already be seen when identifying the 2 by 2 glass drawing process. It will certainly be more complicated to identify a 10 by 20 process. Moreover, numerical problems in parameter estimation will hinder this process.

- It is not clear how to perform optimal input design for intended model applications such as control.
- It is difficult to supply a description (quantification) of the model error of this model class that is suitable for robust control system analysis and design. This topic is of considerable interest and there is much on going research.
- Existing methods of model validation are not control relevant.

These problems have motivated the development of other methods of identification.

Chapter 6

Asymptotic Method; SISO Case

After having studied the various identification methods, the reader has gained considerable knowledge on identification. A big question remains to be answered: *How can we solve the four problems of identification for control in a systematic manner?* The task of this chapter and Chapter 7 is to provide an answer to the question.

As outlined in Chapter 1, the four fundamental problems of identification are

1. **Test design.** Design an identification test using excitation signals so that the parameters of the model are identifiable. If possible, the test should be designed so that the identified model is most suitable for use in control. The optimal test design for control has not yet been studied in previous chapters, although the practical design method has been given in Chapter 3.
2. **Parameter estimation.** Determine the parameters of the model so that the model is most accurate according to a control relevant criterion. This is the most frequently studied topic in literature.
3. **Model order/structure selection.** Determine the order/structure so that the model obtained is most accurate for control. Chapter 5 contains several methods.
4. **Model validation.** Verify if the identified model is suitable for its application, namely, control. *If not, provide a remedy that should include redesigning the identification test.* There is a weak link of identification. Traditional model validation methods such as analysis of residuals and cross-correlations are for checking if the model can best fit the given data. For control applications, it is not sufficient if the identified model is the best process approximation for the given data, it is more important to know whether the data is informative enough for the purpose of control. The link between model validation and test (re)design needs to be emphasized and the control engineer should play an active role in test design.

Because of our limited knowledge about reality, mathematical models can never give an exact description of the process behavior under study. In the identification of industrial processes, undermodeling and disturbances are the main causes of model errors. In the previous decade, robust control theory has been proposed and developed; see Zames (1981), Doyle (1982), Vidyasagar (1985) and Morari and Zafriou (1989). The advantage of robust control is its capability to cope with modeling errors in the analysis and design of control systems. In order to apply robust control theory, one needs not only a nominal process model, but also a suitable description of the modeling errors. These are typically in the form of some bounds on the model parameter variations of the parametric models; or the bounds on the frequency response variations (see Section 6.2). In the last decade, many methods have been proposed for quantifying error bounds of identified models; see, e.g., Goodwin and Salgado (1989), Kosut *et. al.* (1990), van den Boom *et. al.* (1991), Goodwin *et. al.* (1991) and Zhu *et. al.* (1991).

In this chapter, the so-called *asymptotic method* (ASYM) will be introduced which provides systematic solutions to the four identification problems. Using this method, one can estimate not only an accurate parametric model of the process, but also an upper bound of the model errors in the frequency domain. The basic steps of the method consist of a high order model estimation and subsequent model reduction. In this chapter we will present the method for SISO processes; the generalization to MIMO case is quite straightforward and will be treated in the next chapter. First, in Section 6.1, the asymptotic theory of Ljung (1985) will be introduced which explains the frequency domain properties of prediction error methods when the model order increases. All steps of ASYM are based on this theory. In Section 6.2, control relevant optimal test design is studied and both open-loop and closed-loop design formulas are derived using the asymptotic theory. In Section 6.3 identification procedure is outlined where parameter estimation, order selection and error bound estimation are treated. A simulation example is presented in Section 6.4 and Section 6.5 summarizes the chapter.

6.1 The Asymptotic Theory

First review the problem of linear process identification. Consider a SISO linear discrete-time process

$$y(t) = G^o(q)u(t) + H^o(q)e(t) \quad (6.1.1)$$

where

$$G^o(q) = \sum_{k=1}^{\infty} g_k^o \cdot q^{-k} \text{ and } H^o(q) = \sum_{k=0}^{\infty} h_k^o \cdot q^{-k}$$

are the transfer operators of the process and the disturbance respectively. The frequency response of the process is defined as

$$G^o(e^{i\omega}) = \sum_{k=1}^{\infty} g_k^o \cdot e^{-i\omega k} \quad -\pi \leq \omega \leq \pi \quad (6.1.2)$$

In order to be more generic, assume that the identification test is performed in closed-loop; see Figure 6.1.1, where $K(q)$ denotes the feedback controller. We assume that the external test signal $r(t)$ is quasi-stationary (Ljung, 1987) meaning that its spectrum exists. Note that this test signal is *not* the process input $u(t)$.

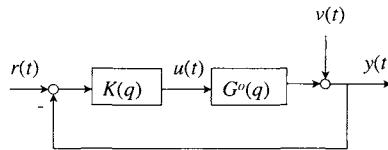


Figure 6.1.1 Identification in closed-loop

The identification problem is to estimate an approximate model from observed input-output data. Denote the data sequence by Z^N :

$$Z^N := y(1), u(1), \dots, y(N), u(N) \quad (6.1.3)$$

If we have a parametrized model (think of the ARX model, ARMAX model, Box-Jenkins model...):

$$y(t) = G(q, \theta)u(t) + H(q, \theta)\varepsilon(t) \quad (6.1.4)$$

where θ is the parameter vector, and $\varepsilon(t)$ is white noise. Then determine the parameters in θ by minimizing the loss function

$$\frac{1}{N} \sum_{t=1}^N \varepsilon(t, \theta)^2 \quad (6.1.5)$$

where $\varepsilon(t, \theta)$ is the prediction error

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t | \theta) = H^{-1}(q, \theta)[y(t) - G(q, \theta)u(t)] \quad (6.1.6)$$

As it has been shown in Chapter 5, this problem statement can cover most of the time domain identification techniques in practice. Particular methods can be obtained by taking a specific model structure.

After the parameter estimation, the transfer function estimates of the process and the disturbance are denoted as:

$$\hat{G}_N^n(e^{i\omega}) = G(\hat{\theta}, e^{i\omega}), \quad \hat{H}_N^n(e^{i\omega}) = H(\hat{\theta}, e^{i\omega}) \quad (6.1.7)$$

Here superscript n denotes the model order and the subscript N is to emphasize that N data samples are used for the estimation.

When the identified model is used in simulation and controller design, we are more concerned with the quality of the frequency response estimates than with the accuracy of the parameters. Since the process is viewed as a black-box, the internal parametrization via θ is merely a vehicle to arrive at this estimate. Then, it is natural to let the model order n depend on the number of observed data samples, $n = n(N)$. Typically, in order to have a model set that is large enough to contain the true transfer function of an industrial process, or to give a good approximation of the true dynamics, we will allow the order n to increase when the number of data samples N increases, but n should be small compared to N . This can be formally expressed by:

$$n(N) \rightarrow \infty \text{ as } N \rightarrow \infty \quad \text{and} \quad n^2(N)/N \rightarrow 0 \text{ as } N \rightarrow \infty \quad (6.1.8)$$

For the identifiability of increasingly higher order models, the process input need to be persistently exciting with sufficiently high order. This can be expressed formally by assuming that the process input $u(t)$ is persistently exciting with any finite order, that is

$$\Phi_u(\omega) > 0 \quad \text{for } -\pi < \omega < \pi. \quad (6.1.9)$$

Result 6.1.1 (Ljung, 1985): Given (6.1.1)-(6.1.9). Suppose that the global minima are obtained in the minimization of the loss function (6.1.5) for all n and N . Then

$$\begin{bmatrix} \hat{G}_N^n(e^{i\omega}) \\ \hat{H}_N^n(e^{i\omega}) \end{bmatrix} \rightarrow \begin{bmatrix} G^o(e^{i\omega}) \\ H^o(e^{i\omega}) \end{bmatrix} \text{ w.p.1 as } N \rightarrow \infty \quad (6.1.10)$$

$$\sqrt{N} \begin{bmatrix} \hat{G}_N^n(e^{i\omega}) - E\hat{G}_N^n(e^{i\omega}) \\ \hat{H}_N^n(e^{i\omega}) - E\hat{H}_N^n(e^{i\omega}) \end{bmatrix} \rightarrow \mathcal{N}(0, P_n(\omega)) \text{ as } N \rightarrow \infty \quad (6.1.11)$$

$$\lim_{n \rightarrow \infty} \frac{1}{n} P_n(\omega) = \Phi_v(\omega) \Phi^{-1}(\omega) \quad (6.1.12)$$

where

$$\Phi(\omega) = \begin{bmatrix} \Phi_u(\omega) & \Phi_{vu}(\omega) \\ \Phi_{uv}(\omega) & R \end{bmatrix}, \quad \Phi_v(\omega) = |H^o(e^{i\omega})|^2 R$$

and R is the variance of the white noise $e(t)$.

Comments on the result

This result show that the model estimates are consistent, and the errors of the transfer functions at each frequency follow a Gaussian distribution. The result is independent of the model structure; the reason for this is that the order is allowed to increase as the number of data points

(test time) increases. It is important to note that the result holds for closed-loop tests with open-loop tests as a special case.

From (6.1.12) one obtains an expression for the asymptotic variance of the process model

$$\text{var}[\hat{G}_N^n(e^{i\omega})] \approx \frac{n}{N} \frac{\Phi_v(\omega)R}{\Phi_u(\omega)R - |\Phi_{ue}(\omega)|^2} \quad (6.1.13)$$

For a clear physical insight, let us do a little bit more calculation. Denote

$$S^o(q) = \frac{1}{1 + G^o(q)K(q)} \quad (6.1.14)$$

as the *sensitivity* function (the transfer function from the disturbance $v(t)$ to the output $y(t)$). From the feedback relation

$$u(t) = K(q)[r(t) - y(t)]$$

we have

$$u(t) = K(q)S^o(q)r(t) - K(q)S^o(q)H^o(q)e(t) \quad (6.1.15)$$

Then from (6.1.15) one obtains

$$\begin{aligned} \Phi_u(\omega) &= |K(e^{i\omega})|^2 |S^o(e^{i\omega})|^2 \Phi_r(\omega) + |K(e^{i\omega})|^2 |S^o(e^{i\omega})|^2 |H^o(e^{i\omega})|^2 R \\ \Phi_{ue}(\omega) &= -K(e^{i\omega})S(e^{i\omega})H^o(e^{i\omega})R \end{aligned}$$

Inserting these two equations into (6.1.13) yields

$$\text{var}[\hat{G}_N^n(e^{i\omega})] \approx \frac{n}{N} \frac{1}{|K(e^{i\omega})|^2 |S^o(e^{i\omega})|^2} \frac{\Phi_v(\omega)}{\Phi_r(\omega)} \quad (6.1.16)$$

This expression is surprisingly simple. It says that the variance of the model at a given frequency is proportional to the (output) noise-to-(external test) signal ratio multiplied by the ratio of model order to the number of data samples, and is inversely proportional to the square of the sensitivity function (the effect of feedback).

Remark. Formula (6.1.16) carries an important message about closed-loop test. It is the test signal outside the loop that counts for model accuracy. For a linear process with linear controller, if there is no excitation from outside the loop, the variance of the model will tend to infinity meaning that the process is not identifiable.

For an open-loop test, one has

$$\Phi_{ue}(\omega) = 0$$

Thus the variance expression (6.1.13) is further simplified to

$$\text{var}[\hat{G}_N^n(e^{j\omega})] \approx \frac{n}{N} \frac{\Phi_v(\omega)}{\Phi_u(\omega)} \quad (6.1.17)$$

Comments on the Establishment of Result 6.1.1

The mathematical proof of Result 6.1.1 is quite involved; see Ljung (1985) and Ljung and Yuan (1985). But it is possible to give physical explanations to readily appreciate the validity of this result.

It is not difficult to see why the estimates are consistent. As the order n increases, the model flexibility will increase. At some moment, the true structure is captured by the model and the consistency is obtained.

To see why the error of the frequency response has a Gaussian distribution with the given variance, let us assume, for the sake of simplicity, that the process is operating in open-loop, the disturbance is white noise with variance R and the input is (known) white noise with variance R_u , and an FIR model is estimated. Then based on the analysis in Section 4.4, we can show that the errors of the n parameter estimates are independent; and they all have the same variance $R/(NR_u)$. The frequency response estimate is the Fourier transform of the n FIR parameters; thus the Fourier transform of the FIR parameter errors are the errors of the transfer function estimate. What the Fourier transform does is simply to create a weighted summation of all the errors, where the modulus of each weight is 1. Therefore, applying the Central Limit Theorem to complex valued random variables, we know that when $n \rightarrow \infty$, the error of the transfer function follows a Gaussian distribution with a variance given by

$$\frac{n}{N} \frac{R}{R_u}$$

This is a special case of (6.1.17).

6.2 Optimal Test Signal Spectrum for Control

In Chapter 3, we have outlined the test design procedure. The theoretical support of this practical test design method will be worked out here. In (theoretical) optimal test design, or optimal input design, one tries to derive a test signal that is optimal in some sense. Early research tried to use the input design in order to improve the accuracy of the parameter estimates (see Mehra, 1974). There are two drawbacks to this approach. First, the optimization procedure for deriving the optimal input is generally very difficult and the optimal formula of the design is a function of the unknown true process model, which makes the design not feasible. Secondly, the intended model application is not addressed in this approach. In identification for control, the test design should be related, as close as possible, to the intended use of the model.

Assume that the identified model will be used for *internal model control* (IMC). It can be shown (Garcia *et. al.*, 1989) that the internal model control scheme is closely related to model predictive control (MPC). In an IMC control scheme, the identified process model is placed in parallel with the process. The difference between the process output and simulated model output is fed back to the controller $Q(q)$. Therefore, a model which can optimally replicate the underlying process behavior for certain given inputs will be very suitable for a internal model control scheme.

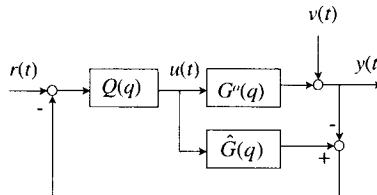


Fig. 6.2.1 Internal model control scheme

The relation between the controller of the classical feedback controller $K(q)$ in Figure 6.1.1 and IMC controller $Q(q)$ is

$$K^o(q) = \frac{Q(q)}{1 - G^o(q)Q(q)} \quad (6.2.1)$$

for the best or "true" controller. When the process model is used, the relation is

$$\hat{K}(q) = \frac{Q(q)}{1 - \hat{G}(q)Q(q)}$$

6.2.1 Open-Loop Test Design for IMC

Denote $u^s(t)$ as the control input to the process (the simulation input to the model) with spectrum $\Phi_u^s(\omega)$. Assume that an open-loop test will be carried out. Let us now derive an optimal test signal $u(t)$ for the high order model and for the purpose of simulation in the IMC scheme. Using the high order model the simulation error is

$$e(t) = [\hat{G}_N^n(q) - G^o(q)]u^s(t) \quad (6.2.3)$$

Note that the disturbance is not included here, so it is not the output error defined before. The error in (6.2.3) has the variance (for the fixed $\hat{G}_N^n(e^{i\omega})$)

$$V_e = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \hat{G}_N^n(e^{i\omega}) - G^o(e^{i\omega}) \right|^2 \Phi_u^s(\omega) d\omega \quad (6.2.4)$$

But now $\hat{G}_N^n(e^{i\omega})$ is a random variable due to disturbance. Hence we will choose the experimental conditions so that the averaged variance

$$EV_e \approx \frac{1}{2\pi} \int_{-\pi}^{\pi} var[\hat{G}_N^n(e^{i\omega})] \Phi_u^s(\omega) d\omega \quad (6.2.5)$$

is small. According to the asymptotic variance expression (6.1.17) it follows that

$$EV_e \approx \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{n}{N} \frac{\Phi_v(\omega)}{\Phi_u(\omega)} \Phi_u^s(\omega) d\omega \quad (6.2.6)$$

The average variance will decrease as the power of input increases. In practice, however, the input power is constrained by several factors: 1) input signal high/low limits, 2) the output variation range and 3) concerns on nonlinearity. This constraint on the input power can be expressed as

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_u(\omega) d\omega \leq C_u \quad (6.2.7)$$

where C_u is a constant.

Then, applying the result of Yuan and Ljung (1985), one can show that the optimal input spectrum which minimizes the mean square of the simulation error for the high order model under constraint (6.2.7) is given by

$$\Phi_u^{op}(\omega) = \mu \sqrt{\Phi_u^s(\omega) \Phi_v(\omega)} \quad (6.2.8)$$

Here μ is a constant which is adjusted such that the equality in (6.2.7) holds, meaning that the maximum allowed input power is used.

Result (6.2.8) says that more signal power should be put at those frequencies where the model will be used more intensively (simulation input has more power) and where more disturbance appears. Note that this spectrum is not related to the process transfer function. This simple result gives clear guidelines for applications because criterion (6.2.5) is closely related to model use in control and because the expression (6.2.8) has very simple form. Take the FIR model as an example. It can be shown (see, e.g., Söderström and Stoica, 1989) that white noise is an optimal test signal if the parameter covariance is minimized. But according to formula (2.2.8), white noise can almost never be optimal if the model is used in IMC control. Although the result is derived for IMC control, the guideline can also be used for other control schemes.

In practice one does not know signal $u^*(t)$ before designing the controller and testing the closed loop control system. This means that some iteration between the identification test and the controller test is necessary for the optimal input design and finally optimal control. The output disturbance $v(t)$ can be measured from the uncontrolled process when keeping the input constant (free-run experiment) or can be estimated as the output error residuals from a previous identification. In practice, if the properties of $u^*(t)$ and $v(t)$ can be obtained for a given process, they can be used for the same class of processes.

6.2.2 Closed-Loop Test Design for IMC

Now, one wishes to study optimal test design in a more realistic situation. The identified model is to be used in an IMC scheme where the reference signal is $r_c(t)$ with spectrum $\Phi_{r_c}(\omega)$. The identification test will be carried out also in an IMC scheme with an existing model $\hat{G}^E(z^{-1})$. The test signal $r(t)$ (not to be confused with the reference signal $r_c(t)$ in IMC control) is applied at the setpoint of the closed-loop system. The problem is to derive the power spectrum $\Phi_r(\omega)$ of test signal $r(t)$ that is optimal in some sense for IMC control for reference signal $r_c(t)$.

It is natural to evaluate the model quality by comparing the output of the controlled system using the true model and that of the system using the identified model. Neglecting the response to the disturbance $v(t)$ we have the system output $y^o(t)$ using the true model as

$$\begin{aligned} y^o(t) &= \frac{G^o(q)K^o(q)}{1 + G^o(q)K^o(q)} r_c(t) \\ &= G^o(q)Q(q)r_c(t) \end{aligned} \quad (6.2.9)$$

The actual output $\hat{y}(t)$ using the identified model is

$$\begin{aligned} \hat{y}(t) &= \frac{G^o(q)\hat{K}(q)}{1 + G^o(q)\hat{K}(q)} r_c(t) \\ &= \frac{G^o(q)Q(q)}{1 + \tilde{G}(q)Q(q)} r_c(t) \end{aligned} \quad (6.2.10)$$

where

$$\tilde{G}(q) = G^o(q) - \hat{G}(q) \quad (6.2.11)$$

Note that we denote $r_c(t)$ as the reference signal during IMC control, and $r(t)$ as the reference signal during identification. A measure of the performance degradation for IMC control can be obtained by analyzing the error signal

$$\begin{aligned} \tilde{y}(t) &= y^o(t) - \hat{y}(t) \\ &= G^o(z^{-1})Q(z^{-1}) \left[1 - \frac{1}{1 + \tilde{G}(z^{-1})Q(z^{-1})} \right] r_c(t) \\ &= \frac{\tilde{G}(z^{-1})}{1 + \tilde{G}(z^{-1})Q(z^{-1})} G^o(z^{-1})Q^2(z^{-1})r_c(t) \end{aligned} \quad (6.2.12)$$

Using first order approximations at zero for $\tilde{G}/(1 + \tilde{G}Q)$ one obtains

$$\tilde{y}(t) \approx \tilde{G}(z^{-1})G^o(z^{-1})Q^2(z^{-1})r_c(t) \quad (6.2.13)$$

Computing the mean square error and using the definition of the power spectrum and (6.1.16)

$$\begin{aligned} E\tilde{y}(t) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{\tilde{y}}(\omega) d\omega \\ &\approx \frac{1}{2\pi} \int_{-\pi}^{\pi} \text{var}\tilde{G}|G^o(e^{i\omega})|^2|Q(e^{i\omega})|^4\Phi_{r_c} d\omega \\ &\approx \frac{n}{N} \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\Phi_v(\omega)}{|K(e^{i\omega})|^2|S^o(e^{i\omega})|^2\Phi_r(\omega)} |G^o(e^{i\omega})|^2|Q(e^{i\omega})|^4\Phi_{r_c}(\omega) d\omega \end{aligned} \quad (6.2.14)$$

Result 6.2.1. Given the process as in (6.1.1), denote $Q(z^{-1})$ as the IMC controller and $\hat{G}^E(z^{-1})$ as an existing process model. An identification test will be carried under IMC control using $Q(z^{-1})$ and $\hat{G}^E(z^{-1})$; see Figure 6.2.1. Then the optimal spectrum for test signal $r(t)$ that minimizes

$$E\tilde{y}(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{\tilde{y}}(\omega) d\omega$$

under output power constraint

$$\int_{-\pi}^{\pi} \Phi_y(\omega) d\omega \leq C_y$$

is

$$\Phi_r^{opt}(\omega) = \mu \left| 1 + \tilde{G}^E(e^{i\omega})Q(e^{i\omega}) \right|^2 \sqrt{\Phi_v(\omega)\Phi_{r_c}(\omega)} \quad (6.2.15)$$

where μ is a constant which is adjusted such that

$$\int_{-\pi}^{\pi} \Phi_y(\omega) d\omega = C_y \quad (6.2.16)$$

and $\hat{G}^E(q)$ is the error of the existing process model (not to be confused with the error of the model to be identified)

$$\tilde{G}^E(q) = G^o(q) - \hat{G}^E(q) \quad (6.2.17)$$

Proof. For the sake of simplicity, the frequency variable ω will be omitted in the proof. The output spectrum can be written as

$$\Phi_y(\omega) = |G^o|^2 |\hat{K}|^2 |S^o|^2 \Phi_r + |S^o|^2 \Phi_v \quad (6.2.18)$$

Define the constant γ as

$$\gamma = C_y - \int_{-\pi}^{\pi} |S^o|^2 \Phi_r d\omega \quad (6.2.19)$$

Then, from (6.2.14) and (6.2.18), the optimization problem becomes

$$\begin{aligned} \min_{\Phi_r(\omega)} & \left\{ \int_{-\pi}^{\pi} \frac{|H^o|^2 \lambda}{|\hat{K}|^2 |S^o|^2 \Phi_r} |G^o|^2 |Q|^4 \Phi_{r_c} d\omega : \right. \\ & \left. \int_{-\pi}^{\pi} |G^o|^2 |\hat{K}|^2 |S^o|^2 \Phi_r d\omega \leq \gamma \right\} \end{aligned} \quad (6.2.20)$$

According to Yuan and Ljung (1985) and Ljung (1987), this problem has the solution

$$\begin{aligned} \Phi_r^{opt} &= \mu \frac{|G^o| |Q|^2}{|\hat{K}| |S^o|} \sqrt{\Phi_v \Phi_{r_c}} \frac{1}{|G^o| |\hat{K}| |S^o|} \\ &= \mu \frac{|Q|^2}{|\hat{K}|^2 |S^o|^2} \sqrt{\Phi_v \Phi_{r_c}} \end{aligned} \quad (6.2.21)$$

where μ is a constant which is adjusted such that

$$\int_{-\pi}^{\pi} |G^o|^2 |\hat{K}|^2 |S^o|^2 \Phi_r d\omega = \gamma \quad (6.2.22)$$

Using the expressions of $S^o(q)$ (6.1.14) and $\hat{K}(q)$ (6.2.2), (6.2.21) becomes

$$\Phi_r^{opt} = \mu \left| 1 + [G^o - \hat{G}^E] Q \right|^2 \sqrt{\Phi_v \Phi_{r_c}} \quad (6.2.23)$$

This proves the result; see also Zhu and van den Bosch (2000).

This result is simple and instructive. It says that the optimal closed-loop test signal at the setpoint is determined by the reference signal spectrum, by the disturbance spectrum, and by the error of the current model. Note that the IMC controller Q is basically the inverse of the process G^o , the term $[G^o - \hat{G}^E]Q$ is basically the relative error of the current model. This implies that the optimal signal spectrum is not sensitive to the error of the current model. Assuming that the current model error is small, then the result can be simplified to

$$\Phi_r^{opt}(\omega) \approx \mu \sqrt{\Phi_v(\omega)\Phi_{r_c}(\omega)} \quad (6.2.24)$$

which is as simple as the result for the open-loop test (Ljung, 1985).

Using the same technique, the optimal signal spectrum, when the power of reference $r(t)$ signal is constrained, can be derived:

Result 6.2.2. Given the process and the IMC controller as in Theorem 1, then the solution to the optimization problem under reference signal power constraint

$$\min_{\Phi_r(\omega)} \left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{\tilde{y}}(\omega) d\omega : \int_{-\pi}^{\pi} \Phi_r(\omega) d\omega \leq C_r \right\} \quad (6.2.25)$$

is

$$\Phi_r^{opt}(\omega) = \mu \left| 1 + \hat{G}^E(e^{i\omega})Q(e^{i\omega}) \right| \left| G^o(e^{i\omega})Q(e^{i\omega}) \right| \sqrt{\Phi_v(\omega)\Phi_{r_c}(\omega)} \quad (6.2.26)$$

where μ is a constant which is adjusted such that

$$\int_{-\pi}^{\pi} \Phi_r(\omega) d\omega = C_r \quad (6.2.27)$$

This result is also instructive and applicable, because, according to (6.2.9), $|G^o(e^{i\omega})Q(e^{i\omega})|$ is the designed transfer function from the output to the setpoint which is known to the user. Assuming the error of the current model is small, we obtain

$$\Phi_r^{opt}(\omega) \approx \mu |G^o(e^{i\omega})Q(e^{i\omega})| \sqrt{\Phi_v(\omega)\Phi_{r_c}(\omega)} \quad (6.2.28)$$

This result tells that the test signal power should be concentrated in the frequency range where the designed system transfer function has large gain, and the reference signal power and the disturbance power are large.

The spectrum $\Phi_{r_c}(\omega)$ and $\Phi_v(\omega)$ are the important ingredient in the optimal spectra. $\Phi_{r_c}(\omega)$ is related to the desired closed-loop response and is determined by the user. As mentioned before, $\Phi_v(\omega)$ can be estimated using the output signal from the open-loop measurement or the output error residual from previous identification.

The optimal spectrum formulas derived from this section form the theoretical basis for the test design approach outlined in Chapter 3. Recently Forssell and Ljung (1998) have derived some

results on closed-loop test design in several different situations. In their scheme, the test signal is applied at the input, both the test signal and the controller for the test are optimized and the weighted sum of input power and output power is constrained. This leads to more complex results and the schemes are less feasible.

6.3 Parameter Estimation and Order Selection

Two things are desired in parameter estimation: 1) model accuracy is high, for example, with minimum variance property and 2) numerical solution is tractable and reliable. As one has learned from Chapter 5, a method that is theoretically more accurate, such as the Box-Jenkins method, is usually numerically more difficult. To develop an estimation method that is both accurate and numerically simple is a challenging task, especially for MIMO processes.

It will be shown in this section that ,by applying the asymptotic theory, one can obtain a method that is highly accurate and, at the same time, numerically simpler and more reliable than other alternatives. The approach is to start with a high order model estimation and then to perform a model reduction. One can avoid numerical problems by combining high order estimation and model reduction.

It is assumed that the input-output data are obtained from a well designed identification test and identifiability conditions are satisfied.

6.3.1 Parameter Estimation

Step 1. High order ARX model estimation

The asymptotic properties of the transfer function estimates are independent of model structures. So, it is sensible to use a model structure that is as simple as possible for the high order model. The equation error model is the most simple parametrization which can supply both the process model and the disturbance model. Estimate a high order equation error (or ARX) model using linear least-squares

$$A(q)y(t) = B(q)u(t) + e(t) \quad (6.3.1)$$

As discussed in Chapter 4, there is analytical solution to this problem and it is numerically simple and reliable. Because global solutions can be obtained for all n and N , then the asymptotic result applies. Denote the process model, disturbance model and the disturbance spectrum as

$$\hat{G}_N^n(q) = \frac{\hat{B}^n(q)}{\hat{A}^n(q)}, \quad \hat{H}_N^n(q) = \frac{1}{\hat{A}^n(q)}, \quad \hat{\Phi}_v(\omega) = \frac{\hat{R}}{\left| \hat{A}^n(e^{i\omega}) \right|^2} \quad (6.3.2)$$

where \hat{R} is the estimated variance of the equation error residual.

Step 2. Model reduction

The ARX model in (6.3.1) is often over-parametrized for process model $G(q)$ meaning that it can be well approximated by a model with a much lower order. Since the variance is proportional to the order n , model reduction can reduce the variance if it is performed properly. Also it is often more convenient to use a compact (reduced) model in controller design.

The asymptotic theory of Section 6.1 shows that, in the frequency domain, the high order model follows approximately a Gaussian distribution with the variance given by (6.1.16). If we view the frequency response of the high order estimates

$$\hat{G}^n(e^{i\omega_1}), \hat{G}^n(e^{i\omega_2}), \dots, \hat{G}^n(e^{i\omega_n}), \text{ where } \omega_k = \frac{k\pi}{n}, k = 1, \dots, n$$

as the noisy observations of the true transfer function, we can then apply the maximum likelihood principle. It can be shown that when $N \rightarrow \infty$, the asymptotic negative log-likelihood function for the process model is given by (Wahlberg, 1989):

$$V = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \hat{G}^n(e^{i\omega}) - \hat{G}^l(e^{i\omega}) \right|^2 \frac{\Phi_u(\omega)R - |\Phi_{ue}(\omega)|^2}{\Phi_v(\omega)R} d\omega \quad (6.3.3)$$

Instead of reproducing the rather lengthy derivation, one can justify this result by observing the fact that the errors are weighted by the inverse of their variances (neglecting the factor N/n); this will lead to an asymptotically efficient (minimum variance) estimate of the frequency response. Because the high order model (observation) follows a Gaussian distribution, one can say that the reduced model is an asymptotic maximum likelihood estimate.

Solving this problem needs a nonlinear minimization algorithm and there are many methods to choose from. In the following we will outline a model reduction algorithm for open-loop data. For open-loop data the log-likelihood function becomes

$$V = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \hat{G}^n(e^{i\omega}) - \hat{G}^l(e^{i\omega}) \right|^2 \frac{\Phi_u(\omega)}{\Phi_v(\omega)} d\omega \quad (6.3.4)$$

We will approach this problem by performing an output error identification.

- 1) **Simulation.** Collect the input signal $\{u(1), u(2), \dots, u(N)\}$, which has been used in the identification test. Filter $u(t)$ by the inverse of the disturbance model

$$u_f = \frac{1}{\hat{H}^n(q)} u(t) = \hat{A}^n(q) u(t)$$

where $u_f(t)$ is used to realize the desired frequency weighting. Then simulate the high order model using the filtered input:

$$\hat{y}^n(t) = \frac{\hat{B}^n(q)}{\hat{A}^n(q)} u_f(t) \quad (6.3.5)$$

This is equivalent to

$$\hat{y}^n(t) = \hat{B}^n(q)u(t) \quad (6.3.6)$$

Thus we obtain the input-output data of the high order model $\hat{G}_N^n(q)$:

$$Z_n^N := [\hat{y}^n(1), u_f(1), \dots, \hat{y}^n(N), u_f(N)]$$

- 2) Parameter estimation.** The parameters of the reduced model are calculated by using an *output error method* which minimizes the following loss function

$$V_{oc}^N = \frac{1}{N} \sum_{t=1}^N \left\{ [\hat{G}^n(q) - \hat{G}^l(q)]u_f(t) \right\}^2$$

Letting $N \rightarrow \infty$ and applying Parseval's identity yield

$$V_{oc}^\infty = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \hat{G}_N^n(e^{i\omega}) - \hat{G}^l(e^{i\omega}) \right|^2 \frac{\Phi_u(\omega)}{\left| \hat{H}(e^{i\omega}) \right|^2} d\omega \quad (6.3.7)$$

This is equivalent to the asymptotic likelihood function (6.3.4), except that the disturbance spectrum and the sensitivity function are replaced by their high order estimates.

In a similar way, a reduced disturbance model can be obtained from the high order estimate $1/\hat{A}^l(q)$. Then we have the reduced process model and the reduced disturbance model as

$$\hat{G}^l(q) = \frac{\hat{B}^l(q)}{\hat{A}^l(q)}, \quad \hat{H}^l(q) = \frac{\hat{C}^l(q)}{\hat{D}^l(q)} \quad (6.3.8)$$

Hence the final model is a Box-Jenkins model.

Comments on ASYM estimation method

The method will be called the asymptotic or ASYM approach. There are two reasons for starting with a high order model estimation. The first is to obtain a consistent estimation for the simple ARX model; the second is to make use of the asymptotic theory which assumes that the order tends to infinity. So, even when we know that the process has a low order, e.g., order 2, we will still use, in this step, a higher order which is necessary for modeling the unmeasured disturbance.

Although one eventually also needs a nonlinear minimization algorithm to find the reduced model, one is now in a much better position for achieving this. First, the influence of the disturbance is reduced greatly when using the data from the simulation of the high order model. Secondly, we have many different ways of obtaining initial values: LS estimate, Steiglitz-McBride estimate, or initial values obtained by some well-known model reduction methods, e.g., (frequency weighted) balanced model reduction or (frequency weighted) Hankel-norm approximation. Thirdly, one can detect poor local minima, because if the minimization algorithm converges to the global minimum, the frequency response of the reduced model should lie in the middle of the fluctuating frequency response of the high order model due to the smoothing effect of model reduction. If this is not the case, a local minimum may occur; and one should use another initial estimate or try another order. All three factors make the method numerically highly reliable.

So far, there is no theoretical result for comparing the accuracy of using a high order ARX model and model reduction with direct estimation of the low order model. However, for FIR model, it can be shown that the model variance using two step estimation is less than or equal to that of direct estimation; see Tjärnström and Ljung (1999).

6.3.2 Model Order Selection

The final output error criterion (FOE) proposed in Chapter 5 can also be used for order selection for the reduced model. However, we will propose another method of order selection based on frequency domain measures.

The asymptotic ML model reduction will generally increase the model quality. If the reduced model is allowed to deviate from the high order model by the same amount as the error of the high order model (measured by its variance), one may expect that the reduced model is closest to the true frequency response. Based on this observation, a method of order selection is simply to choose the order such that the difference between the high order model and the reduced model (in the frequency domain) approximately equals the variance of the high order model. Therefore, the order of the reduced model is determined such that

$$\left| \hat{G}^n(q) - \hat{G}^l(q) \right|^2 \approx \frac{n}{N} \frac{\Phi_u(\omega)\hat{R} - |\hat{\Phi}_{ue}(\omega)|^2}{\hat{\Phi}_v(\omega)\hat{R}} \quad (6.3.9)$$

The selection can be done by visual inspection. Equivalently, the following criterion can be minimized

$$ASYC = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \left| \hat{G}^n(q) - \hat{G}^l(q) \right|^2 - \frac{n}{N} \frac{\Phi_u(\omega)\hat{R} - |\hat{\Phi}_{ue}(\omega)|^2}{\hat{\Phi}_v(\omega)\hat{R}} \right| d\omega \quad (6.3.10)$$

The same idea can be applied for determining the order of the disturbance model. The criterion will be called ASYC (asymptotic criterion) for obvious reasons.

In this selection rule, the selected order is usefully related to the noise-to-signal ratio, and to the test time. For a given process, if the noise level is high and the test time is short, the selected order of the reduced model will be low. For the same process, the selected order will increase if the power of the test signal and/or the test time increases. There is a difference in concept between ASYC and many existing methods of order selection, e.g., the whiteness test of residuals. Using the ASYC, one does not intend to find the true order of the process. Instead, we search for an order so that the best frequency response estimate can be obtained.

6.4 Model Validation using Upper Error Bound

According to the asymptotic theory, the errors of the high order model follows asymptotically a Gaussian (normal) distribution with the variance given by (6.1.16). Therefore, a 3σ upper bound of the errors of the high order model can be defined as follows:

$$\left| G^o(e^{i\omega}) - \hat{G}^n(e^{i\omega}) \right| \leq 3 \sqrt{\frac{n}{N} \frac{\Phi_u(\omega)R - |\Phi_{ue}(\omega)|^2}{\Phi_v(\omega)R}} \text{ w.p. } 99.99\% \quad (6.4.1)$$

This upper bound can also be used for the reduced model $\hat{G}'(q)$, because the model reduction will generally reduce the model error. Thus we have

$$\left| G^o(e^{i\omega}) - \hat{G}'(e^{i\omega}) \right| \leq \bar{\Delta}(\omega) := 3 \sqrt{\frac{n}{N} \frac{\Phi_u(\omega)R - |\Phi_{uc}(\omega)|^2}{\Phi_v(\omega)R}} \text{ w.p. } \geq 99.99\% \quad (6.4.2)$$

The auto and cross-spectra used in (6.4.2) can be estimated using input signal $u(t)$, output error residual $\hat{v}(t)$ and prediction error residual $\hat{c}(t)$.

In literature, an upper error bound with a probability is called a *soft bound*. The upper bound is a simple function of the design variables such as the spectrum of test signal and the test time. This makes the interaction between identification and controller design possible. If the modeling errors at some frequencies are too large for control application, we know from (6.4.2) that we can reduce the errors by:

1. modifying the spectrum of the test signal $\Phi_r(\omega)$ in closed-loop or $\Phi_u(\omega)$ in open-loop; and
2. using more data (increase of the test duration).

Note that the asymptotic theory can also be used to estimate an upper bound for a given model $\hat{G}(q)$. To do this, perform an identification test and estimate a high order model $G_N^n(q)$ whose errors are bounded by (6.4.1). Therefore, via the high order model and its bound we obtain an

upper bound for the given model

$$\begin{aligned} |G^o(e^{i\omega}) - \hat{G}(e^{i\omega})| &\leq |G^o(e^{i\omega}) - \hat{G}^n(e^{i\omega})| + |\hat{G}^n(e^{i\omega}) - \hat{G}(e^{i\omega})| \\ &\leq 3\sqrt{\frac{n}{N} \frac{\Phi_u(\omega)R - |\Phi_{ue}(\omega)|^2}{\Phi_v(\omega)R}} + |\hat{G}^n(e^{i\omega}) - \hat{G}(e^{i\omega})| \text{ w.p. } \geq 99.99\% \quad (6.4.3) \end{aligned}$$

This is more conservative than (6.4.2) because the data has not been used for deriving the model.

6.5 Simulation Studies and Conclusion

Exercise 6.5.1. Check error bound. Given the process

$$y(t) = \frac{1 + .5q^{-1} + .4q^{-2} + 2q^{-3}}{1 + .5q^{-1} - .4q^{-2} - .262q^{-3} - .03q^{-4}} u(t) + v(t)$$

with

$$v(t) = \frac{\alpha}{1 - 0.95q^{-1}} e(t)$$

The input signal is a GBN with average switch time of 10 samples. The test time is 500 samples. The disturbance power is about 10% of that of the noise-free output. Run an open-loop simulation. Identify the second order process using ASYM. Plot the true model frequency response, errors of the high order model and the reduced model, and upper error bound using (6.4.2).

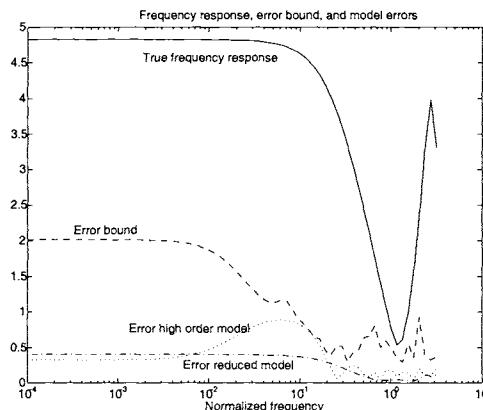


Figure 6.5.1 Check the error bound and errors

Answer. The file for the open-loop ASYM method is `sisoasym.dll` (see `sisoasym.m` for help). The plots are shown in Figure 6.5.1. We find that the error of reduced order is smaller than that of the high order model in most frequencies.

Exercise 6.5.2. Check model accuracy. Given the process

$$y(t) = \frac{1 + .5q^{-1} + .4q^{-2} + 2q^{-3}}{1 + .5q^{-1} - .4q^{-2} - .262q^{-3} - .03q^{-4}} u(t) + v(t)$$

with

$$v(t) = \frac{\alpha}{1 - 0.95q^{-1}} e(t)$$

The input signal is a GBN with an average switch time of 8 samples. The test time is 500 samples. The disturbance power is 10% of that of the noise-free output. Run 50 open-loop simulations. Use each data set to estimate a 4th order output error model, Box-Jenkins model and ASYM model. Compare the following noise-free output-error criteria of the three models

$$V_{OE} = \frac{1}{500} \sum_{t=1}^{500} [G^o(z^{-1})u(t) - \hat{G}(z^{-1})u(t)]^2$$

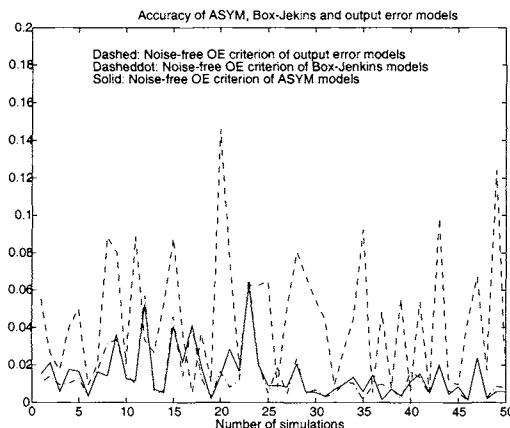


Figure 6.5.2 Compare the accuracy of ASYM, Box-Jenkins and output error models, unbiased case

Answer. The m file for the exercise is given below. The noise-free output error criteria of the models are plotted in Figure 6.5.2. Indeed the accuracy of the ASYM model and that of the Box-Jenkins model are very close and it is higher than the output error model.

```

function [THoe,ErrOE,THbj,ErrBJ,THasym,ErrASYM] = bj_oe_asym(Ao,Bo,N,n,sn,n_sim,Delay
%%%%% bj_oe_asym.m; M file for comparing the OE criteria of BJ, OE and ASYM models
%
% [THoe,ErrOE,THbj,ErrBJ,THasym,ErrASYM] = bj_oe_asym(Ao,Bo,N,n,sn,n_sim,Delay)
%
% This function compares the OE criterion of BJ model and of OE model
%
% Input arguments:
% [Ao, Bo]: Process polynomials
% N: Number of samples to be simulated
% n: Order of identified model
% sn: noise-to-signal ratio
% n_sim: number of simulations
% Delay: Delay, optional
%
% Output arguments:
% THoe: Parameter vector of OE model
% ErrOE: OE Error of OE model
% THbj: Parameter vector of bj model
% ErrBJ: OE Error of BJ model
% THasym: Parameter vector of bj model
% ErrASYM: OE Error of BJ model
%
THoe = [];
THbj = [];
THasym = [];
ErrOE = [];
ErrBJ = [];
ErrASYM = [];
if exist('Delay') == 0
    Delay = 0;
end
for k = 1:n_sim
    %
    % Simulation
    U = gbneng(N,8,1);
    Yo = filter(Bo,Ao,U);
    V = filter(1,[1 -.95],randn(N,1));
    V = V/std(V)*sqrt(sn)*std(Yo);
    Y = Yo + V;

    %
    % Estimate output error model
    thOE = oe([Y U],[n n Delay],15);

```

```

[a,boe,c,d,foe] = th2poly(thOE);
yoe = filter(boe,foe,U);
ErrOE = [ErrOE;cov(Yo-yoe)];
THoe = [THoe; foe boe];

% Estimate Box-Jenkins model
thBJ = bj([Y U],[n 1 1 n Delay],15);
[a,bbj,c,d,fbj] = th2poly(thBJ);
ybj = filter(bbj,fbj,U);
ErrBJ = [ErrBJ;cov(Yo-ybj)];
THbj = [THbj; fbj bbj];
end

% Estimate ASYM model
[Aasym,Basym] = asymest(U,Y,[n n],Delay);
yasym = filter(Basym,Aasym,U);
ErrASYM = [ErrASYM;cov(Yo-yasym)];
THasym = [THasym; Aasym Basym];

```

This chapter introduces the so called ASYM method. The strength of the approach is that it provides systematic solutions to the four problems of identification for control. Based on the asymptotic theory, we are able to provide practically feasible and theoretically optimal or suboptimal solutions at each step. This makes the method highly applicable in practical environments. In the next chapter the method will be extended to the MIMO case.

Chapter 7

Asymptotic Method; MIMO Case

In this chapter, the ASYM method developed in Chapter 6 will be extended to the multi-input multi-output (MIMO) case. In Section 7.1 the MIMO version of the asymptotic theory is presented. The identification procedure is presented in Section 7.2. In Section 7.3, the glass tube process will be identified using ASYM and the result will be compared with those of the prediction methods. Section 7.4 contains conclusions.

7.1 MIMO Version of the Asymptotic Theory

Consider a linear time-invariant discrete-time process with m inputs and p outputs

$$y(t) = G^o(q)u(t) + v(t) \quad (7.1.1)$$

where

$$G^o(q) = \sum_{k=1}^{\infty} G_k^o q^{-k}$$

is called the transfer operator of the process, $y(t)$ is the p -dimensional column output vector at time t , $u(t)$ is the m -dimensional column input vector at time t , $\{G_k^o\}$ is the impulse response of the process, which is a sequence of $p \times m$ matrices, $\{v(t)\}$ is a p -dimensional stochastic stationary process with zero mean values.

The frequency response matrix of the process is defined as

$$G^o(e^{i\omega}) = \sum_{k=1}^{\infty} G_k^o e^{-i\omega k} \quad -\pi \leq \omega \geq \pi \quad (7.1.2)$$

The disturbances vector $v(t)$ comprises filtered white noise signals

$$v(t) = H^o(q)e(t) \quad (7.1.3)$$

where $e(t)$ is a p -dimensional white noise vector and $H^o(q)$ is the $p \times p$ disturbance filter matrix which is stable and minimum phase (its inverse is stable).

Given a parameterized prediction error model

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \quad (7.1.4)$$

and denote the data sequence Z^N as

$$Z^N = y(1), u(1), \dots, y(N), u(N)$$

one can then determine the parameters by minimizing the prediction errors loss function

$$V^N(\theta) = \frac{1}{N} \sum_{t=1}^N \varepsilon^T(t, \theta) \varepsilon(t, \theta) \quad (7.1.5)$$

where $\varepsilon(t, \theta)$ is the prediction error vector

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta) = H^{-1}(q, \theta)[y(t) - G(q, \theta)u(t)] \quad (7.1.6)$$

After parameter estimation, the transfer function estimates are denoted as:

$$\hat{G}_N^n(e^{i\omega}) = G(\hat{\theta}, e^{i\omega}), \quad \hat{H}_N^n(e^{i\omega}) = H(\hat{\theta}, e^{i\omega}) \quad (7.1.7)$$

Several prediction error models are given below and they look exactly like their SISO cases except that the polynomials are replaced by polynomial matrices. The MIMO Box-Jenkins model is given as

$$y(t) = A^{-1}(q)B(q)u(t) + D^{-1}(q)C(q)e(t) \quad (7.1.8)$$

where $A(q)$, $B(q)$, $C(q)$ and $D(q)$ are polynomial matrices with dimensions $p \times p$, $p \times m$, $p \times p$ and $p \times p$ respectively. The parameter vector θ is formed by the coefficients of the polynomial matrices. When using MIMO models, one needs to further specify matrices $A(q)$, $B(q)$, $C(q)$ and $D(q)$ for model identifiability, meaning that the model is uniquely determined if a prediction error method is used and if the data are persistently exciting. For example, one can let $A(q)$ and $D(q)$ be triangular or diagonal polynomial matrices, and the degrees of the polynomial entries follow some rules. This leads to different *canonical forms* of MIMO models which are in general rather complex. In this work, we will try to avoid the study of these canonical forms and use one of the simplest forms: the diagonal form (this is also a canonical form). A diagonal form of the MIMO Box-Jenkins model is that $A(q)$, $C(q)$ and $D(q)$ are diagonal polynomial matrices, and their non zero polynomials are all monic (the highest degree term is 1).

The MIMO ARMAX model is given as

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad (7.1.9)$$

The MIMO ARX model is given as

$$A(q)y(t) = B(q)u(t) + e(t) \quad (7.1.10)$$

For the presentation of the asymptotic theory, we do not restrict ourselves to the diagonal form. So the following result holds for more general models. Assume that all the non-zero polynomials of the matrices of the MIMO model have the same degree n . Note that, unlike the SISO case, this *degree* is in general not the same as the McMillan degree, or the order which is defined as the dimension of the minimal state space realization of the model. Generally, the McMillan degree of the model equals $p \cdot n$.

Result 7.1.1 Given (7.1.1) to (7.1.7). Assume that degree n follows

$$n \rightarrow \infty \text{ as } N \rightarrow \infty \text{ and } n^2/N \rightarrow 0 \text{ as } N \rightarrow \infty$$

and that the input is persistently exciting with any finite order, so that

$$\sigma_{\min}[\Phi_u(\omega)] > 0 \text{ for } -\pi < \omega < \pi$$

where $\sigma_{\min}(\cdot)$ denotes the smallest singular value of the given matrix. And also assume that the global minima are obtained for all n and N . Then

$$[\hat{G}_N^n(e^{i\omega}), \hat{H}_N^n(e^{i\omega})] \rightarrow [G^o(e^{i\omega}), H^o(e^{i\omega})] \text{ w.p.1 as } N \rightarrow \infty \quad (7.1.11)$$

$$\text{col}\sqrt{N} \left[\hat{G}_N^n(e^{i\omega}), \hat{H}_N^n(e^{i\omega}) \right] \rightarrow \mathcal{N}(0, P_n(\omega)) \text{ as } N \rightarrow \infty \quad (7.1.12)$$

$$\lim_{n \rightarrow \infty} \frac{1}{n} P_n(\omega) = [\Phi(\omega)]^{-T} \otimes \Phi_v(\omega) \quad (7.1.13)$$

where

$$\Phi(\omega) = \begin{bmatrix} \Phi_u(\omega) & \Phi_{uc}(\omega) \\ \Phi_{cu}(\omega) & R \end{bmatrix}, \quad \Phi_v(\omega) = H^o(e^{i\omega})R[H^o(e^{i\omega})]^*$$

and R is the covariance of the white noise $e(t)$, $\text{col}(\cdot)$ denotes the vector operator on a matrix, $-T$ means inverse and transpose, \otimes denotes the Kronecker product. Let

$$A = \{a_{ij}\}, \quad B = \{b_{ij}\}$$

be $m \times n$ and $p \times r$ matrices, respectively. The Kronecker product of A and B is defined as an $mp \times nr$ matrix, denoted by $A \otimes B$, so that

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

Result 7.1.1 is the MIMO extension of Ljung (1985); the proof is tedious and can be found in Zhu (1989b).

For an open-loop experiment we have $\Phi_{ue}(\omega) = \Phi_{eu}(\omega) = 0$, and hence

$$\text{col}[\hat{G}_N^n(e^{i\omega})] \rightarrow \mathcal{N}(0, \frac{n}{N} [\Phi_u(\omega)]^{-T} \otimes \Phi_v(\omega)) \text{ as } N \rightarrow \infty \quad (7.1.14)$$

$$\text{col}[\hat{H}_N^n(e^{i\omega})] \rightarrow \mathcal{N}(0, \frac{n}{N} R^{-1} \otimes \Phi_v(\omega)) \text{ as } N \rightarrow \infty \quad (7.1.15)$$

7.2 Asymptotic Method

The MIMO ASYM method will be outlined in this section. Because the asymptotic theory holds for the MIMO case, it is conceptually very straightforward to develop the MIMO version of the method. However, model realization is an additional issue that needs to be treated. Again the method is presented in the order of test design, estimation, order selection and model validation.

It is not yet possible to obtain an MIMO counter part of optimal spectra for the identification test as in Chapter 6. This remains an academic challenge. It is suggested to pick up a slow and dominant output of the process and use the SISO results of Section 6.2. This approach is based on the observation from experience that most often it takes longer to test a slow output than to test a fast output and the model qualities of fast outputs are generally better than that of slow outputs.

The steps of parameter estimation, order/structure selection and model validation will be outlined in the following.

7.2.1 Parameter Estimation

Step 1. Estimation of a high order ARX model

Estimate the parameters of an ARX (equation error) model with a high order

$$A(q)y(t) = B(q)u(t) + e(t) \quad (7.2.1)$$

A diagonal form is recommended, although it is not strictly necessary for ASYM. This means that matrix $A(q)$ is diagonal and the MIMO model is decoupled into p MISO models. After the estimation, we have

$$\hat{G}_N^n(q) = \hat{A}_N^{-1}(q)\hat{B}_N(q) \quad \hat{H}_N^n(q) = \hat{A}_N^{-1}(q) \quad (7.2.2)$$

and the spectrum estimates of the disturbances are

$$\hat{\Phi}_v(\omega) = \hat{A}_N^{-1}(e^{i\omega})\hat{R}\hat{A}_N^{-T}(e^{-i\omega}) \quad (7.2.3)$$

where \hat{R} is the estimated covariance of the equation error residuals.

Step 2. Model reduction

The asymptotic maximum likelihood model reduction can be extended easily to diagonal MIMO models. Denote $\hat{G}_{ij}^n(e^{i\omega})$ and $G_{ij}^o(e^{i\omega})$ as the (i, j) entries of $\hat{G}_N^n(e^{i\omega})$ and $G^o(e^{i\omega})$ respectively. According to the asymptotic theory of Section 7.1, we know that the high order estimates of transfer functions are mutually independent, and each transfer function follows a Gaussian distribution with a variance

$$\text{var}[\hat{G}_{ij}^n(e^{i\omega})] \approx \frac{n}{N} [\Phi^{-1}(\omega)]_{jj} \Phi_{v_i}(\omega) \quad (7.2.4)$$

where $[\Phi^{-1}(\omega)]_{jj}$ is the (j, j) entry of $\Phi^{-1}(\omega)$, and $\Phi_{v_i}(\omega)$ is the spectrum of $v_i(t)$.

For simplicity we also use the diagonal form for the reduced model. This implies that we can perform model reduction for each MISO model separately. It is not difficult to see that the negative log-likelihood function for the i -th sub-model (related to $y_i(t)$) is

$$V_i = \sum_{j=1}^m \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \hat{G}_{ij}^n(e^{i\omega}) - \hat{G}_{ij}^l(e^{i\omega}) \right|^2 \frac{1}{[\Phi^{-1}(\omega)]_{jj} \Phi_{v_i}(\omega)} d\omega \quad (7.2.5)$$

The minimization of this loss function needs a numerical nonlinear search algorithm.

For an open-loop test, assuming the inputs are uncorrelated, the log-likelihood function becomes (see (7.1.14))

$$V_i = \sum_{j=1}^m \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \hat{G}_{ij}^n(e^{i\omega}) - \hat{G}_{ij}^l(e^{i\omega}) \right|^2 \frac{\Phi_{u_j}(\omega)}{\Phi_{v_i}(\omega)} d\omega \quad (7.2.6)$$

In a similar way, model reduction for the disturbance model $1/\hat{A}_{ii}^n(e^{i\omega})$ can be performed.

7.2.2 Model Order/Structure Selection

For simplicity, we will also use diagonal form for the reduced model. Denote l_i as the order (the degree of the polynomials) of the i -th model, then $[l_1, \dots, l_p]$ determines the structure of the diagonal form MIMO model. The following rule is a natural extension of the order selection rule for SISO processes.

For the i -th MISO model: Perform model reduction with various orders and select the order, l_i , such that in the frequency range which is important for control, the following relation holds:

$$\sum_{j=1}^m \left| \hat{G}_{ij}^n(e^{i\omega}) - \hat{G}_{ij}^l(e^{i\omega}) \right|^2 \approx \frac{n}{N} \sum_{j=1}^m [\hat{\Phi}^{-1}(\omega)]_{ii} \hat{\Phi}_{v_j}(\omega) \quad (7.2.7)$$

Equivalently, the following criterion can be minimized

$$ASYC_i = \sum_{j=1}^m \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \left| \hat{G}_{ij}^n(e^{i\omega}) - \hat{G}_{ij}^l(e^{i\omega}) \right|^2 - \frac{n}{N} [\hat{\Phi}^{-1}(\omega)]_{ii} \hat{\Phi}_{v_j}(\omega) \right| d\omega \quad (7.2.8)$$

This criterion will be called the ASYC (asymptotic criterion).

The order of the disturbance model can be selected in a similar way to that in (7.2.8).

7.2.3 Deriving an Upper Bound Matrix for Model Validation

According to the asymptotic theory and (7.2.4), we can define a 3σ bound for $\{G_{ij}^o(e^{i\omega}) - \hat{G}_{ij}^n(e^{i\omega})\}$ as

$$\left| G_{ij}^o(e^{i\omega}) - \hat{G}_{ij}^n(e^{i\omega}) \right| \leq 3 \sqrt{\frac{n}{N} [\Phi^{-1}(\omega)]_{ii} \Phi_{v_j}(\omega)} \quad \text{w.p. } 99.9\% \quad (7.2.9)$$

Because model reduction will in general improve model accuracy, this bound can also be used for the reduced model. Define the error matrix of the reduced model as

$$\Delta(e^{i\omega}) = G^o(e^{i\omega}) - \hat{G}^l(e^{i\omega}) \quad (7.2.10)$$

Then we have an upper bound matrix

$$\bar{\Delta}(\omega) = \{\bar{\Delta}_{ij}(\omega)\} \quad (7.2.11)$$

where

$$\bar{\Delta}_{ij}(\omega) = 3 \sqrt{\frac{n}{N} [\Phi^{-1}(\omega)]_{ii} \Phi_{v_j}(\omega)} \quad (7.2.12)$$

such that

$$|\Delta_{ij}(e^{i\omega})| \leq \bar{\Delta}_{ij}(\omega) \quad \text{for all } i, j \quad \text{w.p. } 99.9\% \quad (7.2.13)$$

The upper bound can be used in robust stability/performance analysis; see, e.g., Skogestad and Postlethwaite (1996). If the model can pass the test, the designed controller can be implemented and tested on the real process. If the model cannot pass the test, one needs either to de-tune the controller or to redesign and carry out an identification test. The upper bound formula (7.2.9) supplies quantitative information for test redesign.

7.2.4 Determine a State Space Realization

The reduced model is given in a diagonal form matrix fraction description (MFD). In the development of the identification method, we have usefully exploited the simplicity of this model structure. One possible drawback of this model structure is that its minimality is not always guaranteed, which means that the sum of the degrees of the sub-models is greater than the McMillan degree of the model. This problem can be solved by converting the model to a state space model and then performing a model reduction on the state space model. For MIMO controller design, it is often convenient to have state space realizations of the process model and the disturbance model. Further model reduction on the state space model will result in a more compact model.

The reduced model is a diagonal form Box-Jenkins model

$$y(t) = [\hat{A}^l(q)]^{-1} \hat{B}^l(q) u(t) + [\hat{D}^l(q)]^{-1} \hat{C}^l(q) \varepsilon(t) \quad (7.2.14)$$

Here $\hat{A}^l(q)$, $\hat{C}^l(q)$ and $\hat{D}^l(q)$ are diagonal matrices and their polynomials are all monic. The transfer operator of the process model is

$$\begin{aligned} \hat{G}^l(q) &= [\hat{A}^l(q)]^{-1} \hat{B}^l(q) \\ &= \text{diag} \left[\frac{1}{\hat{A}_{11}^l(q)} \cdots \frac{1}{\hat{A}_{pp}^l(q)} \right] \begin{bmatrix} \hat{B}_{11}^l(q) & \cdots & \hat{B}_{1m}^l(q) \\ \vdots & \ddots & \vdots \\ \hat{B}_{m1}^l(q) & \cdots & \hat{B}_{mm}^l(q) \end{bmatrix} \end{aligned} \quad (7.2.15)$$

and the transfer operator of the disturbance is

$$\hat{H}^l(q) = \text{diag} \left[\frac{\hat{C}_{11}^l(q)}{\hat{D}_{11}^l(q)} \cdots \frac{\hat{C}_{11}^l(q)}{\hat{D}_{pp}^l(q)} \right] \quad (7.2.16)$$

The following steps are proposed to derive a state space model from the process model $\hat{G}^l(q)$.

- 1) Writing down a state space model for each MISO model. Denote the i -th sub-model of the process as

$$\hat{y}_i(t) = [\hat{A}_{ii}^l(q)]^{-1} \hat{B}_i^l(q) u(t) \quad (7.2.17)$$

where $\hat{A}_{ii}^l(q)$ is a polynomial

$$\hat{A}_{ii}^l(q) = 1 + \hat{a}_{ii,1} q^{-1} + \cdots + \hat{a}_{ii,l_i} q^{-l_i}$$

$\hat{B}_i^l(q)$ is a $1 \times m$ polynomial vector

$$\begin{aligned} \hat{B}_i^l(q) &= [\hat{b}_{i1,1} \cdots \hat{b}_{ip,1}] q^{-1} + \cdots + [\hat{b}_{i1,l_i} \cdots \hat{b}_{ip,l_i}] q^{-l_i} \\ &= \hat{B}_{i,1} q^{-1} + \cdots + \hat{B}_{i,l_i} q^{-l_i} \end{aligned}$$

Then the observer canonical form of the sub-model is simply

$$\begin{cases} x_i(t+1) = A_i x_i(t) + B_i u(t) \\ \hat{y}_i(t) = C_i x_i(t) \end{cases} \quad (7.2.18)$$

with

$$A_i = \begin{bmatrix} -\hat{a}_{ii,1} & 1 & 0 & \cdots & 0 \\ -\hat{a}_{ii,2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & & 1 \\ -\hat{a}_{ii,l_i} & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B_i = \begin{bmatrix} \hat{B}_{i,1} \\ \hat{B}_{i,2} \\ \vdots \\ \hat{B}_{i,l_i} \end{bmatrix}, \quad C_i = [1 \ 0 \ \cdots \ 0]$$

- 2) *Forming a state space model of the total MIMO model.* This is easily done as follows

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ \hat{y}(t) = Cx(t) \end{cases} \quad (7.2.19)$$

with

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_p \end{bmatrix}, \quad B_i = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_p \end{bmatrix}, \quad C = \begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_p \end{bmatrix}$$

- 3) *Performing a model reduction, if necessary.* The order of the total model (7.2.19) is

$$n_t = n1 + n2 + \cdots + np \quad (7.2.20)$$

Theoretically speaking, this order can be higher than the minimal order (McMillan degree) of the state space realization of the model, because it is not always possible to write a given process in a minimal diagonal form. In practice, the identified diagonal form will be minimal even if the true model cannot be written in a minimal diagonal form. This is because the disturbances cause model errors and model reduction takes place in identifying each MISO model. However, it is sensible to perform a model reduction if the total model (7.2.19) can be well approximated by a model with lower order.

The model reduction method based on a balanced realization (Glover, 1984) can be used here. In this method, the state space model is transformed to the so-called balanced realization. In such a realization, all the states are equally observable and controllable. Calculate the singular values of the Hankel matrix (see Appendix A) of the total model, and denote them as

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n_t}$$

If there are several very small Hankel singular values (Appendix A), we can eliminate the states corresponding to these Hankel singular values by simply truncating the balanced state space model. Denote $\hat{G}(e^{i\omega})$ as the transfer function of the reduced model and denote n_l as its order, it is known that the error caused by the balanced model reduction is bounded by (see Glover, 1984)

$$\sup_{\omega} \sigma_{\max} [\hat{G}^l(e^{i\omega}) - \hat{G}(e^{i\omega})] \leq 2 \left[\sum_{k=n_l+1}^{n_t} \sigma_k \right] \quad (7.2.21)$$

Using this relation and the upper bound matrix, it is suggested to determine the reduced order n such that

$$2 \left[\sum_{k=n_l+1}^{n_t} \sigma_k \right] \leq \frac{1}{3} \sup_{\omega} \sigma_{\max} (\bar{\Delta}(\omega)) \quad (7.2.22)$$

Using this rule, one can obtain a more compact model without a sacrifice of model accuracy.

It is important to note that the operations of this subsection should be done for the normalized model, namely, the model identified using scaled input-output data.

7.3 Identification of the Glass Tube Drawing Processes

7.3.1 Identification using White-noise-like PRBS Signals

The glass tube process studied in Chapters 4 and 5 will be used again as a test case. The same data will be used here, where the first 600 samples are used for model estimation and the remaining 669 samples for model validation. The input signals are white-noise-like PRBS; see Figure 7.3.5.

First a high order ARX model with structure [24, 24] is estimated. Then model reduction is performed for each MISO model with orders between 1 and 20. According to the ASYC criterion (7.2.12), the best model has the structure [8, 6]; see Figure 7.3.1. Figure 7.3.2 shows the step responses of the model. Figure 7.3.3 shows the frequency responses of the high order model and upper error bounds. Due to the use of the ASYC criterion, the difference between the high order model and the reduced model is small comparing to the upper bounds. The estimated models are simulated using validation data; and the relative simulation errors are 8.34% and 10.71% respectively; see Figure 7.3.4.

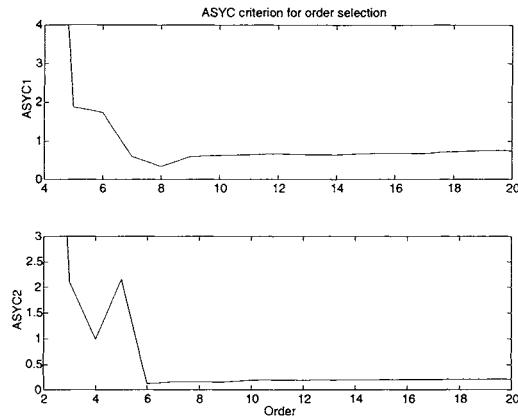


Figure 7.3.1 Order selection using the ASYC criterion. The selected orders are 8 and 6.

It is now a good moment to reflect on the exercise. In section 5.8 we identified the glass tube process using a text book approach. It was rather tedious even for this small MIMO process. Models with different gains can pass residual and correlation tests and the lack of low frequency information would be overlooked if we had not compared step responses. One can imagine it would be difficult to identify a process with more than 10 inputs and 20 outputs using the text book approach. An unexperienced user may easily run into difficulties.

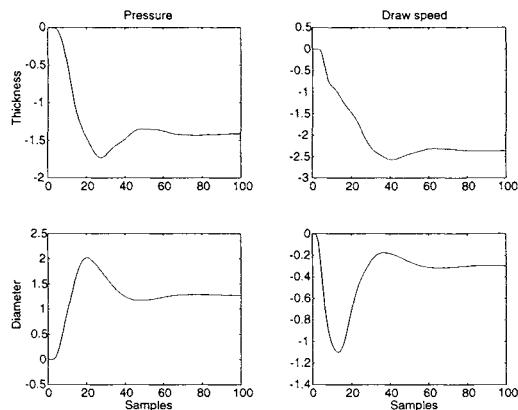


Figure 7.3.2 Model step responses

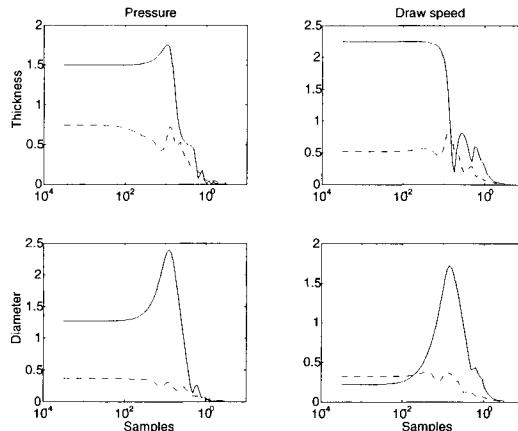


Figure 7.3.3 Frequency responses (solid lines) and upper error bounds (dashed lines) of the glass tube models.

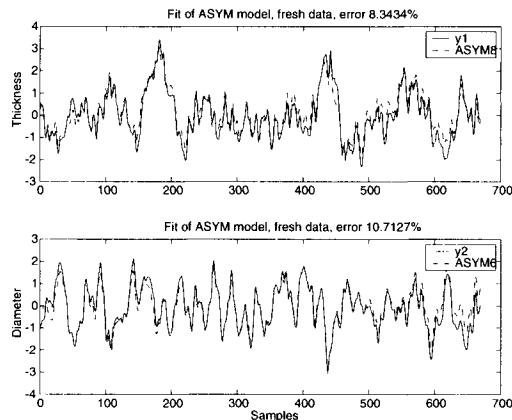


Figure 7.3.4 Fit of ASYM model using validation data.

When using ASYM, the model uncertainty is given by the upper bound matrix. Note that the uncertainty in gains given in Figure 5.8.10 are captured reasonably well by the upper bounds in Figure 7.3.3. So the user becomes aware of the potential problem at low frequencies. Because the upper bounds are related to design variables such as the spectrum of test signals and test time in a simple manner, it will provide information for test redesign when the model cannot pass validation for control, namely, robust stability/robustness test.

So far we have not yet treated the test design problem for this process which will be the topic of next subsection.

7.3.2 Identification using Lowpass PRBS Signals

One can see from the upper bounds that the model errors at low frequencies are large for transfer functions (1, 1) and (2, 2). As mentioned before, this is because white-noise-like PRBS signals are used in the test. According to upper bound formula (7.2.13), model errors at certain frequencies can be reduced by increasing the power of test signals at those frequencies or by increasing the test time. So if one uses colored or lowpass signals, the model quality at low frequencies can be improved. For the glass tube process, optimal spectra for the open-loop test are calculated and they are realized by adding a very slow PRBS signal and a fast PRBS signal; see Figure 7.3.5. One can see a big difference in move patterns between colored PRBS and white-noise-like PRBS. Note that the test design method outlined in Chapter 3 follows the same guideline, although GBN signals are used there.

First a high order ARX model with structure [24, 24] is estimated. Then model reduction is performed for each MISO model with orders between 1 and 20. According to the ASYC criterion, the best model has the structure [8, 12], but model structure [8, 5] will have nearly the same quality; see Figure 7.3.6. Figure 7.3.7 shows the step responses of the model. Figure 7.3.8 shows the frequency responses of the high order model and upper error bounds. One finds that indeed the error bounds at low frequencies are much smaller than that those of white noise models. Figure 7.3.9 shows model fit to the estimation data. This model is very different from the white noise model; compare Figure 7.3.3 and Figure 7.3.7. This is because the second test was done at a very different working point under different wall thickness and diameter.

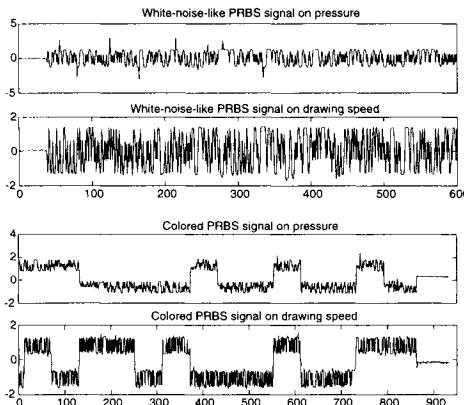


Figure 7.3.5 White-noise-like PRBS signals and colored noise PRBS signals

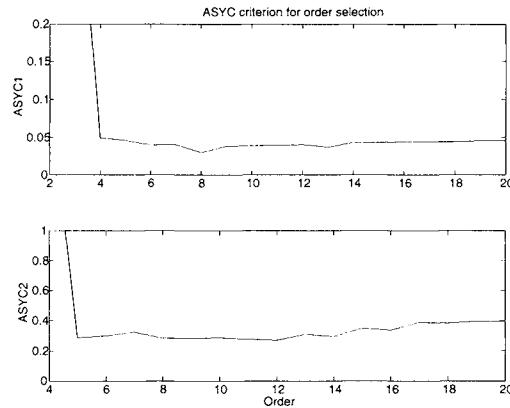


Figure 7.3.6 ASYC criteria for order selection

The reader may now appreciate what ASYM can offer: it tries to solve each of the four problems of identification optimally or sub-optimally for control; model validation and test (re)design are connected via error bounds.

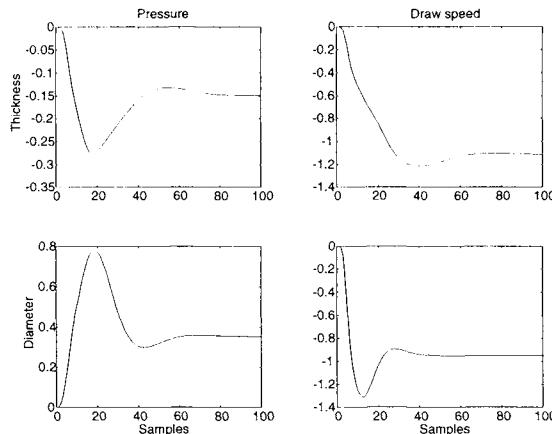


Figure 7.3.7 Model step responses, colored test signals

Here the glass tube process has been used to study identification. The control of the glass tube process can be found in Backx (1987) and Zhu and Backx (1993), where an output error method was used for identification. The model based control improved production considerably: the

standard deviations of diameters and wall thickness were reduced by 50% and the change-over time reduced by 90%!

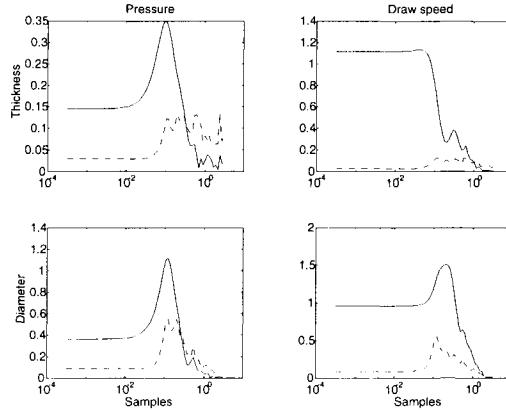


Figure 7.3.8 Frequency responses and upper error bounds, colored test signals

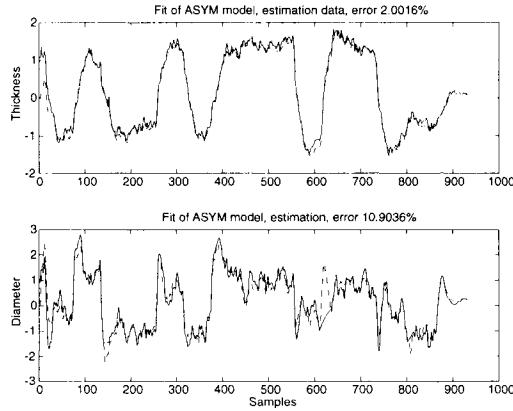


Figure 7.3.9 Fit of ASYM model using estimation data, colored PRBS signals

7.4 Conclusions

In this chapter the ASYM approach is extended to the MIMO case. The strength of the method is its ability to provide systematic solutions to the four problems of identification for control.

This can lead to an automatic identification procedure that is very user friendly; see Zhu (2000b). The method has been tested using the glass tube process and its way of solving the problem has been compared with those of standard text book methods. The reader may find that, in the ASYM approach, more attention is paid to test design, model validation and the connection between the two. Experience has shown that these two steps play more important roles in industrial applications than parameter estimation and order selection do. In Chapter 10 the use of ASYM for process control is further demonstrated.

Chapter 8

Subspace Model Identification of MIMO Processes

By Michel Verhaegen

8.1 Introduction

Many parameter estimation algorithms used in system identification are based on numerical schemes to solve parametric optimization problems. Apart from the fact that the user has to make a selection on a *particular* model parametrization, the *iterative* nature of many of these optimization schemes requires *accurate* initial estimates. On the one hand, both selections can have a critical influence on the results of the optimization run and hence on the *quality* of the identified model. On the other hand, providing the user with reliable information on both selection items has long remained an open and challenging research topic.

Subspace identification methods have the potential to provide extremely useful information in the two critical selections mentioned above. Furthermore, a vast amount of practical evidence has shown that the results obtained by the non-iterative subspace identification schemes do not need further improvement in iterative parametric optimization methods.

In this chapter, we highlight the fundamental nature of subspace identification algorithms. Since the latter are based on elementary *linear algebra* results, a summary of the relevant matrix analysis tools is given in Appendix A.

To follow the tread of the book, we start outlining the nature of subspace identification algorithms first for the special case of using step response measurements neglecting errors on the data. Step responses are often used in industrial applications in order to acquire initial information to design dedicated identification experiments. We start the chapter by formulating the identification problem considered for general input and perturbation conditions. A crucial step in the analysis

and solution of subspace identification methods is to relate input and output data to the system matrices in a structured manner so both data and model information are represented as matrices and not just as vectors and matrices as is the case in the classical definition of state space models. This is done in Section 8.3. The step input response is treated in Section 8.4. The generalization to different and more general input sequences is analyzed in Section 8.5.1. A special section, Section 8.6, is devoted to the analysis of perturbations considered in Section 8.2 in a subspace identification context. In addition to the identification of dynamic systems operating in *open-loop*, extensions to address the identification in *closed-loop* is given as well. Finally in Section 8.8 we summarize some extensions to the identification of nonlinear systems.

8.2 Definition of the State Space Identification Problem

Consider the block schematic representation given in Figure 8.2.1. Here the plant \mathbf{P} denotes the unknown system to be identified. It is assumed that the mathematical relationship between the measurable input $u(k)$ and the observed output $y(k)$ is described by the following state space model:

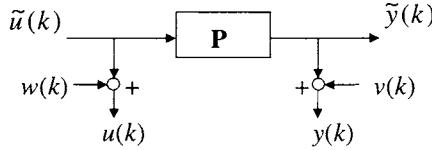


Figure 8.2.1 Block-schematic representation of a plant \mathbf{P} and generic perturbations

$$\begin{aligned} x(k+1) &= Ax(k) + B\tilde{u}(k) + f(k) \quad \text{for } k \geq 0; \quad x(0) = x_0; \quad x(k) \in \mathbb{R}^n \\ y(k) &= Cx(k) + D\tilde{u}(k) + v(k) \quad \tilde{u}(k) \in \mathbb{R}^m; \quad y(k) \in \mathbb{R}^\ell \\ u(k) &= \tilde{u}(k) + w(k) \end{aligned} \quad (8.2.1)$$

where $f(k)$, $w(k)$ and $v(k)$ are assumed to be mutually correlated ergodic, zero-mean, white noise sequences satisfying:

$$E\left[\begin{bmatrix} f(t) \\ w(t) \\ v(t) \end{bmatrix} \left[\begin{matrix} f(s)^T & w(s)^T & v(s)^T \end{matrix} \right]\right] =: \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{12}^T & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{13}^T & \Sigma_{23}^T & \Sigma_{33} \end{bmatrix} \delta_{s,t}$$

with $E[\cdot]$ denoting the mathematical expectation and $\delta_{s,t}$ is the dirac function satisfying,

$$\delta_{s,t} =: \begin{cases} 1 & s = t \\ 0 & s \neq t \end{cases}$$

Further, it is assumed that the state space quadruple of system matrices (A, B, C, D) is minimal (Kailath, 1980).

Based on this description of the plant \mathbf{P} , the identification problem considered in this chapter is defined as:

Given the input-output time sequences

$$\{u(k), y(k)\}_{k=0}^N = [u(0), y(0) \ u(1), y(1) \ \cdots \ u(N), y(N)],$$

the problem is to estimate in a *statistically consistent* manner:

1. the system order n , and
2. the system matrices (A, B, C, D) and the initial state $x(0)$ up to an unknown similarity transformation.

In the subsequent three Sections 8.3, 8.4 and 8.5.1 we analyze the above identification problem assuming the additive perturbations $f(k), w(k)$ and $v(k)$ to be zero. This allows us to first introduce the basic principles of subspace model identification from a simple algebraic viewpoint.

8.3 Definition of the Data Equation

An important and critical step prior to the design (and use) of subspace identification algorithms is to find an *appropriate relationship* between the measured data sequences, on one hand, and the matrices (or parameters) that define the model, on the other hand. In this section and Sections 8.4, 8.5.1, we will treat the identification problem stated in Section 8.2 for the special case where the input-output description of the plant \mathbf{P} is given as:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{8.3.1}$$

This model, however, only relates the (local) quantities $\{x(k+1), x(k), y(k), u(k)\}$. To relate larger segments from the given input-output time records to matrix functions of the model matrices $\{A, B, C, D\}$, let us consider the system response of the multivariable system (8.3.1) the first k time instances:

$$\left\{ \begin{array}{lcl} x(1) &= Ax(0) &+ Bu(0) \\ x(2) &= A^2x(0) &+ ABu(0) &+ Bu(1) \\ x(3) &= A^3x(0) &+ A^2Bu(0) &\dots \\ \vdots & & & \\ x(k) &= A^kx(0) &+ A^{k-1}Bu(0) &\dots &+ ABu(k-2) &+ Bu(k-1) \end{array} \right.$$

$$\left\{ \begin{array}{lcl} y(0) & = & Cx(0) + Du(0) \\ y(1) & = & CAx(0) + CBu(0) \quad \dots \\ y(2) & = & CA^2x(0) + CABu(0) \quad \dots \\ \vdots & & \\ y(k-1) & = & CA^{k-1}x(0) + CA^{k-2}Bu(0) \quad \dots + CBu(k-2) + Du(k-1) \end{array} \right.$$

This sequence of expressions allows the specification of the following relationship between the input-output segments $\{u(k)\}_{k=0}^{s-1}$, $\{y(k)\}_{k=0}^{s-1}$,

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(s-1) \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix}}_{\Gamma_s} x(0) + \underbrace{\begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & & \ddots & & \\ CA^{s-2}B & \cdots & CB & D \end{bmatrix}}_{H_s} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(s-2) \\ u(s-1) \end{bmatrix} \quad (8.3.2)$$

In the sequel, the matrix Γ_s will be referred to as the *extended observability matrix*. Eq. (8.3.2) relates *vectors*, derived from the input and output data sequences and the (unknown) initial condition $x(0)$, to the *matrices* Γ_s, H_s , derived from the system matrices (A, B, C, D) . So any attempt to retrieve information on the latter from the given vectors will fail. However, observe that since the underlying system is time-invariant, we can relate the same '*model*' *matrices* Γ_s, H_s to the shifted variants of the '*input-output*' *vectors*. Consider for example a shift over $k+1$ samples, we obtain:

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+s-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix} x(k) + \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & & \ddots & & \\ CA^{s-2}B & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+s-2) \\ u(k+s-1) \end{bmatrix} \quad (8.3.3)$$

Then we can combine the relationships (8.3.2) and (8.3.3) for different shifts k (permitted by available input-output samples) as follows:

$$\begin{bmatrix} y(0) & y(1) & \cdots & y(N-s+1) \\ y(1) & y(2) & \cdots & y(N-s+2) \\ \vdots & & & \vdots \\ y(s-1) & y(s) & \cdots & y(N) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix} [x(0) \ x(1) \ \cdots \ x(N-s+1)] + \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & & \ddots & & \\ CA^{s-2}B & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u(0) & u(1) & \cdots & u(N-s+1) \\ u(1) & u(2) & \cdots & u(N-s+2) \\ \vdots & & & \vdots \\ u(s-1) & u(s) & \cdots & u(N) \end{bmatrix} \quad (8.3.4)$$

The above *data equation* is denoted compactly as,

$$Y_{0,s,N} = \Gamma_s X_{0,1,N-s+1} + H_s U_{0,s,N} \quad (8.3.5)$$

where the first entry of the pair of subscripts of the (block-) Hankel matrices $Y_{0,s,N}$ and $U_{0,s,N}$ only refer to the time index of its top left entry, the second refers to the number of block-rows in the Hankel matrices and the third refers to the time index of the bottom right entry. The same holds for the Hankel matrix $X_{0,1,N-s+1}$.

The data equation (8.3.5) now is a relationship between matrices constructed from the data and matrices constructed from the system matrices. So the hope is that this 'equivalence' in representation allows to derive information on the system matrices (A, B, C, D) from that of the data matrices, such as $Y_{0,s,N}, U_{0,s,N}$. This idea is explored first for the special case the input is a step function.

8.4 Analysis of Step Response Measurements

Transforming the Data Equation

The step response of the system, restricted for the sake of brevity to a single-input, multiple-output (SIMO) system, is the response to the special input defined as:

$$u(k) = \begin{cases} 1 & \text{for } k \geq 0 \\ 0 & \text{for } k < 0 \end{cases} \quad (8.4.1)$$

For this special case, let $\mathbb{E}_{N'} \in \mathbb{R}^{N'}$ (for $N' = N - s + 2$) denote the vector with all entries equal to one. Then we can write the data equation (8.3.5):

$$Y_{0,s,N} = \Gamma_s X_{0,s,N} + (H_s \mathbb{E}_s) \mathbb{E}_{N'}^T \quad (8.4.2)$$

The question now is, having information only on $Y_{0,s,N}$ and \mathbb{E}_N , whether we can decompose $Y_{0,s,N}$ again into a sum of 2 components that bear information on either of the two original components of the sum in Eq. (8.4.2). One such (orthogonal) decomposition is,

$$\begin{aligned} Y_{0,s,N} &= Y_{0,s,N} \left(I - \frac{\mathbb{E}_{N'}}{\sqrt{N'}} \frac{\mathbb{E}_{N'}^T}{\sqrt{N'}} \right) + \left(Y_{0,s,N} \frac{\mathbb{E}_{N'}}{\sqrt{N'}} \right) \frac{\mathbb{E}_{N'}^T}{\sqrt{N'}} \\ &= \Gamma_s X_{0,s,N} \left(I - \frac{\mathbb{E}_{N'}}{\sqrt{N'}} \frac{\mathbb{E}_{N'}^T}{\sqrt{N'}} \right) + \left(Y_{0,s,N} \frac{\mathbb{E}_{N'}}{\sqrt{N'}} \right) \frac{\mathbb{E}_{N'}^T}{\sqrt{N'}} \end{aligned} \quad (8.4.3)$$

This follows from the fact that $\mathbb{E}_{N'}^T \mathbb{E}_{N'} = N'$. The last expression shows that the columns of the transformed data matrix $Y_{0,s,N} \left(I - \frac{\mathbb{E}_{N'}}{\sqrt{N'}} \frac{\mathbb{E}_{N'}^T}{\sqrt{N'}} \right)$ are indeed formed by a linear combination of that of the extended observability matrix. In mathematical terms, this result shows that the column space of the matrix $Y_{0,s,N} \left(I - \frac{\mathbb{E}_{N'}}{\sqrt{N'}} \frac{\mathbb{E}_{N'}^T}{\sqrt{N'}} \right)$ is contained in that of the matrix Γ_s . That both are *equal* is shown in the next subsection.

As an illustration, we display in Figure 8.4.1, the *orthogonal* decomposition in Eq. (8.4.3) for the special case $N = s = 2$.

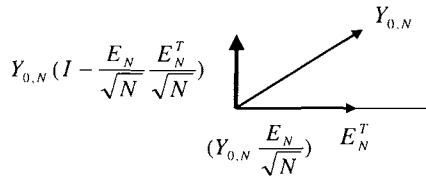


Figure 8.4.1 Illustration of Eq. (8.4.3) for $N = s = 2$

8.4.1 Deriving the System Order n and the System Matrices

That one of the matrices in the decomposition of $Y_{0,s,N}$ in Eq. (8.4.3) has a special structure is revealed in the following theorem.

Theorem 1 Let the SIMO system given by Eq. (8.3.1) be minimal, let A be asymptotically stable (a.s.), that is all eigenvalues of A have modulus smaller than 1, let the input be step as defined in Eq. (8.4.1) and let $x(0) = 0$, then:

$$(i) \lim_{N \rightarrow \infty} \frac{1}{N'} \sum_{k=0}^{N'-1} x(k) = (I - A)^{-1} B \quad (8.4.4)$$

$$(ii) \lim_{N \rightarrow \infty} Y_{0,s,N} (I - \frac{1}{N'} \mathbb{E}_{N'} \mathbb{E}_{N'}^T) = -\Gamma_s (I - A)^{-1} [B \ AB \ A^2 B \ \dots] \quad (8.4.5)$$

Proof: Since the system is a.s., the state converges to a stationary value \bar{x} , which satisfies:

$$\bar{x} = A\bar{x} + B \Rightarrow \bar{x} = (I - A)^{-1} B$$

Because the state sequence is deterministic, its mean value also equals:

$$\bar{x} = \lim_{N \rightarrow \infty} \frac{1}{N'} \sum_{k=0}^{N'-1} x(k)$$

which concludes the proof of (i). For part (ii) note that since $(\mathbb{E}_N^T \mathbb{E}_N) = N$,

$$\begin{aligned} \lim_{N' \rightarrow \infty} Y_{0,s,N}(I - \frac{1}{N'} \mathbb{E}_{N'} \mathbb{E}_{N'}^T) &= \lim_{N' \rightarrow \infty} \Gamma_s X_{0,s,N} \left(I - \frac{1}{N'} \mathbb{E}_{N'} \mathbb{E}_{N'}^T \right) \\ &= \lim_{N' \rightarrow \infty} \Gamma_s \left(X_{0,s,N} - \frac{1}{N'} X_{0,s,N} \mathbb{E}_{N'} \mathbb{E}_{N'}^T \right) \\ &= \lim_{N' \rightarrow \infty} \Gamma_s \left(X_{0,s,N} - \frac{1}{N'} \sum_{k=0}^{N'-1} x(k) \mathbb{E}_{N'}^T \right) \end{aligned}$$

and using (i) and the explicit expression for $X_{0,s,N}$ in terms of the system matrices (A, B) , the right hand side of the last equation equals:

$$\begin{aligned} &= \Gamma_s \left([0 \mid B \mid AB + B \mid A^2B + AB + B \mid \cdots] - (I - A)^{-1} B \mathbb{E}_{N'}^T \right) \\ &= \Gamma_s \left[-(I - A)^{-1} B \mid B - (I - A)^{-1} B \mid AB + B - (I - A)^{-1} B \mid \cdots \right] \\ &= \Gamma_s \left[-(I - A)^{-1} B \mid B - B - AB - A^2B - \cdots \mid AB + B - B - AB - A^2B - \cdots \mid \cdots \right] \\ &= -\Gamma_s (I - A)^{-1} \left[B \mid AB \mid A^2B \mid \cdots \right] \end{aligned}$$

where we have made use of the series expansion,

$$(I - A)^{-1} = I + A + A^2 + A^3 + \cdots$$

which converges for A a.s. \square

Theorem 1 highlights that for the special case the input $u(k)$ is a step function, the system (8.3.1) is minimal, the matrix A is asymptotically stable and the number of block rows of the Hankel matrix $Y_{0,s,N}$, that is the integer s , satisfies,

$$s > n \tag{8.4.6}$$

then denoting the rank of the matrix M by $\text{rank}(M)$, the following conditions hold:

1. $(I - A)$ is invertible,
2. $\text{rank}(\Gamma_s (I - A)^{-1}) = n$ and
3. $\text{rank} \left[B \mid AB \mid \cdots \mid A^n B \mid \cdots \right] = n$

Based on these immediate conclusions from Theorem 1, an application of Sylvester's inequality (A.3.5), shows that:

$$\text{rank} \left(\lim_{N' \rightarrow \infty} Y_{0,s,N}(I - \frac{1}{N'} \mathbb{E}_{N'} \mathbb{E}_{N'}^T) \right) = n \tag{8.4.7}$$

The combination of the results (8.4.4) and (8.4.7) show that the column space of the matrix $\lim_{N \rightarrow \infty} Y_{0,s,N}(I - \frac{1}{N'} \mathbb{E}_{N'} \mathbb{E}_{N'}^T)$ *not only* is contained in that of the matrix Γ_s , **but** is equal. For if it is a proper subspace of the column space of Γ_s , its rank has to be smaller than n . This result is denoted as,

$$\text{range} \left(\lim_{N \rightarrow \infty} Y_{0,s,N}(I - \frac{1}{N'} \mathbb{E}_{N'} \mathbb{E}_{N'}^T) \right) = \text{range}(\Gamma_s) \quad (8.4.8)$$

As a consequence of the above exposure and Theorem 1, a singular value decomposition of the matrix $Y_{0,s,N}(I - \frac{1}{N'} \mathbb{E}_{N'} \mathbb{E}_{N'}^T)$ would provide us, as outlined in Appendix A.3, with three important sources of information:

1. The number of non-zero singular values equals the rank of Γ_s and thus the order n of the underlying system to be identified;
2. Let the SVD be denoted as,

$$Y_{0,s,N} \left(I - \frac{1}{N'} \mathbb{E}_{N'} \mathbb{E}_{N'}^T \right) = U_n S_n V_n^T \quad (8.4.9)$$

such that $\text{rank}(S_n) = n$, then,

$$\lim_{N \rightarrow \infty} U_n = \Gamma_s T = \begin{bmatrix} CT \\ CTT^{-1}AT \\ \vdots \\ CT(T^{-1}AT)^{s-1} \end{bmatrix} = \begin{bmatrix} C_T \\ C_T A_T \\ \vdots \\ C_T A_T^{s-1} \end{bmatrix} \quad (8.4.10)$$

for T a non-singular (arbitrary) transformation in $\mathbb{R}^{n \times n}$;

3. From the SVD (8.4.9) and (8.4.4), it follows,

$$\begin{aligned} \lim_{N \rightarrow \infty} S_n V_n^T &= -T^{-1}(I - A)^{-1} [B \mid AB \mid \cdots \mid A^n B \mid \cdots] \\ &= -\left(T^{-1}(I - A)T \right)^{-1} T^{-1} [B \mid AB \mid \cdots \mid A^n B \mid \cdots] \\ &= -(I - A_T)^{-1} [B_T \mid A_T B_T \mid \cdots \mid A_T^n B_T \mid \cdots] \end{aligned} \quad (8.4.11)$$

The system matrices (A, B, C) can be computed (up to an unknown similarity transformation T) from Eqs. (8.4.10) and (8.4.11). Using a Matlab (The Mathworks [77]) like way of denoting subparts of matrices, the matrix C_T follows from:

$$C_T = \lim_{N \rightarrow \infty} U_n(1 : \ell, :) \quad (8.4.12)$$

The matrix A_T is computed by solving the following over-determined equation, which due to condition (8.4.6) has a unique solution,

$$\left(\lim_{N \rightarrow \infty} U_n(1 : (s-1)\ell, :) \right) A_T = \lim_{N \rightarrow \infty} U_n(\ell + 1 : s\ell, :) \quad (8.4.13)$$

Let \bar{V}_n^T denote $(I - A_T)S_n V_n^T$, then the matrix B_T follows from (8.4.11),

$$B_T = \lim_{N \rightarrow \infty} \bar{V}_n^T(:, 1) \quad (8.4.14)$$

The above exposure can easily be summarized in an algorithmic implementation that consists of a few lines of Matlab code. Further, the extension to the MIMO case is trivial by considering the step response of each entry of the input vector $u(k)$ individually.

Remark 2 To use the above theory, we basically need to work with matrices (operators) with an infinite number of columns. This can be avoided when the stationary value of the output $y(k)$ is known. Denote this stationary value by \bar{y} , then the operator $\lim_{N \rightarrow \infty} Y_{0,s,N}(I - \frac{1}{N} \mathbb{E}_{N'} \mathbb{E}_{N'}^T)$ in Eq. (8.4.4) is replaced by the matrix

$$Y_{0,s,N} = \begin{bmatrix} \bar{y} \\ \bar{y} \\ \vdots \\ \bar{y} \end{bmatrix} \mathbb{E}_{N'}^T$$

and all subsequent results derived from this theorem hold for finite $N \geq n + s - 2$. \square

Remark 3 For the special case the input is an impulse, defined again for the SIMO case as,

$$u(k) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases} \quad (8.4.15)$$

the data equation has the special form,

$$Y_{0,s,N} = \Gamma_s X_{0,s,N} + H_s \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

This form of the data equation immediately shows that the column space of the matrix $Y_{1,s,N}$ is contained in that of Γ_s . Inspecting the matrix $X_{1,1,N-s+1}$ more closely for the case the initial conditions x_0 are zero, yields the following result:

$$X_{1,1,N-s+1} = [B \ AB \ A^2B \ \cdots]$$

and therefore,

$$Y_{1,s,N} = \Gamma_s [B \ AB \ A^2B \ \cdots]$$

A simple SVD of the matrix $Y_{1,s,N}$ allows to determine the system matrices (A_T, B_T, C_T) in a similar way as outlined above. This is Kung's implementation of the famous Ho-Kalman realization algorithms (Ho and Kalman, 1966 [58]).

Remark 4 The number of block-rows s in the definition of the Hankel matrices $U_{0,s,N}$ and $Y_{0,s,N}$ is made based on a rough estimate of the order of the underlying system. This requires some engineering judgement. Past experience with the class of algorithms described in this chapter has shown that selecting s is not a critical problem. \square

8.5 Subspace Identification using Generic Input Signals

8.5.1 Deriving the Column Space of the Matrix Γ_s

When the input is of a more general nature than a step or an impulse, it is still possible to derive the data equation as given in Eq. (8.3.5). Furthermore, making use of a basic tool from linear algebra, namely the RQ factorization (see Appendix A), we can perform a similar decomposition of $Y_{0,s,N}$ as in Eq. (8.4.3) that reveals the column space of Γ_s .

For that purpose, let us denote the following RQ factorization:

$$\begin{bmatrix} U_{0,s,N} \\ Y_{0,s,N} \end{bmatrix} = \begin{bmatrix} R_u & 0 \\ R_{yu} & R_y \end{bmatrix} \begin{bmatrix} Q_u \\ Q_y \end{bmatrix} \quad (8.5.1)$$

Then we can perform the same (additive) decomposition of $Y_{0,s,N}$ as in (8.4.3) now with the matrix Q_u (the matrix with row space equal to that of $U_{0,s,N}$) replacing $\frac{1}{\sqrt{N}}\mathbf{E}_N^T$. This decomposition is written as,

$$Y_{0,s,N} = Y_{0,s,N}(I - Q_u^T Q_u) + (Y_{0,s,N} Q_u^T) Q_u$$

Using the RQ factorization, the data equation (8.3.5) can be written as,

$$Y_{0,s,N} = \Gamma_s X_{0,s,N} + (H_s R_u) Q_u \quad (8.5.2)$$

That in this case the column space of the matrix $Y_{0,s,N}(I - Q_u^T Q_u)$ is again contained in that of the matrix Γ_s is shown in the following theorem.

Theorem 5 *Let the input-output data of the system (8.3.1) be stored in the (block-) Hankel matrices $U_{0,s,N}$, $Y_{0,s,N}$ respectively as in Eqs. (8.3.4) and (8.3.5), and let the RQ factorization (8.5.1) be given, then:*

$$\begin{aligned} Y_{0,s,N}(I - Q_u^T Q_u) &= R_y Q_y \\ &= \Gamma_s X_{0,s,N} Q_y^T Q_y \end{aligned} \quad (8.5.3)$$

Proof: From the RQ factorization (8.5.1) we can express $Y_{0,s,N}$ as:

$$Y_{0,s,N} = R_y Q_y + R_{yu} Q_u \quad (8.5.4)$$

Further, it follows from the orthogonality of the matrix $\begin{bmatrix} Q_u \\ Q_y \end{bmatrix}$ that,

$$\begin{bmatrix} Q_u^T & Q_y^T \end{bmatrix} \begin{bmatrix} Q_u \\ Q_y \end{bmatrix} = I \Rightarrow (I - Q_u^T Q_u) = Q_y^T Q_y$$

and

$$\begin{bmatrix} Q_u \\ Q_y \end{bmatrix} \begin{bmatrix} Q_u^T & Q_y^T \end{bmatrix} = I \Rightarrow Q_u Q_u^T = 0 \quad Q_y Q_y^T = I$$

The right hand side of the above 2 implications and the expression (8.5.2) are used to derive the following result,

$$\begin{aligned} Y_{0,s,N}(I - Q_u^T Q_u) &= Y_{0,s,N} Q_y^T Q_y \\ &= \Gamma_s X_{0,s,N} Q_y^T Q_y \end{aligned}$$

Using again the right hand side of the above 2 implications and the expression (8.5.4) for $Y_{0,s,N}$ we arrive at,

$$\begin{aligned} Y_{0,s,N}(I - Q_u^T Q_u) &= Y_{0,s,N} Q_y^T Q_y \\ &= R_y Q_y \end{aligned}$$

which completes the proof. \square

The theorem shows that the column space of the matrix $Y_{0,s,N}(I - Q_u^T Q_u)$, and in the same way that of the matrix R_y , is *contained in the matrix* Γ_s . To show that both column spaces are *equal* is far more difficult to prove than for the special input cases considered in Section 8.4. From a practical point of view, the condition generally holds and depends on the persistency of excitation (the richness) of the input (Verhaegen and Dewilde, 1992 [114], Verhaegen, 1993 [111], Verhaegen, 1994 [112] and Jansson and Wahlberg, 1998). In the next section, we assume that the input is such that the condition

$$\text{rank}(R_y) = n$$

holds.

8.5.2 The Calculation of the System Matrices

If we denote the SVD of the matrix R_y as:

$$R_y = U_n S_n V_n^T \quad (8.5.5)$$

with $\text{rank}(S_n) = n$, then the important relationship (8.4.10) holds for finite N . From the matrix U_n we can compute the pair (A_T, C_T) as highlighted in Eqs. (8.4.12), (8.4.13). The calculation of the pair (B_T, D) and the initial condition $x_T(0)$ is discussed next.

When the pair (A_T, C_T) is known, the output $y(k)$ of the system given by (8.3.1) can be written as,

$$y(k) = C_T A_T^k x_T(0) + \left(\sum_{\tau=0}^{k-1} u(\tau)^T \otimes C_T A_T^{k-\tau-1} \right) \text{vec}(B_T) + u(k)^T \otimes \text{vec}(D) \quad (8.5.6)$$

where \otimes and $\text{vec}(\cdot)$ resp. denote the Kronecker product and the vector resulting by storing the columns of the matrix (\cdot) on top of each other. When we define the vector θ and the matrix Φ_N as follows,

$$\theta := \begin{bmatrix} x_T(0) \\ \text{vec}(B_T) \\ \text{vec}(D) \end{bmatrix}$$

$$\Phi_N := \begin{bmatrix} C_T & 0 & u_0^T \otimes I_\ell \\ C_T A_T & u_0^T \otimes C_T & u_1^T \otimes I_\ell \\ \vdots & \vdots & \vdots \\ C_T A_T^{N-1} & \sum_{\tau=0}^{N-1} u_\tau^T \otimes C_T A_T^{N-1-\tau} & u_N^T \otimes I_\ell \end{bmatrix} \quad (8.5.7)$$

then we can accumulate (8.5.6) into the following over-determined set of equations:

$$Y_{0,N,1} = \Phi_N \theta$$

from which we can solve the unknown quantities in θ by solving the least squares problem:

$$\min_{\theta} \|Y_{0,N,1} - \Phi_N \theta\| \quad (8.5.8)$$

where $\|\cdot\|$ denotes the Euclidean norm.

Example 6 To illustrate the formulation in Eq. (8.5.6), consider the SISO second order system:

$$x(k+1) = Ax(k) + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(k)$$

$$y(k) = Cx(k) + du(k)$$

Further consider the related systems:

$$\begin{cases} x_1(k+1) = Ax_1(k) & x_1(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ y_1(k) & = Cx_1(k) \end{cases}$$

$$\begin{cases} x_2(k+1) = Ax_2(k) & x_2(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ y_2(k) & = Cx_2(k) \end{cases}$$

$$\begin{cases} x_3(k+1) = Ax_3(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k) & x_3(0) = 0 \\ y_3(k) & = Cx_3(k) \end{cases}$$

$$\begin{cases} x_4(k+1) = Ax_4(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) & x_4(0) = 0 \\ y_4(k) = Cx_4(k) \end{cases}$$

Then, by the linearity of the system, the output $y(k)$ is written as,

$$y(k) = [y_1(k) \ y_2(k)] x(0) + [y_3(k) \ y_4(k)] \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + u(k)d$$

□

8.6 Treatment of Additive Perturbations

When additive perturbations, like $f(k), w(k), v(k)$ are present as in the generic identification configuration as depicted in Figure 8.2.1 of Section 8.2, the scheme discussed in Section 8.5.1 produces biased estimates (Verhaegen, 1993 [110]).

One possible solution for overcoming this is the use of instrumental variables (Ljung, 1987). A number of subspace identification variants have been proposed in this context, see e.g. Verhaegen, (1993) [110], Verhaegen, (1994) [111], Peternell *et. al.* (1996) [87] and Van Overschee and De Moor (1996) [108]. In this chapter, one recently proposed variant will be summarized (Chou and Verhaegen 1997 [26]).

The main reason for this is its wide range of applicability indicated by the presence of all perturbations $f(k), w(k), v(k)$ as defined in Section 8.2 in consistently estimating the column space of the matrix Γ_s . For subsequent calculations of the system matrices $[B_T \ D]$ and $x_T(0)$ case dependent solutions are currently proposed. Two of these cases will be summarized in this section. Namely for the open-loop and *closed-loop* identification case where we have $w(k) \equiv 0$. Other cases, such as where we restrict ourselves to the use of periodic inputs and/or $w(k) \neq 0$ can be analyzed equivalently, but for the sake of brevity are not analyzed in this chapter.

8.6.1 Deriving a Consistent Estimate of the Column Space of Γ_s

When the perturbations $f(k), w(k), v(k)$ defined in Section 8.2 are present, a consistent estimate of the column space of Γ_s can be obtained as summarized in the next theorem.

Theorem 7 *Let the input-output data of the system (8.2.1) be stored in the (block-) Hankel matrices $U_{0,s,N}, U_{s,s,N}, Y_{0,s,N}$ and $Y_{s,s,N}$ respectively, and let the RQ factorization be given as:*

$$\lim_{N \rightarrow \infty} \frac{1}{N} \begin{bmatrix} U_{s,s,N} \\ Y_{s,s,N} \end{bmatrix} \begin{bmatrix} U_{0,s,N}^T & Y_{0,s,N}^T \end{bmatrix} = \begin{bmatrix} R'_u & 0 \\ R'_{yu} & R'_y \end{bmatrix} \begin{bmatrix} Q'_u \\ Q'_y \end{bmatrix} \quad (8.6.1)$$

then,

$$R'_y Q'_y = \lim_{N \rightarrow \infty} \frac{1}{N} \Gamma_s X_{s,s,N} [U_{0,s,N}^T \ Y_{0,s,N}^T] (Q'_y)^T Q'_y \quad (8.6.2)$$

Proof: For the perturbations considered in Section 8.2, the data equation reads:

$$Y_{s,s,N} = \Gamma_s X_{s,s,N} + H_s^u \tilde{U}_{s,s,N} + H_s^f F_{s,s,N} + V_{s,s,N}$$

with H_s^u the lower triangular Toeplitz matrix H_s defined in Eqs. (8.3.4) and (8.3.5), H_s^f as H_s defined for the quadruple $(A, I, C, 0)$ and $\tilde{U}_{s,s,N}$, $F_{s,s,N}$ and $V_{s,s,N}$ (block-) Hankel matrices defined from $\tilde{u}(k)$, $f(k)$ and $v(k)$ respectively, similarly to the definition of $U_{s,s,N}$ from $u(k)$ in Eqs. (8.3.4) and (8.3.5).

By the definition of $\tilde{u}(k)$, we also have:

$$U_{s,s,N} = \tilde{U}_{s,s,N} + W_{s,s,N}$$

Since $f(k)$, $w(k)$, $v(k)$ are zero-mean white noise processes, we have that:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \begin{bmatrix} U_{s,s,N} \\ Y_{s,s,N} \end{bmatrix} [U_{0,s,N}^T \ Y_{0,s,N}^T] = \lim_{N \rightarrow \infty} \frac{1}{N} \begin{bmatrix} \tilde{U}_{s,s,N} \\ \Gamma_s X_{s,s,N} + H_s^u \tilde{U}_{s,s,N} \end{bmatrix} [U_{0,s,N}^T \ Y_{0,s,N}^T]$$

The left hand side corresponds to the 'noise-free' case considered in Section 8.5.1 for the input matrix

$$\lim_{N \rightarrow \infty} \frac{1}{N} \tilde{U}_{s,s,N} [U_{0,s,N}^T \ Y_{0,s,N}^T]$$

and the output matrix,

$$\lim_{N \rightarrow \infty} \frac{1}{N} \Gamma_s X_{s,s,N} [U_{0,s,N}^T \ Y_{0,s,N}^T] + \lim_{N \rightarrow \infty} \frac{1}{N} H_s^u \tilde{U}_{s,s,N} [U_{0,s,N}^T \ Y_{0,s,N}^T]$$

An application of Theorem 5 to these input-output matrices, which no longer have a (block-) Hankel structure, completes the proof of this Theorem. \square

The calculation of the system matrices $[A_T \ C_T]$ proceeds as indicated by Eqs. (8.4.12) and (8.4.13), provided the rank of the matrix R'_y equals n . An analysis of the conditions (of the persistency of excitation of the input $u(k)$) necessary to guarantee that this rank condition of R'_y is satisfied is complicated and not yet fully understood. Nevertheless, in many practical studies the approach offers reliable information about the order of an underlying linear system.

8.6.2 Calculation of the System Matrices $[B_T \ D]$ under the Presence of Process and Measurement Noise: Open-Loop

For the case where the error-term $w(k) \equiv 0$, the approach outlined in Section 8.5.2 can still be used to compute consistent estimates of the matrices $(B_T \ D)$ and $x_T(0)$.

When the pair (A_T, C_T) is known, the output $y(k)$ of the system given by Eq. (8.2.1) can be written as:

$$\begin{aligned} y(k) &= C_T A_T^{k-j} x_T(j) + \left(\sum_{\tau=j}^{k-1} u(\tau)^T \otimes C_T A_T^{k-\tau-1} \right) \text{vec}(B_T) + u(k)^T \otimes \text{vec}(D) \\ &\quad + C_T \sum_{\tau=j}^{k-1} A_T^{k-\tau-1} f(\tau) + v(k) \end{aligned} \quad (8.6.3)$$

Let the matrix Φ_N be defined as in Eq. (8.5.7) and let the matrix Ψ_N be defined as:

$$\Psi_N := \begin{bmatrix} 0 \\ C_T f(0) \\ \vdots \\ C_T \sum_{\tau=0}^{N-1} A_T^{N-1} f(\tau) \end{bmatrix}$$

then we can accumulate the measurements $y(k)$ in the following over-determined set of equations,

$$Y_{1,N,1} = \Phi_N \theta + \Psi_N + V_{1,N,1} \quad (8.6.4)$$

Solving this over-determined set of equations in least-squares sense results in a *consistent* estimate of the parameter vector θ , since the columns of the matrix Φ_N (which are determined by the input sequence $u(k)$) and those of the matrices $\Psi_N, V_{1,N,1}$ are statistically independent.

8.6.3 Calculation of the System Matrices $[B_T D]$ under the Presence of Process and Measurement Noise: Closed-Loop

The approach outlined in Section 8.6.1 to compute a consistent estimate of the column space of the matrix Γ_s is still valid when the input and output sequences $u(k), y(k)$ are recorded in the closed-loop configuration, depicted in Figure 8.6.1, see Chou and Verhaegen (1997) [26].

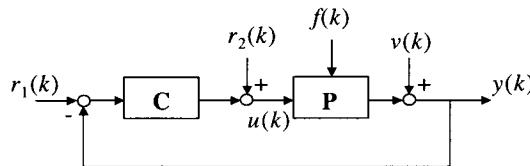


Figure 8.6.1 Block-schematic representation of a plant P in a closed-loop configuration

The calculation of the matrix pair $[B_T, D]$ and $x_T(0)$ via the solution of the over-determined equations (8.6.4) does *not*, however, give consistent estimates. A possible solution to this is to

compute a matrix Φ'_N as in Eq. (8.5.7) with the input $u(k)$ replaced by $r_2(k)$ or replaced by an estimate obtained from an estimate of the system from $[r_1(k)^T \ r_2(k)^T]^T$ to $u(k)$. This matrix is then used as an instrumental variable matrix to produce a consistent estimate by solving the equations:

$$\lim_{N \rightarrow \infty} \frac{1}{N} ((\Phi'_N)^T \Phi_N) \theta = \lim_{N \rightarrow \infty} \frac{1}{N} ((\Phi'_N)^T Y_{1,N,1})$$

The additional condition that the matrix $\lim_{N \rightarrow \infty} \frac{1}{N} ((\Phi'_N)^T \Phi_N)$ is invertible still needs to be checked.

Having an initial estimate of the matrices $[\hat{B}_T^0, \hat{D}^0]$ and $\hat{x}_T^0(0)$, an iterative procedure can be defined to possibly improve this estimate.

At the i -th (for $i \geq 1$) iteration of this procedure, assume we have the estimate $[\hat{B}_T^{i-1}, \hat{D}^{i-1}]$ and $\hat{x}_T^{i-1}(0)$ available in addition to the estimate of the pair $[A_T, C_T]$. Using the sequences $r_1(k)$ and $r_2(k)$ we compute an estimate of $u(k)$ and use this estimate to generate the matrix Φ'_N . This allows us to update the parameter vector θ .

8.7 A Simulation Study

Exercise 8.7.1. First write an M file that estimates a subspace model using the SMI Toolbox (Haverkamp and Verhaegen). Then perform simulations for the following process:

$$y(t) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7} u(t) + v(t)$$

with

$$v(t) = \frac{\alpha}{1 - 0.95q^{-1}} e(t)$$

The input signal is a GBN with average switch time of 8 samples. The test time is 500 samples. The disturbance power is 10% of that of the noise-free output. Run 50 open-loop simulations. Use each data set to estimate a 2nd order output error model, Box-Jenkins model and subspace model. Compare the following noise-free output-error criteria of the three models

$$V_{OE} = \frac{1}{500} \sum_{t=1}^{500} [G^o(z^{-1})u(t) - \hat{G}(z^{-1})u(t)]^2$$

and find out the accuracy of the subspace model. Also compare the speed of the three methods.

Answer. The M file for subspace method is called `moesp.m` which is shown below. The M file for the simulation is called `bj_oe_sm1.m` which is very similar to `bj_oe_asym.m` of Section 6.4 and it is not shown here. The noise-free output error criteria of the models are plotted in Figure

8.7.1. We find that the Box-Jenkins model is most accurate. The accuracy of the output error model is slightly higher than that of the subspace model. On the other hand, the times needed to perform 50 times estimations of the OE model, the Box-Jenkin model and the subspace model are 29 seconds, 20 seconds and 2 seconds respectively. This shows that the subspace method has a much higher speed of computation.

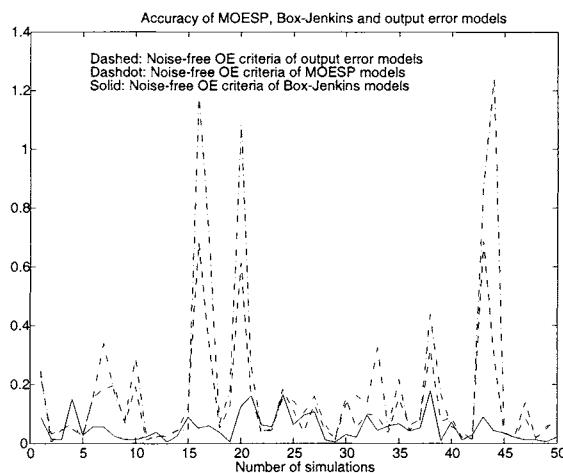


Figure 8.7.1 Comparing the accuracy of subspace, Box-Jenkins and output error models

```

function [A,B,C,D,x0] = moesp(U,Y,n);
%
%%% M file moesp.m
%
% This function estimate state space model using subapce method MOESP.
%
% Inputs arguments:
% U: Input vector or matrix (column wise)
% Y: Output vector or matrix (column wise)
% n: Order of the model
%
% Output arguments:
% [A,B,C,D]: State space model
% x0: Initial state
%
s = n+1;
[S,R]=dordpi(U,Y,s);

```

```
[A,C]=dmodpi(R,n);
[B,D]=dac2bd(A,C,U,Y);
x0=dinit(A,B,C,D,U,Y);
```

More simulations using the subspace method and comparisons will be performed in Section 10.4.

8.8 Summary of Extensions

Various extension of the MOESP class of subspace identification methods have been proposed or are under full development. An important focus of the current research activities is the extension to various classes of nonlinear systems. We list here:

1. Hammerstein systems (Verhaegen and Westwick, 1996 [114])
2. Wiener systems (Westwick and Verhaegen, 1996[119], Verhaegen, 1998 [112])
3. Wiener-Hammerstein systems
4. Bilinear systems (Verdult *et. al.*, 1998 [109])
5. Quadratic systems

The important rational under the current development is to transfer, as much as possible, the algorithmic strength characteristic for linear subspace identification schemes to non-linear systems. For making these tools easily accessible, a Matlab toolbox (referred to as the SMI Toolbox) is under development (Haverkamp and Verhaegen [55]).

Chapter 9

Nonlinear Process Identification

After more than three decades of research and development, linear system identification has reached its mature state. The main problem now is the transfer of technology to industries. Nonlinear system identification is more challenging and it has received less attention compared to linear system identification. From a control point of view, being able to identify nonlinear models will extend the working range of model based controllers.

There are several approaches to nonlinear system identification. One way is to use theoretically sound nonlinear functions and to develop identification schemes for these models. Identification using Volterra series, neural networks and nonlinear ARMAX models belong to this methodology. The advantage of this approach is the ability to obtain a global model of the underlying system. The main difficulty of the approach is the high cost in identification tests and computation. Another approach is to combine linear dynamic models with static or memoryless nonlinear functions. This type of models are called *block-oriented nonlinear models*. There are several advantages when using block oriented models: 1) low cost in identification tests; 2) low cost in identification and control computations and 3) it is easy to comprehend and to incorporate *a priori* process knowledge. The shortcoming of block-oriented models is the lack of global description of the system. The third approach is to construct global models by combining several local models which is called *operating regime based identification* or *multiple model approach*. The local models can be linear models or simple nonlinear models. The motivation of the operating regime based identification is to obtain the advantages of both first and second approaches.

In this chapter we will study the identification of block-oriented nonlinear models. Three types will be treated: Hammerstein model (Section 9.1), Wiener model (Section 9.2), and combined Hammerstein-Wiener model (Section 9.3).

9.1 Identification of Hammerstein Models

The Hammerstein model is a block-oriented nonlinear model with a static nonlinear function followed by a linear dynamic block. Although very simple, this model structure can represent and approximate many real life processes. Examples are: control systems with linear processes and nonlinear actuators and industrial processes with variant gain. The identification of Hammerstein models was perhaps first suggested by Narendra and Gallman (1966). They proposed an iterative scheme that approximately minimizes the output-error criterion. The iteration for the linear part and the nonlinear part are done separately, which has shown good convergence when the disturbance is white noise. Since then several other parametric model identification methods for this model class have been developed. Chang and Luus (1971) treated the nonlinear model as a multiple-input and single output linear model and used the equation-error method for parameter estimation; Hsia (1977) proposed using the generalized least-squares (GLS) method; Stoica and Söderström (1982) have used the instrumental variable (IV) method. Beyer and co-workers (1979) studied closed-loop identification of Hammerstein models using least-squares method, GLS method and maximum-likelihood method and applied the results to model the dynamic behavior of a blood pressure control system. Eskinat and co-workers (1991) studied the use of the Hammerstein model for the identification of industrial processes where they also compared the method of Narendra and Gallman (1966) with the output-error method based on a nonlinear search scheme. Most of the previous work concentrates on parameter estimation of the Hammerstein models and leaves test design, order selection and model validation to the user.

We will study parametric model identification of SISO Hammerstein models. The *asymptotic* (ASYM) method for linear model identification in Chapter 6 will be extended and used to solve the problem. We will address all four problems of identification. Simulation studies will be used to demonstrate the method. This section is based on Zhu (2000c)

9.1.1 Model Parametrization and Problem Statement

A nonlinear Hammerstein model in discrete-time is given in Figure 9.1.1. The input $u(t)$ is first scaled by the static nonlinear function $f(u)$ and then passes the linear time-invariant transfer function $G(q)$. It is assumed that 1) the nonlinear function $f(u)$ is continuous; 2) the linear block $G(q)$ is stable and 3) that the process output is disturbed by a stationary stochastic process $\{v(t)\}$.

For parametric model identification, it is necessary to parametrize the model in some way. One of the parametrizations for identification is to use polynomial function for the nonlinear block and the Box-Jenkins model for the linear and disturbance part:

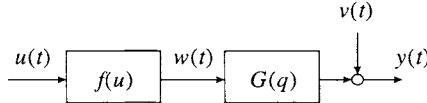


Figure 9.1.1 Hammerstein model

$$f[u(t)] = \beta_1 u(t) + \beta_2 u^2(t) + \dots + \beta_m u^m(t) \quad (9.1.1)$$

$$G(q) = \frac{b_0 + b_1 q^{-1} + \dots + b_n q^{-l}}{1 + a_1 q^{-1} + \dots + a_n q^{-l}} = \frac{B(q)}{A(q)} \quad (9.1.2)$$

$$v(t) = H(q)e(t) = \frac{1 + c_1 q^{-1} + \dots + c_n q^{-l}}{1 + d_1 q^{-1} + \dots + d_n q^{-l}} e(t) = \frac{C(q)}{D(q)} e(t) \quad (9.1.3)$$

Denote the parameter vector as

$$\theta = [\beta_1, \beta_2, \dots, \beta_m, a_1, \dots, a_l, b_0, b_1, \dots, b_l, c_1, \dots, c_n, d_1, \dots, d_l]^T$$

Assuming that $\beta_1 \neq 0$, then without loss of generality we can set $\beta_1 = 1$.

With these notations the process model can be written in a more compact form as

$$y(t) = \frac{B(q)}{A(q)} \sum_{k=1}^m \beta_k u^k(t) + \frac{C(q)}{D(q)} e(t) \quad (9.1.4)$$

It is known that polynomials have several drawbacks. They do not extrapolate well; high degree polynomials have oscillatory behavior and parameter estimation is often numerically ill-conditioned; see Lancaster and Šalkauskas (1986). Piecewise linear or linear spline functions were used by Pajunen (1992) and Wigren (1993) for Wiener model identification. The problems with polynomials do not occur with linear splines, but linear splines are not smooth meaning that their derivative are not continuous at their break points or knots. To overcome this last problem, we can use cubic splines to model the nonlinear block.

Let K denote a set of knots $\{u_1, u_2, \dots, u_m\}$ which are real numbers and satisfy

$$u_1 = u_{\min} < u_2 < u_3 < \dots < u_m = u_{\max}$$

A cubic spline function defined for all real numbers u is given as

$$w(u) = \sum_{k=2}^{m-1} \beta_k |u - u_k|^3 + \beta_m + \beta_{m+1} u + \beta_{m+2} u^2 + \beta_{m+3} u^3 \quad (9.1.5)$$

where $[\beta_2, \beta_3, \dots, \beta_{m+3}]$ are fixed real numbers, namely, the parameters to be estimated. Here m is called the number of knots which can be seen as the "degree" or "order" of the cubic splines. Note that the number of parameters of the cubic splines is $m + 2$. It is easy to verify that the first and second derivatives of the function are continuous and hence the function is smooth.

The identification problem is almost the same in using either polynomials or cubic splines. Therefore, polynomials will be used in the sequel for all the discussions.

Now, the *identification problem* for model (9.1.4) can be stated as follows:

- 1) **Test design.** Design an identification test using excitation signals so that the parameters of model (7.1.4) are identifiable. If possible, the test should be designed so that the identified model is most suitable for model application. Denote the collected input-output data as

$$\{u(1), y(t), u(2), y(2), \dots, u(N), y(N)\}$$

where N is the number of samples.

- 2) **Parameter estimation.** Determine the parameters of model (9.1.4) using the test data by minimizing the loss function

$$V_{PE} = \sum_{t=1}^N \varepsilon^2(t) \quad (9.1.6)$$

where

$$\varepsilon(t) = H^{-1}(q)\{y(t) - G(q)f[(u(t)]\} = \frac{D(q)}{C(q)}[y(t) - \frac{B(q)}{A(q)} \sum_{k=1}^m \beta_k u^k(t)] \quad (9.1.7)$$

- 3) **Order selection.** Determine the orders l and m so that the obtained model is the most accurate for control.
- 4) **Model validation.** Verify if the identified model is suitable for its application, namely, control. *If not, provide a remedy that should include redesigning the identification test.*

9.1.2 Test Design

For the identification of linear models, it is sufficient to consider only frequency content of the test signal. Therefore, often binary test signals are used for linear model identification. For the identification of Hammerstein models, binary test signals applied on the process input can only identify the nonlinear polynomial of degree two. Therefore, in order to uniquely determine the coefficients of the nonlinear polynomial with higher degrees, one needs to use multiple-level test

signals. This implies one should consider both the amplitude distribution and the frequency distribution of the test signals for the identifiability of the Hammerstein model.

Signal Amplitude Distribution.

It is easy to verify that to uniquely determine the nonlinear polynomial of degree m , the amplitude of the test input should have at least $m + 1$ levels. Many signals satisfy this requirement; examples are staircase signals, multiple level pseudo-random signals, multiple sinusoids, white noises and filtered white noises. When the number of signal levels is sufficiently high, the next issue is how to distribute the signal amplitude over its variation range. The amplitude distribution will influence the accuracy of the model when there is process disturbance and unmodeled dynamics. The general guideline is that the test input amplitude should be similar to, but richer than, the input signal during typical process operations; and use higher density in areas where high model accuracy is desired. A normal distribution of signal amplitude will put more weight on the area around the mean value of input at the cost of other areas; a uniformly distributed signal will treat the whole signal range equally. If there is not enough *a priori* knowledge about the process, it is recommended to use uniformly distributed test signals for identifying Hammerstein models. We call a multiple level signal uniformly distributed when the occurrence frequencies at all levels are equal.

Signal Power Spectrum

So far, we do not have an optimal spectrum for nonlinear systems. So the linear optimal design formulas in Chapter 6 will be borrowed here. Open-loop optimal spectrum of the process input is given as

$$\Phi_u^{opt}(\omega) = \mu \sqrt{\Phi_u^{con}(\omega)\Phi_v(\omega)} \quad (9.1.8)$$

where $\Phi_v(\omega)$ is the power spectrum of the disturbance $v(t)$ and $\Phi_u^{con}(\omega)$ is the power spectrum of the control input and μ is a constant adjusted so that the power of the input is constrained. The closed-loop optimal spectrum of the test signal at the setpoint is given as

$$\Phi_u^{opt}(\omega) \approx \mu \sqrt{\Phi_r(\omega)\Phi_v(\omega)} \quad (9.1.9)$$

where $\Phi_r(\omega)$ is the power spectrum of the reference signal and μ is a constant adjusted so that the power of the output is constrained. Although these results are based on some simplified assumptions, they are very instructive and can be used as guidelines.

Duration of Identification Tests

It is common sense that the identification test time should be sufficiently long so that the effects of unmeasured disturbance can be averaged out. A shorter test can be used for processes with low noise level and/or a simple model with a small number of parameters; a longer test time is necessary when the process noise level is high and/or many parameters need to be determined. Based on many simulation studies and industrial applications, it was recommended in Chapter 3 that a typical test for linear model identification in a very noisy environment will take 10 to 18 times the process settling time for a given set of process inputs; and up to 10 inputs can be

excited simultaneously. In general, a Hammerstein model has more parameters than a linear model for the same process. This fact may require a longer test time. On the other hand, for a given process, the signal-to-noise ratio of a test for nonlinear model identification is, in general, higher than that of a test for linear model identification, because the input amplitude range for the former is larger. It is, therefore, recommended that the test time for Hammerstein model identification can be the same as or slightly longer than that for linear model identification, namely, 16 to 25 times the process settling time. Note that this is only a recommendation for identification tests. For a given class of processes, the test time can be adjusted according to experience and model validation.

Recommended Tests

Based on the above discussions, the following test designs are proposed for Hammerstein model identification. Here, it is assumed that the input amplitude range is known from process operation experience and the test time is 16 - 25 times the (average) process settling time. These tests are not exclusive and many other tests can be proposed. Note that only one type of test is necessary for a given application.

- 1) *Staircase test.* The advantage of this type of signal is that process behavior (gain and time constants) in each area can be watched, which is helpful for understanding the dynamics of the process.
 - *The number of stairs.* This number should be greater than or equal to the highest degree of nonlinear polynomials to be identified.
 - *Width of stairs.* Denote $\tau = T_s/4$ (T_s is the 98% settling time). Then use stair width τ for 1/3 of the test period, stair width 2τ for 1/3 of the test period and stair width 3τ for 1/3 of the test period. It is recommended to mix the stair width when creating the test signal.
- 2) *Generalized multiple-level noise (GMN).* This is a multiple-level generalization of GBN of Tulleken (1990).
 - *Amplitude distribution.* As discussed before, it is preferred that the amplitude has a uniform distribution over the signal range. The signal amplitude can have discrete values in the signal range; it can also have a continuous distribution. In the first case, the number of levels should be greater than or equal to the degree of the nonlinear polynomial of to be identified.
 - *Average switch time.* Denote T_{sw} as the average switch time of the signal. Then set $T_{sw} = T_s/3$ where T_s is the 98% settling time.
- 3) **Filtered white uniform noise.** The advantage of this kind of signals is the flexibility in shaping the signal spectra. Any spectrum can be realized by using a proper filter. The previous two signal types do not have this freedom. Often a first order low-pass filter will

be suitable for a good design. It is easy to design the shaping filter in continuous-time. Denote the filter as

$$F(s) = \frac{K}{\tau_f s + 1}$$

where K is the filter gain and τ_f is the time constant. Denote T_s as the process 98% settling time, then, using the same design rule as for linear model identification will lead to

$$\tau_f = T_s/3$$

The filter gain K can be adjusted so that the designed signal spans the whole amplitude range of process input. Note that this kind of signal is continuous and hence it the nonlinear polynomial will be identifiable.

When a closed-loop identification test is used, it is recommended to apply the test signals on the setpoint of the system. Note that signals with binary amplitude such as GBN can also be used in a closed-loop test, because the process input will become a continuous signal due to control action.

9.1.3 Parameter Estimation

The prediction error defined in (9.1.7) is highly nonlinear in the model parameters. Direct minimization of the loss function (9.1.6) can be costly and may run into numerical problems. It is highly desirable to reduce this complexity by looking for some simpler numerical schemes, such as the relaxation algorithm. Eskinat and co-workers have shown that the simple method of Narendra and Gallman (1966) is very robust and produced better results than the output error method using numerical optimization routine. The essence of the method of Narendra-Gallman is to simplify the optimization by separating the estimation of the linear and the nonlinear part of the model at each iteration, which is called the relaxation algorithm.

It is known that any linear prediction error model structure can be approximated arbitrarily well by an ARX or equation error model with sufficiently high order; see, e.g., Ljung (1987). Based on this observation, let us approximate the linear Box-Jenkins model by a high order ARX model

$$y(t) = \frac{B^n(q)}{A^n(q)} \sum_{k=1}^m \beta_k u^k(t) + \frac{1}{A^n(q)} e(t) \quad (9.1.10)$$

or

$$A^n(q)y(t) = B^n(q) \sum_{k=1}^m \beta_k u^k(t) + e(t) \quad (9.1.11)$$

where $A^n(q)$ and $B^n(q)$ are polynomials of the delay operator q^{-1} and n is the order of ARX model which is much higher than the order l of the corresponding Box-Jenkins model.

The loss function for parameter estimation of model (7.1.11) becomes

$$V_{ARX} = \sum_{t=1}^N \varepsilon^2(t) \quad (9.1.12)$$

where

$$\varepsilon(t) = A^n(q)y(t) - B^n(q) \sum_{k=1}^m \beta_k u^k(t) \quad (9.1.13)$$

Now the error $\varepsilon(t)$ is linear in the parameters of $A^n(q)$ and bilinear in the parameters of $B^n(q)$ and $f(u)$. This is much simpler than the original error form in (9.1.6). The property of bilinear in the parameters can be exploited and the following relaxation algorithm can be used for parameter estimation.

Estimating the Hammerstein Model with High-Order ARX Model

Initialization. One way to initialize the iteration is to set $B^n(q) = 1$ and estimate $A^n(q)$ and $f(u)$ using linear least-squares (linear regression). Another way is to set $f(u) = u$ and estimate $A^n(q)$ and $B^n(q)$ using linear least-squares.

Iteration. Denote $\hat{A}_{(i)}^n(q)$, $\hat{B}_{(i)}^n(q)$ and $\hat{f}_{(i)}(u) = \hat{\beta}_{1(i)}u + \hat{\beta}_{2(i)}u^2 + \dots + \hat{\beta}_{m(i)}u^m$ as the estimates from iteration i , then

- 1) Calculate $\hat{A}_{(i+1,1)}^n(q)$ and $\hat{B}_{(i+1)}^n(q)$ parameters for fixed $\hat{f}_{(i)}(u)$ by minimizing the following loss function

$$V_{ARX}^{(i+1,1)} = \sum_{t=1}^N \{ A_{(i+1,1)}^n(q)y(t) - B_{(i+1)}^n(q)[\sum_{k=1}^m \hat{\beta}_{k(i)} u^k(t)] \}^2 \quad (9.1.14)$$

- 2) Calculate $\hat{A}_{(i+1,2)}^n(q)$ and $[\hat{\beta}_{1(i+1)}, \hat{\beta}_{2(i+1)}, \dots, \hat{\beta}_{m(i+1)}]$ parameters for fixed $\hat{B}_{(i+1)}^n(q)$ by minimizing the following loss function

$$V_{ARX}^{(i+1,2)} = \sum_{t=1}^N \{ A_{(i+1,2)}^n(q)y(t) - \sum_{k=1}^m \beta_{k(i+1)} [\hat{B}_{(i+1)}^n(q) u^k(t)] \}^2 \quad (9.1.15)$$

Go back to 1). Stop when convergence occurs.

Both steps are linear least-squares problems. Note that $A^n(q)$ is updated twice in each iteration, because the error is linear in its parameters. To some extent, the relaxation algorithm looks like the Steiglitz-McBride method studied in Chapter 5. One knows that the Steiglitz-McBride method is an approximation of output error method and it is, in general, not a minimization algorithm. However, the relaxation algorithm here is a minimization algorithm as stated as follows.

Result 9.1.1. Assume that the input is persistently exciting with order greater than $2n$ and the number of its amplitude levels is greater than m . Then the relaxation algorithm of (9.1.14) and (9.1.15) minimizes the criterion in (9.1.12).

Proof. It is easy to verify that the relaxation algorithm is a special case of the so-called *separable least-squares* problem studied in Golub and Pereyra (1973). Then the equivalence of the relaxation algorithm to the minimization of the original criterion (9.1.12) follows directly from Theorem 2.1 of Golub and Pereyra (1973).

Model Reduction for the Linear ARX Model

The obtained Hammerstein model with a high order ARX linear part is unbiased, provided that the process can be modelled by a Hammerstein model, but the variance error of the model is high due to its high order. Model reduction can be used to reduce the variance error. This follows the same idea as in developing the ASYM method in Chapters 6 and 7.

Assume that the output of the nonlinear function $w(t)$ is known exactly, order n of the ARX model is sufficiently high and the signal $w(t)$ is sufficiently rich (persistently exciting), then it can be shown (see Ljung, 1985 and Section 6.1) that the estimated frequency response of the high order model is unbiased and its error follows a Gaussian distribution with variance given as

$$\text{var}[\hat{G}^n(e^{i\omega})] \approx \frac{n}{N} \frac{\Phi_v(\omega)R}{\Phi_w(\omega)R - |\Phi_{we}(\omega)|^2} \quad (9.1.16)$$

where $\hat{G}^n(e^{i\omega})$ is the frequency response of the high order ARX model, $G^o(e^{i\omega})$ is that of the true model, n is the order of the ARX model, N is the number of data points, $\Phi_v(\omega)$ is the power spectrum of output disturbance, R is the variance of white noise $\{e(t)\}$ that generated the output disturbance, $\Phi_w(\omega)$ is the power spectrum of signal $w(t)$ and $\Phi_{we}(\omega)$ is the cross-spectrum between the white noise $\{e(t)\}$ and $w(t)$. Note that this result holds for both open-loop and closed-loop tests.

Then based on (9.1.16), one can apply maximum likelihood model reduction by minimizing (see Section 6.3)

$$V = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \hat{G}^n(e^{i\omega}) - G^l(e^{i\omega}) \right|^2 \frac{\Phi_w(\omega)R - |\Phi_{we}(\omega)|^2}{\Phi_v(\omega)R} d\omega \quad (9.1.17)$$

where $G^l(e^{i\omega})$ is the frequency response of the reduced model to be calculated.

The minimization of the loss function (9.1.17) needs nonlinear search algorithm. It can be done directly in the frequency domain; it can also be converted to a time-domain parameter estimation problem; see Section 6.3. Note that estimated signals of $w(t)$, $v(t)$, and $e(t)$ are needed in implementing the model reduction. If a disturbance model is needed in control, it can be obtained by a model reduction on $1/\hat{A}^n(q)$ using the same idea as for the process model.

9.1.4 Order Selection

Three orders need to be determined here: the order of the ARX model n , the degree of the nonlinear polynomial m and the order of the reduced linear model l . Experience has shown that the order of ARX model should be greater than 10 for most applications. Often the end result is not sensitive to the high order n . The selection of m and l will be discussed in the following subsections.

In Section 5.6, the final output error (FOE) criterion was proposed for order selection. The FOE is control relevant and does not need validation data. Hence it will be used for order selection here. The FOE for the nonlinear polynomial or cubic spline is

$$FOE(m) = \frac{N+d}{N-d} V_{oe} = \frac{N+(2n+m)}{N-(2n+m)} \frac{1}{N} \sum_{t=1}^N [y(t) - \hat{G}^n(q) \sum_{k=1}^m \hat{\beta}_k u^k(t)]^2 \quad (9.1.18)$$

where V_{oe} is the output-error (or simulation error) criterion evaluated on the estimation data set. Here $\frac{N+d}{N-d}$ is again used to correct for the overfit and V_{oe} is used to reflect the model purpose, namely, control.

The FOE for order selection for the reduced linear model $\hat{G}^l(q)$, is given as

$$FOE(l) = \frac{N+d}{N-d} V_{oe} = \frac{N+(2l+m)}{N-(2l+m)} \frac{1}{N} \sum_{t=1}^N [y(t) - \hat{G}^l(q) \sum_{k=1}^m \hat{\beta}_k u^k(t)]^2. \quad (9.1.19)$$

9.1.5 Model Validation

Here we try to solve the model validation problem for the linear part only, by assuming that the nonlinear function $f(u)$ and signal $w(t)$ are known perfectly. Then according to the asymptotic theory, the high order ARX model is unbiased, its frequency response follows a Gaussian distribution with variance given by (9.1.16). Therefore, a 3σ upper bound of the errors can be defined for the high order model

$$|\hat{G}^n(e^{i\omega}) - G^o(e^{i\omega})| \leq \bar{\Delta}(\omega) = 3 \sqrt{\frac{n}{N} \frac{\Phi_v(\omega) R}{\Phi_w(\omega) R - |\Phi_{we}(\omega)|^2}} \quad \text{w.p. 99.9\%} \quad (9.1.20)$$

Again estimated signals of $w(t)$, $v(t)$, and $e(t)$ are needed to calculate various spectra. The same upper bound will also be used for the reduced model $\hat{G}^l(e^{i\omega})$, because model reduction will, in general, reduce the model error. The use of the upper bound for validation is the same as that for linear processes.

This approach is theoretically not completely sound, because the errors of the nonlinear part are not accounted for. However, when used with caution, it can be effective in applications. Due to the errors in nonlinear function estimation, one may suggest that the errors of the linear part are larger than those given by the linear asymptotic theory. Based on this reasoning, one can use 4σ bound instead of 3σ bound.

To extend the asymptotic theory to the Hammerstein model is still an academic challenge.

9.1.6 Simulation Studies

In this section simulations will be used to verify parameter estimation of the proposed method and, if possible, compare the results with other methods proposed in the literature.

Example 9.1.1. In this example we will study the performance of parameter estimation.

Given a simulated process:

$$y(t) = \frac{1 + .5q^{-1} + .4q^{-2} + 2q^{-3}}{1 + .5q^{-1} - .4q^{-2} - .26q^{-3} - .03q^{-4}} [u(t) + 3u^2(t) + 2u^3(t)] + v(t)$$

where

$$v(t) = \frac{\alpha}{1 - 0.9q^{-1}} e(t)$$

$e(t)$ is a white Gaussian noise and α is a constant that is used to obtain a signal-to-noise ratio of 10% in power at the output. The noise-free part of this example is the same as the first example in Narendra and Gallman (1966), the difference is that we use colored noise disturbance which is more realistic and also more difficult to identify than the white noise case. Open-loop simulations are carried out.

The input signal used is a GMN signal with an average switch time of 10 samples in the range of $[-1.5, 1.5]$. The number of samples is 700. In total 20 open-loop simulations are carried out. The ASYM parameter estimation proposed in this work is studied. A method that is similar to the iteration scheme of Narendra and Gallman (1966) is also used in estimation where the linear model is estimated using a true output error method. The results are given in the following table where each estimated parameter is given by the mean values and the standard deviations calculated from the 20 estimates. The original algorithm of Narendra and Gallman (1966) does not converge for the simulated data. The reason is that the Steiglitz-McBride method they use for the linear part does not perform well when $v(t)$ is colored noise; see Stoica and Söderström (1981) and Section 5.1.

Parameters	True values	Output-error model	ASYM model
a_1	0.5000	0.4928 ± 0.0309	0.4979 ± 0.0205
a_2	-0.4000	-0.3990 ± 0.0314	-0.3945 ± 0.0171
a_3	-0.2600	-0.2535 ± 0.0258	-0.2543 ± 0.0207
a_4	-0.0300	-0.0263 ± 0.0332	-0.0285 ± 0.0168
b_o	1.0000	0.9532 ± 0.4663	0.9902 ± 0.1499
b_1	0.5000	0.4454 ± 0.2004	0.4768 ± 0.1042
b_2	0.4000	0.3573 ± 0.1464	0.3899 ± 0.0757
b_3	2.0000	1.9144 ± 0.8907	1.9921 ± 0.2655
β_1	1.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
β_2	3.0000	4.3589 ± 3.2823	3.0906 ± 0.4939
β_3	2.0000	3.2902 ± 2.9771	2.0854 ± 0.4236

One can see that the variance of the ASYM model is smaller than that of the output error model. This is because, when the output disturbance is colored noise, the ASYM model is minimum variance estimate while the output-error model is not. The simulation accuracy of the ASYM model is higher than that of the output-error model. The average *noise-free output-error criteria* of the 20 estimates are:

- 0.33% of the variance of the output for the output-error model; and
- 0.15% of the variance of the output for the ASYM model.

For this example, both methods converged in 3 iterations.

Example 9.1.2. In this simulation we will check the approximation of the nonlinear function using both polynomials and cubic splines. Simulated process:

$$y(t) = \frac{q^{-1} + .5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} \cdot f[u(t)]$$

with

$$\begin{aligned} f[u(t)] &= u(t) & -1 \leq u(t) \leq 1 \\ &= \text{sign}[u(t)] * 1 & |u(t)| \geq 1 \end{aligned}$$

GMN input signal with $T_{sw} = 10$, the range of $u(t)$ is $[-2.5, 2.5]$, open-loop noise-free simulation, 1000 samples are simulated.

Both polynomials and cubic splines with the same number of parameters are used to fit the nonlinear function (saturation). The result is better when using cubic splines. The relaxation

algorithms converged in 3 iterations. The fit of the linear part is nearly perfect (not shown). The fit of the nonlinear function using cubic splines is shown in Figure 7.1. One can say that the fit is very good and the extrapolation is reasonable. In this example, the polynomial function can not extrapolate at all.

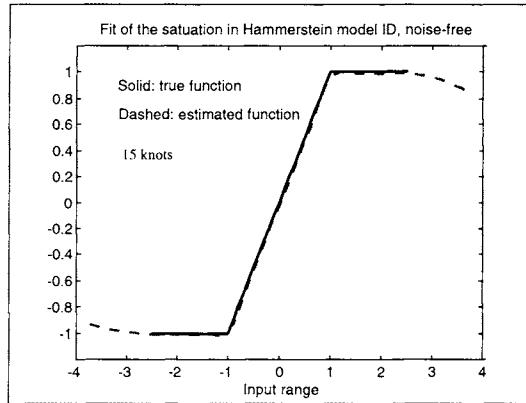


Figure 9.1.2 Fit of the nonlinear function using cubic splines

Example 9.1.3. In this simulation we will test the performance of parameter estimation when the data are collected from closed-loop tests with a high level of disturbance. Given a simulated process:

$$y(t) = \frac{q^{-1} + .5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} \cdot f[u(t)] + v(t)$$

with

$$\begin{aligned} f[u(t)] &= u(t) & -0.5 \leq u(t) \leq 0.5 \\ &= \text{sign}[u(t)]0.5 & |u(t)| \geq 0.5 \end{aligned}$$

and

$$v(t) = \frac{\alpha}{1 - 0.9q^{-1}} e(t)$$

Disturbance level is 13% (in power) at $y(t)$, GMN at setpoint, $T_{sw} = 10$, the range of u is $[-1.58, 2.33]$, closed-loop test using a PI controller, 1000 samples are simulated.

Cubic splines are used for fitting the nonlinear function. Again the relaxation algorithm converged in 3 iterations. The results are shown in Figures 9.1.3 and 9.1.4, where the gain has

been corrected (redistributed between linear and nonlinear blocks). The result is satisfactory considering the high level of disturbance and the closed-loop test does not impose any difficulty in parameter estimation. Note however, when the level of disturbance is high, extrapolation is difficult using cubic splines. Process knowledge is needed for extrapolating the nonlinear function.

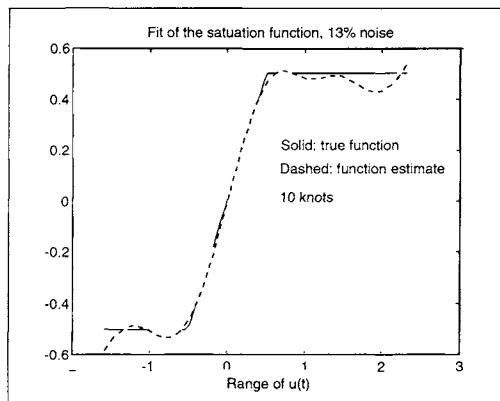


Figure 9.1.3 Fit of the nonlinear function

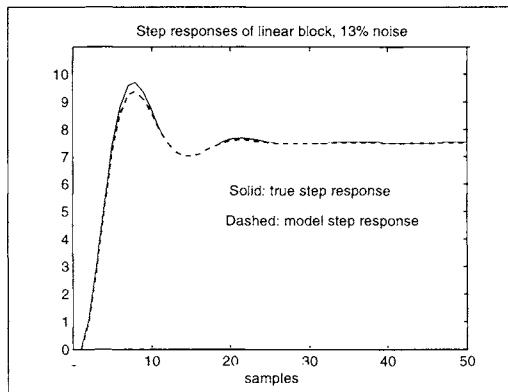


Figure 9.1.4 Fit of the linear block

9.2 Identification of Wiener Models

A Wiener model is a nonlinear model with a linear dynamic block followed by a static nonlinear function. Identification of Hammerstein models is easier than that of Wiener models, because the nonlinearity of the equations with respect to model parameters is more severe for Wiener models than for Hammerstein models. Many researchers have studied the parametric identification of the Hammerstein model. The identification of Wiener models, especially parametric model identification, has received less attention. Some work has been done on the nonparametric identification of Wiener models; see, e.g., Billings and Fakhouri (1982) and Griblicki (1992). The nonparametric method is rather restrictive for use in real life processes, because very specific test signals are needed and model accuracy is low due to high variance error. In parametric estimation of Wiener models, Verhaegen (1998) combines the state space model of the linear part and a parametric nonlinear function, then uses numerical optimization in several steps to estimate model parameters; Wigren (1993) proposed a recursive algorithm where the nonlinear function is approximated with a piecewise linear function. This section is based on Zhu (1999).

9.2.1 Model Parametrization and Problem Statement

A nonlinear Wiener model in discrete-time is given in Figure 9.2.1. The input $u(t)$ first passes the linear time-invariant transfer function $G(q)$ and then the output signal is scaled by the static nonlinear function $f(w)$. It is assumed that: 1) the nonlinear function $f(\cdot)$ is continuous, invertible and monotone (this assumption is more restrictive than that of the Hammerstein model); 2) the output of the linear part is disturbed by a stationary stochastic process $\{v(t)\}$.

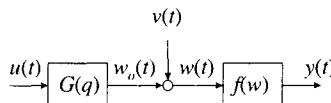


Figure 9.2.1 Wiener model

The model is given in the following form:

$$y(t) = f[G(q)u(t) + v(t)] \quad (9.2.1)$$

Note that here we have introduced the disturbance term $v(t)$ before the nonlinear block which is different from the normal assumption that a disturbance acts at the process output. This model implies *asymmetrical* output disturbance: the output disturbance is large when the process gain is large and is small when the gain is small. This assumption is more realistic from a process operation point of view: a larger gain implies that the process is sensitive and hence the disturbance will have strong influence at its output; a small gain means that the process is

insensitive and the disturbance will have little influence at its output. Greblicki (1992) uses a similar block diagram for nonparametric identification where the disturbance is assumed to be white noise and independent of the input signal, which is more restrictive than our assumption.

The measurement noise cannot be modelled properly when one puts the disturbance term before the linear block. In practice the influence of process disturbance is often greater than that of the measurement noise. The influence of measurement noise will be studied in simulations later.

For model parametrization, again, the linear part with disturbance will be parametrized using the Box-Jenkins model:

$$w(t) = G(q)u(t) + H(q)e(t) \quad (9.2.2)$$

where

$$\begin{aligned} G(q) &= \frac{B(q)}{A(q)} = \frac{b_0 + b_1q^{-1} + \dots + b_nq^{-n}}{1 + a_1q^{-1} + \dots + a_nq^{-n}} \\ H(q) &= \frac{C(q)}{D(q)} = \frac{1 + c_1q^{-1} + \dots + c_nq^{-n}}{1 + d_1q^{-1} + \dots + d_nq^{-n}} \end{aligned}$$

For the nonlinear block, as motivated in the previous section, we will use cubic splines. Let K denote a set of knots $\{w_1, w_2, \dots, w_m\}$ which are real numbers and satisfy

$$w_1 = w_{\min} < w_2 < w_3 < \dots < w_m = w_{\max}$$

A cubic spline function defined for all real numbers w is given as

$$y(w) = \sum_{k=2}^{m-1} \alpha_k |w - w_k|^3 + \alpha_m + \alpha_{m+1}w + \alpha_{m+2}w^2 + \alpha_{m+3}w^3 \quad (9.2.3)$$

where $[\alpha_2, \alpha_3, \dots, \alpha_{m+3}]$ are fixed real numbers, namely, the parameters to be estimated. Note that the signal $w(t)$ can not be measured, hence an arbitrary gain may be distributed between the linear block and the nonlinear block. This means that the parameters of the model cannot be uniquely determined without introducing further constraint. A solution to this problem is to fix the gain of the nonlinear block. For example one can let $\alpha_{m+1} = 1$.

Now, the *identification problem* for the Wiener model can be stated as follows:

- 1) **Test design.** Design an identification test using excitation signals so that the parameters of the model in (9.2.1) - (9.2.3) are identifiable. If possible, the test should be designed so that the identified model is most suitable for control application.
- 2) **Parameter estimation.** Determine the parameters of the model using the input-output data from the test by minimizing the loss function

$$V = \sum_{t=1}^N \varepsilon^2(t) \quad (9.2.4)$$

where

$$\varepsilon(t) = H^{-1}(q)[f^{-1}(y(t)) - G(q)u(t)] = \frac{D(q)}{C(q)}[f^{-1}(y(t)) - \frac{B(q)}{A(q)}u(t)] \quad (9.2.5)$$

- 3) **Order selection.** Determine the orders l and m so that the obtained model is the most accurate for control.
- 4) **Model validation.** Verify if the identified model is suitable for control. *If not, provide a remedy that should include redesigning the identification test.*

9.2.2 Test Design

For Wiener model identification, binary test signals applied on the process input can also be used. However, in order to obtain an accurate estimate of the nonlinear function, multiple-level test signals are preferred. This implies that one should consider both the amplitude distribution and the frequency distribution of the test signals for the identifiability of the Wiener model. Note that the nonlinear function of a Wiener model is at the output, it is more effective to influence the output amplitude distribution in the test design. This can be realized using a closed-loop test and applying the test signal at the setpoint of the controlled system. If a closed-loop test is not feasible, then the desired output amplitude distribution can be realized indirectly in an open-loop test.

Test signal amplitude distribution and power spectrum, test time and recommended test signals are the same as for the Hammerstein model as described in the previous section.

9.2.3 Parameter Estimation

The error defined in (9.2.5) is highly nonlinear in the model parameters. Thus, direct minimization of the loss function (9.2.4) can be costly and may run into numerical problems. Based on the same reasoning as for the Hammerstein model, let us approximate the linear part of the Wiener model with the Box-Jenkins structure in (9.2.2) using a high order ARX model

$$w(t) = \frac{B^n(q)}{A^n(q)}u(t) + \frac{1}{A^n(q)}e(t) \quad (9.2.6)$$

Substitute $w(t)$ in (9.2.1) using ARX model, take the inverse of the nonlinear function on both sides and then multiply them by $A^n(q)$. It follows that

$$A^n(q)f^{-1}(y(t)) = B^n(q)u(t) + e(t) \quad (9.2.7)$$

Then, the loss function for parameter estimation for model (9.2.7) becomes

$$V_{ARX} = \sum_{t=1}^N \varepsilon^2(t) \quad (9.2.8)$$

where

$$\varepsilon(t) = A^n(q)f^{-1}(y(t)) - B^n(q)u(t) \quad (9.2.9)$$

Now let us parametrize the inverse function using a cubic spline function in the similar way as for $f(w)$

$$w = \sum_{k=2}^{m_2-1} \gamma_k |y - y_k|^3 + \gamma_{m_2} + \gamma_{m_2+1}y + \gamma_{m_2+2}y^2 + \gamma_{m_2+3}y^3$$

where m_2 is the number of knots and can also be called the degree of the inverse function. Then equation (9.2.9) becomes

$$\varepsilon(t) = A^n(q) \left[\sum_{k=2}^{m_2-1} \gamma_k |y(t) - y_k|^3 + \gamma_{m_2} + \dots + \gamma_{m_2+3}y(t)^3 \right] - B^n(q)u(t) \quad (9.2.10)$$

Note that the error $\varepsilon(t)$ is linear in the parameters of $B^n(q)$ and bilinear in the parameters of $A^n(q)$ and $f^{-1}(y)$. This is much simpler than the original error form. Therefore, one can use the following relaxation algorithm for the estimation of the model.

The use of the nonlinear inverse for the Wiener model identification is not new in literature. Greblicki (1992) used the nonlinear inverse for nonparametric identification of the Wiener model.

Estimating Nonlinear Inverse and High-Order ARX Model

Initialization.

Method 1. Set $A^n(q) = 1$ and estimate $B^n(q)$ and $f^{-1}(y)$ sequentially using the linear least-squares.

Method 2. Set $f^{-1}(y) = y$ and estimate $A^n(q)$ and $B^n(q)$ using the linear least-squares.

Iteration. Denote $\hat{A}_{(i)}^n(q)$, $\hat{B}_{(i)}^n(q)$ and $\hat{f}_{(i)}^{-1}(y(t))$ as the estimates from iteration i , then

- 1) Calculate $\hat{A}_{(i+1)}^n(q)$ and $\hat{B}_{(i+1)}^n(q)$ for fixed $\hat{f}_{(i)}^{-1}(y(t))$ by minimizing

$$V_{ARX}^{(i+1,1)} = \sum_{t=1}^N \{ \hat{A}_{(i+1)}^n(q) \hat{f}_{(i)}^{-1}(y(t)) - \hat{B}_{(i+1)}^n(q)u(t) \}^2 \quad (9.2.11)$$

- 2) Calculate $[\hat{\gamma}_{2(i+1)}, \hat{\gamma}_{3(i+1)}, \dots, \hat{\gamma}_{m_2+3(i+1)}]$ for fixed $\hat{A}_{(i+1)}^n(q)$ and $\hat{B}_{(i+1)}^n(q)$ by minimizing

$$V_{ARX}^{(i+1,2)} = \sum_{t=1}^N \{ \hat{A}_{(i+1)}^n(q) f_{(i+1)}^{-1}(y(t)) - \hat{B}_{(i+1)}^n(q)u(t) \}^2 \quad (9.2.12)$$

Go back to 1). Stop when convergence occurs.

Note that both steps are linear least-squares problems which are numerically simple and reliable. One can also update $B^n(q)$ twice at each iteration as in (9.1.14) and (9.1.15), because the error is linear in its parameters.

The reader can see that the relaxation algorithm is numerically the same as that for the Hammerstein model. Therefore, Result 9.1.1. applies here as well. This means that the iteration minimizes the loss function (9.2.8) if it converges.

Calculate the Nonlinear Function from its Inverse

The determination of the nonlinear function $f(w)$ from the estimate of its inverse $\hat{f}^{-1}(y)$ is an approximation problem. This can be done using a linear least-squares estimate.

The number of knots or degree m of the cubic spline function can be determined by visual inspection of the fit of the nonlinear function where process knowledge can be used to help the selection. The process knowledge can also be used to extrapolate the nonlinear function at this step. In general, degree m can be either higher or lower than the degree m_2 . The selection of m_2 is more complicated due to the involvement of the disturbance and will be discussed later.

Model Reduction for the Linear ARX Model

The idea of model reduction for the Hammerstein model will also be used here. Assume that the nonlinear inverse is known exactly, order n of the ARX model is sufficiently high and the input signal $u(t)$ is persistently exciting, then it has been shown in Chapter 6 that the estimated frequency response of the high order model is unbiased and its error follows a Gaussian distribution with variance given as

$$\text{var}[\hat{G}^n(e^{i\omega})] \approx \frac{n}{N} \frac{\Phi_v(\omega)R}{\Phi_u(\omega)R - |\Phi_{ue}(\omega)|^2} \quad (9.2.13)$$

where $\hat{G}^n(e^{i\omega})$ is the frequency response of the estimated high order ARX model, n is the order of the ARX model, N is the number of data points, $\Phi_v(\omega)$ is the power spectrum of output disturbance, R is the variance of white noise $\{e(t)\}$ that generated disturbance $\{v(t)\}$, $\Phi_u(\omega)$ is the power spectrum of input $u(t)$ and $\Phi_{ue}(\omega)$ is the cross-spectrum between the white noise $\{e(t)\}$ and $u(t)$ (due to feedback).

Then using the same reasoning as for the ASYM and for the Hammerstein model, the reduced model can be calculated by minimizing the following likelihood function

$$V = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \hat{G}^n(e^{i\omega}) - G^l(e^{i\omega}) \right|^2 \frac{\Phi_u(\omega)R - |\Phi_{ue}(\omega)|^2}{\Phi_v(\omega)R} d\omega \quad (9.2.14)$$

where $G^l(e^{i\omega})$ is the frequency response of the reduced model to be calculated.

The minimization of the loss function (9.2.14) needs a nonlinear search algorithm. Note that estimated signals of $v(t)$ and $e(t)$ are needed in implementing the model reduction which, in turn, needs the estimates of $w^o(t)$ and $w(t)$.

If a disturbance model is needed in control, it can be obtained by a model reduction on $1/\hat{A}^n(q)$ using the same idea as for the process model.

9.2.4 Order Selection

Order Selection for the Nonlinear Inverse

Here, the degree m_2 of the nonlinear inverse will be determined during the parameter estimation with a high order ARX model. We will propose the following *final output-error criterion* (FOE)

$$FOE_y(m_2) = \frac{N+d}{N-d} V_{yoe} = \frac{N+(2n+m_2)}{N-(2n+m_2)} \frac{1}{N} \sum_{t=1}^N \{y(t) - \hat{f}[\hat{G}^n(q)u(t)]\}^2 \quad (9.2.15)$$

where V_{yoe} is the output-error (or simulation error) criterion evaluated on the estimation data set. Here $\frac{N+d}{N-d}$ is again used to correct for the overfit and V_{yoe} is used to reflect the model use in control. Evaluating $FOE_y(m_2)$ needs the estimate of nonlinear function $f(w)$. This can be avoided by using an equivalent criterion

$$FOE_w(m_2) = \frac{N+d}{N-d} V_{woc} = \frac{N+(2n+m_2)}{N-(2n+m_2)} \frac{1}{N} \sum_{t=1}^N [\hat{f}^{-1}(y(t)) - \hat{G}^n(q)u(t)]^2 \quad (9.2.16)$$

where V_{woc} is the output-error criterion for signal $w(t)$ before the nonlinear block evaluated on the estimation data set. It is easy to see that

$$FOE_y(m_2) = \hat{f}[FOE_w(m_2)]$$

provided that

$$\hat{f}[\hat{f}^{-1}(y(t))] = y(t) \quad (9.2.17)$$

Now, because the nonlinear function is monotone, the minimum of $FOE_w(m_2)$ implies the minimum of $FOE_y(m_2)$.

Order Selection for the Reduced Linear Model

Again FOE is used for order selection for the reduced linear model $\hat{G}^l(q)$. Two equivalent criteria can be used. When using the criterion for $y(t)$, one gets

$$FOE_y(l) = \frac{N+d}{N-d} V_{yoe} = \frac{N+(2l+m_2)}{N-(2l+m_2)} \frac{1}{N} \sum_{t=1}^N \{y(t) - \hat{f}[\hat{G}^n(q)u(t)]\}^2. \quad (9.2.18)$$

and when using the criterion for $w(t)$, one gets

$$FOE_w(l) = \frac{N+d}{N-d} V_{woe} = \frac{N+(2l+m_2)}{N-(2l+m_2)} \frac{1}{N} \sum_{t=1}^N [\hat{f}^{-1}(y(t)) - \hat{G}^n(q)u(t)]^2 \quad (9.2.19)$$

9.2.5 Model Validation

The model validation problem for linear part is outlined here, that follows the same idea as for the Hammerstein model case. Assume that the nonlinear function and hence $w(t)$ is known perfectly. Then according to the asymptotic theory, the high order ARX model is unbiased, its frequency response follows a Gaussian distribution with variance given by (9.2.13). Therefore, a 3σ upper bound of the errors defined for the high order model is

$$|\hat{G}^n(e^{i\omega}) - G^o(e^{i\omega})| \leq \bar{\Delta}(\omega) = 3 \sqrt{\frac{n}{N} \frac{\Phi_v(\omega)R}{\Phi_u(\omega)R - |\Phi_{ue}(\omega)|^2}} \quad \text{w.p. 99.9\%} \quad (9.2.20)$$

Again estimated signals $v(t)$, and $e(t)$ are needed to calculate various spectra. The same upper bound will also be used for the reduced model $\hat{G}'(e^{i\omega})$.

Due to the errors in nonlinear function estimation, one may expect that the errors of the linear part is larger than those given by the linear asymptotic theory. Hence, one can use 4σ bound instead.

9.2.6 Simulation Studies

First the noise-free data is used to test the correctness of the identification scheme. Then open-loop and closed-loop test data is used to show the effectiveness of the proposed method. Also the effect of measurement noise is tested.

Example 9.2.1. In this example we will study the performance of parameter estimation in the open-loop case. Given a simulated process:

$$w(t) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} u(t) + v(t)$$

$$y(t) = f(w(t)) = w(t) + 2w^3(t)$$

and

$$v(t) = \frac{\alpha}{1 - 0.9q^{-1}} e(t)$$

where $\{e(t)\}$ is zero mean white Gaussian noise.

The input signal used is a GMN signal with an average switch time of 8 samples in the range of $[0.0, 0.4]$. The number of samples is 500. The range of $w(t)$ is $[-0.5, 3.5]$. The disturbance is 10% (in power) at the output, 8.5% at $w(t)$.

The fit of nonlinear function is shown in Figure 9.2.2. The fit of the step response of the linear block (after gain correction) is shown in Figure 9.2.3. The result is very satisfactory when considering the relatively high level of disturbance.

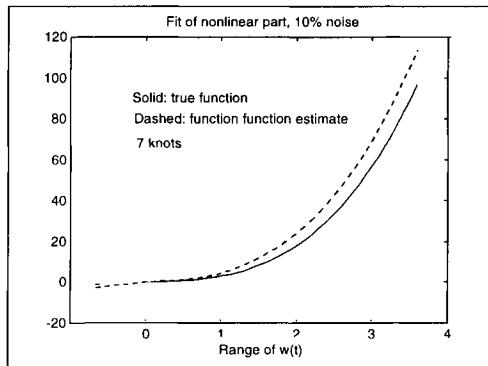


Figure 9.2.2 Fit of nonlinear function, simulation 1-2.

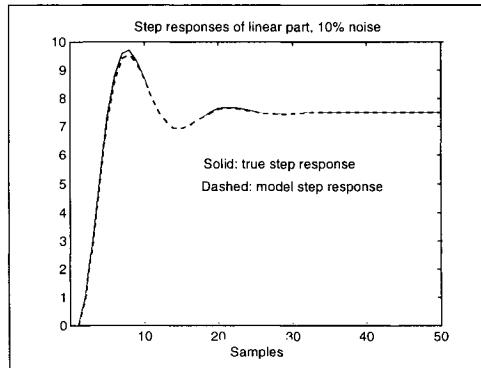


Figure 9.2.3 Fit of linear block, Simulation 1-2

Example 9.2.2. In this example more difficult situations are simulated, namely, a closed-loop

test and with measurement noise present. Simulated process

$$\begin{aligned} w(t) &= \frac{q^{-1} + .5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} u(t) + v(t) \\ y(t) &= \frac{w(t)}{\sqrt{0.1 + 0.9w^2(t)}} + e_2(t) \\ v(t) &= \frac{\alpha}{1 - 0.9q^{-1}} e(t) \end{aligned}$$

where $e_2(t)$ represents measurement noise.

Simulation 9.2.2-1. Disturbance enters at $w(t)$, the output of the linear block. Noise level is 10% in power at the output of the linear block and 12% in power at the process output. The measurement noise $e_2(t)$ is zero. A GBN with an average switch time of 10 samples is applied at the setpoint, the range of w is $[-0.64, 2.27]$. Closed-loop test. 1000 samples are used for identification.

The fit of the nonlinear function is shown in Figure 9.2.4. The fit of the step response of the linear block (after gain correction) is shown in Figure 9.2.5. The algorithm works well for the closed-loop data.

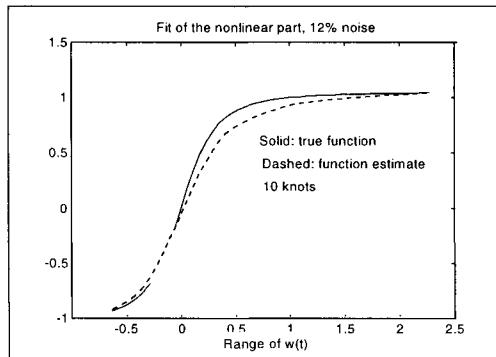


Figure 9.2.4 Fit of nonlinear function, Simulation 9.2.2-1

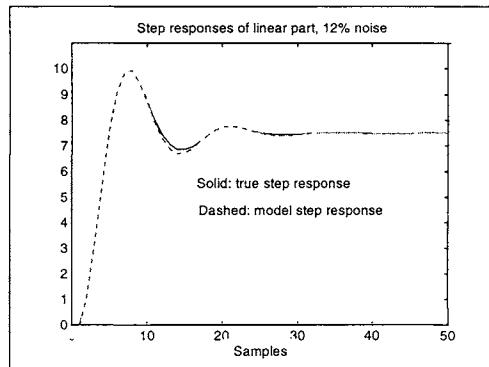


Figure 9.2.5 Fit of linear block, Simulation 9.2.2-1

Simulation 9.2.2-2. Disturbance $v(t)$ enters at $w(t)$, the output of the linear block. Noise level is 7% in power at the output of the linear block and 12% in power at the output. Measurement noise $e_2(t)$ enters at process output. $e_2(t)$ is white Gaussian noise and its level is 3% in power at the process output. A GBN, with average switch time of 10 samples, is applied at the setpoint, the range of w is $[-0.48, 1]$. Closed-loop test. 1000 samples are used for identification. The fit of the nonlinear function is shown in Figure 9.2.6. The fit of the step response of the linear block (after gain correction) is shown in Figure 9.2.7. One can say that, in this example, the algorithm is robust against measurement noise.

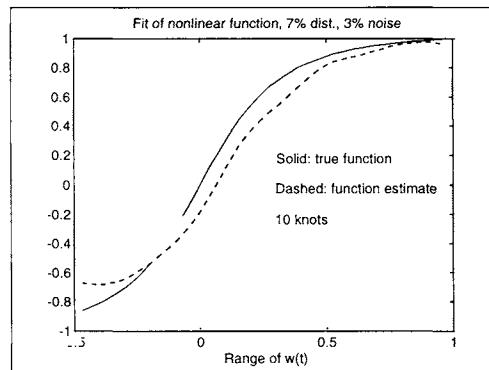


Figure 9.2.6 Fit of nonlinear function, Simulation 9.2.2-2

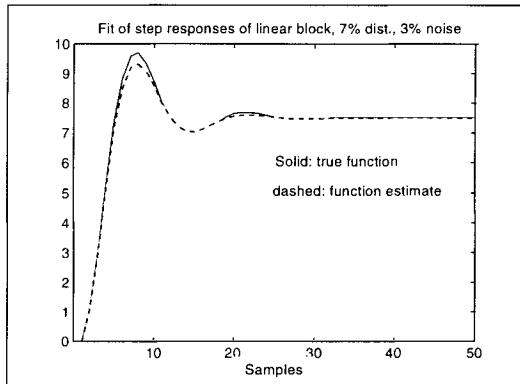


Figure 9.2.7 Fit of linear block, Simulation 9.2.2-2

In the above simulation studies, the relaxation algorithm converged in about 5 iterations.

9.3 Identification of NLN Hammerstein-Wiener Model

In this section a combination of Hammerstein and Wiener models will be proposed and its identification problems solved. The combined model consists of a linear block embedded between two static nonlinear gains; see Figure 9.3.1. For obvious reasons, this model will be called NLN Hammerstein-Wiener model. In a process control environment, the NLN model can be motivated by considering that the input nonlinear block $f_1(u)$ represents actuator nonlinearity and the output nonlinear block $f_2(w)$ represents process nonlinearity. Mathematically, because the NLN model includes both the Hammerstein model and the Wiener model as its special cases, it will approximate nonlinear systems better than either of these two models. Note that the disturbance $v(t)$ enters the process before the output nonlinearity. The motivation for this is the same as that for the Wiener model.

Very little has been done in the study of this model structure, perhaps due to its difficulties. Falkner (1988) proposed an iterative scheme to identify a nonparametric NLN model. The disturbance or noise is not considered explicitly. Chernyshov (2000) proposed an output error method for the identification of a nonparametric NLN model. To the best of our knowledge, parametric identification of the NLN model has not been studied in control literature. This section is based on Zhu (2001).

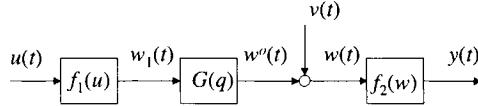


Figure 9.3.1 NLN Hammerstein-Wiener model

9.3.1 Model Parametrization and Problem Statement

The equations that describe the NLN model in Figure 9.3.1 are

$$\begin{aligned} w_1(t) &= f_1(u(t)) \\ w(t) &= G(q)w_1(t) + v(t) \\ y(t) &= f_2(w(t)) = f_2[G(q)f_1(u(t)) + v(t)] \end{aligned} \quad (9.3.1)$$

It is assumed that: 1) the nonlinear functions $f_1(u)$ and $f_2(w)$ are continuous; 2) function $f_2(w)$ is monotone and invertible; 3) the output of the linear part is disturbed by a stationary stochastic process $\{v(t)\}$.

Just as for the Hammerstein and Wiener models, the nonlinear functions will be parametrized using cubic splines and the linear block with the disturbance will be parametrized using a Box-Jenkins model. A cubic spline function for $f_1(u)$ is given as

$$w_1(u) = f_1(u) = \sum_{k=2}^{m_1-1} \beta_k |u - u_k|^3 + \beta_{m_1} + \beta_{m_1+1} u + \beta_{m_1+2} u^2 + \beta_{m_1+3} u^3 \quad (9.3.2)$$

The parameters in (9.3.2) are defined in the same way as that of the Hammerstein model in (9.1.5). A cubic spline function for $f_2(w)$ is given as

$$y(w) = f_2(w) = \sum_{k=2}^{m-1} \alpha_k |w - w_k|^3 + \alpha_m + \alpha_{m+1} w + \alpha_{m+2} w^2 + \alpha_{m+3} w^3 \quad (9.3.3)$$

The parameters in (9.3.3) are defined in the same way as those of the Wiener model in (9.2.3). The Box-Jenkins model for the linear part is

$$w(t) = G(q)w_1(t) + H(q)e(t) \quad (9.3.4)$$

where

$$\begin{aligned} G(q) &= \frac{b_0 + b_1 q^{-1} + \dots + b_n q^{-n}}{1 + a_1 q^{-1} + \dots + a_n q^{-n}} = \frac{B(q)}{A(q)} \\ H(q) &= \frac{1 + c_1 q^{-1} + \dots + c_n q^{-n}}{1 + d_1 q^{-1} + \dots + d_n q^{-n}} = \frac{C(q)}{D(q)} \end{aligned}$$

Note that intermediate signals $w_1(t)$ and $w(t)$ cannot be measured, hence two arbitrary gains may be distributed between the linear block and the two nonlinear blocks. This means that the parameters of the model cannot be uniquely determined without introducing further constraint. A solution to this problem is to fix the gains of the nonlinear blocks. For example, one can let $\alpha_{m+1} = 1$ and $\beta_{m+1} = 1$.

Now, the *identification problem* for the NLN Hammerstein-Wiener model can be stated as follows:

- 1) **Test design.** Design an identification test using excitation signals so that the parameters of the model in (9.3.1) - (9.3.4) are identifiable. If possible, the test should be designed so that the identified model is most suitable for control application.
- 2) **Parameter estimation.** Determine the parameters of model using the input-output data from the test by minimizing the loss function

$$V = \sum_{t=1}^N \varepsilon^2(t) \quad (9.3.5)$$

where

$$\varepsilon(t) = H^{-1}(q)[f_2^{-1}(y(t)) - G(q)f_1(u(t))] = \frac{D(q)}{C(q)}[f_2^{-1}(y(t)) - \frac{B(q)}{A(q)}f_1(u(t))] \quad (9.3.6)$$

- 3) **Order selection.** Determine the orders l , m_1 and m so that the obtained model is most accurate for control.
- 4) **Model validation.** Verify if the identified model is suitable for control. *If not, provide a remedy that includes redesigning the identification test.*

In the sequel, only the parameter estimation is studied. The other three problems can be solved in the same way as for the Hammerstein and Wiener models. The reader is now in a position to figure them out himself.

9.3.2 Parameter Estimation

The error defined in (9.3.6) is even more nonlinear than the Hammerstein or Wiener models. One may wonder if the same technique can be used to reduce the degree of nonlinearity and to use the relaxation algorithm again. The answer is, fortunately, yes.

Approximate the linear part with the Box-Jenkins structure in (9.3.4) by a high order ARX model

$$\begin{aligned} w(t) &= \frac{B^n(q)}{A^n(q)}w_1(t) + \frac{1}{A^n(q)}e(t) \\ &= \frac{B^n(q)}{A^n(q)}f_1(u(t)) + \frac{1}{A^n(q)}e(t) \end{aligned} \quad (9.3.7)$$

where $\{e(t)\}$ is white Gaussian noise with zero mean.

Substitute $w(t)$ in (9.3.1) using the ARX model, take the inverse of the nonlinear function $f_2(w)$ on both sides and then multiply them by $A^n(q)$. It then follows that

$$A^n(q)f_2^{-1}(y(t)) = B^n(q)f_1(u(t)) + e(t) \quad (9.3.8)$$

Then, the loss function for parameter estimation for model (9.3.8) becomes

$$V_{ARX} = \sum_{t=1}^N \varepsilon^2(t) \quad (9.3.9)$$

where

$$\varepsilon(t) = A^n(q)f_2^{-1}(y(t)) - B^n(q)f_1(u(t)) \quad (9.3.10)$$

Now parametrize the inverse $f_2^{-1}(y)$ by a cubic spline

$$f_2^{-1}(y) = \sum_{k=2}^{m_2-1} \gamma_k |y - y_k|^3 + \gamma_{m_2} + \gamma_{m_2+1}y + \gamma_{m_2+2}y^2 + \gamma_{m_2+3}y^3$$

where m_2 is number of knots and can also be called the degree of the inverse function. Writing out the nonlinear functions in equation (9.3.10) yields

$$\begin{aligned} \varepsilon(t) &= A^n(q) \left[\sum_{k=2}^{m_2-1} \gamma_k |y(t) - y_k|^3 + \gamma_{m_2} + \gamma_{m_2+1}y(t) + \gamma_{m_2+2}y^2(t) + \gamma_{m_2+3}y^3(t) \right] \\ &\quad - B^n(q) \left[\sum_{k=2}^{m_1-1} \beta_k |u(t) - u_k|^3 + \beta_{m_1} + \beta_{m_1+1}u(t) + \beta_{m_1+2}u^2(t) + \beta_{m_1+3}u^3(t) \right] \end{aligned} \quad (9.3.11)$$

Note that the error $\varepsilon(t)$ is bilinear in the parameters of $B^n(q)$, $A^n(q)$, $f_1(u)$ and $f_2^{-1}(y)$. Once more, one can use the relaxation algorithm for parameter estimation.

Estimating the Nonlinear Functions and the High-Order ARX Model

Initialization.

Set $f_1(u) = u$ and $f_2^{-1}(y) = y$ and estimate $A^n(q)$ and $B^n(q)$ using linear least-squares.

Iteration. Denote $\hat{A}_{(i)}^n(q)$, $\hat{B}_{(i)}^n(q)$, $\hat{f}_{1(i)}(u)$ and $\hat{f}_{2(i)}^{-1}(y)$ as the estimates from iteration i , then

- 1) Calculate the parameters of $\hat{f}_{1(i+1)}[(u(t)]$ for fixed $\hat{f}_{2(i)}^{-1}[y(t)]$, $\hat{A}_{(i)}^n(q)$ and $\hat{B}_{(i)}^n(q)$ by minimizing

$$V_{ARX}^{(i+1,1)} = \sum_{t=1}^N \{\hat{A}_{(i)}^n(q)\hat{f}_{2(i)}^{-1}[y(t)] - \hat{B}_{(i)}^n(q)f_{1(i+1)}[u(t)]\}^2 \quad (9.3.12)$$

- 2) Calculate the parameters of $\hat{f}_{2(i+1)}^{-1}[(y(t)]$ for fixed $\hat{f}_{1(i+1)}[(u(t)]$, $\hat{A}_{(i)}^n(q)$ and $\hat{B}_{(i)}^n(q)$ by minimizing

$$V_{ARX}^{(i+1,2)} = \sum_{t=1}^N \{\hat{A}_{(i)}^n(q)\hat{f}_{2(i+1)}^{-1}[y(t)] - \hat{B}_{(i)}^n(q)\hat{f}_{1(i+1)}[u(t)]\}^2 \quad (9.3.13)$$

- 3) Calculate $\hat{A}_{(i+1)}^n(q)$ and $\hat{B}_{(i+1)}^n(q)$ for fixed $\hat{f}_{1(i+1)}(u)$ and $\hat{f}_{2(i+1)}^{-1}(y)$ by minimizing

$$V_{ARX}^{(i+1,3)} = \sum_{t=1}^N \{A_{(i+1)}^n(q)\hat{f}_{2(i+1)}^{-1}[y(t)] - B_{(i+1)}^n(q)\hat{f}_{1(i+1)}[u(t)]\}^2 \quad (9.3.14)$$

Go back to 1). Stop when convergence occurs.

All the three steps are linear least-squares problems which are numerically simple and reliable. Again Result 9.1.1 applies here, which means that the iteration minimizes the loss function (9.3.9) if it converges.

Calculate the Nonlinear Function $f_2(y)$

The determination of the nonlinear function $f_2(w)$ from the estimate of its inverse $\hat{f}_2^{-1}(y)$ is an approximation problem. This can be done using a linear least-squares estimate. The number of knots or degree m of cubic spline function can be determined by visual inspection of the fit of the nonlinear function where process knowledge can be used to help the selection.

Model Reduction for the Linear ARX Model

The same idea of model reduction for the Hammerstein and Wiener models will also be used here; see the last two sections.

9.3.3 Simulation Studies

The performance of the NLN model estimation algorithm is demonstrated here. The process is given as

$$y(t) = f_2 \left[\frac{0.5q^{-1} + 0.25q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} \cdot f_1[u(t)] + v(t) \right]$$

with

$$\begin{aligned} f_1[u(t)] &= \frac{u(t)}{\sqrt{0.1 + 0.9u^2(t)}} \\ f_2(w(t)) &= w(t) + 0.2w^3(t) \\ v(t) &= \frac{\alpha}{1 - 0.9q^{-1}} e(t) \end{aligned}$$

The range of $u(t)$ is $[-0.7, 2.5]$. As can be seen in Figure 9.3.2, this process has severe nonlinearities at both the input and the output.

Simulation 1. First noise free simulation is used to check the correctness of the algorithm. Open-loop test is used and 1500 samples are simulated. GMN signal with $T_{sw} = 10$ is applied at the input. The range of $u(t)$ is $[-0.7, 2.5]$. The true model order of the linear part is used; the degrees of the two nonlinear functions are both 12. The algorithm converges in about 15 iterations. The identified model is almost perfect. The fit error is 0.07% in power at $w(t)$ and 0.04% at $y(t)$. The model plots are not shown. For comparison, some linear models are also identified. The fit error of a linear Box-Jenkins model is 26.8% and that of a linear output error model is 25.5%.

Simulation 2. The same as in Simulation 1, but a disturbance $v(t)$ is added at $w(t)$. Disturbance level is 3% (in power) at $w(t)$ and is about 10% at $y(t)$.

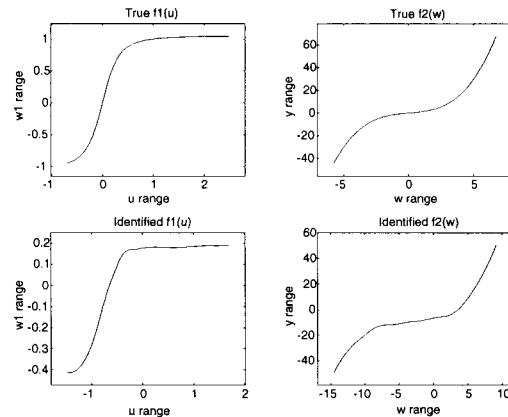


Figure 9.3.2 True and identified nonlinear functions. Note that differences in ranges are caused by removing the mean values of the input and output and by non-uniqueness of the gain distributions.

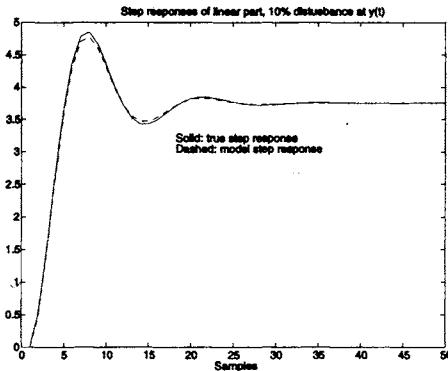


Figure 9.3.3 True and estimated step responses of the linear block. Model gain is corrected.

The true model order of the linear part is used; the degrees of the two nonlinear functions are both 12. The true and identified nonlinear functions are plotted in Figure 9.3.2 and the true step response of linear part and its estimation are shown in Figure 9.3.3. One can see that the identified model is still very good considering the disturbance level. The algorithm converges in about 15 iterations.

Simulation 3. The same as Simulation 2, but white Gaussian measurement noise is added at output $y(t)$. Noise level is 1% (in power) at $y(t)$. This is to test the robustness of the algorithm against measurement noise that is not included in the NLN model (9.3.1).

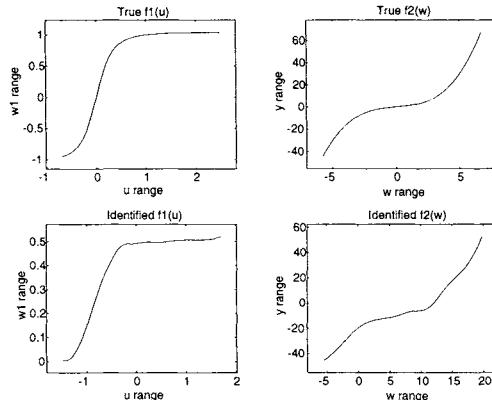


Figure 9.3.4 True and identified nonlinear functions. Note that differences in ranges are caused by removing the mean values of the input and output and by non-uniqueness of the

gain distributions.

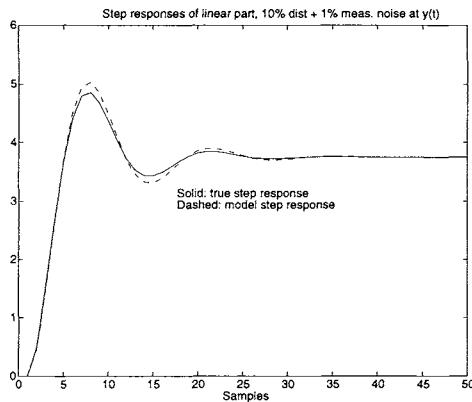


Figure 9.3.5 True and estimated step responses of the linear block. Model gain is corrected.

The true model order of the linear part is used; the degrees of the two nonlinear functions are both 10. The results are shown in Figure 9.3.4 and Figure 9.3.5. The model accuracy is lower than that in Simulation 2. This is caused by the measurement noise. However, the resulting model is still good when considering the difficult situation of nonlinearity and disturbance/noise. The algorithm converges in about 15 iterations.

9.3.4 MISO Extension

The NLN model can be extended to multi-input single-output (MISO) case easily if there is no joint nonlinearity among the inputs. Assume m inputs, then the block diagram of a MISO NLN model is given in Figure 9.3.6.

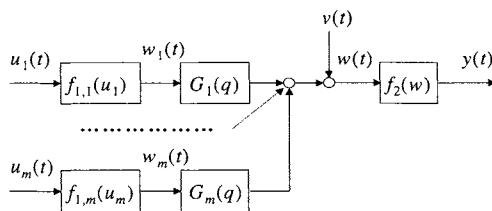


Figure 9.3.6 MISO NLN model

The equations of the MISO NLN model are

$$\begin{aligned} w(t) &= G_1(q)f_{1,1}(u_1(t)) + \dots + G_m(q)f_{1,m}(u_m(t)) + v(t) \\ y(t) &= f_2(w(t)) \end{aligned} \quad (9.3.15)$$

It is assumed that: 1) the nonlinear functions $f_{1,1}(u_1), \dots, f_{1,m}(u_m)$ and $f_2(w)$ are continuous; 2) function $f_2(w)$ is monotone and invertible; 3) the output of the linear part is disturbed by a stationary stochastic process $\{v(t)\}$. Again the linear part with the disturbance will be parametrized in a MISO Box-Jenkins model and the nonlinear functions will be parametrized using cubic splines; see the last subsection for details.

It is straightforward to extend the identification algorithm (9.3.12) to (9.3.14) for the MISO NLN model (9.3.15), which will not be repeated here. The same hold for MISO Hammerstein and MISO Wiener models.

9.4 Conclusions and Recommendations

In this chapter the identification of block-oriented nonlinear models is studied. Three types of models have been studied in detail: Hammerstein model, Wiener model and NLN Hammerstein-Wiener model. The ASYM method is extended to provide systematic solutions to the four problems of identification for both open-loop and closed-loop operations. In test design, it was emphasized that, in general, multiple amplitude signals should be used. An important result in this chapter is in parameter estimation. The idea is to manipulate the model structure so that the errors are bilinear-in-the-parameters and simple relaxation algorithms can be used. For model order selection, the final output-error (FOE) criterion is used. Upper error bound for the linear part is used for model validation. The effectiveness of the proposed methods has been shown in various simulation examples. The MISO extension of the algorithms is straightforward provided that there is no joint input nonlinearity.

The method developed in this chapter will certainly extend the capability of linear models. However, they will not solve the problems of all nonlinear processes. The reason is that only static nonlinearities are included in this model class and they are combined in a relatively simple manner.

When the reader has a nonlinear process that cannot be modelled by these block-oriented models, he can still try to use the guidelines and techniques developed in this chapter:

- Use process knowledge to reduce model complexity
- Put the disturbance in a position that is physically motivated and model it as colored noise if possible
- Manipulate/approximate the model so that the error is linear- or bilinear-in-the -parameters.

When all the above techniques can be applied for the given process, the reader will be able to derive an identification method that is theoretically accurate and numerically simple and reliable.

Nonlinear process identification remains a fascinating field of research. We are still in a begin phase. One of the interesting topics is to develop parametric models that can model the change of dynamics as a function of working point(s).

Chapter 10

Applications of Identification in Process Control

This chapter is about application. First an industrial control project approach is outlined and the use of identification in various steps in control projects will be discussed. Current industrial identification methods are commented on and possible improvements are pointed out. Model based PID tuning is discussed and a laboratory example is shown. Then the identification of ill-conditioned processes is studied based on simulations. Finally two industrial MPC control projects cases will be described where the ASYM method was used for model identification. The purpose of this chapter is twofold: to provide guidelines for industrial process control and in particular identification, and to show that modern identification approach is very cost-effective. It will be useful to study this chapter together with Chapter 1 and Chapter 3. It is assumed that the reader is familiar with PID control and MPC control.

10.1 A Project Approach to Advanced Process Control

To carry out an industrial advanced process control (APC) project is a challenging task. The main difficulties are the large scale and complexity of the problem, high level of disturbances, limits on additional allowed disturbances for testing and some degree of nonlinearity. A project approach has been developed by several APC/MPC control vendors and operation companies in the last 15 years that have been proven successful in the refining/petrochemical industry; see Cutler and Hawkins (1988) and Rchalet (1993). The phases of the project are:

- feasibility study and functional design;
- pre-test;
- identification test and model identification;

- controller tuning, simulation and commissioning;
- maintenance.

Experience has shown that the success rate of projects is high if a project team follows these steps seriously; the rate of success is low otherwise. Although this approach has been developed within the refining/petrochemical industry, it can also be used as a guideline or reference for other industries.

Feasibility Study and Functional Design

This step involves studying the process unit operation practice, reviewing unit operation economics, identifying process constraints and performing benefit analysis of the MPC project. Considerable time and effort is spent on the plant and with the operation personnel for gathering relevant information. P&ID's (pipe and instrument drawings) are reviewed and historical operating data are logged and studied. Discussions with operation managers, process engineers and operators will take place. Based on the collected information, one or more preliminary MPC controllers are designed that reflect the desired unit operation improvements. Operation performance improvement of MPC control is assessed. An MPC control is used to maintain stable unit operating, to reduce the effect of disturbances, and to bring the unit to a more profitable working range. This, in turn, results in the reduction of energy and material consumption and the increase of the throughput of valuable products. The economic benefits of the MPC control are then estimated. A Functional Design Report is composed, containing the following:

- Summery of unit operation and economics;
- MPC control strategy and preliminary controller design that includes
 - control objectives
 - economic objectives
 - list of manipulated variables (MV's)
 - list of measured disturbance variables (DV's)
 - list of controlled variables (CV's)
- data base configuration and programing;
- recommendations for instrument modification that includes adding tags in data collection, display screens;
- recommendations of additional analyzers or inferential calculations;
- benefit analysis;
- project planning and cost estimates.

The work should be carried out by an experienced control engineer who understands unit operation and its economics.

An MPC project will be justified if there are enough benefits. The pay-back time of an MPC project is between 3 to 9 months in the refinery/petrochemical industry.

Pre-Test

This is another on-site phase of the project. Objectives of the pre-test are:

- to inspect and, if necessary, to tune PID loops which are MV's;
- to identify and fix instrument/actuator problems such as sticking valves;
- to obtain rough estimates of time to steady state (settling time) and some model gains;
- to determine proper signal amplitudes (step sizes) for identification tests;
- to obtain rough estimates of the long time delays.

To achieve these objectives, steps are applied at all MV's in open-loop operation. For each MV, at least one step up and one step down are necessary. More steps may be necessary for PID tuning and/or for determining process gain and settling time. Small step sizes can be sufficient for PID tuning. Larger step sizes are needed for information about process settling time and gains.

There are some good practices in pre-test that can save project time. Collecting and studying historical data can be useful in identifying good and bad regulatory control loops and in determining process settling time (time to steady state) and some process gains. Asking good questions to experienced operators can provide information on regulatory control loops, on process settling time, delays and suitable step sizes (amplitudes) for some MV's.

Single variable identification can be used to tune the regulatory PID loops; see Section 10.3.

The information obtained during the pre-test is used to design the identification test and to give recommendations for additional instrument work.

Identification Test and Model Identification

This is the crucial phase of the project, because accurate models are essential for the success of an MPC project. Often an identification test is called a step test. This is because traditionally steps are used as test signals in a series of single variable tests; see Cutler and Hawkins (1988) and Richarlet (1993). We will no longer use this name because in the new test approach, other types of signals, such as GBN signals, are used. The traditional step tests are carried out manually in open-loop. The new tests are typically multivariable and (partially) in closed-loop, and are carried out automatically by a test program. During a test, operator intervention is allowed and encouraged in order to stabilize the unit operation. See Chapter 3 for the details of identification

test design. During the tests, data will be collected and preliminary identification will be carried out. The intermediate identification results are used to access the state of the on-going test and to decide when to stop the test.

When a test is finished and data collected, model identification will be performed using an identification software package. Traditionally, after a long step test, an equally long time is needed to perform identification and hence it was often defined as a separate project phase. This is caused by two things. First, many disturbances occur during the long step test. Data slicing is used to cut the bad portion out, which is a tedious process. Secondly, there is no systematic approach to model order selection (or, settling time determination) and model validation/selection. Hence much time is spent using a trial-and-error approach. In the new approach, the test is much shorter and operator intervention and closed-loop control will maintain stable operation. The task of data slicing is minimal when using multivariable automated tests. In model identification, the problems of parameter estimation, order selection and model validation are solved in a systematic manner and it can be finished in a fraction of the (already shorter) test time. Therefore, in the new approach, the identification test and model identification can be combined as one phase.

In general, identification tests will not disturb normal unit operations.

It is this part of the project that will benefit from the methods developed in this book; see Sections 10.5 and 10.6.

Controller Tuning, Simulation and Commissioning

The designed controller is tuned according to the desired specifications. The tuning of an MPC controller considers first the parameters related to unit operation requirements, these include MV high/low limits, MV rate of change limits, MV ideal resting values (for economic optimization), CV high/low limits (zones) or setpoints. Then, the parameters related to control performance such as loop speed and robustness are considered; these parameters include MV weighting factors, CV weighting factors or closed-loop settling time of the CV's. Some MPC controllers can set CV priorities when their control requirements are in conflict conditions. The controller nominal performance is checked in simulation where the identified model is used to represent the process. In simulation, first the stability of the controlled system is checked. There should be no oscillations and some of the CV's must be decoupled reasonably. The constraint handling are checked by narrowing the MV and/or CV limits; CV conflict situations are simulated in order to see how the controller can resolve them.

When a simulated system shows a good control performance, the developed controller can be loaded to the control computer and turned on gradually. First the controller is set to a simulation or off-line mode. In this mode, the controller will read the MV and CV measurements, simulate the CV movements and show the desired MV actions, but the process MV's are not yet connected to the MPC control outputs. In this way the control engineer can verify if the model predictions are accurate and control actions are relevant. When proper control movements are shown, the MV's will be turned on, meaning they will be connected to controller outputs. Initially, the

high/low limits of the MV's are set very close to each other in order to restrict the controller's freedom. This is to let the operators gain confidence with the controller. These limits are then gradually relaxed so that the controller will be able to push the unit to its most beneficial operating point. The controller performance will be tested, which includes disturbance attenuation, constraint handling and setpoint following. It may be necessary to adjust the tuning parameters such as MV weights and CV weights (or CV settling times), this is because the on-line controller will perform a little differently than the simulated one due to disturbances, model errors and nonlinearities. Occasionally, one can also modify some model gains. Normally, increasing model gains will lead to slower control actions and decreasing model gains will lead to more aggressive control actions. Usually only a few poor model gains, indicated by model validation, can be modified. If many model transfer functions are of poor quality, gain modification will no longer work. It is then recommended to perform an improved identification test.

Operators need to know how MPC works and how to operate the unit with the controller MPC on-line. Hence operator training is necessary. This happens during and at the end of the commissioning.

A fall-back mechanism must be implemented so that when the MPC controller fails, whatever the reason, the control system will automatically enter a safe operation mode.

Controller Maintenance

The life span of an MPC controller is between 2 to 3 years. Therefore, MPC maintenance is important to prevent the loss of benefits due to control performance degradation over time. The control performance degradation is caused by slow process changes such as heat exchangers fouling, catalyst deactivation and process modification or revamp. These changes will increase the error of the existing model, which, in turn, leads to poor control performance. The main task of maintenance is to re-identify the process model and to re-commission the MPC controller. It may also include MPC design modification when operating objectives change.

Traditionally, in MPC control maintenance, a new open-loop step test will be performed and new models identified, which is, again, very time consuming and may disturb the unit operation. Then the controller is simulated and commissioned. The techniques developed in this book can play an important role in controller maintenance. For model identification, it is recommended to carry out a multivariable and closed-loop test using the current MPC controller. Although the MPC is no longer optimal, it can often be used to stabilize the operation during the identification test. In this way, the test time will be much shorter and unit operation will not be disturbed. Also a partial closed-loop test can be performed with part of the MPC turned on or using several PID loops.

10.2 Identification Requirements for Process Control

In this section we will review the characteristics of industrial processes and discuss their identification requirements. We will again use the processes of the refinery/petrochemical industry for the discussion. From control point of view, these process units can be characterized as follows.

Large scale with complex dynamics. A small MPC controller will have 3 to 5 MV's and 5 to 10 CV's; a large size MPC will have over 20 MV's and 35 CV's. Some CV's, such as product qualities, are very slow with time constants of several hours, and other CV's, such as valve positions are very fast with time constant of a few minutes or even a few seconds. Inverse response (non-minimum phase zeros), oscillating behavior and time delays exist. Control/identification methods that work for single variable processes or processes with a couple of inputs and outputs can fail when used here.

Dominant slow dynamics. The time to steady state of a typical product quality model ranges from one hour to several hours. This requires a long time for the identification test.

High level unmeasured disturbances. Typical sources of unmeasured disturbances are feed composition variations and weather changes. These are irregular variations. Too large test signal amplitudes are not permitted because they will cause off-specification of products and/or nonlinearity.

Local nonlinearity. Although in general linear models are relevant for MPC for a given range of operation, some nonlinear behavior may still show up. Examples are CV's that are very pure product qualities and valve positions close to their limits. An identification test should be done with great care when severe nonlinearities are to be avoided.

Based on these observations we will outline the special needs in identification for process control and comment on the existing methods.

Identification tests. Current practice in the MPC industry is to use a series of single variable step tests for model identification. The tests are carried out manually. The advantage of this test method is that the control engineer can watch many step responses and can learn about the process behavior in an intuitive way. The first problem of the single variable step tests is its high cost in time and in manpower. For example, to test a crude unit atmospheric tower will take two to three weeks around the clock. The second problem is that the data from single variable tests may not contain good information about the multivariable character of the process. Test signal design is another important problem. Steps are often too slow and do not contain enough power at middle and high frequencies. In the other extreme, a textbook test method uses white-noise-like PRBS signals that are too fast and do not contain enough power at low frequencies. For the MPC applications, a good design method is to choose test signals that minimize the prediction error (often called simulation error in identification literature) under CV constraints. Ideally, if possible, the identification test should be done in closed-loop operation. One advantage of a closed-loop test is obvious: the controller helps to keep the CV's within their limits. Another advantage is that a model identified from a closed-loop operation can give

better control performance; see Hjalmarsson *et. al.* (1996) for the SISO case and Section 10.4 for the MIMO case.

There were some misunderstandings on test methods and identifiability issues. One misunderstanding says that *the MV signals must be uncorrelated*. The right statement is that MV correlations will, in general, not do any harm on identifiability provided that no MV is linearly dependent on the others. Another misunderstanding says that *the process is not identifiable if closed-loop data are used*, because feedback will cause correlations between MV's and between unmeasured disturbances and MV's. It is well known that the process is identifiable in closed-loop, provided that excitation signals are applied from outside the loops; see Gustavsson *et. al.* (1977). In Section 10.4 we will show that MV correlations and closed-loop test will in fact improve model quality for control.

Model structure, parametric or nonparametric. Some traditional industrial identification software packages are based on finite impulse response (FIR) model. This model class has inherent problems for processes with slow dynamics. Model variance error is high due to its high order; yet bias error is often not negligible due to truncation. Therefore, parametric or compact models are preferred for identification of processes with slow dynamics. This is true also for MPC controllers that use non-parametric models.

One misunderstanding on model structure says that *a parametric model is not capable of modeling complex dynamics such as under-damped (oscillating) dynamics and inverse responses (nonminimumphase zeros)*. In fact a linear time-invariant process can be approximated arbitrarily well by a parametric model provided that the model orders or structure indices are chosen properly.

Model structure, with or without disturbance model. As mentioned before, high level unmeasured disturbances are present in process control environments. This means that when model accuracy is concerned, an estimation criterion that includes a disturbance model will be better than a criterion without the disturbance model. The prediction error methods and frequency domain maximum likelihood methods belong to the first class; while output error criterion belongs to the second class; see Section 5.5. Moreover, the prediction error criterion and maximum likelihood criterion will work properly for closed-loop data (model estimates are consistent); while the output error criterion will deliver a biased model when using closed-loop data. Finally, a disturbance model can be used to improve short and middle term prediction in MPC control.

Model validation. The goal of model validation is to test whether the model is good enough for control and to provide advice for re-identification if the identified model is not valid for its intended use. Commonly used methods of model validation are simulation using estimation data or fresh data, whiteness tests residuals and the tests on the independence between the residuals and past MV's. These methods only tell how well the model agrees with the test data. They can neither quantify the model quality with respect to the purpose of closed-loop control, nor can they give good advice for re-identification. The trial-and-error approach in this step makes industrial identification a very expensive practice; see Section 10.1. In linear model identification, a suitable approach for model validation is to quantify model errors in the frequency domain

and to find a relationship between this kind of error description and control performance; see Chapters 6 and 7.

10.3 PID Autotuning using Process Identification

Although MIMO model based control such as MPC is becoming more popular in process control, most control loops are still of the PID type. PID tuning is also part of the pre-test in an MPC project. Therefore, proper tuning of PID loops is very important to maintain good performance of the overall process control system. In this section a model based PID autotuning method is outlined. The method uses the ASYM method for identification and IMC tuning rules for controller tuning.

Step 1. Identification

Given a PID loop that needs to be retuned, perform a closed-loop test by adding a test signal at the setpoint or at the input and collect the data. Then identify a model using the ASYM method as in Chapter 6. Denote the process model as $\hat{G}(q)$ and the upper error bound as $\bar{\Delta}(\omega)$. Because the PID tuning rules in the next step uses up to second order models, reduce the model $\hat{G}(q)$ to order two if its order is higher than two. Finally convert the model to continuous-time (s -domain) and denote it as $\hat{G}(s)$.

Step 2. Robust PID Tuning

There are many model-based PID tuning rules, such as dominant pole placement, optimization by minimizing integral square error (ISE) or integral absolute error (IAE), and internal model control (IMC) tuning; see Åström and Hägglund (1995). Here we will introduce IMC tuning rules introduced by Rivera *et. al.* (1986). The idea is to use the two step IMC design method to derive PID parameters based on a low order model of the process. The PID parameters are determined so that the closed-loop behavior approximates the behavior of a first order filter

$$f(s) = \frac{1}{\tau_{cl}s + 1} \quad (10.3.1)$$

For controller tuning, the user only needs to specify the time constant τ_{cl} of the filter. A large time constant leads to a slow response and a more robust controller; a small time constant leads to a fast response, but a less robust controller. The closed-loop load response exhibits no oscillation or overshoot, which minimizes controller interactions and enhances process disturbance rejection. Tuning formulae for various process models are given in a table; see Rivera *et. al.* (1986) and Chien and Fruehauf (1990). Therefore, when a process model is identified, it is straightforward to obtain PID parameters using the IMC tuning rules. The closed-loop system can be simulated using the model and the controller. Industrial experience of IMC tuning rules is very positive; see Chien and Fruehauf (1990).

Because model errors are inevitable in process identification, a good control performance according to simulation does not necessarily mean a good performance in reality. The robustness of

the controlled system against model errors can be analyzed using the upper error bound $\bar{\Delta}(\omega)$. Denote $C(s)$ as the controller, then it can be shown (see, e.g., Rivera *et. at.*, 1986) that the controlled system is robustly stable for all the errors bounded by the upper bound if

$$\frac{|C(i\omega)|}{|1 + \hat{G}(i\omega)C(i\omega)|} \cdot \bar{\Delta}(\omega) < 1 \text{ for } \omega \in [0, \infty] \quad (10.3.2)$$

Also the performance of the true system will be close to the simulation if the left hand side of (10.3.2) is much smaller than 1, for example, smaller than 0.4.

Nonlinear PID Control

The above procedure can easily be generalized to nonlinear processes if block-oriented models can be identified as in Chapter 9. The idea is to first compensate the nonlinearity using inverse of the nonlinear function, then tune the PID using the linear model. The block diagrams of nonlinear PID control using the Hammerstein model and the Wiener model are given in Figures 9.3.1 and 9.3.2 respectively.

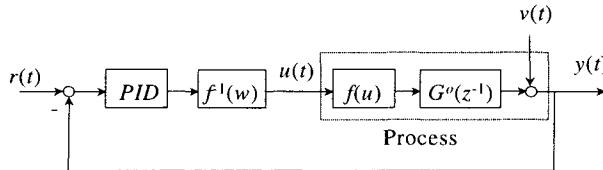


Figure 9.3.1 Nonlinear PID control with Hammerstein model where $f^{-1}(w)$ is the inverse of the nonlinear function

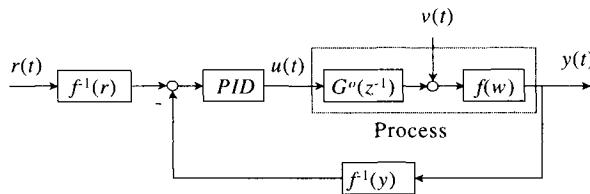


Figure 9.3.2 Nonlinear PID control with Wiener model where $f^{-1}(r)$ and $f^{-1}(y)$ are the inverses of the nonlinear function

Identification and Control of a pH Process

This is a laboratory process. The pH process consists of a continuous stirred tank reactor (CSTR) with two input streams and one output stream. The scheme is shown in figure 9.3.3.

The first input flow consists of solution of strong acid and the second flow consists of a solution of strong base. The acid flow has a constant rate and the rate of base flow can be adjusted using a controlled pump. These two flows react with each other and produce a pH value. The pH of the solution inside the CSTR is measured by using a pH sensor. The base flow rate is used to control the pH value of the solution inside the tank.

A closed-loop identification test has been carried out. A staircase test signal with varying step length is applied at the pH setpoint; see Figure 9.3.4. A Wiener model is identified using the data. The linear model part has a order of 2, but a first order model is almost as good. The nonlinear part has degree 10. Figure 9.3.5 shows the identified nonlinear gain which shows that the gain decreases as the pH value increases. Figure 9.3.6 shows the fit of the model to part of the data.

Based on the identified Wiener model, a nonlinear PID controller is designed and tested for the pH process. In the control scheme, the inverse of the nonlinear gain is placed in the feedback path and before the setpoint. Figure 9.3.7 shows the control result of the nonlinear PID controller; Figure 9.3.8 shows the result of a linear PID controller. One can see that the system with linear controller becomes slower when pH value is high, but with the nonlinear controller the performance is nearly the same for low and high pH values. See Erol (1999) for more details on the study.

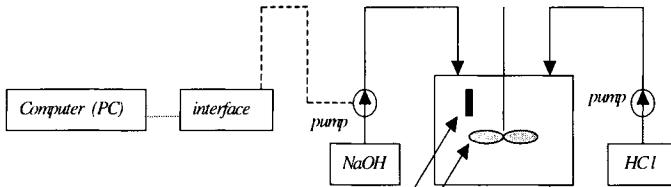


Figure 9.3.3 The pH control setup

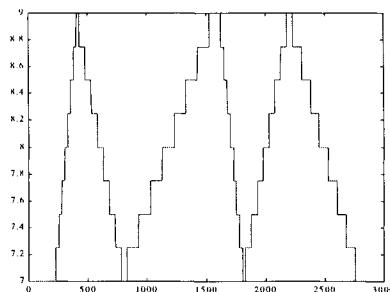


Figure 9.3.4 Test signal at setpoint

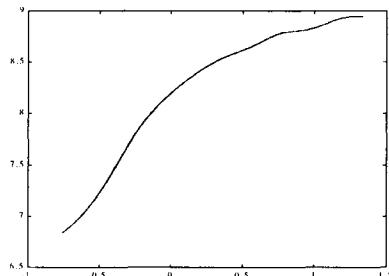


Figure 9.3.5 Identified nonlinear curve

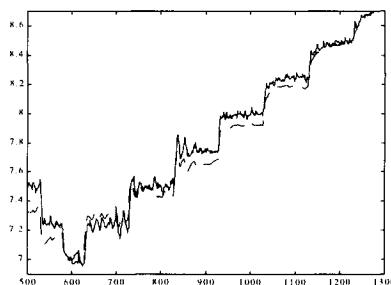


Figure 9.3.6 Model fit to the data. Solid line: measured pH; dashed line: simulated pH

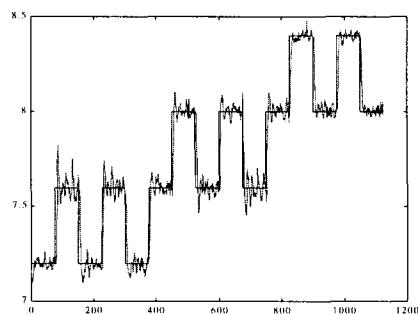


Figure 9.3.7 Control performance of the nonlinear PI control

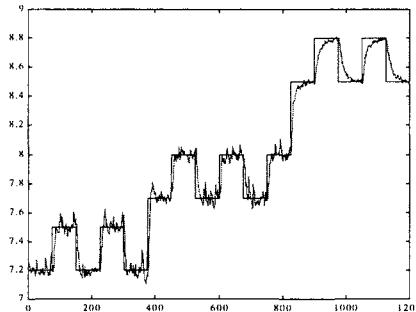


Figure 9.3.8 Control performance of the linear PI controller

10.4 Identification of Ill-Conditioned Processes

Unlike single-input single-output (SISO) processes, multivariable (MIMO) processes have "interactions" or "directions" among certain outputs. If some interactions are very strong, it will be called an ill-conditioned process. For controlling these outputs independently, it is important that the low-gain direction of the model, or, the small difference between certain outputs, should be identified with high accuracy. This is a difficult task because the information about the low-gain direction is buried in the noise in a common open-loop test. The problem of identifying dynamic models for control of ill-conditioned processes has received some attention recently. Jacobsen (1994) has studied the influence of model parameterization and of the test design. He concluded that a MISO structure is not suitable for the problem and a closed-loop test or an open-loop test with correlated inputs should be used. Koun and MacGregor (1993) have developed a test approach using highly correlated test signals.

In this section we will study various ways to improve the estimate of the low-gain direction of the model. We will check the influence of model structures, different identification methods and test methods. As an example of ill-conditioned processes, a linearized model of a high-purity distillation column is used. In Section 10.4.1 the problem is stated using the high purity distillation column. In Section 10.4.2 various model structures and identification methods are compared using normal open-loop and closed-loop tests. In Section 10.4.3 a simple and practical open-loop test method will be proposed and tested. This section is based on Stec and Zhu (2001).

10.4.1 High Purity Distillation Column as an Ill-Conditioned Process

The process to be identified is a binary distillation column. Its detailed description can be found in Skogestad (1997). The linearized version of the model has 82 states; see Jacobsen (1994). The column is running in the so called LV-configuration. The two MV's are reflux flow rate L and boilup flow rate V ; CV's are distillate composition D_y and bottom composition B_x . The level in the condenser is controlled using the distillate flow and the bottom level is controlled using bottom product flow. Figure 10.4.1 shows the schematic diagram of the column.

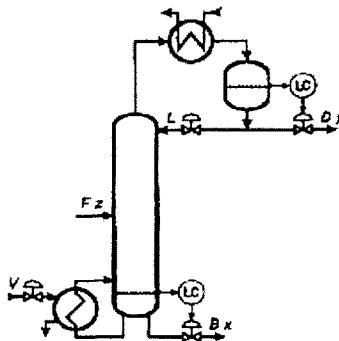


Figure 10.4.1 High purity distillation column

Frequency responses of the model are given as

$$\begin{bmatrix} D_y(i\omega) \\ B_x(i\omega) \end{bmatrix} = G(i\omega) \begin{bmatrix} L(i\omega) \\ V(i\omega) \end{bmatrix} \quad (10.4.1)$$

where

$$G(i\omega) = \begin{bmatrix} g_{11}(i\omega) & g_{12}(i\omega) \\ g_{21}(i\omega) & g_{22}(i\omega) \end{bmatrix}$$

After singular value decomposition the matrix $G(i\omega)$ can be written as

$$G(i\omega) = W(i\omega)\Sigma(\omega)V^*(i\omega) \quad (10.4.2)$$

where

$$W(i\omega) = [w_1(i\omega) \quad w_2(i\omega)], \quad \Sigma(\omega) = \text{diag}(\sigma_1(\omega) \quad \sigma_2(\omega)), \quad V(i\omega) = [v_1(i\omega) \quad v_2(i\omega)]$$

Here $W(i\omega)$ is the left singular unitary matrix with $w_1(i\omega)$ and $w_2(i\omega)$ being the output directions (vectors), $V(i\omega)$ is the right singular unitary matrix with $v_1(i\omega)$ and $v_2(i\omega)$ being the input

directions (vectors), and $\Sigma(\omega)$ is the real diagonal singular value matrix with singular values in a descending order $\sigma_1(\omega) > \sigma_2(\omega)$.

The large singular value $\sigma_1(\omega)$ is called high gain of $G(i\omega)$ with $v_1(i\omega)$ and $u_1(i\omega)$ being the corresponding input and output directions respectively. Similarly, the small singular value $\sigma_2(\omega)$ is called low gain of $G(i\omega)$ with $v_2(i\omega)$ and $u_2(i\omega)$ being the corresponding input and output directions respectively.

The condition number is given by the ratio of upper and lower singular values

$$\gamma(G(i\omega)) = \frac{\sigma_1(\omega)}{\sigma_2(\omega)} \quad (10.4.3)$$

A process is said to be ill-conditioned if $\sigma_1(\omega) \gg \sigma_2(\omega)$ or $\gamma(G(i\omega)) \gg 1$ in some frequency range. However, a large condition number does not necessarily mean that the two outputs (CV's) are strongly interactive. For example, consider a diagonal process with two SISO subprocesses. The condition number can be arbitrarily large when the numerical ranges of the two SISO sub-processes are very different, due to the use of engineering units. But the two CV's are not interactive at all. To resolve the problem, one can normalize the model by scaling the inputs and outputs. The scaling for an input or output signal is done by dividing the original signal by a scaling factor. The scaling factor can be the normal operation range of the signal, or, the standard deviation of the signal during an identification test. For a normalized model, all the inputs (MV's) all have the same or similar numerical ranges and the same holds for the outputs (CV's). If a normalized model $G(i\omega)$ has a large condition number and is ill-conditioned, then the two CV's will be strongly interactive. From now on, it is assumed that $G(i\omega)$ is a normalized model.

As a good measure of interaction level that is independent of scaling, we can use the relative gain array (RGA); see Morari and Zafiriou (1989). The definition of the elements of the RGA is given as

$$\lambda_{ij} = g(i\omega)[G^{-1}(i\omega)]_{ij}$$

For a 2×2 matrix, we can describe the level of interactions in the system with just one parameter λ_{11} . A large value of λ_{11} means that the plant is strongly interactive. This is independent of the model scaling.

The singular value and RGA plots of the high purity column are shown in Figure 10.4.2. It is easy to see that the process is very ill-conditioned and strongly interactive at low frequencies. Because of that, such a process requires multivariable techniques for identification and control system design. The directionality as well as interactivity of the system is weaker at medium and high frequencies. However, it is the low frequency behavior of the process that plays the greater role from the control point of view. The directionality of high purity distillation columns is caused by difference between reactions of the plant on a change in internal and external flows (Skogestad, 1997). Changes in external flows (distillate and bottom flow rates) affect final product compositions significantly more strongly than similar changes in internal flows. That

corresponds respectively to the high and low gain direction of the plant. The low gain direction dynamics is much faster than the dynamics of the high gain direction and has a dominant time constant of about 15 minutes. The slower direction has a time constant of about 194 minutes and it is the dominant time constant of the process.

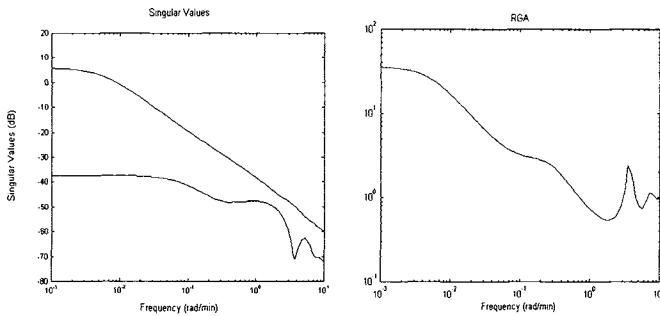


Figure 10.4.2 Singular value plots (left) and RGA plot (right) of the high purity distillation model

10.4.2 Identification of the High Purity Column

We will study the influence of identification testing methods, model parameterization and identification methods. Simulating the distillation column model generates both open-loop and closed-loop test data. The duration of a single experiment is 10000 minutes and the sampling period is 2 minutes. It results in 5000 samples for each test. Outputs are disturbed by low-pass filtered uncorrelated white noises (1st order filter with a time constant of 100 minutes). To study the influence of disturbance, three disturbance levels are simulated: signal-to-noise ratios (SNR's) for each output are 3.3, 10 and 33 in terms of variances. These test setups are realistic for the given process; see Zhu (1998) for industrial test conditions. For the sake of space, not all results are shown.

The model structure is roughly defined as follows:

- 1) SISO structure: Each transfer function is parameterized independently
- 2) MISO structure: There is a common model for each output
- 3) MIMO structure: There is a common model for all outputs.

Various SISO, MISO and MIMO parameter estimation methods are used and compared to each other. These are:

- 1) ARMAX method, MISO structure, common denominator polynomial per output
- 2) BOX-JENKINS method, SISO structure
- 3) Subspace method MOESP (Chapter 8), MIMO and MISO structures.

Remark. The ARMAX and Box-Jenkins models can be interpreted as maximum likelihood estimates and they have the highest accuracy in a stochastic framework. But to estimate ARMAX and Box-Jenkins models for a MIMO structure is difficult. The subspace method is numerically simple and reliable and it can estimate the MIMO structure easily. The accuracy, however, is not highest, because it is not yet clear what error criterion it uses.

For the validation of the models we check the fit of the singular values, RGA and the performance of the corresponding MPC controllers. For each situation 20 simulations are performed. The singular value and RGA plots of all the 20 simulations are plotted together with the true ones. A model will be considered "unbiased" if the true singular value and RGA plots are in the middle of their 20 estimates. A model will be considered have small variance if the 20 singular value and RGA plots form narrow bands. The step responses of closed-loop MPC system are used to verify the model quality for use in control. Two MPC controllers are designed such that the closed-loop dynamics have a dominant time constant of 60 and 10 minutes respectively. When the simulated closed-loop step responses are close to the "true" ones, we say that the model has a high quality for control or has a high performance.

Identification using a Normal Open-loop Test

First we use two independent GBN signals on inputs of the plant simultaneously. Both GBN signals have an average switch time of 300 minutes. The amplitude is chosen somewhat arbitrarily as 0.1. The process model is linear so the amplitude has no influence on estimation results.

None of the applied estimation methods can create a good model for control. While the high gain can be estimated easily, the low gain is very poorly estimated in all cases. Figure 10.4.3 shows the results for 20 models for the signal-to-noise ratio of 33 which is a very low noise level. All the methods and model structure give poor results.

We have experienced a number of numerical problems in parameter estimation such as instability of models or lack of algorithm convergence. The signs of matrix gain determinant for the model and for the plant also differed from each other in several cases.

The reason for such poor estimation results is that the information of the low gain direction was dominated by the noise when using normal uncorrelated test inputs. Figure 10.4.4 shows the directionality of the output signals. Thus, we conclude that the problem is caused by poor data and not related to identification methods or model structure.

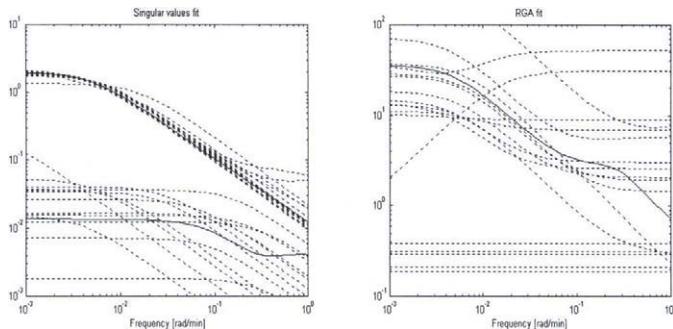


Figure 10.4.3 Singular values (left) and RGA's of MIMO MOESP models from 20 simulations, SNR = 33. Solid lines are the true values

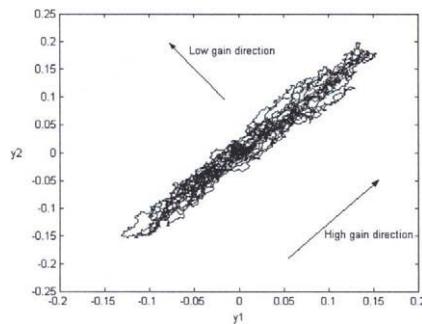


Figure 10.4.4 Excitation of output directions of a normal open loop test

Closed-loop Identification

It is clear that the low gain information needs to be excited more in order to obtain good estimates. To this end, strongly correlated test inputs with larger amplitudes are needed. One way to create this correlation is to use feedback in a closed-loop test. The need for closed-loop experiments in such a case is discussed in Chou *et. al.* (1999) and Jacobsen (1994). When controller gain is sufficiently high, the following approximation for process input signals is derived

(Chou *et. al.*, 1999):

$$\begin{aligned} u(i\omega) &\approx V(i\omega)\Sigma^{-1}(\omega)W^T(-i\omega)r(i\omega) \\ &= [v_1(i\omega) \quad v_2(i\omega)] \begin{bmatrix} \frac{1}{\sigma_1(\omega)} & \\ & \frac{1}{\sigma_2(\omega)} \end{bmatrix} \begin{bmatrix} w_1^T(-i\omega) \\ w_2^T(-i\omega) \end{bmatrix} \begin{bmatrix} r_1(i\omega) \\ r_2(i\omega) \end{bmatrix} \end{aligned}$$

where $r(i\omega)$ denotes the vector of setpoint signals. One can see that, for an ill-conditioned process, because $\sigma_1(\omega) \gg \sigma_2(\omega)$, the inversion of the singular value matrix will make the input signals contain mainly the low gain direction component.

We used two PI controllers as proposed in Chou *et. al.*, (1999). They are tuned such that both output directions of the plant are sufficiently excited ($K = 20$, $T_i = 20$). As the reference signals, two independent GBN signals were used and their average switch times were both 300 minutes. Figure 10.4.5 shows the closed-loop test data in the output space. Comparing them with Figure 10.4.4 we can see that the low gain direction is excited as well as the high gain direction.

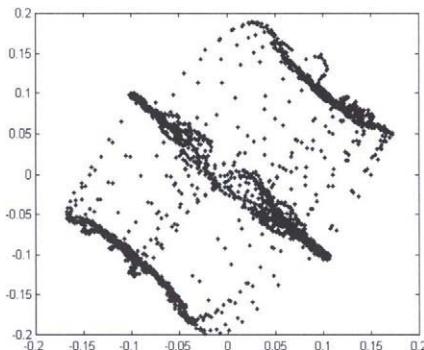


Figure 10.4.5 Excitation of output directions in the closed-loop test

All the above-mentioned estimation methods have been tested. It is soon turned out that the SISO structure used by the Box-Jenkins method is not suitable for the problem; the low gain direction cannot be identified with sufficient accuracy. Therefore, only the results of MISO and MIMO structures are shown below.

1) Two 4th order MISO ARMAX model, SNR = 10

Two 4th order MISO ARMAX models are estimated using the closed-loop data. The results are shown in Figures 10.4.6 and 10.4.7. The estimates are very good and the control performance

of the model is very high. The models are a little bit biased as can be seen from the plots. Numerical convergence problems are encountered when higher than 4th order ARMAX models are used.

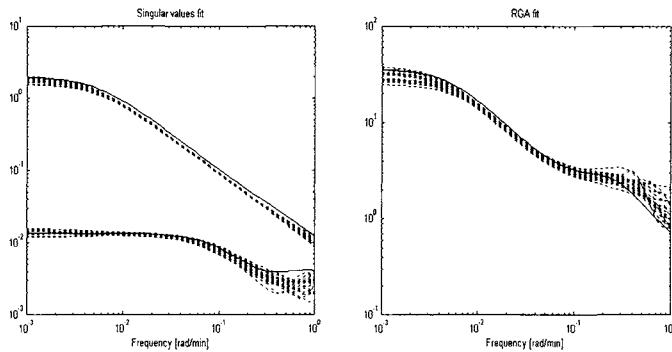


Figure 10.4.6 Singular values (lest) and RGA's of 2x4th order MISO ARMAX models from 20 simulations, SNR = 10. Solid lines are the true values

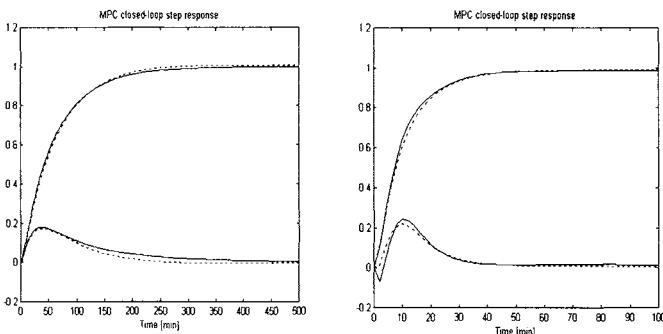


Figure 10.4.7 Step responses of closed-loop MPC systems. Left: the desired closed-loop time constant is 60 minutes. Right: the closed-loop time constant is 10 minutes. Solid lines: true values; dashed lines: 2x4th order MISO ARMAX model simulation

2) Two 2nd order MISO state space model (MOESP method), SNR = 10

The results are shown in Figures 10.4.8 and 10.4.9. The results are worse than those of the MISO ARMAX model. We have also experienced some large estimation errors. Two models had a positive sign of the determinant of the gain matrix while it should have been negative.

The models are biased. Numerical problem occurred and errors become larger when using higher orders.

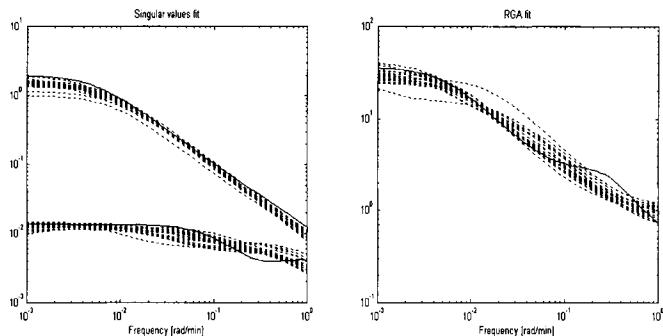


Figure 10.4.8 Singular values (lest) and RGA's of 2x2nd order MISO state space models from 20 simulations, SNR = 10. Solid lines are the true values

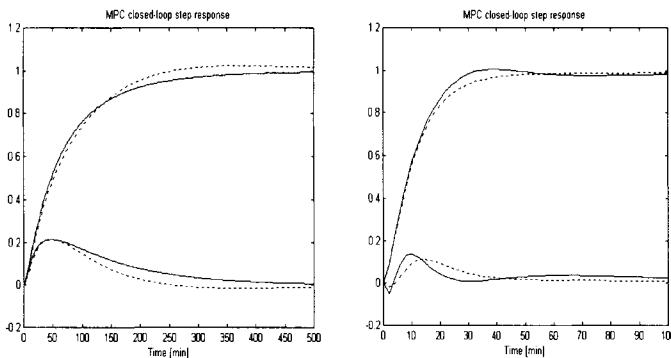


Figure 10.4.9 Step responses of closed-loop MPC systems. Left: the desired closed-loop time constant is 60 minutes. Right: the closed-loop time constant is 10 minutes. Solid lines: true values; dashed lines: 2x2nd order MISO state space model simulation

3) MIMO 2nd Model (MOESP method), SNR = 10

The results are better than the MISO state space model; but slightly worse than the MISO ARMAX model; see Figures 10.4.10 and 10.4.11. Some bias can be seen from the plots. However, using a MIMO model with higher orders resulted in a number of numerical problems and large estimation errors.

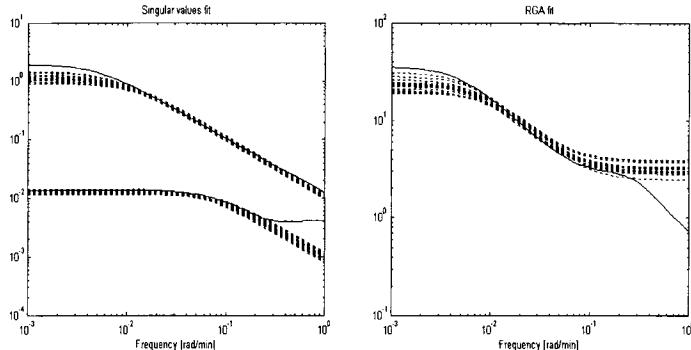


Figure 10.4.10 Singular values (lest) and RGA's of 2nd order MIMO state space models from 20 simulations, SNR = 10. Solid lines are the true values

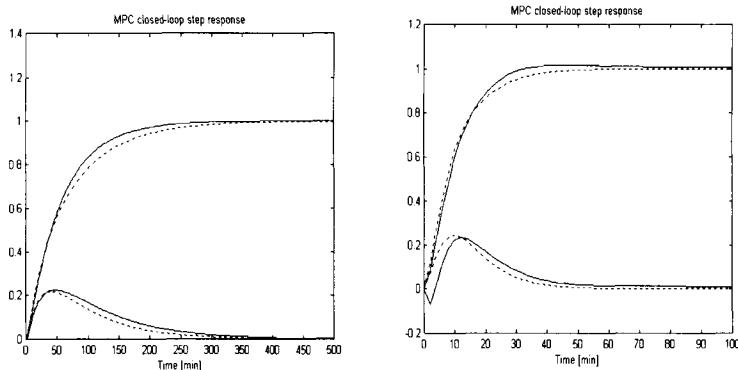


Figure 10.4.11 Step responses of closed-loop MPC systems. Left: the desired closed-loop time constant is 60 minutes. Right: the closed-loop time constant is 10 minutes. Solid lines: true values; dashed lines: 2nd order MIMO state space model simulation

Summary of Closed-loop Identification

- If the test data contains good information about low gain direction, both MISO and MIMO models can be used to identify ill-conditioned processes.
- For the same estimation method, in our case the subspace method MOESP, a MIMO model structure is better than a MISO structure.

- The MISO ARMAX model has the highest quality. This may be due to its lower variance for closed-loop data.
- A SISO model structure is not suitable for ill-conditioned process identification.
- The ARMAX method has numerical problems when higher than 4th order models are used; the subspace method has numerical problems when order is higher than 2. One can conclude that ARMAX is numerically more robust than the subspace method for the given problem.

10.4.3 A Simple Open-Loop Test Method

If feedback controllers are not available, one can design an open-loop test that excites more information about the low gain direction. In order to increase the SNR in the low gain output direction, one can replace the standard uncorrelated input signals with low gain directed signals of much higher amplitudes. Koung and MacGregor (1993) have suggested an optimal design method that minimizes the mismatch in multivariable model structure, where the knowledge of low gain direction is needed for the design (not practical). Jacobsen (1994) suggested a design method using an estimate of low gain direction. Both methods result in highly correlated test signals that are almost linearly dependent.

These studies show an important fact: the low gain input direction of an ill-conditioned process is very near the direction $[1, 1]^T$, the direction that the two test signal are linearly dependent. This fact can be used to simplify the test design. We propose the following test method that is much simpler to implement. The test consists of two steps.

Step 1. Test for high gain direction

Perform an open-loop test using two uncorrelated signals. Estimate a model using the data. The test need not be very long, because good estimation of high gain direction is easy to obtain.

Step 2. Test for both high gain and low gain directions

Perform an open-loop test using two correlated signals. The test signals have two kinds of periods, namely, low amplitude uncorrelated periods and high amplitude identical signal periods. The high amplitude and identical periods are used to create the strong correlation needed to identify the low gain direction. The small amplitude and independent periods are for identifying the high gain direction and for the persistent excitation condition. Note that the signals are not persistently exciting for the MIMO problem if they are 100% correlated or linearly dependent. In a high amplitude period, the sign of the two signals should be so that they move each output in opposite directions. The ratio of the two amplitudes should be that their effects at the two outputs nearly cancel out. This can be determined from the model identified from Step 1. The two signals should have the same sign for distillation columns, because increasing reflux has opposite effect as increasing the steam (boilup). The size of the high amplitudes can

be obtained by gradually increasing them up to input amplitude constraints, or until output variation becomes large enough.

Figure 10.4.12 shows the test signals of Step 2 where the high amplitudes are 20 times those of the low amplitudes. Note that the high amplitudes of identical inputs will not cause large variations of the outputs due to the strong interactions of the process. In Figure 10.4.13 one can see that the low gain direction is excited more than a normal open-loop test. This test method is easy to implement and also easy to comprehend. It requires only knowledge about the high gain directions of the process which is easily obtained. No quantitative knowledge about low gain direction is necessary, because we know it is near $[1, 1]^T$.

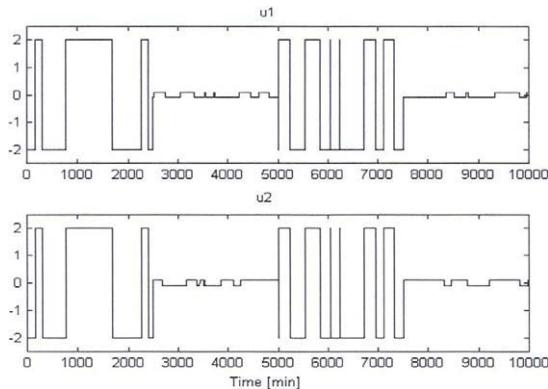


Figure 10.4.12 Open loop test signals using low amplitude uncorrelated signals and high amplitude identical signals

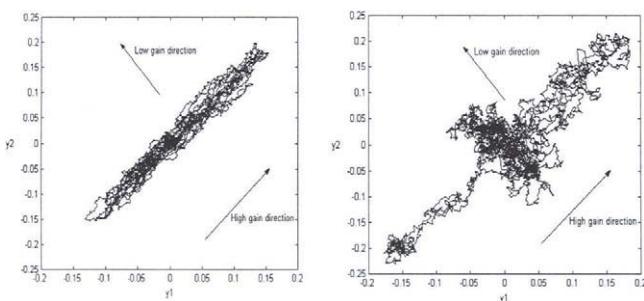


Figure 10.4.13 Output directions of a normal open loop test (left) and proposed open loop test (right)

Simulation results are shown below.

1) Two 4th order MISO ARMAX model, SNR = 10

The results are shown in Figures 10.4.14 and 10.4.15. The fit of the singular values and the RGA, especially in the medium frequency range, is good. On the other hand, poor estimates can be observed in the high frequency range. This does not affect the control performance. The model error at high frequencies can be reduced by adding some high frequency components in the test signal, which is easy to implement.

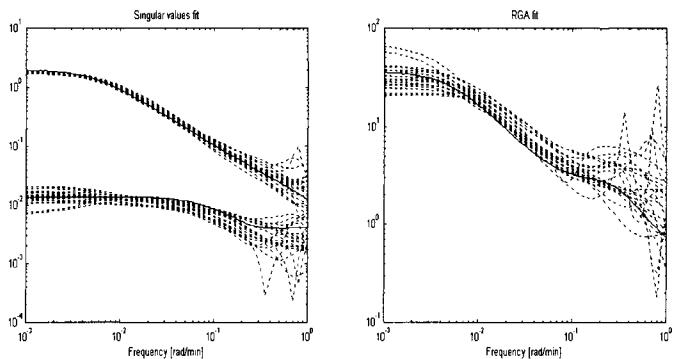


Figure 10.4.14 Singular value plots (left) and RGA plots of 2×4 th order MISO ARMAX models from 20 simulations, $\text{SNR} = 10$. Solid lines: true values. Dashed lines: estimates.

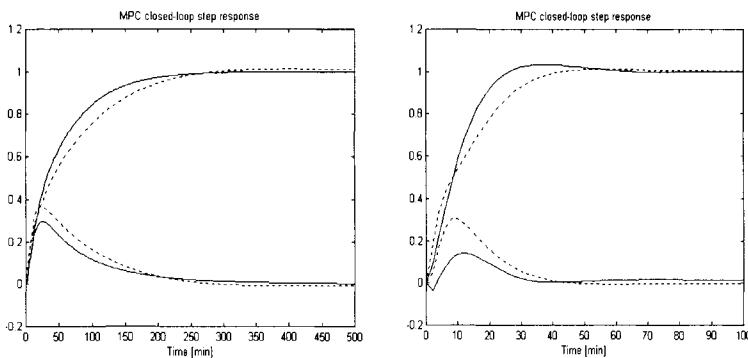


Figure 10.4.15 Step responses of closed-loop MPC systems. Left: the desired closed-loop time constant is 60 minutes. Right: the closed-loop time constant is 10 minutes. Solid lines are the true values and dashed lines are 2×4 th order MISO ARMAX model simulations.

2) Two 2nd order MISO state space model (MOESP method), SNR = 10

The results are shown in Figures 10.4.16 and 10.4.17. We have experienced some numerical problems during parameter estimation. We were not able to obtain any model at all for some simulation runs because of the lack of convergence. Thus, there are fewer than 20 models presented in figure 10.4.16.

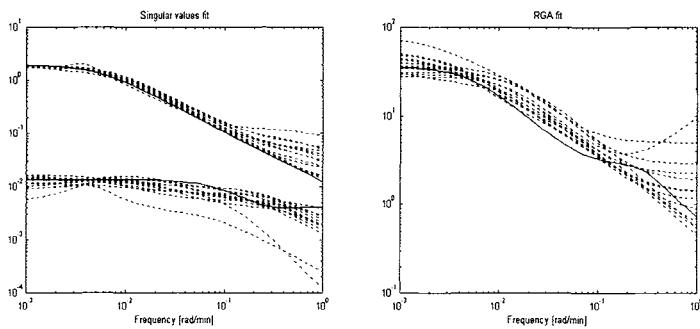


Figure 10.4.16 Singular value plots (left) and RGA plots of 2x2nd order MISO state space models from some (fewer than 20) simulations, S/N = 10. Solid lines: true values. Dashed lines: estimates.

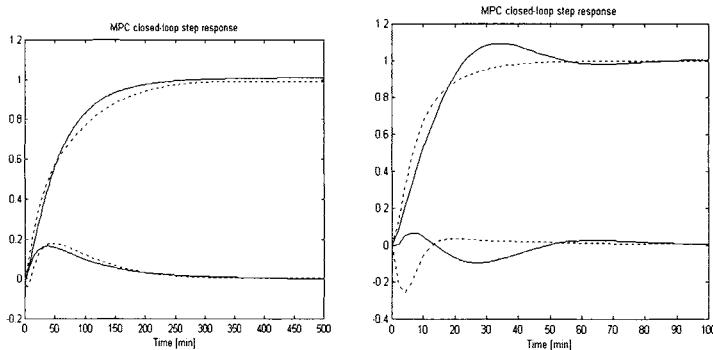


Figure 10.4.17 Step responses of closed-loop MPC systems. Left: the desired closed-loop time constant is 60 minutes. Right: closed-loop time constant is 10 minutes. Solid lines: true values. Dashed lines: 2x2nd order MISO state space model simulations.

3) MIMO 2nd order state-space model (MOESP method), SNR = 10

The results are shown in Figures 10.4.18 and 10.4.19. The subspace method using the MIMO structure has good estimation of process models, but it fails occasionally. If it succeeds, model variance is smaller than that of the MISO ARMAX model. Control performance is slightly better than that of the MISO ARMAX model.

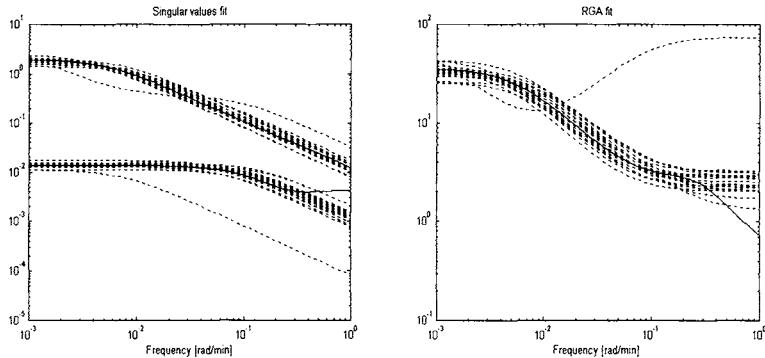


Figure 10.4.18 Singular value plots (left) and RGA plots of 2nd order MIMO state space model from 20 simulations, SNR = 10. Solid lines: true values. Dashed lines: estimates.

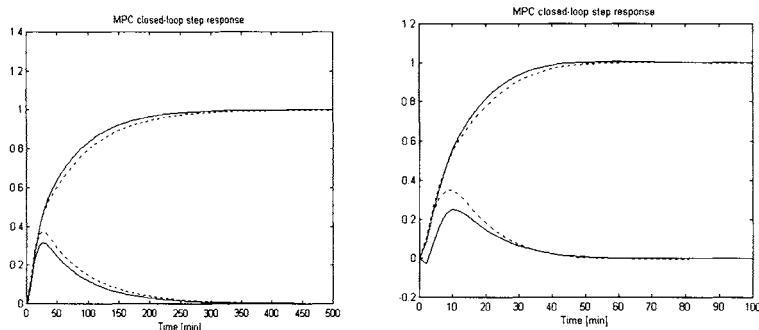


Figure 10.4.19 Step responses of closed-loop MPC systems. Left: the desired closed-loop time constant is 60 minutes. Right: closed-loop time constant is 10 minutes. Solid lines: true values. Dashed lines: MIMO 2nd order state space model simulations.

Summary on the new open-loop test

- For the open-loop identification, the MIMO state space model performs slightly better than the MISO ARMAX model in terms of variance. This can be attributed to the advantage of MIMO structure.

- Again, the SISO model structure is not suitable for the open-loop test. Therefore, a SISO structure should be avoided in identifying ill-conditioned processes.
- The prediction error method (ARMAX) is numerically more reliable and robust than the MOESP method for the given problem. Outliers (few very poor models) are obtained using state space models.

An ill-conditioned problem is a specific problem for multivariable processes. It has been shown that closed-loop tests and correlated inputs are better than open-loop tests and uncorrelated inputs.

10.5 Identification of a Crude Unit for MPC

The crude unit consists of three distillation columns in series: an atmospheric tower, a splitter and a stabilizer column. Two DMC controllers have been installed for the crude unit, one for atmospheric column and one for splitter/stabilizer. In this section we will only discuss the first controller. Figure 10.5.1 shows a simplified process flow diagram of the atmospheric tower. The column performs the initial distillation of the crude oil into various boiling range fractions. The column has four side draws each with its own side draw stripper:

- Kerosene (side draw 1)
- In-line blended automotive diesel (AD) (combined side draws 1,2,3,4 and an intermediate gas oil (IGO) from the vacuum unit)
- Heavy gasoil (side draw 4).

The column has whole straight run naphtha (WSR) as the top product and a bottom residue is the feedstock for the vacuum unit. Moreover the column has a top reflux flow and three pump-around flows of which the top pump-around (TPA) and bottom pump-around (BPA) exchange heat with stabilizer and splitter columns respectively.

The operating economics of a crude unit generally dictate that the unit should run as close as possible to the product specifications. More specifically the DMC control objectives are:

Minimum naphtha mode:

- Minimize naphtha flow.
- Maximize kero flow at minimum flashpoint and maximum 90% point specifications.

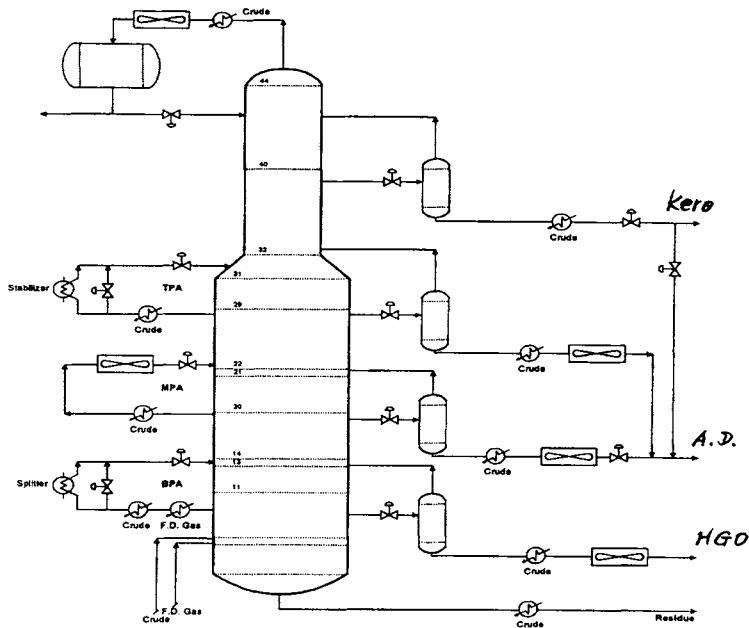


Figure 10.5.1 Simplified flow diagram of crude atmospheric tower distillation column

Off minimum naphtha mode:

- Maximize naphtha flow at maximum 90% point specification of the HSR-naphtha.
- Kero 90% point at maximum specification. Flashpoint is no longer a control target.

Always:

- Maximize automotive diesel yield up to process limits and cloudpoint specifications at the expense of heavy gasoil (HGO).
- Maximize HGO yield up to process limits and color specification at the expense of residue yield.
- Maximize heater outlet temperature.
- Maximize crude feed flow up to process limits.

- Minimize column pressure.
- Maximize TPA flow to maximize preheat up to stabilizer energy needs.
- Maximize BPA flow to maximize preheat up to splitter needs.

In order to achieve above mentioned control objectives, the following MV's, DV's and CV's have been selected:

Manipulating variables (MV's):

1. 40FRC62S: Crude feed flow rate
2. 40FRC63S: Crude feed flow rate
3. 40FRC96S: Crude feed flow rate
4. 40TRC50S/60S: Combined coil outlet temperature
5. 40FC184S: 1st side draw flow rate
6. 40FC185S: 2nd side draw flow rate
7. 40FC186S: 3rd side draw flow rate
8. 40FC187S: 4th side draw flow rate
9. 40FRC18S: 1st side stripper steam flow rate
10. 40FRC14S: 2nd side stripper steam flow rate
11. 40FRC15S: 3rd side stripper steam flow rate
12. 40FRC38S: Top reflux flow rate
13. 40FRC10S: Light reflux flow rate
14. 40FRC90S: Medium reflux flow rate
15. 40FRC11S: Heavy reflux flow rate
16. 40FRC35S: Splitter feed flow rate
17. 40FRC79S: 1st side flow rate to automotive diesel
18. 40FC170S: 4th side flow rate to automotive diesel
19. 2FRC22S: IGO flow rate to automotive diesel

Disturbance variables (DV's):

1. 40PR8: Steam pressure
2. 40SRC4: Compressor feed
3. 40FR98: Flash drum off-gas flow

Controlled variables (CV's):

1. 2FRC22X: IGO flow valve position
2. 2LRC4X: IGO drum level control valve position
3. 40FRC79X: 40FRC79 valve position
4. 40FRC79S: Flow rate to MEROX
5. 40FRC9X: 40FRC9 valve position

6. 40FC170X: 40FC170 valve position
7. 40FRC1S: 1st side draw to tank flow rate
8. 40FRC1X: 40FRC1 valve position
9. 40LRC9X: 1st side stripper level control output
10. 40LRC7X: 2nd side stripper level control output
11. 40LRC8X: 3rd side stripper level control output
12. 40LRC6X: 4th side stripper level control output
13. 40LRC5X: Bottom level control output
14. 40FRC7S: Residue flow rate
15. 40FC184: 40FC184 valve position
16. 40FC185: 40FC185 valve position
17. 40FC186: 40FC186 valve position
18. 40FC187: 40FC187 valve position
19. 40PCT4/40TRC4: Top temperature
20. 40AR4: Kero flashpoint
21. 40KERO90/40TBPSC1: Kero 90% point
22. 40AR6: Added diesel cloud point
23. 40GASOIL: Total gasoil flow
24. 40AR46: 4th side color
25. 40LRC16: D-2 level
26. 40TRC11X: Splitter reboiler bypass flow
27. 40TRC13X: Stabilizer reboiler bypass flow
28. 40FRC62X: 40FRC62 valve position
29. 40FRC63: 40FRC63 valve position
30. 40FRC96: 40FRC96 valve position
31. 40CRDHVO: Total crude feed
32. Furnace A duty
33. Furnace B duty
34. 40FZDTO: Flash zone delta temperature
35. 40AR5/40TBPHSR: HSR 90% point
36. 40PRC15: Suction pressure 40K30

Initially a single variable step test approach was used. The step test took 14 days around the clock. The identified models were used in the DMC controller and the control performance was not satisfactory. It was believed that model quality was one of the causes of control problems. This does not mean that the traditional identification approach cannot solve the problem. If longer test was used, the model quality could have been high enough for the DMC control. But a longer test was difficult to plan due to the circumstances. This situation motivated the search for more effective and efficient identification approaches. The following are the results of ASYM identification.

A. Identification Tests

It was decided that 13 of the 19 MV's will be tested and their models identified using ASYM. Independent GBN signals are used as test signals. The average switch time of all the GBN signals are set to 50 minutes. The amplitudes of the GBN signals are chosen such that they will generate data with good enough signal-to-noise ratio but will not disturb the product quality. Two open-loop tests were designed and carried out

Test 1, MV Amplitude (top/top)

40TRC50S	2 °C
40FC184S	200 M3/D
40FC185S	200 M3/D
40FC186S	200 M3/D
40FC187S	200 M3/D
40FRC35S	0.4 KM3/D
40FRC79S	150 M3/D
40FC170S	150, 100 M3/D
40FRC22S	100 M3/D

Test 2, MV Amplitude (top/top)

40FRC38S	0.2 KM3/D
40FRC10S	0.2 KM3/D
40FRC90S	0.3 KM3/D
40FRC11S	0.2 KM3/D
40FRC35S	0.6 KM3/D

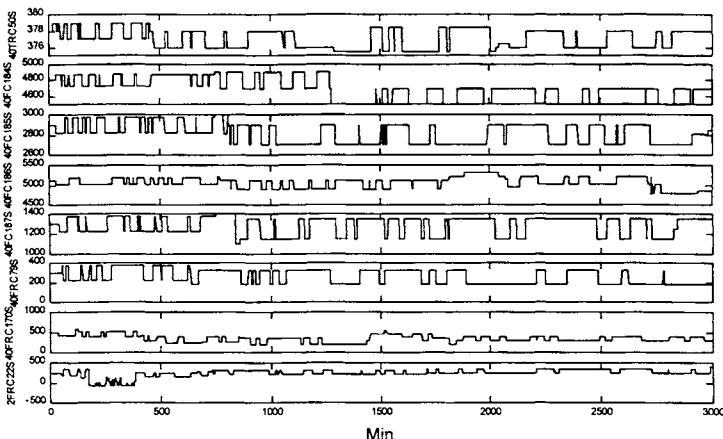


Figure 10.5.2 Part of the MV's during Test 1

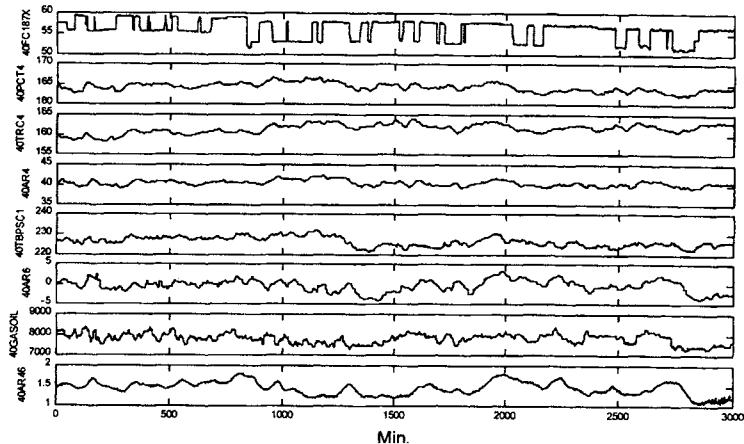


Figure 10.5.3 Part of the CV's during Test 1

Each test lasted for about two days, so the total test time was 4 days. The tests did not cause any product quality problems. Figure 10.5.2 shows the plots of 8 MV's from Test 1; Figure 10.5.3 shows some of CV's of Test 1. During the test the operators have adjusted the average setpoints for many MV's in order to maintain stable unit operation.

B. Process Models and Model Validation

Identification calculation using ASYM is straightforward. In Figure 10.5.4 the model step responses of 8 CV's for Test 1 are plotted; in Figure 10.5.5 their frequency responses and upper error bounds are plotted.

The models are graded according to the relative sizes of their upper bounds. More specifically, identified transfer functions are graded in A (very good, bound $\leq 30\%$ model), B (good, $30\% \text{model} < \text{bound} \leq 60\% \text{model}$), C (marginal, $60\% \text{model} < \text{bound} \leq 90\% \text{model}$), and D (poor, or, no model exists, $\text{bound} > 90\% \text{model}$).

Based on extensive simulations and project experience, A grade and B grade models can be used in the controller, provided that the process is not very ill-conditioned for important CV's. C grade and D grade models are not relevant for MPC control and they are treated as follows:

1. Zero them when there are no relationships between the MV/CV pairs. This can be determined by using the process knowledge and cross checking.
2. If a transfer function is expected and needed in the control, redesign a test or modify the on going test in order to improve the accuracy of these models.

Using the upper bound formula (7.2.13) we can easily give some guidelines for improving the test design:

- doubling the amplitudes of test signals or quadrupling the test time will halve the error over all frequencies;
- doubling the average switch time of GBN signals will halve the error at low frequencies and double the error at high frequencies.

Model validation results using upper error bounds agreed very well with the process knowledge. Most of the A and B grade models are used in the DMC controller.

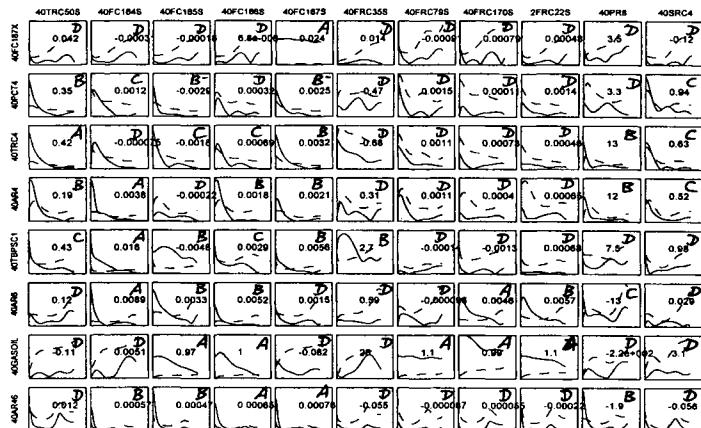
C. DMC Commissioning and its Performance

The identified models are very accurate for the working range tested. As a result, the DMC controller can be tuned very fast. The average MV move suppression factor (MV weighting) is about 1. After the model identification, it took two weeks to commission the atmospheric column DMC controller. The controller has been on-line since then. Control results were pointing at a much higher yield pattern at the expense of residue. The variance of the product qualities has been reduced dramatically.



Time [min], total 200.

Figure 10.5.4 Step responses of part of the models from Test 1. The numbers in the middle of the plots are the model gains



Norm. freq [rad/min], total 0.785398.

Figure 10.5.5 Frequency responses and error bounds of part of the models from Test 1.

10.6 Closed-Loop Identification of a Deethanizer

The process is a Deethanizer which belongs to the cold side of an ethylene unit. The deethanizer separates C2 and lighter from C3 and heavier. The light product leaves the column overhead as vapor distillate and the heavy product exits the column as bottom liquid flow. The column operates in a high purity range.

A DMC controller was designed and to be commissioned as part of an APC and optimization project. The purpose of the deethanizer DMC is to reduce the variations of product qualities while respecting process operation constraints.

MV's of the controller:

- Reflux: Reflux flow setpoint
- Steam: Reboiler steam flow setpoint
- Preheater: Feed preheater flow setpoint

DV:

- Feed: Column feed flow

Main CV's of the controller:

OverheadC3: Overhead C3 composition
DeltaPress: Column pressure difference
BotTemp: Bottom temperature
TopTemp: Top temperature
TrayTemp: A tray temperature

The experience of column operation has shown that the tray temperature TrayTemp is an important variable. It should be controlled in a range of 6 degrees for normal operation. When the tray temperature is outside this range, it will be difficult to control the column. This can be due to process nonlinearity.

Initially, an open-loop multivariable test was carried out; see the left sides of Figures 10.6.1 and 10.6.2. Independent GBN signals were used as test signals. The following have happened during the open-loop test:

- The tray temperature varied in a range of about 20 degrees, which was far beyond the normal operation range. The MV amplitudes determined during the pretest session turned out to be too big.
- The tray temperature became too high at about sample 600. The operator closed the PI control loop in order to bring it back. The control action reduced the steam flow to a very low level, which may excited nonlinearity in the data.
- The preheater flow could not be moved during the most part of the test period due to high level of disturbance introduced by reflux and steam.
- The column pressure difference became too high after sample 2200, which indicated the column was in flooding. The overhead C3 composition increased during this period.
- The second half of the test had disturbed normal unit operation.

The data have been used for identification using several software packages. Data slicing was used to remove the portion during the column flooding. When the identified models were used in the DMC controller, the closed-loop performance was not satisfactory. During the control commissioning it was quickly decided that the best solution would be to retest the plant in order to obtain a better data set.

Remark: The difficulty of the test is caused more by the high purity characteristics of the column than by the multivariable test approach. A single variable open test will face the same problem for high purity distillation columns. In general a well designed multivariable test will not cause more disturbance than a conventional single variable step test as can be seen from the previous section.

Finally, it was decided to carry out a partial closed-loop test with the tray temperature controlled by the steam flow using an existing PI controller; see the right hand sides Figures 10.6.1 and

10.6.2 for the test data. In the test, independent GBN signals are applied at reflux, preheater and the tray temperature setpoint. The closed-loop test can be summarized as follows:

- CV variations were much smaller than the CV variations of the open-loop test. The tray temperature was kept within a range of 7 degrees during the closed-loop test.
- All the three MV's could be tested according to the plan. The test did not disturb normal operation.
- Much less operator intervention had taken place, which implies easy test.
- During the second half of the test, the feed flow had to be reduced by about 20% due to production planning. The operator could handle this easily, thanks to the tray temperature controller.

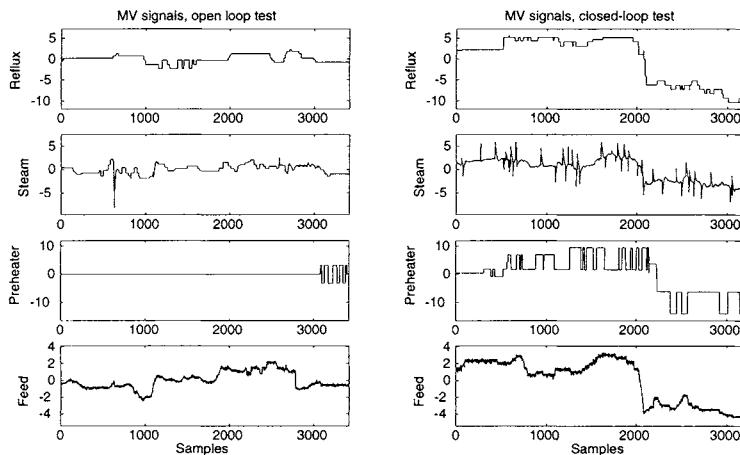


Figure 10.6.1 MV plots of the open loop test and closed-loop test. The data are normalised by subtracting their mean values and by dividing by some factors. For comparison, the same scaling factor is used for both open-loop test and closed-loop test for each MV

The ASYM method was used to identify the models using the closed-loop data. The DMC commissioning using the closed-loop model went smoothly and the controller has been on-line since then without major problems.

Due to the change of feed flow during the closed-loop test, the usable data for identification is less than desired. Fortunately, the model from this very short test period is good enough for control.

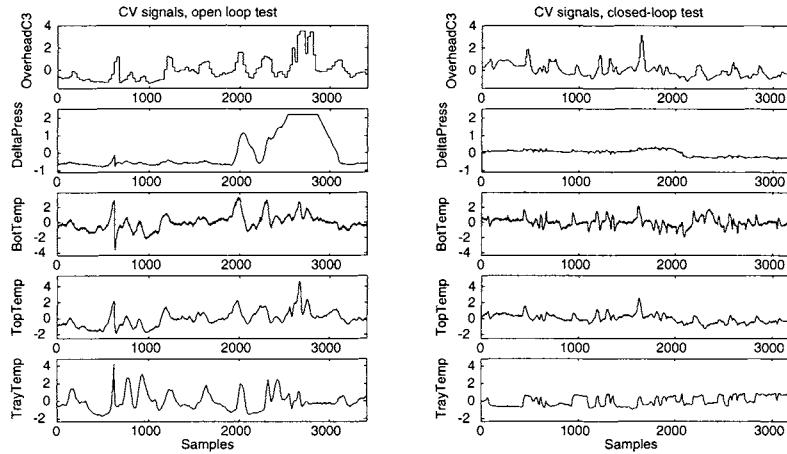


Figure 10.6.2 CV plots of the open loop test and closed-loop test. The data are normalised by subtracting their mean values and by dividing by some factors. For comparison, the same scaling factor is used for both open-loop test and closed-loop test for each CV

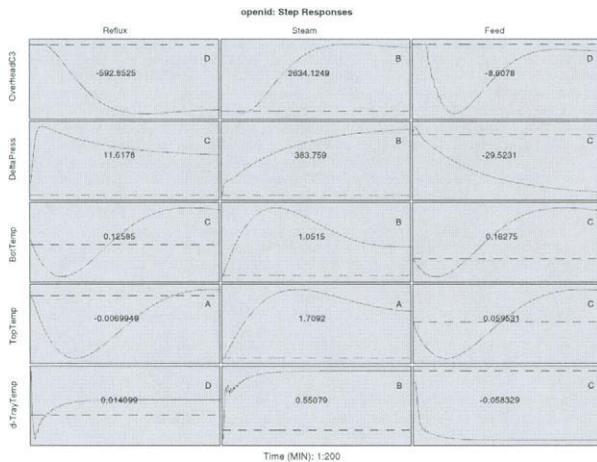


Figure 10.6.3 Model step responses identified using open loop data

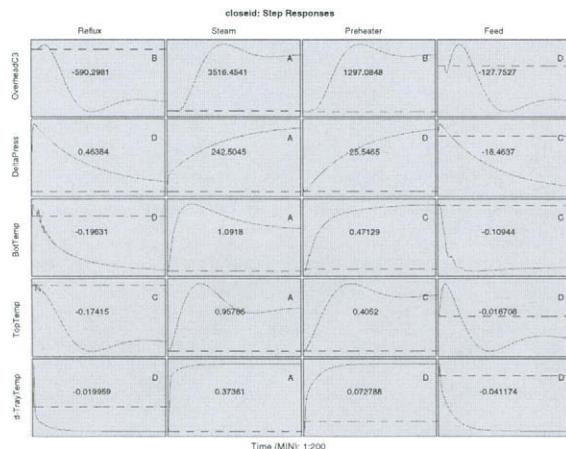


Figure 10.6.4 Model step responses identified from closed-loop data

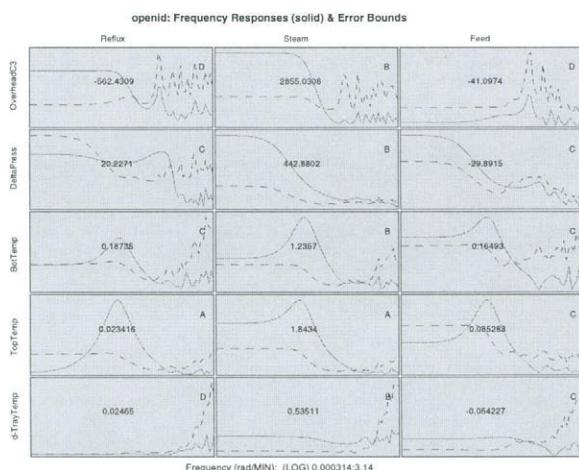


Figure 10.6.5 Model frequency responses and upper bounds. Model identified using open loop data

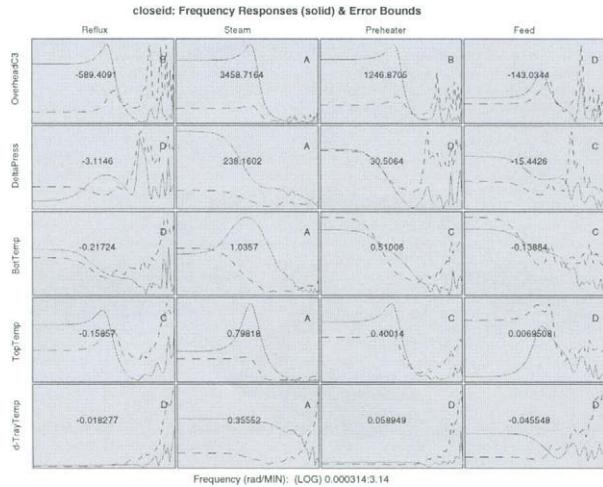


Figure 10.6.6 Model frequency responses and upper bounds. Model identified using closed-loop data

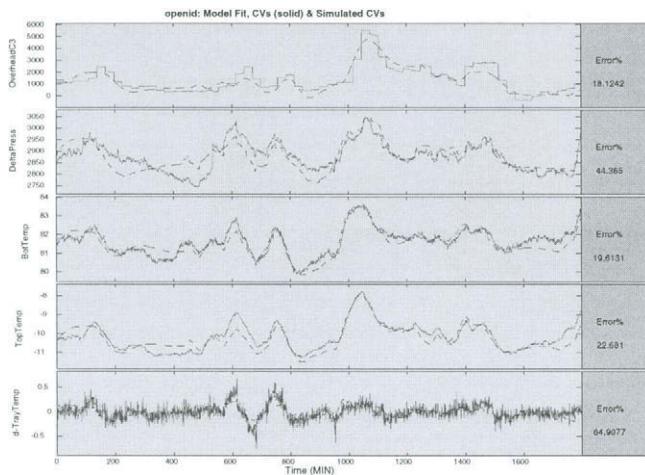


Figure 10.6.7 Model fit of the open loop model

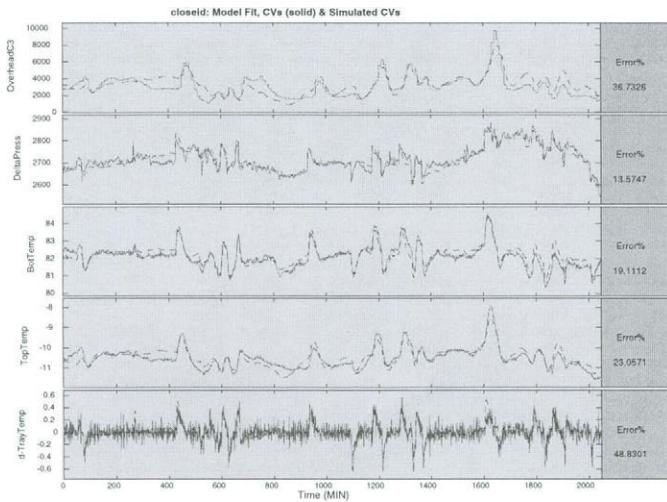


Figure 10.6.8 Model fit of the closed-loop model

For comparison, the identification results of the open-loop data and the closed-loop data are shown in Figures 10.6.3 to 10.6.8. Note that the tray temperature TrayTemp is identified as an integral process and the models of its derivative are shown. One can see that the quality of the closed-loop model is higher than that of the open-loop model according to the upper error bounds. The difference can also be seen using the process knowledge. For example, the gains between the reflux and both top and bottom temperatures should be negative. The closed-loop model correctly determines these. However, the gain of the open-loop model is positive for the bottom temperature and is nearly zero for the top temperature. The overall model fit of the closed-loop model is slightly better than that of the open-loop data.

10.7 Conclusions and Perspectives

In this chapter identification is viewed from an industrial process control perspective. It is shown that identification is, though very important, a very small part of an industrial MPC project. To carry out an MPC project remains a very challenging task. To understand the production economics and unit operation, and to communicate with operating personnel are equally important as the technical knowledge of identification and control.

MPC technology has been widely and successfully applied in the refinery/petrochemical industry. Current experience shows that process modeling and identification become a bottleneck during an MPC project, due to the high costs in tests and difficulties in model identification and validation.

A modern identification approach can solve the problem. The elements of the *modern* approach are automatic multivariable closed-loop tests, parametric models and error bounds for validation. These are not new concepts for academic researchers, but they are not yet widely applied in MPC projects. The benefits of the new approach are: improved model quality; reduced disturbance to unit operation; reduced time (over 60%) in tests and data analysis; and user friendliness. Due to the reduced cost of identification, more process units will be justified for MPC applications. Besides developing models for MPC, identification can also be used for tuning regulatory control loops and for developing inferential models.

The development of identification can also have an impact on MPC controllers. Adaptive or semi-adaptive MPC control can be realized if an identification method is implemented in a recursive form. Nonlinear MPC can be used if nonlinear models are identified using, for example, block-oriented models.

Unlike the refinery/petrochemical industry, other process industries do not use MPC or multivariable control as a standard technology. One reason is that to develop and maintain a multivariable control system is very costly and it requires personnel with very high skills in modeling and control. If the benefit of a multivariable control system are not substantially higher than its cost, it will not be justified economically. Another reason can be that the potential benefit of multivariable control is not fully realized in many industries. Most multivariable controllers are model based. Often the main cost and difficulty of a multivariable control project is due to modeling and identification. It is the hope of the author that, by the introduction of more efficient identification methods, more industries will apply multivariable control technology.

Chapter 11

Model Based Fault Detection and Isolation

By Qinghua Zhang

The very initial purpose of system identification, as developed in the field of automatic control, is related to controller design. However, as a theory for model fitting and parameter estimation, it has many other applications, inside or even beyond the field of automatic control. In the hierarchy of modern automation systems as described in Chapter 1, there is a layer for diagnosis and supervision. The approaches already developed for this purpose can be roughly classified into the fields of artificial intelligence and control system theory. The approaches in the second category are usually *model based*, as they exploit the knowledge expressed by a mathematical model. In this chapter, we present some model based approaches to fault detection and isolation for the purpose of diagnosis and supervision. The relation of these approaches with system identification is twofold: on the one hand, they require a nominal model characterizing the fault-free system, often obtained by system identification; on the other hand, the used algorithms are sometimes inspired by those of system identification.

11.1 Introduction

The increasing complexity of modern engineering systems has motivated the development of fault detection and isolation (FDI) approaches, due to the requirement on reliability, availability, cost efficiency, human and material safety in such systems. The problem of FDI is important not only for safety-critical systems (nuclear systems, aircrafts, etc.), but also for many modern industrial automation systems. There are typically two kinds of applications of FDI: one is to quickly react to sudden failures for the purpose of safety, the other is to monitor slow degradations for the

purpose of maintenance optimization. In this chapter we use the term *fault* to indifferently refer to failures and degradations.

The common idea for fault detection is to check the consistency related to some redundancy in the monitored system. A primitive approach is to use physical redundancy. For example, two or more sensors may be installed to measure the same signal. If the outputs of these sensors are not consistent, then a fault in at least one of the sensors is detected. This approach is effective and conceptually simple, but its implementation may be excessively expensive and cumbersome. Alternatively, the *model-based* approaches exploit the redundancy provided by a mathematical model of the monitored system. As a model describes some relationship among system variables, checking the consistency of the model with the measurements provides solutions to FDI. Typically, the (parametric) model-based approaches assume that the faults correspond to changes in some parameters of the considered mathematical model. The problem of FDI thus amounts to the detection and isolation of parameter changes. Model-based FDI approaches have been studied for more than two decades. Recently, due to the growing need for automatic supervision of industrial systems, researches on FDI have been rapidly developed, as demonstrated by a large number of publications, for instance, the survey papers of Frank (1990), [40] Isermann (1993) [61] and Bassseville (1998) [10], and the books Patton *et. al.* (1989), [86]Basseville and Nikiforov (1993), [13]Gertler (1993) [43] and Chen and Patton (1999). [22]

Now it is widely acknowledged that the problem of FDI can be split into two steps: *residual generation* and *residual evaluation*. In this chapter, both generation and evaluation of residuals are addressed. A (FDI) residual is a transformation of raw data, ideally being zero in the fault-free case, and different from zero in the faulty case. For example, if it is known that some measured variables y_1, \dots, y_m are governed by an equation $f(y_1, \dots, y_m) = 0$, then $r = f(y_1, \dots, y_m)$ can be considered as a residual candidate. Note that this residual r is based on a redundancy among the variables y_1, \dots, y_m stated by the equation $f(y_1, \dots, y_m) = 0$. In a complex system, it is sometimes possible to find such equations concerning a small part of the system. The resulting residuals make the problem of fault isolation easier. In more general situations, a mathematical model may involve some unmeasured variables. The design of residuals is then less straightforward, as a residual must be evaluated from measured signals for the purpose of FDI.

Another practical difficulty is that a residual is never strictly zero, even in the fault-free case, due to various measurement and modeling errors. For this reason, residual evaluation is not a trivial task for the purpose of FDI. Typically the designed residual should be close to zero in the fault-free case, and significantly different from zero (in some sense) in the faulty case. Statistical methods are often used for residual evaluation due to the uncertainties contained in residuals. The application of statistical methods often requires a stochastic model describing the uncertainties in the monitored system. Typically, the detection and isolation of additive faults in linear dynamic systems are considered with a fully stochastic modeling (see Section 11.2.2). The theory for FDI in this situation is quite complete. In other situations, in particular when nonlinear dynamic systems are considered, a fully stochastic approach is much more difficult. For this reason, the detection and isolation of faults in nonlinear systems are often studied in a

deterministic framework. The evaluation of residuals generated through a purely deterministic approach has more difficulties. In Section 11.4.2, we describe a method for residual evaluation based on very restricted knowledge on the stochastic properties of the residuals. This method can be efficiently combined with residuals generated through a deterministic approach, together with some mild assumptions.

11.2 Residuals for Linear Systems with Additive Faults

In this section, we present some well established results for linear time invariant systems. Based on well developed tools in linear system theory, residual generation for linear time invariant systems is the most mature part of the FDI theory. The current state of the art allows the design of residuals with known stochastic properties. We start by a short review on existing methods, then we focus on a particular design method of robust residuals.

11.2.1 Review of Existing Methods

In the beginning of the 1970s, the idea of using analytical redundancy provided by a mathematical model for generating residuals was already known (Beard, 1971, [14] and Jones, 1973 [64]). Since then, the detection and isolation of *additive* faults occurring in linear time invariant (LTI) systems have been extensively studied through both deterministic and stochastic approaches. Most methods for residual generation developed so far can be classified into two categories: parity relation based methods (Chow and Willsky, 1984 [27]) and observer based methods (Clark *et. al.*, 1975 [29] and Frank, 1987 [39]). There are also some transfer function based methods which are equivalent to methods of the above two categories (Gertler, 1988 [42]).

Parity Relation based Residual Generation

The concept of parity relation was first introduced in static models with parallel redundancy. For simplicity, let us ignore all measurement and modeling uncertainty. Assume that n physical variables, represented by $x \in \mathbb{R}^n$, are measured by m sensors, denoted by $y \in \mathbb{R}^m$, with the known relationship

$$y = Cx$$

where C is a known $m \times n$ matrix with full column rank. If $m > n$, there is some redundancy among the sensors. This redundancy can be exploited to generate residuals as follows. Let V be a matrix such that $VC = 0$. If the measurement equation $y = Cx$ is correct, then $Vy = VCx = 0$ for any value of x . If the measurement equation is not correct ($y = Cx$ is affected by an additive term or by some change in C), then one can expect that $Vy \neq 0$. Therefore, for some

appropriately designed matrix V , the residual $r = Vy$ can be used for detecting faults in these sensors.

For dynamic systems, parity relations based on temporal redundancy can be used for residual generation. This method is usually studied for discrete-time systems. Consider a linear discrete-time state-space system:

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (11.2.1)$$

where $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}^m$, $u(t) \in \mathbb{R}^p$, and A, B, C, D are constant matrices of appropriate sizes. Packing up model (11.2.1) from $t = k - s$ to $t = k$ yields

$$Y(k) - HU(k) = Ox(k-s) \quad (11.2.2)$$

with

$$Y(k) = \begin{pmatrix} y(k-s) \\ y(k-s+1) \\ \vdots \\ y(k) \end{pmatrix} \quad U(k) = \begin{pmatrix} u(k-s) \\ u(k-s+1) \\ \vdots \\ u(k) \end{pmatrix}$$

$$H = \begin{pmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{s-1}B & CA^{s-2}B & \cdots & D \end{pmatrix} \quad O = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^s \end{pmatrix}$$

Then similarly to the static case, for some matrix V satisfying $VO = 0$, a residual can be designed as

$$r(k) = V [Y(k) - HU(k)]$$

Such a residual is zero whenever equation (11.2.2) holds. The designed matrix V should also satisfy some fault sensitivity condition so that the faults to be detected affect the behavior of the residual $r(k)$.

Observer based Residual Generation

With observer based methods, the basic idea for residual generation is to estimate the state variable and the output of the monitored system, then the error between the estimated output and the measured output can be considered as a residual. It is useful to formulate the design of

observers in a general framework in order to exploit the design freedoms to generate residuals robust to disturbances. Consider a continuous-time state space system

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Assume that we are interested in estimating $Lx(t)$ with a given matrix L , instead of estimating $x(t)$ itself. Such a general observer can be formulated in the following form:

$$\begin{cases} \dot{z}(t) = Fz(t) + Ky(t) + Ju(t) \\ w(t) = Gz(t) + Ry(t) + Su(t) \end{cases}$$

where $z = Tx$ with some transformation matrix T . For $w(t)$ to be an asymptotic estimate of $Lx(t)$, i.e.,

$$\lim_{t \rightarrow \infty} w(t) = Lx(t)$$

the necessary and sufficient conditions on the design matrices F, K, J, G, R, S, T are (O'Reilly, 1983 [84] and Chen and Patton, 1999 [22])

$$\begin{cases} F \text{ has stable eigenvalues} \\ TA - FT = KC \\ J = TB - KD \\ RC + GT = L \\ S + RD = 0 \end{cases}$$

For the purpose of residual generation, let $L = C$ and $\hat{y}(t) = w(t) + Du(t)$. Then a residual is obtained as

$$r(t) = Q[y(t) - \hat{y}(t)]$$

with some chosen weighting matrix Q .

It is obvious that many freedoms are left on the choice of the design matrices. Typically these freedoms are exploited to design residuals with desired robustness to disturbances. Some well known design methods for robust residual design are based on *unknown input observers* and on *eigenstructure assignment*; see Chen and Patton (1999) [22]. There is also an elegant method for solving this problem with *descriptor Kalman filters* (i.e., Kalman filters for implicit linear dynamic systems); see Benveniste *et. al.* (1993) [15]. In the following we recall a particular method for robust residual design due to Nikoukhah (1994) [82]. The reason for this choice is that it is based on elementary algebraic operations and the designed residuals not only are robust to disturbances, but also have the property of a white noise in the fault-free case. The white property of the residuals is important for the design of efficient statistical evaluation methods.

11.2.2 Innovation Residuals

In this section, we describe a method for the design of a particular class of residuals due to Nikoukhah (1994) [82], called innovation residuals. Let us consider the continuous-time state space model with additive faults, noises and unknown disturbances:

$$\begin{cases} \dot{x}(t) = Ax(t) + B_1u(t) + B_2\nu(t) + B_3w(t) + B_4f(t) \\ y(t) = Cx(t) + D_1u(t) + D_2\nu(t) + D_3w(t) + D_4f(t) \end{cases} \quad (11.2.3)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $y(t) \in \mathbb{R}^m$ the outputs, $u(t) \in \mathbb{R}^p$ the inputs, $\nu(t) \in \mathbb{R}^q$ some disturbances (also called unknown inputs), $w(t) \in \mathbb{R}^l$ a white noise with unit covariance, $f(t) \in \mathbb{R}^s$ the vector of faults, and A, B_i, C, D_i ($i = 1, 4$) are known constant matrices of appropriate sizes.

Remarks:

1. $f(t) = 0$ in the fault free case, and $f(t) \neq 0$ otherwise.
2. The “directions” of the disturbances B_2, D_2 are assumed known, though the disturbances ν are unknown.
3. The white noise vector $w(t)$ is not a strict mathematical object for continuous-time systems, but significant for engineers. It can be loosely considered as a stationary process independent in time.

We look for residuals $r(t)$ generated by filtering the input $u(t)$ and output $y(t)$ through a linear time invariant (LTI) filter V . More specifically, let $r(s), u(s)$ and $y(s)$ be respectively the Laplace transform of $r(t), u(t)$ and $y(t)$, $r(s)$ is generated as

$$r(s) = V(s) \begin{pmatrix} u(s) \\ y(s) \end{pmatrix}$$

We are particularly interested in residual generators specified by the following definition.

Definition 8 A residual generator V is called an innovation filter for system (11.2.3) if

1. it is stable and generating a residual $r(t)$ of minimum dimension such that, in the fault-free case (when $f = 0$), $r(t)$ is zero mean, white, decoupled from $u(t)$ and $\nu(t)$; and
2. for any LTI filter V' generating a residual decoupled from $u(t)$ and $\nu(t)$, there exists a filter L such that $V'(s) = L(s)V(s)$.

Note that condition 2 in this definition guarantees that the innovation filter does not destroy any useful information for the purpose of FDI. A residual generated by an innovation filter is called an *innovation residual*.

Innovation Filter Decomposition

As the input u is assumed totally known, intuitively, the design of innovation filter for system (11.2.3) should be related to the design for the same system in the case of $u = 0$. In other words, it is possible to start the innovation filter design for the system without input (by ignoring B_1 and D_1), then to complete it by considering the effect of u through B_1 and D_1 . This fact is formulated by the following theorem.

Theorem 9 (Nikoukhah, 1994) $V(s)$ is an innovation filter for system (11.2.3) if and only if there exists a stable full row rank rational matrix $W(s)$ such that

$$V(s) = W(s) \begin{pmatrix} -B_1 & 0 \\ -D_1 & I \end{pmatrix} \quad (11.2.4)$$

$$\text{Row-Im } W(s) = \text{Left-Ker} \begin{pmatrix} -sI + A & B_2 \\ C & D_2 \end{pmatrix}$$

and

$$\Psi = W(s) \begin{pmatrix} B_3 \\ D_3 \end{pmatrix} (B_3^T \quad D_3^T) W(-s)$$

is a constant matrix.

Note that Row-Im and Left-Ker mean row-image and left-kernel of a matrix, respectively.

According to this theorem, the innovation filter design amounts to the design of $W(s)$ from the matrices A, C, B_2, B_3, D_2, D_3 , and to adding the effect of B_1, D_1 according to equation (11.2.4). Following Nikoukhah (1994) [82], $W(s)$ is called a *prefilter* for the innovation filter design.

The design of the prefilter $W(s)$ is still a difficult problem. It is easier when there is no disturbance (when $B_2 = 0, D_2 = 0$), as then the well-known Kalman filter is a valid innovation filter. Let us first consider the disturbance free case, then we will see how the design problem with disturbance can be solved by an algorithm based on the solution for the disturbance free case.

For the design to be simple, we introduce a regularity condition.

Definition 10 The system (A, B, C, D) is said regular if

$$\begin{pmatrix} -j\omega I + A & B \\ C & D \end{pmatrix}$$

has full row rank for all $\omega \in \mathbb{R} \cup \{\infty\}$.

The innovation filter design problem is more complicated without this regularity condition and is not always solvable. See Nikoukhah (1994) [82] for the details.

Innovation Filter Design, the Disturbance-free Case

Now let us consider the disturbance free system $(A, B_1, B_3, C, D_1, D_3)$. For this system, it is obvious that the well known Kalman filter is an innovation filter, *i.e.*, the innovation of the Kalman filter is zero mean, white and decoupled from $u(t)$. In the following, we formulate this innovation filter $V(s)$ and the corresponding prefilter $W(s)$. The latter will be useful for the design of innovation filter for the system with disturbances.

First consider the system without input, *i.e.*, system (A, B_3, C, D_3) . The existence of the Kalman filter for this system is guaranteed by the following lemma.

Lemma 11 (Kucera, 1991). *If (A, C) is detectable (*i.e.*, the unobservable modes are stable) and (A, B_3, C, D_3) is regular, let*

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = \begin{pmatrix} B_3 \\ D_3 \end{pmatrix} \begin{pmatrix} B_3^T & D_3^T \end{pmatrix}$$

then the Riccati equation

$$(A - SR^{-1}C)P + P(A - SR^{-1}C)^T - PC^T R^{-1}CP + Q - SR^{-1}S^T = 0$$

has a solution P such that $A + KC$ is stable with $K = -(PC^T + S)R^{-1}$.

Under the conditions of Lemma 11, the *steady state* Kalman filter generates the innovation

$$r_1(s) = V_1(s)y(s) \tag{11.2.5}$$

with

$$V_1(s) = C(sI - A - KC)^{-1}K + I$$

where K is as in Lemma 11.

Now consider the system $(A, B_1, B_3, C, D_1, D_3)$ with input $u(t)$. The innovation generated by the steady state Kalman filter is

$$r(s) = V(s) \begin{pmatrix} u(s) \\ y(s) \end{pmatrix} \tag{11.2.6}$$

with

$$V(s) = C(sI - A - KC)^{-1}(-B_1 - KD_1, K) + (-D_1, I)$$

Now put (11.2.5) into the form of (11.2.6) with zero input:

$$r_1(s) = (C(sI - A - KC)^{-1}, V_1(s)) \begin{pmatrix} 0 \\ y(s) \end{pmatrix}$$

Then

$$W(s) = (C(sI - A - KC)^{-1}, V_1(s)) = C(sI - A - KC)^{-1}(I, K) + (0, I) \quad (11.2.7)$$

is in fact a prefilter for the system $(A, B_1, B_3, C, D_1, D_3)$, as it can be easily verified that

$$\begin{aligned} W(s) \begin{pmatrix} -B_1 & 0 \\ -D_1 & I \end{pmatrix} &= C(sI - A - KC)^{-1}(I, K) \begin{pmatrix} -B_1 & 0 \\ -D_1 & I \end{pmatrix} + (0, I) \begin{pmatrix} -B_1 & 0 \\ -D_1 & I \end{pmatrix} \\ &= C(sI - A - KC)^{-1}(-B_1 - KD_1, K) + (-D_1, I) \\ &= V(s) \end{aligned}$$

Innovation Filter Design with Disturbance

The innovation filter design in the presence of disturbance can be reduced to the case without disturbance with the aid of the following results.

Theorem 12 (Nikoukhah, 94) (*Nikoukhah, 1994*). *A prefilter $W(s)$ exists for system (11.2.3) if and only if it can be expressed as*

$$W(s) = W_r(s)\Gamma$$

where $W_r(s)$ is a prefilter for $(A_r, B_{3r}, C_r, D_{3r})$ with

$$\begin{pmatrix} -sI + A_r & B_{3r} \\ C_r & D_{3r} \end{pmatrix} = \Gamma \begin{pmatrix} -sI + A & B_2 & B_3 \\ C & D_2 & D_3 \end{pmatrix} \begin{pmatrix} \Phi & 0 \\ 0 & I \end{pmatrix} \quad (11.2.8)$$

for some full row rank matrix Γ and full column rank matrix Φ .

Recall the sizes of the matrices:

$$A \in \mathbb{R}^{n \times n}, C \in \mathbb{R}^{m \times n}, B_2 \in \mathbb{R}^{n \times q}, B_3 \in \mathbb{R}^{n \times l}, D_2 \in \mathbb{R}^{m \times q}, D_3 \in \mathbb{R}^{m \times l}$$

The reduced system has the dimension as

$$A_r \in \mathbb{R}^{n_r \times n_r}, C_r \in \mathbb{R}^{m_r \times n_r}, B_{3r} \in \mathbb{R}^{n_r \times l}, D_{3r} \in \mathbb{R}^{m_r \times l}$$

where n_r, m_r are to be determined by an algorithm. Accordingly, $\Gamma \in \mathbb{R}^{(n_r+m_r) \times (n+m)}, \Phi \in \mathbb{R}^{(n+q) \times n_r}$.

Based on this theorem, the design of the prefilter $W(s)$ for system (11.2.3) amounts to finding the above matrices Γ and Φ . In order to present the algorithm for this purpose, we need to introduce the following lemma.

Lemma 13 (Nikoukhah, 94) Let E, F be a pair of arbitrary matrices of equal size. There exist a full row rank matrix M and a full column rank matrix N of appropriate sizes such that, if we let

$$E' = MEN, \quad F' = MFN$$

E' has full column rank and

$$\text{Left-Ker}(sE - F) = [\text{Left-Ker}(sE' - F')]M$$

Such matrices M, N can be found by the following algorithm.

Algorithm 14 (Nikoukhah, 94)

Initialization: $E' = E, \quad F' = F, \quad M = I, \quad N = I$

While E' does not have full column rank, do

- Find an orthogonal matrix $\Sigma = (\Sigma_1, \Sigma_2)$ such that $(E'_1, 0) = E'(\Sigma_1, \Sigma_2)$ with full column rank E'_1 . One possible solution is to make a singular value decomposition of E' and to let the columns of Σ be the right singular vectors.
- Calculate $(F'_1, F'_2) = F'(\Sigma_1, \Sigma_2)$.
- Find the highest full row rank matrix L such that $LF'_2 = 0$. For instance, L^T is composed of the left singular vectors of F'_2 associated to the zero singular values.
- Let $E' = LE'_1, \quad F' = LF'_1, \quad M = LM, \quad N = N\Sigma_1$.

The matrix M is constructed by products of full row rank matrices L , so M has full row rank. Similarly the constructed N has full column rank. The algorithm ends in a finite number of iterations, since the number of columns of E' is reduced by one or more at each iteration. The relationship $\text{Left-Ker}(sE - F) = [\text{Left-Ker}(sE' - F')]M$ is guaranteed by the fact that L is always taken as the admissible highest rank matrix.

Now we are ready to state the algorithm finding the reduced system $(A_r, B_{3r}, C_r, D_{3r})$ and the corresponding matrices Γ, Φ .

Algorithm 15 (Nikoukhah, 94)

- Let

$$E = -\begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}, \quad F = -\begin{pmatrix} A & B_2 \\ C & D_2 \end{pmatrix}$$

where the sizes of the blocks of E are as those of F .

- Apply Algorithm 14 to get the matrices

$$E' \in \mathbb{R}^{(n_r+m_r) \times n_r}, F' \in \mathbb{R}^{(n_r+m_r) \times n_r}, M \in \mathbb{R}^{(n_r+m_r) \times (n+m)}, N \in \mathbb{R}^{(n+q) \times n_r}$$

Note that the sizes n_r, m_r are results of Algorithm 14 and determine the dimension of the reduced system as follows.

- Find an invertible matrix T such that

$$TE' = \begin{pmatrix} -I \\ 0 \end{pmatrix}$$

Such a solution can be obtained by performing the QR decomposition

$$E' = (Q_1, Q_2) \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$$

and let

$$T = \begin{pmatrix} -R_1^{-1}Q_1^T \\ Q_2^T \end{pmatrix}$$

- Let $\Gamma = TM$, $\Phi = N$. Note that $\Gamma \in \mathbb{R}^{(n_r+m_r) \times (n+m)}$, $\Phi \in \mathbb{R}^{(n+q) \times n_r}$.

- Finally, let

$$\begin{pmatrix} A_r \\ C_r \end{pmatrix} = -TF', \quad \begin{pmatrix} B_{3r} \\ D_{3r} \end{pmatrix} = \Gamma \begin{pmatrix} B_3 \\ D_3 \end{pmatrix}$$

with $A_r \in \mathbb{R}^{n_r \times n_r}$, $C_r \in \mathbb{R}^{m_r \times n_r}$, $B_{3r} \in \mathbb{R}^{n_r \times l}$, $D_{3r} \in \mathbb{R}^{m_r \times l}$.

It is straightforward to check that the constructed matrices A_r, C_r, B_{3r}, D_{3r} satisfy equation (11.2.8).

Let us summarize the design of innovation filter for system (11.2.3).

- Apply Algorithms 14 and 15 to construct the reduced system $(A_r, B_{3r}, C_r, D_{3r})$ and the matrix Γ .
- Apply formula (11.2.7) to $(A_r, B_{3r}, C_r, D_{3r})$ to design a prefilter $W_r(s)$.
- An innovation filter for system (11.2.3) is given by

$$V(s) = W_r(s)\Gamma \begin{pmatrix} -B_1 & 0 \\ -D_1 & I \end{pmatrix}$$

Residual Sensitivity to Faults

The procedure presented above for the design of innovation filter only ensure the whiteness of $r(t)$ and its decoupling from the inputs $u(t)$ and the disturbances $\nu(t)$. After the design of such a innovation filter, we have to check if it is sensitive to the faults to be detected, *i.e.*, to ensure that the presence of a fault $f(t) \neq 0$ does lead to a non zero $r(t)$. It is easy to verify that a designed innovation filter $V(s)$ is *fault sensitive* if the transfer function

$$V(s) \begin{pmatrix} 0 \\ C(sI - A)^{-1}B_4 + D_4 \end{pmatrix} \quad (11.2.9)$$

is not zero.

There exist also some alternative definitions of fault sensitivity: $V(s)$ is said *completely fault sensitive* or *strictly fault sensitive*, if the transfer function (11.2.9) has full rank for s over the field \mathbb{R} or \mathbb{Q} (rational numbers), respectively (Massoumnia *et. al.* 1989 [76]).

It is known that, as far as innovation filters are concerned, for a given system, the fault sensitivity does not depend on a particular choice of innovation filter (Nikoukhah, 1994 [83]). In this sense, all the innovation filters are equivalent.

Residual Design for Fault Isolation

So far we have seen how to design residuals for the detection of all the faults $f(t) \in \mathbb{R}^s$. In general such a residual $r(t)$ is sensitive to all the components of $f(t)$. For the purpose of fault isolation, we want to design residuals sensitive only to part of the components of $f(t)$, but decoupled to the other components. By designing different residuals only sensitive to different parts of the components of $f(t)$, the fault isolation problem can be solved.

Let us arbitrarily partition $f(t)$ into $(f_1^T(t), f_2^T(t))^T$. In order to design a residual sensitive to $f_1(t)$, but decoupled from $f_2(t)$, just consider $f_2(t)$ as part of disturbances. More specifically, in model (11.2.3), replace $f(t)$ by $f_1(t)$ and $\nu(t)$ by $(\nu^T(t), f_2^T(t))^T$, and modify the matrices B_2, B_4, D_2, D_4 accordingly. Then apply the above design procedure for robust innovation filter. The resulting residual will satisfy the desired property. The design of residuals for fault isolation can thus be achieved.

11.3 Residuals for Non Additive Faults in Nonlinear Systems

We first recall some results for non additive faults in linear systems before presenting methods for nonlinear systems.

11.3.1 Non Additive Faults in Linear Systems

The detection and isolation of non additive faults, even in linear systems, are a more difficult problem and much less results are available than for the case of additive faults.

Non additive faults are generally modeled as changes in some parameters of a system. It is thus natural to use parameter estimation based approaches to detect and isolate such faults (Isermann, 1984 [60] and Isermann 1997 [62]). The basic idea is to perform on-line identification of the monitored system and to compare the identified parameter to its nominal value. For computational efficiency of on-line identification, the equation error model is preferred. The on-line identification of other models, in particular the state space model, is computationally much more expensive.

For the detection of non additive faults in linear state space systems, it is possible to generate residuals like in the case of additive faults. In the fault-free case, the behavior of the residuals is the same. However, in the presence of non additive faults, the behavior of these residuals is more complicated. Generally the fault sensitivity analysis is more difficult and it is not clear how to design residuals for fault isolation. Another problem is that the statistical properties of the residuals are not well known in the presence of non additive faults.

It is worth mentioning the methods for the detection of non additive faults in linear systems proposed in Basseville *et. al.* (1993) [12] related to some instrumental variable identification method, and in Basseville *et. al.* (2000) [11] based on results on subspace identification method. In both cases, only the local statistical properties of the residuals are well established.

11.3.2 Residual Generation for Nonlinear Systems

There are relatively few results available for the detection and isolation of faults in nonlinear systems. This fact is a consequence of the difficulties in nonlinear system analysis. As it is very difficult to study nonlinear systems in a stochastic framework, most known methods of residual generation for nonlinear systems are developed in a deterministic framework. A drawback of this practice is the lack of knowledge on the statistical properties of the generated residuals. Nevertheless, we will see later in this chapter that, thanks to the asymptotic local approach to change detection, the evaluation of such residuals for the detection and isolation of small faults can often be achieved in a well established statistical framework.

As for linear systems, one of the difficulties for residual generation is the presence of unknown variables in system models, usually unmeasured state variables. Typically there are two approaches to deal with unknown variables, namely *elimination* and *estimation*. Accordingly, we present two methods for residual generation.

Elimination of Unknown Variables

The parity relation for linear dynamic systems (see Section 11.2.1) is in fact obtained by the elimination of unmeasured state variables. It is a natural idea to extend this method to nonlinear systems. Usually, nonlinear continuous-time models are manipulated for variable elimination rather than discrete-time models, as existing differential-algebraic tools are helpful for the former. Note that in the discrete-time case, the unknown variables are eliminated based on the temporal redundancy among the delayed variables. In the continuous-time case, it is the redundancy among the time derivatives of the variables which is exploited. This implies that the computation of the resulting residuals will require the estimation of time derivatives of the measured signals which is a noise sensitive task.

At present, it is known that the elimination of unknown variables can be achieved for systems with polynomial nonlinearities. In Guernez *et. al.* (1997) [51] the Gröbner basis is used for this purpose. Here we summarize the method proposed in Zhang *et. al.* (1998) [126] which is based on Ritt's algorithm and has the advantage to automatically manage the differentiation of the involved variables.

This method allows to deal with systems modeled by differential-algebraic equations (DAE). Assume that a system is modeled by a set of equations of the form:

$$f_i(x(t), u(t), y(t), \theta, p) = 0, \quad i = 1, 2, \dots, n \quad (11.3.1)$$

where $u(t)$ is the inputs, $y(t)$ the outputs, $x(t)$ the unknown internal variables, p the operator of differentiation in time, and f_i a polynomial parameterized by θ . Note that the usual state space model is a particular case of the differential-algebraic model (11.3.1).

Working with polynomial differential-algebraic equations is for the purpose of applying techniques in differential-algebra. The restriction to polynomial nonlinearity is not as restrictive as it seems. As a matter of fact, many equations with non polynomial nonlinearities can be transformed into polynomial differential-algebraic equations. For example, the equation

$$y = \sin x$$

is equivalent to

$$y^2 + y^2 \dot{x}^2 = x^2$$

apart from some singular solutions.

The main problem is the elimination of the unknown variables $x(t)$ through differential-algebraic manipulations. The basic idea is as follows. For a given value of θ , if there exists $x(t)$ such that $u(t)$ and $y(t)$ satisfy the set of equations (11.3.1), then the pair $u(t)$ and $y(t)$ is said to be a solution of (11.3.1). If we differentiate with respect to time one of the equations (11.3.1), then the solution of (11.3.1), $u(t)$ and $y(t)$, together with the associated $x(t)$, also satisfies the new equation obtained by differentiation. We can continue to differentiate this equation or

other ones in (11.3.1). This way, more and more equations are obtained involving $u(t), y(t), x(t)$ and their time derivatives. It is obvious that these equations are redundant. It is possible, among a finite number of such equations, to eliminate $x(t)$ and its time derivatives through algebraic manipulations. The procedure for differentiation and elimination is often very lengthy. Fortunately, computer implementations of Ritt's algorithm, can be used for this purpose (Zhang *et. al.* 1998 [126]). See Ljung and Glad (1994) [72] to understand the detail of the algorithm.

Assume that, after eliminating $x(t)$ and its time derivatives, we obtain

$$g(u(t), y(t), \theta, p) = 0 \quad (11.3.2)$$

where g is a polynomial or a vector of polynomials. From this stage, it is natural to generate a residual as $r(t) = g(u(t), y(t), \theta, p)$. However, it is known that this residual is efficient only if the monitored faults are modeled as additive terms in equation (11.3.2). When faults are generally modeled as changes in the parameter vector θ , it is recommended to generate the residual as

$$\begin{aligned} r(t) &= \frac{1}{2} \frac{\partial}{\partial \theta} [g^T(u(t), y(t), \theta, p) g(u(t), y(t), \theta, p)] \\ &= \left(\frac{\partial g(u(t), y(t), \theta, p)}{\partial \theta} \right)^T g(u(t), y(t), \theta, p) \end{aligned}$$

The residual generated in this way has the particularity to have the same dimension as θ , whatever is the dimension of g .

The advantages of this method based on the elimination of $x(t)$ are its generality for a large class of nonlinear systems and its systematic character. It also has serious drawbacks: the elimination algorithm often generate bulky polynomials, numerical differentiation of sampled signals is very sensitive to noise, the generated residuals may be badly conditioned. These drawbacks have limited the application of this method to complex systems.

Estimation of Unmeasured Variables

Instead of eliminating the unknown variables $x(t)$, we can also estimate them with an observer. Usually observers are designed for state space systems of the form

$$\begin{cases} \dot{x}(t) = f(\theta, x(t), u(t)) \\ y(t) = h(\theta, x(t), u(t)) \end{cases} \quad (11.3.3)$$

where $x(t)$ is the state vector, $u(t)$ the inputs, $y(t)$ the outputs, f and h are two differentiable nonlinear functions parameterized by θ .

The design of observer for nonlinear systems is a difficult problem and at present there is no satisfactory general method. Nevertheless, some recent progresses have been reported; see Walcott *et. al.* (1987), [118] Gauthier *et. al.* (1992), [41] Ciccarella *et. al.* (1993), [28] Marino and Tomei (1995), [74] Moraal and Grizzle (1995), [80] and Nikoukhah (1998) [83]. Besides

the extended Kalman filters or similar observers with local convergence, there are typically two situations where observers with global convergence can be designed: observers with linear error dynamics for systems which can be linearized by coordinate transformation and by output injection (Marino and Tomei, 1995) [74]; and high gain observers for systems with some particular structure satisfying Lipschitz conditions (Gauthier *et. al.* (1992) [41]).

Assume that an observer for system (11.3.3) is available. Typically, it has the form

$$\dot{\hat{x}}(t) = g(\theta, \hat{x}(t), u(t), y(t))$$

where $\hat{x}(t)$ is the state estimate and g is a designed nonlinear function. As the observer is designed such that (under noise-free assumption)

$$\lim_{t \rightarrow \infty} [\hat{x}(t) - x(t)] = 0$$

under continuity condition, the output estimate $\hat{y}(t) = h(\theta, \hat{x}(t), u(t))$ has the property

$$\lim_{t \rightarrow \infty} [\hat{y}(t) - y(t)] = 0$$

It is natural to generate a residual as $r(t) = \hat{y}(t) - y(t)$. It is known, however, that this residual is in general not sufficient to detect and to isolate the faults modeled as changes in the parameter vector θ (Basseville and Nikiforov, 1993) [13]. It is more powerful to generate residuals as

$$\begin{aligned} r(t) &= \frac{1}{2} \frac{\partial}{\partial \theta} \left[(\hat{y}(t, \theta) - y(t))^T (\hat{y}(t, \theta) - y(t)) \right] \\ &= \left(\frac{\partial \hat{y}(t, \theta)}{\partial \theta} \right)^T (\hat{y}(t, \theta) - y(t)) \end{aligned}$$

where the dependence of $\hat{y}(t, \theta)$ on θ has been made explicit (Zhang and Basseville, 1998) [124].

The computation of the residual $r(t)$ requires $\partial \hat{y}(t)/\partial \theta$ which is calculated as

$$\frac{\partial \hat{y}(t)}{\partial \theta} = \frac{\partial h(\theta, \hat{x}(t), u(t))}{\partial \hat{x}} \frac{\partial \hat{x}(t)}{\partial \theta} + \frac{\partial h(\theta, \hat{x}(t), u(t))}{\partial \theta}$$

where $\partial \hat{x}(t)/\partial \theta$ is in turn obtained as a solution of the differential equation

$$\frac{d}{dt} \left(\frac{\partial \hat{x}(t)}{\partial \theta} \right) = \frac{\partial g(\theta, \hat{x}(t), u(t), y(t))}{\partial \hat{x}} \frac{\partial \hat{x}(t)}{\partial \theta} + \frac{\partial g(\theta, \hat{x}(t), u(t), y(t))}{\partial \theta}$$

Compared to the elimination approach, the estimation of $x(t)$ with observers has the advantage to be more robust to measurement noise. Its main drawback is the difficulty for nonlinear observer design in general.

11.4 Residual Evaluation

Though it is widely acknowledged in the literature that the problem of FDI is typically solved in two steps, namely residual generation and residual evaluation, most publications on FDI focus on the first step. Here we present some statistical methods for residual evaluation, first in the case with a well known stochastic model, then in the case with poor statistical knowledge.

11.4.1 Evaluation of Residuals with Known Stochastic Model

Typically, for linear systems with additive faults, it is possible to solve the problem of FDI through a fully stochastic approach.

Statistical residual evaluation is usually performed in discrete time. Assume that a sequence of residual $r(t)$, $t = 1, 2, 3, \dots$, is available. It may be generated with a discrete time model or generated with a continuous time model and then sampled. In order to apply statistical tests, we need to know the probability density function (PDF) of the residual sequence. We first assume the PDF is known in both fault-free and faulty cases, then only in fault-free case.

Known PDF in Fault Free and Faulty Cases

A typical situation where residuals with known PDF can be designed is linear systems with additive faults. For example, it can be shown (Nikoukhah, 1994) [82] that the innovation residual described in Section 11.2.2 satisfies

$$r(s) = G_w(s)w(s) + G_f(s)f(s)$$

with

$$G_w(s) = V(s) \begin{pmatrix} 0 \\ C(sI - A)^{-1}B_3 + D_3 \end{pmatrix} \quad G_f(s) = V(s) \begin{pmatrix} 0 \\ C(sI - A)^{-1}B_4 + D_4 \end{pmatrix}$$

In the fault free case, $r(s) = G_w(s)w(s)$. By assumption $w(t)$ is a quite noise. The conditions of Theorem 9 ensure that

$$\Psi = G_w(s)G_w^T(-s)$$

is a constant matrix, therefore $r(t)$ is also a white noise. Assume that $w(t)$ is Gaussian with unit covariance, then $r(t)$ is also Gaussian with covariance matrix Ψ . The PDF of $r(t)$ is thus completely known. In the faulty case, assume that the profile of a particular fault $f(t)$ is fully known, so is $f(s)$, then $r(s) - G_f(s)f(s)$ has the same behavior as the fault-free $r(s)$. From this observation, the PDF of $r(t)$ can be readily derived. If several fault profiles $f(t)$ are considered, then the computation must be performed for each fault.

In what follows, \mathbf{H}_0 means the fault free hypothesis and \mathbf{H}_1 the hypothesis of the presence of a particular fault. Denote by r_1^k the residual sequence $r(1), r(2), \dots, r(k)$. Let $p_0(r_1^k)$ and $p_1(r_1^k)$ be the PDF of r_1^k under \mathbf{H}_0 and \mathbf{H}_1 , respectively. Then we can formulate the (log) likelihood ratio test

$$S(r_1^k) = \log \frac{p_1(r_1^k)}{p_0(r_1^k)} \quad (11.4.1)$$

Then it can be shown that

$$\begin{aligned} \mathbf{E}(S(r_1^k)|\mathbf{H}_0) &= \int S(r_1^k)p_0(r_1^k)dr_1^k < 0 \\ \mathbf{E}(S(r_1^k)|\mathbf{H}_1) &= \int S(r_1^k)p_1(r_1^k)dr_1^k > 0 \end{aligned}$$

Obviously, the decision between \mathbf{H}_0 and \mathbf{H}_1 can be made from the sign of $S(r_1^k)$. In practice, the decision is made by comparing $S(r_1^k)$ to a threshold λ : if $S(r_1^k) \geq \lambda$, then \mathbf{H}_1 is decided; otherwise \mathbf{H}_0 is decided. The choice of the threshold λ is based on a trade-off between the *false alarm probability* P_F and the *detection probability* P_D defined as follows.

$$\begin{aligned} P_F(S, \lambda) &= \int_{S(r_1^k) \geq \lambda} p_0(r_1^k)dr_1^k \\ P_D(S, \lambda) &= \int_{S(r_1^k) \geq \lambda} p_1(r_1^k)dr_1^k \end{aligned}$$

In general a larger threshold λ leads to a smaller false alarm probability, but also a smaller detection probability.

In the test (11.4.1), it is assumed that either \mathbf{H}_0 or \mathbf{H}_1 is true for all the time instants $t = 1, \dots, k$. Such problem is called *off-line* hypothesis testing. For quick detection of sudden faults, *on-line* hypothesis testing should be considered: the hypothesis \mathbf{H}_0 is assumed true from the beginning to some instant, then \mathbf{H}_1 becomes true. This problem can be easily solved when the residual sequence is independent in time, as then the PDF can be factorized. (Note that when the residual sequence is not independent, a similar result can in principle be obtained using conditional probability density functions.) Let k be the current time instant, and q the time when \mathbf{H}_1 becomes true. Then the likelihood ratio test writes

$$\begin{aligned} S(r_1^k) &= \sum_{t=1}^{q-1} \log \frac{p_0(r(t))}{p_0(r(t))} + \sum_{t=q}^k \log \frac{p_1(r(t))}{p_0(r(t))} \\ &= \sum_{t=q}^k \log \frac{p_1(r(t))}{p_0(r(t))} \end{aligned}$$

As q is unknown, the likelihood ratio test is modified as

$$\tilde{S}(r_1^k) = \max_{1 \leq q \leq k} \sum_{t=q}^k \log \frac{p_1(r(t))}{p_0(r(t))} \quad (11.4.2)$$

When $\tilde{S}(r_1^k)$ exceeds a chosen threshold λ , the value of q maximizing $S(r_1^k)$ is considered as the estimated change time.

According to formula (11.4.2), all the history of $r(t)$ has to be memorized for computing $\tilde{S}(r_1^k)$. Fortunately, the following efficient recursive algorithm with very small memory requirement can be used for computing $\tilde{S}(r_1^k)$:

$$\begin{aligned}\tilde{S}(r_1^0) &= 0 \\ \tilde{S}(r_1^k) &= \tilde{S}(r_1^{k-1}) + \left[\log \frac{p_1(r(k))}{p_0(r(k))} \right]_+ \quad k = 1, 2, 3, \dots\end{aligned}$$

with

$$[x]_+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

It only needs to store the value of $\tilde{S}(r_1^{k-1})$ for computing $\tilde{S}(r_1^k)$, and the last time instant when $\tilde{S}(r_1^t)$ became positive from zero for estimating the change time. This algorithm is known as the Page-Hinkley algorithm (Basseville and Nikiforov, 1993) [13].

Known PDF in Fault Free Case Only

It is more realistic to assume that the PDF of a residual is fully known only in the fault free case, since the PDF in the faulty case is rarely fully known in practice. Let us assume that the PDF of the residual is parameterized with unknown parameters. For example, in system (11.2.3), let the fault profile $f(t)$ be parameterized by θ , say $f(t, \theta)$. For any given value of θ , the innovation residual $r(t)$ described in Section 11.2.2 has known PDF, exactly as in the case with known $f(t, \theta)$. Therefore, the PDF of $r(t)$ is also parameterized by θ . For a residual sequence r_1^k , denote by $p_\theta(r_1^k)$ the parameterized PDF. Let θ_0 be the parameter corresponding to the case $f(t) = 0$, i.e., the fault free case, then the fault free PDF is $p_{\theta_0}(r_1^k)$. In the faulty case, $p_\theta(r_1^k)$ is known up to a parameter θ , therefore in the test θ is replaced by its maximum likelihood estimate:

$$S(r_1^k) = \max_{\theta} \log \frac{p_\theta(r_1^k)}{p_{\theta_0}(r_1^k)}$$

This test is known as the *generalized likelihood ratio* (GLR) test. It is compared with a threshold in order to make the decision between \mathbf{H}_0 and \mathbf{H}_1 .

The above formulation is for off-line hypothesis testing, i.e., it is assumed that either \mathbf{H}_0 or \mathbf{H}_1 is true for all the time instants $t = 1, \dots, k$. For on-line hypothesis testing, we should consider a change from \mathbf{H}_0 to \mathbf{H}_1 at $t = q$ by searching for $q = 1, \dots, k$ maximizing $S(r_1^k)$. Due to the double maximization (in θ and in q), in general there is no simple recursive algorithm, even when the residual sequence is independent in time. In practice, the search for the optimal value of q

has to be restricted to a finite sliding window. A simpler solution is to give up the maximization in q and to set $q = k - \tau$ with a chosen value τ . It amounts to compute $S(r_1^k)$ in a sliding data window.

There is one situation where the on-line hypothesis testing algorithm can be simplified. If the unknown parameter θ is a scalar number, a critical value of this parameter, say θ_1 , can often be specified in practice. It means that the values of θ between θ_0 and θ_1 are tolerable, a fault should be detected only if the value of θ exceeds θ_1 . In this case, in the PDF $p_\theta(r_1^k)$ under \mathbf{H}_1 , the parameter θ can be set to a pre-specified value, the maximization in θ is then avoided. The problem to be solved becomes similar to the one in the case with known PDF and the efficient recursive Page-Hinkley algorithm can also be applied.

11.4.2 Evaluation of Residuals through the Asymptotic Local Approach

Outside the framework of linear systems with additive faults, usually the designed residuals do not have well known statistical properties. Therefore the above statistical tests no longer apply. It is then important to develop statistical methods for residual evaluation with a larger extent of applicability. The asymptotic local approach to change detection offers interesting results for this purpose.

In order to present the asymptotic local approach to change detection, we have to clearly distinguish the parameters of the system from those of the model, respectively noted as Θ , and θ . It is the parameters Θ which really govern the system behavior, whereas the parameters θ are only an image of Θ established by the mathematical modeling procedure. So strictly speaking, a fault occurred in the system should correspond to a change in Θ , not in θ , though quite often this difference is not clearly emphasized. In most practical situations, we only have access to θ , not to Θ .

Residual Generated with a Full Model

For simplicity, we first consider the case where the residual $r(t)$ is generated with a “full model”, *i.e.*, the parametrization of the model with θ coincides with the system parameters Θ . Moreover, we assume that the nominal parameter value θ_0 is unbiased, *i.e.*, $\theta_0 = \Theta_0$, with Θ_0 governing the fault-free system. The more realistic situation where $r(t)$ is generated with a reduced model will be considered later.

The residuals presented in this chapter can all be formally put into the following form.

$$\begin{cases} \dot{\xi}(t) = \phi(\theta_0, \xi(t), u(t), y(t)) \\ r(t) = \psi(\theta_0, \xi(t), u(t), y(t)) \end{cases} \quad (11.4.3)$$

where $\xi(t)$ is some auxiliary variable of finite dimension. For example, the innovation residual presented in Section 11.2.2 is generated with a transfer function $V(s)$, it can also be realized in state space form, then $\xi(t)$ corresponds to the state variable.

For shorter notations, let $Z(t)$ collect the variables $\xi(t), u(t), y(t)$, then we formally write

$$r(t) = \gamma(\theta_0, Z(t))$$

where in general γ is a nonlinear function. It is assumed that $r(t)$ only has poorly known stochastic property.

Now assume that $Z(t)$ is sampled at $t = 1, 2, \dots, N$. Let us define

$$\zeta_N(\theta) \triangleq \frac{1}{\sqrt{N}} \sum_{t=1}^N \gamma(\theta, Z(t)) \quad (11.4.4)$$

We will see that, under mild assumptions on $r(t)$ as stated below, the quantity $\zeta_N(\theta)$ has well established asymptotic statistical properties. Therefore, $\zeta_N(\theta)$ is called *normalized residual*, whereas $r(t)$ *primary residual*.

Now we state the assumptions which the asymptotic local approach relies on. The essential assumption is that the sampled $r(t)$ is stationary and that

$$\begin{aligned} \mathbf{E}(\gamma(\theta_0, Z(t)) | \Theta = \theta_0) &= 0 \\ \mathbf{E}(\gamma(\theta_0, Z(t)) | \Theta \in U(\theta_0) \setminus \{\theta_0\}) &\neq 0 \end{aligned} \quad (11.4.5)$$

where $U(\theta_0)$ a neighborhood of θ_0 . Note that the expectation depends on Θ , since $Z(t)$ is governed by it. This assumption states that the residual $r(t)$ should have zero mean when no fault occurs, and non zero mean in the presence of a (small) fault.

We also assume that the sensitivity matrix

$$M(\theta_0) \triangleq \mathbf{E}\left(\left[\frac{\partial}{\partial \theta} \gamma(\theta, Z(t))\right]_{\theta=\theta_0} \middle| \Theta = \theta_0\right) \quad (11.4.6)$$

exists and has full column rank, and that the covariance matrix

$$\Sigma(\theta_0) \triangleq \lim_{N \rightarrow \infty} \mathbf{E}(\zeta_N(\theta_0) \zeta_N^T(\theta_0) | \Theta = \theta_0) \quad (11.4.7)$$

exists and is positive definite.

Consider two hypotheses:

$$\mathbf{H}_0 : \Theta = \theta_0$$

$$\mathbf{H}_1 : \Theta = \theta_0 + \frac{\eta}{\sqrt{N}}$$

where η is an unknown vector of the same dimension as θ . Note that \mathbf{H}_0 is the fault-free hypothesis and \mathbf{H}_1 the small fault hypothesis for large sample length N . The difference between \mathbf{H}_0 and \mathbf{H}_1 is assumed to be proportional to $1/\sqrt{N}$ in order to establish the central limit theorem as follows. Under the assumptions (11.4.5), (11.4.6), (11.4.7) and some weak mixing conditions, when the sample length $N \rightarrow \infty$, the normalized residual $\zeta_N(\theta_0)$ tends to a Gaussian vector:

$$\zeta_N(\theta_0) \rightsquigarrow \begin{cases} \mathcal{N}(0, \Sigma(\theta_0)) & \text{under } \mathbf{H}_0 \\ \mathcal{N}(-M(\theta_0)\eta, \Sigma(\theta_0)) & \text{under } \mathbf{H}_1 \end{cases} \quad (11.4.8)$$

See Benveniste *et. al.*, (1987) [16] for a rigorous statement of the result and its proof.

Note that under both hypotheses, the limiting distribution of $\zeta_N(\theta_0)$ is Gaussian with the same covariance matrix, the only difference under the two hypotheses is in the mean vector. Based on this result, the detection and isolation of small changes in Θ are asymptotically equivalent to the detection and isolation of changes in the mean of a Gaussian vector. Therefore, no matter how the residual $r(t)$ is generated, if the above assumptions are satisfied, then its evaluation for the detection and isolation of small faults can be solved as a standard statistical testing problem in the Gaussian framework.

Residual Generated with a Reduced Model

Reduced models are used in most practical situations where the model parameterization θ is different from Θ . We assume that the dimension of Θ is finite, but larger than that of θ . In order to deal with residuals generated with reduced models, the previously presented asymptotic local approach is extended in Zhang *et. al.* (1994) [125]. Roughly speaking, with some modified assumptions, a similar central limit theorem is obtained so that the same statistical tests can be applied to the evaluation of the residuals in the present case. We summarize the result as follows.

The primary residual $r(t)$ is still assumed to be generated by the generator (11.4.3) and the normalized residual is defined as (11.4.4). Note that the residual $r(t)$ depends both on the (nominal) model parameter θ and on the system parameter Θ . We write

$$r(t) = \gamma(\theta, Z(t), \Theta)$$

to show this double dependence. Let Θ_0 be the value of Θ governing the fault-free system and θ_0 the nominal model parameter value. Under appropriate smoothness conditions, assume that the matrix

$$M(\theta_0) \triangleq \mathbf{E} \left(\left[\frac{\partial}{\partial \theta} \gamma(\theta, Z(t), \Theta) \right]_{\theta=\theta_0} \middle| \Theta = \Theta_0 \right) \quad (11.4.9)$$

exists and has full column rank. Then, according to the implicit function theorem, in a neighborhood of Θ_0 , a function

$$\theta = \beta(\Theta)$$

is defined by the equation

$$\mathbf{E}(\gamma(\theta, Z(t), \Theta) | \Theta) = 0$$

It is obvious that $\theta_0 = \beta(\Theta_0)$, though in general the function β is unknown.

Now the assumption (11.4.5) is replaced by

$$\begin{aligned}\mathbf{E}(\gamma(\theta_0, Z(t)) | \beta(\Theta) = \theta_0) &= 0 \\ \mathbf{E}(\gamma(\theta_0, Z(t)) | \beta(\Theta) \in U(\theta_0) \setminus \{\theta_0\}) &\neq 0\end{aligned}\tag{11.4.10}$$

and accordingly

$$\Sigma(\theta_0) \triangleq \lim_{N \rightarrow \infty} \mathbf{E}(\zeta_N(\theta_0) \zeta_N^T(\theta_0) | \Theta = \Theta_0)\tag{11.4.11}$$

The hypotheses are now formulated as

$$\begin{aligned}\mathbf{H}_0 : \beta(\Theta) &= \theta_0 \\ \mathbf{H}_1 : \beta(\Theta) &= \theta_0 + \frac{\eta}{\sqrt{N}}\end{aligned}$$

After these modifications, the asymptotic property (11.4.8) of the normalized residual $\zeta_N(\theta_0)$ still holds. See [125] for the details.

Note that the new hypotheses are formulated in terms of $\beta(\Theta)$ instead of Θ itself. This means that, due to the use of a reduced model, the changes of Θ not affecting its image $\beta(\Theta)$ are not detectable. It is thus important to verify if the chosen model is compatible with the faults to be detected.

Statistical Tests on Gaussian Vectors

We have seen that under mild assumptions on the primary residual, the normalized residual has asymptotic Gaussian distribution, under both \mathbf{H}_0 and \mathbf{H}_1 . Now we recall some well known statistical tests for detecting and isolating changes in the mean of a Gaussian vector.

Let us first simplify the notations. Assume that the sample length N is sufficiently large so that $\zeta = \zeta_N(\theta_0)$ follows the Gaussian distribution

$$\zeta \sim \mathcal{N}(-M\eta, \Sigma)$$

The fault detection problem is making a decision between the two hypotheses: $\eta \neq 0$ and $\eta = 0$. It is well known that this problem can be solved by the generalized likelihood ratio (GLR) test. See, e.g., Basseville and Nikiforov, (1993) [13]. The test is in a quadratic form:

$$S = \zeta^T \Sigma^{-1} M (M^T \Sigma^{-1} M)^{-1} M^T \Sigma^{-1} \zeta\tag{11.4.12}$$

This test is referred to as the *global test*, as it detects all non zero components of η . Because of the Gaussian distribution of ζ , the global test S has a χ^2 -distribution. Its number of degrees of freedom is equal to the dimension of η . When $\eta = 0$, the χ^2 -distribution is central, and when $\eta \neq 0$, its non-centrality parameter is $\delta = \eta^T M^T \Sigma^{-1} M \eta$.

Note that when the matrix M is square invertible, the test becomes simply

$$S = \zeta^T \Sigma^{-1} \zeta$$

The detection rule is

- if $S \leq \lambda$, then no fault
- if $S > \lambda$, then fault

where the choice of the threshold λ can be guided by the χ^2 -table to achieve a desired false alarm probability.

Now let us consider the tests for fault isolation, *i.e.*, the tests allowing to isolate non zero components of η . Of course, as we have seen, fault isolation can be achieved in the residual generation step, where a set of residuals is generated, each of them is sensitive only to part of the faults. As this kind of fault isolation is not always possible, here we consider fault isolation through statistical tests.

As the estimation of the number of non zero components is a difficult task, we suggest to make different assumptions on this number and to solve the fault isolation problem under each of the assumptions. This way we will be able to provide information such as: if there is one non zero component of η , the most likely one is η_1 ; if there are two non zero components, the most likely pair is (η_1, η_1) , and so on. So in what follows we assume that the number of non zero components of η is known.

Let us partition η into two sub-vectors, η_a and η_b . The problem is to make a decision between the hypotheses $\eta_a \neq 0$ and $\eta_a = 0$. Without loss of generality, assume that η_a consists of the first components of η , *i.e.*,

$$\eta = \begin{pmatrix} \eta_a \\ \eta_b \end{pmatrix}$$

Remark that for the problem formulated above, nothing about the value of η_b is assumed. Typically there are two different methods to solve the problem, based on different assumptions on η_b . The first one is the *sensitivity test*, assuming $\eta_b = 0$; and the second one is the *minmax test*, assuming the least favorable value of η_b for making the decision.

According to the partition of η , make the partition

$$M = [M_a, M_b]$$

so that $M\eta = M_a\eta_a + M_b\eta_b$. Let

$$\mathbf{F} = M^T \Sigma^{-1} M$$

and make the partition

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_{aa} & \mathbf{F}_{ab} \\ \mathbf{F}_{ba} & \mathbf{F}_{bb} \end{pmatrix} = \begin{pmatrix} M_a^T \Sigma^{-1} M_a & M_a^T \Sigma^{-1} M_b \\ M_b^T \Sigma^{-1} M_a & M_b^T \Sigma^{-1} M_b \end{pmatrix}$$

The sensitivity test on η_a writes

$$\tilde{S}_a = \tilde{\zeta}_a^T \mathbf{F}_{aa} a^{-1} \tilde{\zeta}_a$$

with $\tilde{\zeta}_a = M_a^T \Sigma^{-1} \zeta$. It is a χ^2 -test whose number of degrees of freedom is equal to the dimension of η_a .

The minmax test writes

$$S_a^* = \zeta_a^{*T} \mathbf{F}_a^{*-1} \zeta_a^*$$

with $\zeta_a^* = \tilde{\zeta}_a - \mathbf{F}_{ab} \mathbf{F}_{bb}^{-1} \tilde{\zeta}_b$, $\tilde{\zeta}_a = M_a^T \Sigma^{-1} \zeta$, $\tilde{\zeta}_b = M_b^T \Sigma^{-1} \zeta$ and $\mathbf{F}_a^* = \mathbf{F}_{aa} - \mathbf{F}_{ab} \mathbf{F}_{bb}^{-1} \mathbf{F}_{ba}$.

See Basseville, (1997) [9] for some details about these tests.

In order to perform fault isolation, for a given number of non zero components of η , i.e., the dimension of η_a , make all the possible partitions of η . For each of the partitions, a sensitivity (or minmax) test (or minmax) is computed. There are two alternatives to make decisions. One is to compare each test to a threshold and to decide if the considered η_a is non zero. The other is, among all the possible partitions for a given dimension of η_a , find the partition leading to the highest sensitivity (or minmax) test. This partition then indicates the most likely faulty configuration.

11.5 Industrial Applications

In this section we present two industrial applications of FDI methods. As for all application of model based methods, a mathematical model of the considered system must be first established, often with the aid of methods for system identification. In the presentation of the FDI methods in the previous sections, a monolithic model of the considered system is assumed, for instance a complete state space model. Models of real systems may be in more complex form. Sometimes, different components of a system can be described by decoupled equations. Such particular properties should be first exploited for fault isolation before applying more complex methods, as illustrated in the first example.

11.5.1 Detection and Isolation of Faults in a Thermal Power Plant

Power plants are complex systems with many components. The example presented in this section concerns the sub-system for fuel combustion and steam generation in a thermal power plant. The FDI algorithms have been implemented in the framework of a computer assisted operation system for thermal power plants through a research project of Electricité de France and Institut de Recherche en Informatique et Systèmes Aléatoires. In the results presented in this section, the detection and isolation of two faults are considered: the leakage of feed water in the economizer and the fouling of the heat exchangers inside the combustion chamber.

Simplified Mathematical Model

Let us first introduce some notations.

p :	pressure in the water-steam tank
p_N :	nominal value of p
q_w :	flow of feed water
q_v :	flow of steam out of the water-steam tank
h_w :	enthalpy of water
h_v :	enthalpy of steam
ρ_p :	pressure stabilizing coefficient
V :	equivalent volume of the water-steam tank
T_w :	temperature of water
q_f :	flow of fuel
k_f :	calorific coefficient of the fuel
γ :	distribution coefficient of the calorific flow
δ :	normal loss of feed water flow
η :	equivalent flow of feed water leakage

The FDI algorithms are based on the following model of the considered sub-system:

$$q_w = q_v + \delta + \eta \quad (11.5.1)$$

$$\frac{dp}{dt} = [(q_w - \eta)h_w(T_w, p) - q_v h_v(p) + k_f \gamma q_f (1 - \rho_p(p - p_N))] / V \quad (11.5.2)$$

where the enthalpies $h_w(T_w, p)$ and $h_v(p)$ are respectively nonlinear functions of T_w, p and of p .

The first equation comes from the fact that the water flow entering into the water-steam tank should equal the steam flow getting out of the tank, assuming an equilibrium state in the tank. The term δ is due to various loss of water flow in the feeding pipes and in the tank under normal working conditions. The term η models the feed water leakage. When there is no leakage, $\eta = 0$.

The second equation comes from a simplified energy balance, modified with the term $\rho_p(p - p_N)$ to ensure the stability of the model. The combustion chamber fouling is modeled as changes in the value of the parameter γ : $\gamma = \gamma_0$ when there is no fouling, and $\gamma = \gamma_0 + \tilde{\gamma}$ otherwise.

The variables p, q_w, q_v, T_w, q_f are all measured with a sampling period of 5 seconds.

Residual Generation and Evaluation

It is obvious that equation (11.1) can be directly used to detect feed water leakage modeled by η . The detection of combustion chamber fouling with equation (11.1) is more complicated, due to the fact that the feed water leakage η influences also equation (11.1), therefore the residual monitoring combustion chamber fouling generated from equation (11.1) would be affected by feed water leakage. A solution to this problem is to perform an algebraic rejection of η in equation (11.1). By substituting equation (11.1) into (11.1), we obtain

$$\frac{dp}{dt} = [(q_v + \delta)h_w(T_w, p) - q_v h_v(p) + k_f \gamma q_f (1 - \rho_p(p - p_N))] / V \quad (11.5.3)$$

For notational convenience, let u be the vector collecting the measured variables q_v, q_f, T_w and

$$f(p, u, \gamma) = [(q_v + \delta)h_w(T_w, p) - q_v h_v(p) + k_f \gamma q_f (1 - \rho_p(p - p_N))] / V$$

Discretize equations (11.1) and (11.5.3) as follows:

$$q_w(t) = q_v(t) + \delta + \eta \quad (11.5.4)$$

$$p(t+1) = p(t) + \tau f(p(t), u(t), \gamma) \quad (11.5.5)$$

where t is the discrete time instant and the discretization step $\tau = 5$ seconds.

Because of modeling and measurement errors, the measured signals cannot strictly satisfy these two equations. The variables in these equations should be considered as theoretic variables, and should be distinguished from the measured signals. For this reason, we use q_w^m to note the measured q_w , p^m to note the measured p , and so on. Then we have the measurement equations

$$q_w^m(t) = q_w(t) + \varepsilon_w(t) \quad (11.5.6)$$

$$q_v^m(t) = q_v(t) + \varepsilon_v(t) \quad (11.5.7)$$

$$p^m(t) = p(t) + \varepsilon_p(t) \quad (11.5.8)$$

$$u^m(t) = u(t) + \varepsilon_u(t) \quad (11.5.9)$$

where the ε 's are noises due to modeling and measurement errors.

Based on equation (11.1), the residual monitoring feed water leakage is generated by

$$r(t) = q_w^m(t) - q_v^m(t) - \delta \quad (11.5.10)$$

where $q_w^m(t)$ and $q_v^m(t)$ are measured signals, and the parameter δ can be estimated from the signals $q_w^m(t)$ and $q_v^m(t)$ measured when there is no leakage. According to equations (11.1), (11.1) and (11.1),

$$r(t) = \begin{cases} \varepsilon_w(t) - \varepsilon_v(t) & \text{if } \eta = 0 \\ \varepsilon_w(t) - \varepsilon_v(t) + \eta & \text{if } \eta \neq 0 \end{cases}$$

Therefore the feed water leakage η affects the residual $r(t)$ in an additive manner by changing the mean of $r(t)$. This is a convenient situation for applying statistical detection methods.

For the monitoring of the combustion chamber fouling, we apply an observer based approach. The observer has the form

$$\hat{p}(t+1) = \hat{p}(t) + \tau f(\hat{p}(t), u^m(t), \gamma) + K_o(p^m(t) - \hat{p}(t)) \quad (11.5.11)$$

where $\hat{p}(t)$ is the estimation of $p(t)$ and K_o is a parameter known as the observer gain.

The nominal value of γ is identified over a set of fault free data by

$$\gamma_0 = \arg \min_{\gamma} \frac{1}{N - k_0 + 1} \sum_{t=k_0}^N (p^m(t) - \hat{p}(t))^2$$

where N is the data sample size and k_0 is the instant from which the observer (11.5.11) is considered as having converged.

Now we define the residual

$$\begin{aligned} s(t) &= -\frac{1}{2} \frac{\partial}{\partial \gamma} (p^m(t) - \hat{p}(t))^2 \\ &= \frac{\partial \hat{p}(t)}{\partial \gamma} (p^m(t) - \hat{p}(t)) \end{aligned}$$

where $\frac{\partial \hat{p}(t)}{\partial \gamma}$ is recursively computed through the differentiation of equation (11.5.11) in γ :

$$\begin{aligned} \frac{\partial \hat{p}(k+1)}{\partial \gamma} &= \frac{\partial \hat{p}(t)}{\partial \gamma} + \tau \frac{\partial}{\partial \hat{p}} f(\hat{p}(t), u^m(t), \gamma) \frac{\partial \hat{p}(t)}{\partial \gamma} \\ &\quad + \tau \frac{\partial}{\partial \gamma} f(\hat{p}(t), u^m(t), \gamma) - K_o \frac{\partial \hat{p}(t)}{\partial \gamma} \end{aligned}$$

Because γ_0 minimizes the mean square error, the derivative of the square errors $s(t)$ has zero mean when $\gamma = \gamma_0$. Moreover, with the asymptotic local approach, for large N , the distribution of

$$\zeta_N = \frac{1}{\sqrt{N - k_0 + 1}} \sum_{t=k_0}^N s(t) \quad (11.5.12)$$

is asymptotically Gaussian. A small change in γ only affects the mean of ζ_N , not its variance.

The evaluation of the residual $r(t)$ for feed water leakage can clearly be handled by the Page-Hinkley algorithm with a specification on the maximum tolerable value of η . For the evaluation of the residual $s(t)$, in principle we should work with the ζ_N as defined in (11.5.12) according to the local approach. In order to obtain an efficient on-line algorithm, we also apply the Page-Hinkley algorithm to the residual $s(t)$ as if it was independent in time and Gaussian distributed.

A sample of the measured signals $p^m, q_w^m, q_v^m, T_w^m, q_f^m$ is shown in Figure 11.5.1. The abscissa is the time in minutes. A feed water leakage is artificially simulated at about the 47th minute. A change in the behavior of the signals after the 47th minute is visible, however, it is similar to the change due to working point change at the beginning of the data sample. So it is not obvious to detect this feed water leakage by visual inspection of the signals.

The two Page-Hinkley tests monitoring feed water leakage and combustion chamber fouling are shown in Figure 11.5.2. The thresholds of these two tests are set to 20. The first test starts to grow after the occurrence of the feed water leakage and exceeds the threshold. The second test stays close to zero, before and after the feed water leakage.

In the data depicted in Figure 11.5.3, two faults are simulated. A feed water leakage occurs at about the 36th minute and a combustion chamber fouling at about the 58th minute. The two tests shown in Figure 11.5.4 react as expected. The feed water leakage fault is first detected, then combustion chamber fouling fault.

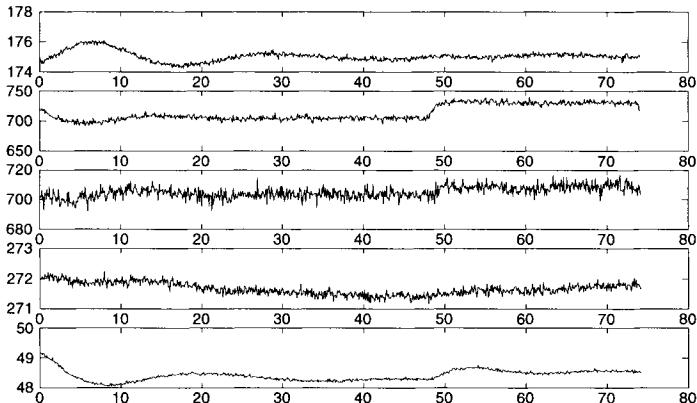


Figure 11.5.1 Data with a feed water leakage started at the 47th minute. From top to bottom:
 $p^m, q_w^m, q_v^m, T_w^m, q_f^m$.

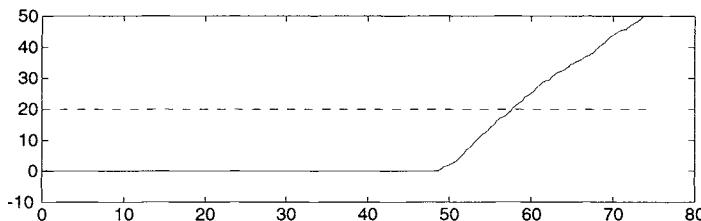


Figure 11.5.2a The Page-Hinkley test monitoring feed water leakage. The data are depicted in Figure 11.5.1 and the threshold is shown by the dashed line

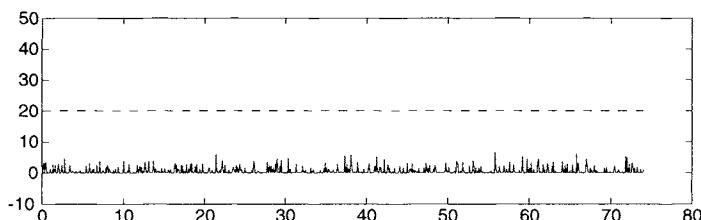


Figure 11.5.2b The Page-Hinkley test monitoring combustion fouling. The data are depicted in Figure 11.5.1 and the threshold is shown by the dashed line

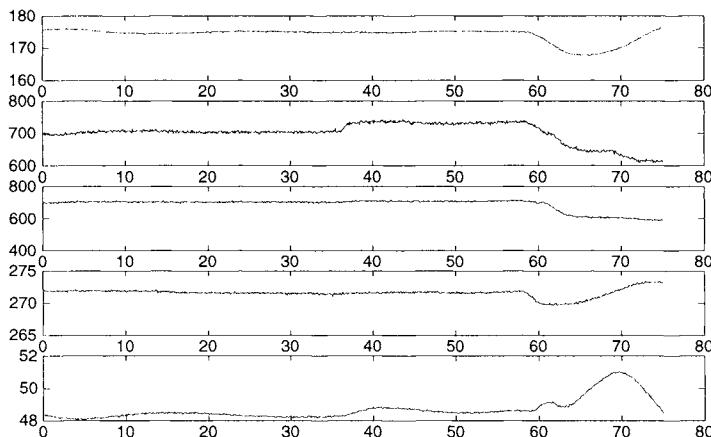


Figure 11.5.3 Data with a feed water leakage started at the 36th minute and a combustion chamber fouling at the 58th minute. From top to bottom: p^m , q_w^m , q_v^m , T_w^m , q_f^m .

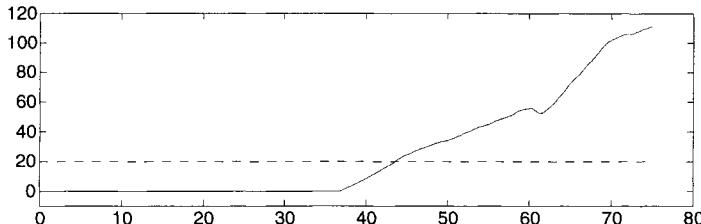


Figure 11.5.4a The Page-Hinkley test monitoring feed water leakage. The data are depicted in Figure 11.5.3 and the threshold is shown by the dashed line

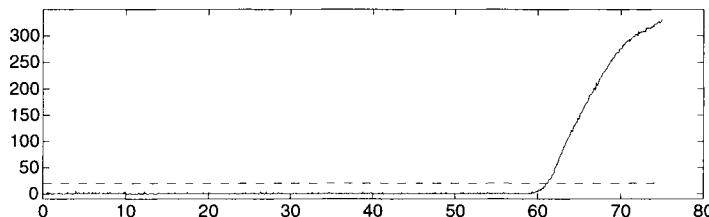


Figure 11.5.4b The Page-Hinkley test monitoring combustion fouling. The data are depicted in Figure 11.5.3 and the threshold is shown by the dashed line

These results illustrate the performance of the two tests for the detection and isolation of the two considered faults in the thermal power plant.

11.5.2 Gas Turbine Monitoring

We present in this section an example of fault detection and isolation for a gas turbine. Gas turbines can be used as jet engine or as rotating motor. In our study a rotating turbine motor is considered. The entire system is mainly composed of a compressor, 10 combustion chambers and an expansion turbine. This presentation is extracted from Mathis, (1994) [76] and has been part of a research project of European Gas Turbine SA, Alcatel-Alsthom-Recherche and Institut de Recherche en Informatique et Systèmes Aléatoires. The purpose is to monitor slow degradations of the system for maintenance optimization.

A Simple Mathematical Model

We are particularly interested in the monitoring of the 10 combustion chambers where the fuel is burned after being mixed with the air compressed by the compressor. The produced gas with

high pressure and high temperature is then guided into the expansion turbine to drive the rotor of the system.

The ten combustion chambers are uniformly located on a circle perpendicular to the rotation axis of the turbine. The temperature profile in the combustion chambers is an important indicator of the functioning of the system, but cannot be directly measured, due to the high temperature resulting from the combustion. Nevertheless, the mean value of this temperature profile, T_e , can be estimated indirectly from available measurements. Denote this temperature profile by $p_e(\varphi)$ with φ the angular coordinate. With an appropriately chosen Gaussian shape function $g(\cdot)$, the temperature profile is modeled as

$$p_e(\varphi) = T_e + \sum_{i=1}^{10} a_i g(\varphi - \phi_i^c)$$

where ϕ_i^c is the angular coordinate of the i -th combustion chamber, a_i is a coefficient. The temperature profile at the exhaust of the expansion turbine is measured by 18 thermocouples. Let y_j be the measurement given by the j -th thermocouple, then it can be modeled as

$$y_j = \frac{T_s}{T_e} \left(T_e + \sum_{i=1}^{10} a_i g(\phi_j^t - \phi_i^c - \Phi(T_e, T_s, \omega)) \right) + b \cos(\phi_j^t - \phi^s) + w_j$$

where T_s is the mean value of the temperature profile at the exhaust, ϕ_j^t is the angular coordinate of the j -th thermocouple, ω is the rotation velocity of the turbine, Φ is a known function allowing to compute the phase shift of the temperature profile due to the rotation of the turbine, the term $b \cos(\phi_j^t - \phi^s)$ takes into account the dissymmetry of the expansion turbine, w_j is the noise due to modeling and measurement errors.

Directly available measurements are $y_j(t), \omega(t)$, with the sampling period of 15 seconds. $T_s(t)$ is computed by averaging the 18 thermocouples, whereas $T_e(t)$ is computed from some other available measurements. The model parameters are identified by minimizing the mean of square errors over a data set.

Residual Generation and Evaluation

In order to detect and isolate temperature drop in the combustion chambers, we design residuals monitoring the changes in the a_i coefficients. Let

$$f_j(\theta, T_e, T_s, \omega) = \frac{T_s}{T_e} \left(T_e + \sum_{i=1}^{10} a_i g(\phi_j^t - \phi_i^c - \Phi(T_e, T_s, \omega)) \right) + b \cos(\phi_j^t - \phi^s)$$

be the model output corresponding to the measurement y_j , $\theta = (a_1, \dots, a_{10})^T$. Consider the residual

$$r(t) = \frac{1}{2} \frac{\partial}{\partial \theta} \sum_{j=1}^{18} [f_j(\theta, T_e(t), T_s(t), \omega(t)) - y_j(t)]^2$$

Because the nominal parameter θ_0 is identified by minimizing the mean of square errors over a data set, the gradient of the mean of square errors, evaluated at θ_0 , is zero. It follows that $r(t)$ has zero mean on the same data set. All changes in a_i will be reflected by the mean of $r(t)$ if the system is persistently excited.

This residual is evaluated through the asymptotic local approach as described in Section 11.4.2. The global test for fault detection (11.4.12) is applied to each block of 2000 data points. Some results are shown in Figure 11.5.5 where for each block of 2000 data points the temperature T_e averaged over the data block and the global test are shown. It can be observed that the test has an increasing tendency. With the threshold tuned to 400, the global test starts to exceed the threshold around the 250th data block. This behavior is due to the progressive fouling of the combustion chambers. It indicate that a cleaning operation should be considered during the second half of the time interval where the data are collected.

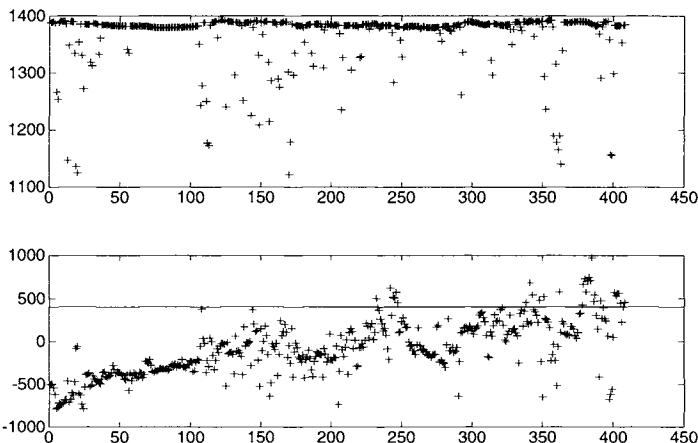


Figure 11.5.5 Detection of slow combustion chamber fouling. Top: the temperature T_e averaged over each block of 2000 data points. Bottom: global test on each block of 2000 data points.

Because no data with fault is available, a simulator is used to modify some data sets in order to simulate temperature drop in combustion chambers. First, a drop of 20 degrees in the combustion chamber number 1 is simulated. The global test applied to blocks of 2000 data points is shown in Figure 11.5.6. In most situations the test exceeds the threshold tuned to 400. Under the assumption that a fault occurs only in one of the combustion chambers, the minmax test is used to test the ten possible hypotheses. The highest test value among the ten tests indicate the most likely faulty combustion chamber. The result is also shown in Figure 11.5.6. The faulty combustion chamber is correctly isolated for 97% of the data blocks.

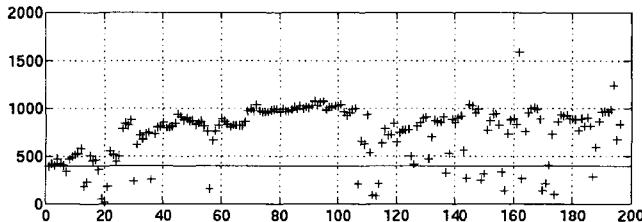


Figure 11.5.6a Detection and isolation of a single fault: global test on each block of 2000 samples.

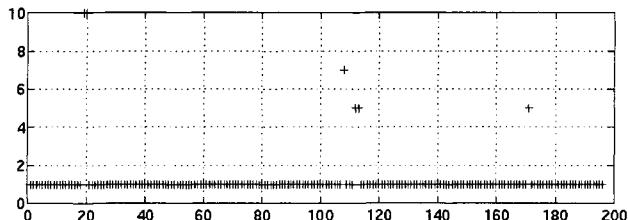


Figure 11.5.6b Detection and isolation of a single fault: the number of the faulty combustion chamber indicated by the minmax test.

Now a temperature drop of 20 degrees in the combustion chambers number 1 and number 3 is simulated. The global test applied to blocks of 2000 data points is shown in Figure 11.5.7. For all the data blocks, the test exceeds the threshold tuned to 400. Under the assumption that faults occur in two combustion chambers, the fault isolation result obtained by the minmax test is shown in Figure 11.5.7. The faulty pair of combustion chambers is successfully isolated for 98% of the data blocks.

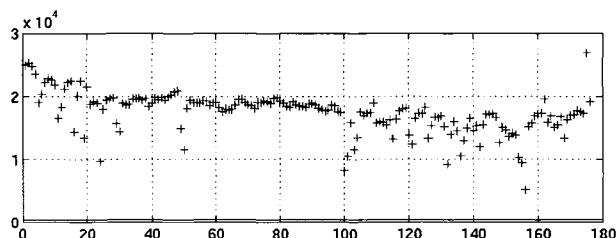


Figure 11.5.7a Detection and isolation of two faults: global test on each block of 2000 data points.

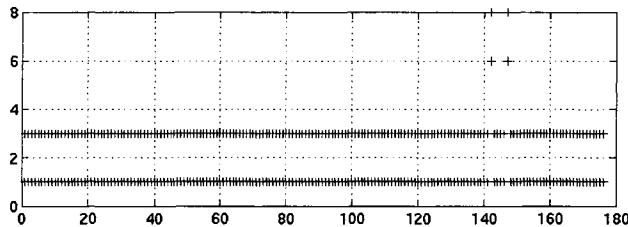


Figure 11.5.7b Detection and isolation of two faults: the pair of the faulty combustion chambers indicated by the minmax test.

For the fault isolation results shown above, we have assumed that the number of faults is known. In practice, this number is not known a priori. Such results can be used to indicate first the most probable single faulty combustion chamber, then the most probable faulty pair, and so on. Nevertheless, a method for estimating the number of faults has been proposed in Mathis, (1994) [76].

Appendix A

Refresher on Matrix Theory

This appendix presents some matrix results that are used in the book.

A.1 Definitions and Some Basic Properties of Matrices

A real matrix A of dimension $m \times n$ (m -by- n) is defined as

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \quad (\text{A.1.1})$$

where a_{ij} is a real number. Denote $\mathbb{R}^{m \times n}$ as all real $m \times n$ matrices, then we can express the matrix in (A.1.1) more compactly as $A \in \mathbb{R}^{m \times n}$. A real matrix with dimension $n \times 1$ is called a real n -vector:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (\text{A.1.2})$$

Similarly, this can also be expressed as $\mathbf{x} \in \mathbb{R}^n$.

The basic matrix computations include the transposition:

$$A^T = \begin{bmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{mn} \end{bmatrix} \quad (\text{A.1.3})$$

the addition of two matrices of the same dimension

$$A + B = \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix} \quad (\text{A.1.4})$$

the multiplication by a scalar α

$$\alpha A = \begin{bmatrix} \alpha a_{11} & \cdots & \alpha a_{1n} \\ \vdots & \ddots & \vdots \\ \alpha a_{m1} & \cdots & \alpha a_{mn} \end{bmatrix} \quad (\text{A.1.5})$$

and the multiplication of two matrices $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{r \times n}$

$$AB = \begin{bmatrix} \sum_{k=1}^r a_{1k}b_{k1} & \cdots & \sum_{k=1}^r a_{1k}b_{kn} \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^r a_{mk}b_{k1} & \cdots & \sum_{k=1}^r a_{mk}b_{kn} \end{bmatrix} \quad (\text{A.1.6})$$

A square matrix has equal number of rows and columns. The inverse of a square non-singular matrix A is denoted as A^{-1} and satisfies $A^{-1}A = AA^{-1} = I$. Here I is the identity matrix with diagonal elements being 1's and off-diagonal elements 0's. It is easy to show that $IA = AI = A$. The trace of a square matrix A is the sum of its diagonal elements and is denoted as $\text{tr}A$. Denote $\det A$ as the determinant of A and $\text{adj}A$ as the adjugate of A , then

$$A^{-1} = \frac{\text{adj}A}{\det A} \quad (\text{A.1.7})$$

Let A and B be matrices with compatible dimensions and their inverses exist, we have

$$(AB)^T = A^T B^T \quad (\text{A.1.9})$$

and

$$(AB)^{-1} = A^{-1}B^{-1} \quad (\text{A.1.10})$$

The matrix inversion lemma. Let A , B , C and D be matrices with compatible dimensions and the inverses of A and C exist. Then

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \quad (\text{A.1.11})$$

This can be proved by multiply the right hand side in (A.11) by $A + BCD$, which results in the identity matrix.

A.2 Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors. Let A be a square $n \times n$ matrix. The eigen values of A are denoted as $\lambda_1, \lambda_2, \dots, \lambda_n$ and they are the solutions to the characteristic equation

$$\det(A - \lambda I) \quad (\text{A.2.1})$$

The right eigenvector t_i corresponding to the eigenvalue λ_i is the non-zero solution to

$$At_i = \lambda_i t_i, \text{ or, } (A - \lambda_i I)t_i = 0 \quad (\text{A.2.2})$$

Similarly, the corresponding left eigenvector q_i is the nonzero solution to

$$q_i^T A = \lambda_i q_i^T, \text{ or, } q_i^T (A - \lambda_i I) = 0 \quad (\text{A.2.2})$$

The eigen values λ_i of the real matrix A has following properties:

1. The eigen values of the real matrix A are either real or in complex conjugate pairs.
2. The sum of the eigenvalues of A equals the trace of A

$$\text{tr}A = \sum_{i=1}^n \lambda_i \quad (\text{A.2.3})$$

3. The product of the eigenvalues of A is equals to its determinant

$$\det A = \prod_{i=1}^n \lambda_i \quad (\text{A.2.4})$$

4. The eigen values of a diagonal matrix

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ a_{n1} & \cdots & \cdots & a_{nn} \end{bmatrix}$$

are equal to its diagonal elements $a_{11}, a_{22}, \dots, a_{nn}$.

A.3 The Singular Value Decomposition and QR Factorization

Singular Value Decomposition (SVD)

A matrix $Q \in \mathbb{R}^{m \times m}$ is *orthogonal* if $Q^T Q = I$.

If a matrix $Q = [q_1 \ \cdots \ q_m] \in \mathbb{R}^{m \times m}$, with $q_i \in \mathbb{R}^m$, then q_i form an orthonormal basis for \mathbb{R}^m .

The existence of the SVD is summarized in the next result.

Theorem A.3.1 (Golub and van Loan, 1989). *Let $A \in \mathbb{R}^{m \times n}$, then there exists orthogonal matrices:*

$$U = [u_1 \ \cdots \ u_m] \in \mathbb{R}^{m \times m} \quad \text{and} \quad V = [v_1 \ \cdots \ v_n] \in \mathbb{R}^{n \times n}$$

such that,

$$U^T A V = \begin{bmatrix} \sigma_1 & & 0 & | & 0 \\ & \ddots & & | & \vdots \\ 0 & & \sigma_p & | & 0 \\ \hline 0 & & & | & 0 \end{bmatrix} \in \mathbb{R}^{m \times n} \quad p \leq \min\{m, n\} \quad (\text{A.3.1})$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > 0$ and where some of the zero matrices may be empty.

The SVD of a matrix allows to characterize a number of geometric properties of a matrix representing a linear transformation. These are the (co-)range and (co-)null space or (co-)kernel of a matrix and are defined as:

Let $A \in \mathbb{R}^{m \times n}$, then we have:

$$\begin{aligned} \text{range}(A) &= \{y \in \mathbb{R}^m | y = Ax \text{ for some } x \in \mathbb{R}^n\} \\ \text{rank}(A) &= \dim(\text{range}(A)) \\ \text{kern}(A) &= \{x \in \mathbb{R}^n : Ax = 0\} \\ \text{co-range}(A) &= \text{kern}(A^T) \\ \text{co-kern}(A) &= \text{range}(A^T) \end{aligned} \quad (\text{A.3.2})$$

Given a matrix A , a basis for the different geometric spaces defined in Definition A.3.1 can reliably be derived from the SVD of A .

Let the SVD of $A \in \mathbb{R}^{m \times n}$ be denoted as in Theorem A.3.1 and let $p \leq n$, then

$$\begin{aligned} \text{range}(A) &= \text{span}\{u_1, \dots, u_p\} \subset \mathbb{R}^m \\ \text{rank}(A) &= p \\ \text{kern}(A) &= \text{span}\{v_{p+1}, \dots, v_n\} \subset \mathbb{R}^n \\ \text{co-range}(A) &= \text{span}\{u_{p+1}, \dots, u_m\} \subset \mathbb{R}^m \\ \text{co-kern}(A) &= \text{span}\{v_1, \dots, v_p\} \subset \mathbb{R}^n \end{aligned} \quad (\text{A.3.3})$$

Finally, one important application to be used in Chapter 8 is the column (or row) compression of a low rank matrix. Let A be as given in Theorem A.3.1, then the column compression of A is given as:

$$AV = [A_1 \ 0] \quad (\text{A.3.4})$$

with A_1 a full column rank matrix. With the SVD of A , the matrix A_1 is explicitly given as:

$$A_1 = [\sigma_1 u_1 \ \cdots \ \sigma_p u_p] \quad (\text{A.3.5})$$

A.3.2 The Rank of a Product of Two Matrices

Though the SVD is a numerically reliable tool to assess the (numerical) rank of a matrix, for analysis of the rank of a matrix that is the product of two matrices a useful result is the so called *Sylvester's inequality* [65].

Lemma A.3.2 (Sylvester's inequality, Kailath, 1980). *Let $M_1 \in R^{m \times n}$ and $M_2 \in R^{n \times p}$, then:*

$$\rho(M_1) + \rho(M_2) - n \leq \rho(M_1 M_2) \leq \min(\rho(M_1), \rho(M_2)) \quad (\text{A.3.5})$$

A.3.3 The QR Factorization of a Matrix

Let A be a $m \times n$ matrix with $m \geq n$. The QR factorization of A is given by

$$A = QR \quad (\text{A.3.6})$$

where Q is an $m \times m$ orthogonal matrix and R is an $m \times n$ upper triangular matrix.

A.4 The Hankel Matrix of a Linear Process

In recent years, the Hankel matrix has played a very important role in identification, model reduction and robust control (H_∞ control). The Hankel matrix $\boldsymbol{\kappa}$ of a discrete-time process $G(q)$ is defined as the double infinite matrix

$$\boldsymbol{\kappa} = \begin{bmatrix} G_1 & G_2 & G_3 & \cdots \\ G_2 & G_3 & G_4 & \cdots \\ G_3 & G_4 & G_5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (\text{A.4.1})$$

where $\{G_k\}_{k=1,\dots,\infty}$ is the impulse response matrix of $G(q)$.

Denote n as the minimal order of $G(q)$ (also called Mcmillan degree which is the minimum order of state space realisation of $G(q)$), then it is well known that

$$\text{rank}(\boldsymbol{\kappa}) = n \quad (\text{A.4.2})$$

The n singular values of $\boldsymbol{\kappa}$, $h_1(G), \dots, h_n(G)$, are called Hankel singular values (HSV) of the process $G(q)$; and, by convention, they are ordered as

$$h_1(G) \geq h_2(G) \geq \cdots \geq h_{n+1}(G) \quad (\text{A.4.3})$$

where $h_1(G)$ is also called the *Hankel norm* of $G(q)$.

Bibliography

- [1] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Autom. Control*, Vol. AC-19, pp. 716-723.
- [2] Akaike, H. (1981). Modern development of statistical methods. In P. Eykhoff, ed.; *Trends and Progress in System Identification*, Pergamon Press, Elmsford, New York. Åström, K.J. and P. Eykhoff (1971). System identification - a survey. *Automatica*, Vol. 7, pp. 123-162.
- [3] Åström, K.J. and T. Bohlin (1965). Numerical identification of linear dynamic systems from normal operating records. IFAC Symposium on Self-Adaptive Systems, Teddington, England.
- [4] Åström, K.J. and P. Eykhoff (1971). System identification - a survey. *Automatica*, Vol. 7, pp. 123-162.
- [5] Åström, K.J. and T. Hägglund (1995). *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America. Research Triangle Park, N.C.
- [6] Åström, K.J. and B. Wittenmark (1984). *Computer Controlled Systems: Theory and Design*. Prentice-Hall, Englewood Cliffs, N.J.
- [7] Backx, A.C.P.M. (1987). *Identification of an Industrial Process: A Markov Parameter Approach*. Dr. dissertation, Dept. EE., Eindhoven University of Technology, Eindhoven, The Netherlands.
- [8] Backx, A.C.P.M. and A.A.H. Damen (1989). Identification of industrial MIMO processes for fixed controllers-part 1 and part 2. *Journal A*, Vol. 30, pp. 3-12, and pp. 33-43.
- [9] Basseville, M. (1997). Information criteria for residual generation and fault detection and isolation. *Automatica*, 33(6):783-803.
- [10] Basseville, M. (1998). On-board component fault detection and isolation using the statistical local approach. *Automatica*, 34(11):1391-1415.
- [11] Basseville, M., M. Abdelghani and A. Benveniste (2000). Subspace-based fault detection algorithms for vibration monitoring. *Automatica*, 36(1):101-109.

- [12] Basseville, M., A. Benveniste, B. Gach-Devauchelle, M. Goursat, D. Bonnecase, P. Dorey, M. Prevosto and M. Olagnon (1993). Damage monitoring in vibration mechanics: issues in diagnostics and predictive maintenance. *Mechanical Systems and Signal Processing*, 7(5):401–423.
- [13] Basseville, M. and I. Nikiforov (1993). *Detection of Abrupt Changes – Theory and Applications*. Information and System Sciences Serie. Prentice Hall, Englewood Cliffs, N.J.
- [14] Beard, R. V. (1971). *Failure accommodation in linear system through self reorganization*. PhD thesis, MIT, USA.
- [15] Benveniste, A., M. Basseville, L. El Ghaoui, R. Nikoukhah and A. Willsky (1993). On the use of descriptor systems for failure detection and isolation. In *IFAC'93 World Congress*, Sydney.
- [16] Benveniste, A., M. Basseville and G. Moustakides (1987). The asymptotic local approach to change detection and model validation. *IEEE Trans. on Automatic Control*, 32:583–592.
- [17] Beyer, J., G. Gens and J. Wernstedt (1979). Identification of nonlinear systems in closed loop. *Proceedings of 5th IFAC Symp. on Identification and System Parameter Estimation*, Darmstadt, pp. 661-667.
- [18] Billings, S.A. and S.Y. Fakhouri (1982). Identification of systems containing linear dynamics and static nonlinear elements. *Automatica*, Vol. 18, No. 1, pp. 15-26.
- [19] Box, G.E.P. and G.M. Jenkins (1970). *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco.
- [20] Chang, F.H.I. and R. Luus (1971). A noniterative method for identification using Hammerstein model. *IEEE Trans. Autom. Control*, Vol. AC-16, pp.464–468.
- [21] Chen, H.-W. (1995). Modeling and identification of parallel nonlinear systems: structural classification and parameter estimation methods. *Proceedings of the IEEE*, Vol. 83, No. 1, pp. 39-66.
- [22] Chen, J. and R.J. Patton (1999). *Robust model-based fault diagnosis for dynamic systems*. Kluwer Academic Publishers, Boston, Dordrecht, London.
- [23] Chernyshov, K. (2000). Identification of stochastic N-L-N systems using monotone correlations. *Proceedings of 12th IFAC Symposium on System Identification*, 21-23 June, 2000, Santa Barbara, USA.
- [24] Chien, I.-L. and P.S. Fruehauf (1990). Consider IMC tuning to improve controller performance. *Process Control*. October, 1990, pp 33-41.
- [25] Chou, C.T., H.H.J. Bloemen, V. Verdult, T.J.J. van den Boom, T. Backx and M. Verhaegen(1999). Nonlinear identification of high purity distillation columns. Internal report, University of Twente, The Netherlands.

- [26] Chou, C.T., and M. Verhaegen (1997). Subspace algorithms for the identification of multivariable dynamic errors-in-variables models. *Automatica*, Vol. 33, No. 10, pp. 1857-1869.
- [27] Chow, E. Y. and A.S. Willsky (1984). Analytical redundancy and the design of robust detection systems. *IEEE Trans. on Automatic Control*, 29(7):603-614.
- [28] Ciccarella, G., M.D. Mora and A. Germani (1993). A Luenberger-like observer for nonlinear systems. *International Journal of Control*, 57(3):537-556.
- [29] Clark, R. N., D.C. Fosth and V.M. Walton (1975). Detecting instrument malfunctions in control systems. *IEEE Trans. Aero. Electron. Syst.*, 11:465-473.
- [30] Clarke, D.W. (1967). Generalized least squares estimation of parameters of a dynamic model. *Proceedings of First IFAC Symposium on Identification in Automatic Control Systems*, Prague, paper 3.17.
- [31] Cutler, C.R. and R.B. Hawkins (1988). Application of a large predictive multivariable controller to a hydrocracker second stage reactor. *Proceedings of American Control Conference*, pp. 284-291.
- [32] Doyle, J.C. (1982). Analysis of feedback systems with structured uncertainty. *IEE Proceedings*, Part D, Vol. 129, pp. 242-250.
- [33] Erol, F. (1999). Nonlinear PID control using Hammerstein or Wiener models. M.Sc. Thesis. Faculty of Electrical Engineering, Eindhoven University of Technology, The Netherlands.
- [34] Eskinat, E., S.H. Johnson and W.L. Luyben (1991). Use of Hammerstein models in identification of nonlinear systems. *AIChE Journal*, Vol. 37, No. 2, pp. 255-268.
- [35] Eykhoff, P. (1974). *System Identification: Parameter and State Estimation*. John Wiley & Sons, New York.
- [36] Falkner, A.H. (1988). Iterative technique in the identification of a non-linear system. *International Journal of Control*, Vol. 48, No.1 pp. 385-396.
- [37] Forssell U. and L. Ljung (1998). Identification for control: some results on optimal experiment design. *Report LiTH-ISY-R-2014*. Dept. EE, Linkoping University, Sweden.
- [38] Forssell U. and L. Ljung (2000). Identification of unstable systems using output error and Box-jenkins model structures. *IEEE Trans. Autom. Control*. Vol 45, pp 131-147.
- [39] Frank, P. M. (1987). Fault diagnosis in dynamic system via state estimation - a survey. In Tzafestas, Singh, and Schmidt, editors, *System fault diagnostics, Reliability and Related Knowledge-Based Approaches*, volume 1, pages 35-98. D. Reidel Presss.
- [40] Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge based redundancy - a survey and new results. *Automatica*, 26:459-474.

- [41] Gauthier, J.-P., H. Hammouri and S. Othman (1992). A simple observer for nonlinear systems, application to bioreactors. *IEEE Trans. on Automatic Control*, 37(6):875–880.
- [42] Gertler, J. J. (1988). Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine*, 8(6):3–11.
- [43] Gertler, J. J. (1998). *Fault detection and diagnosis in engineering systems*. Marcel Dekker, Inc., New York, Basel, Hong Kong.
- [44] Gevers, M. and L. Ljung (1986). Optimal experiment design with respect to the intended model application. *Automatica*, Vol. 22, pp. 543-554.
- [45] Glover, K. (1984). All optimal Hankel-norm approximations of linear multivariable systems and their L_∞ -error bounds. *Int. J. Control*, Vol. 39, pp. 1115-1193.
- [46] Golub, G. and C. van Loan (1989). *Matrix Computations*. John Hopkins.
- [47] Golub, G.H. and V. Pereyra (1973). The differentiation of pseudo-inverse and nonlinear least squares problems whose variables separate. *SIAM J. Numer. Anal.*, Vol. 10, No. 2, pp 413-432.
- [48] Goodwin, G.C. and M.E. Salgado (1989). A stochastic embedding approach for quantifying uncertainty in the estimation of restricted complexity models. *Int. J. Adaptive Control and Signal Processing*, Vol.3, pp. 333-356.
- [49] Goodwin, G.C., M. Gevers and B. Ninness (1992). Quantifying the error in estimated transfer functions with application to model order selection. *IEEE Trans. Autom. Control*, Vol. 37, No. 7, pp. 913-928.
- [50] Greblicki, W. (1992). Nonparametric identification of Wiener systems. *IEEE Trans. Information Theory*, Vol. 38, No. 5, pp. 1487-1493.
- [51] Guernez, C., J.P. Cassar and M. Staroswiecki (1997). Extension of parity space to nonlinear polynomial dynamic systems. In *Proc. Safeprocess'97*, Hull, UK.
- [52] Guo, L. and L. Ljung (1994). The role of model validation for assessing the size of the unmodelled dynamics. *Proceedings of the 33rd Conference on Decision and Control*, Lake Buena Vista, Florida, pp. 3894-3899.
- [53] Gustavsson, I., L. Ljung and T. Soderstrom (1977). Identification of processes in closed loop-Identifiability and accuracy aspects. *Automatica*, Vol. 13, pp. 59-75.
- [54] Hakvoort, R.G. and P.M.J. Van den Hof (1997). Identification of probabilistic uncertainty regions by explicit evaluation of bias and variance errors. *IEEE Trans. Autom. Control*, Vol. AC-42, no. 11, pp. 1516-1528.
- [55] Haverkamp, B. and M. Verhaegen. SMI Toolbox: state space model identification software for multivariable dynamical systems, v. 0.9, <http://www.sce.tn.utwente.nl/index2.html>.

- [56] Hjalmarsson, H., M. Gevers, F. de Bruyne (1996). For model-based control design, closed-loop identification gives better performance. *Automatica*, Vol. 32, No. 12, pp. 1659-1673.
- [57] Hsia, T.C. (1977). System Identification: Least-Squares mehtods. Lexington Books, Lexington.
- [58] Ho, B.L. and R.E. Kalman (1960). Effective construction of linear, state-variable models from input/output functions, *Regelungstechnik*, Vol. 14, pp. 545-548.
- [59] Isermann, R. (1984). Fault detection based modeling and estimation methods – a survey. *Automatica*, Vol. 20, pp. 387-404.
- [60] Isermann, R. (1984). Process fault detection based on modelling and estimation methods: a survey. *Automatica*, 20(4):387-404.
- [61] Isermann, R. (1993). Fault diagnosis of machines via parameter estimation and knowledge processing - tutorial paper. *Automatica*, 29(4):815–836.
- [62] Isermann, R. (1997). Supervision, fault detection and fault diagnosis methods - an introduction. *Control Engineering Practice*, 5(5):639–652.
- [63] Jacobsen, E.W. (1994). Identification for Control of Strongly Interactive Plants. AIChE Annual Meeting, San Fransisco.
- [64] Jansson, M. and B. Wahlberg, B. (1998). On consistency of subspace identification meth- ods. *Automatica*, Vol. 34, No. 12, pp. 1507-1519.
- [65] Jones, H.L. (1973). *Failure detection in linear systems*. PhD thesis, MIT.
- [66] Kailath, T. (1980). *Linear Systems*. Prentice Hall, Englewood Cliffs, N.J.
- [67] Kosut, R.L., M. Lau and S. Boyd (1990). Identification of systems with parametric and nonparametric uncertainty. *Proc. of American Control Conference*, San Diego, June 1990, pp. 2412-2417.
- [68] Koung, C.-W. and J.F. MacGregor (1993). Design of identification experiments for robust control. A geometric approach for bivariate processes. *Ind. Eng. Chem. Res.*, Vol. 32, pp. 1658-1666.
- [69] Kučera, V. (1991). *Anylysis and Design of Discrete Linear Control Systems*. Prentice Hall, Hemel Hempstead, UK.
- [70] Lancaster and Šalkauskas (1986). *Curve and Surface Fitting: An Introduction*. Academic Press, London.
- [71] Ljung, L. (1985). Asymptotic variance expressions for identified black-box transfer function models. *IEEE Trans. Autom. Control*, Vol. AC-30, pp. 834-844.

- [72] Ljung, L. (1987, 2nd Edition, 1999). *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, N.J.
- [73] Ljung, L. and T. Glad (1994). On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2):265–276.
- [74] Ljung, L. and Z.D. Yuan (1985). Asymptotic properties of black-box identification of transfer functions. *IEEE Trans. Autom. Control*, Vol. AC-30, pp.514-530.
- [75] Marino, R. and P. Tomei (1995). *Nonlinear control design*. Information and system sciences. Prentice Hall, London, New York.
- [76] Massoumnia, M.A., G.C. Vergheese and A.S. Willsky (1989). Failure detection and identification. *IEEE Trans. Automatic Control*, 34(3):316–321.
- [77] Mathis, G. (1994). *Surveillance de turbine à gaz*. PhD thesis, Université de Rennes 1. In French.
- [78] *Matlab User's Guide*. The MathWorks.
- [79] Mehra, R.K. (1974). Optimal input signals for parameter estimation in dynamic systems—survey and new results. *IEEE Trans. Autom. Control*, Vol. AC-19, pp. 753-768.
- [80] Morari, M. and E. Zafiriou (1989). *Robust Process Control*. Prentice-Hall, Englewood Cliffs, N.J.
- [81] Moraal, P. E. and J.W. Grizzle (1995). Observer design for nonlinear systems with discrete-time measurements. *IEEE Trans. Autom. Control*, 40(3):395–404.
- [82] Narendra, K.S. and P.G. Gallman (1966). An iterative method for the identification of nonlinear systems using Hammerstein model. *IEEE Trans. Autom. Control*, Vol. AC-11, No. 31, pp. 546-550.
- [83] Nikoukhah, R. N. (1994). Innovations generation in the presence of unknown inputs: application to robust failure detection. *Automatica*, 30(12):1851–1867.
- [84] Nikoukhah, R.N. (1998). A new methodology for observer design and implementation. *IEEE Trans. Autom. Control*, 43(2):229 –234.
- [85] O'Reilly, J. (1983). *Observers for linear systems*. Academic Press.
- [86] Pajunen, G. (1992). Adaptive control of Wiener type nonlinear systems. *Automatica*. Vol. 28, No. 4, pp. 781-785.
- [87] Patton, R., P. Frank and R. Clark editors (1989). *Fault Diagnosis in Dynamic Systems - Theory and Application*. International Series in Systems and Control Engineering. Prentice Hall International, London, UK.

- [88] Peternell, K., W. Scherrer, M. Deister (1996). Statistical analysis of novel subspace identification methods, *Signal Processing* special issue, Vol. 52, pp. 161-177.
- [89] Richalet, J. (1993). Industrial applications of model based predictive control. *Automatica*, Vol. 29, No. 5, pp. 1251-1274.
- [90] Rivera, D.E., M. Morari and S. Skogestad (1986). Internal model control 4. PID controller design. *Ind. Eng. Chem. Process Des. Dev.*, Vol. 25, pp. 252-265.
- [91] Schoukens, J. and R. Pintelon (1991). *Identification of Linear Systems: A Practical Guide-line to Accurate Modeling*. Pergamon Press, London.
- [92] Schroeder. M. (1970). Sythesis of low-peak factor signals and binary sequences with low autocorrelation. *IEEE Trans. Inform. Theory*, Vol. IT-16, pp. 85-89.
- [93] Söderström, T. and P. Stoica (1982). Some properties of the output error method. *Automatica*, Vol. 18, pp. 93-99.
- [94] Söderström, T. and P. Stoica (1989). *System Identification*. Prentice-Hall, New York.
- [95] Söderström, T. (1974). Convergence properties of the generalised least squares identification method. *Automatica*, Vol. 10, pp. 617-626.
- [96] Steiglitz, K. and L.E. McBride (1965). A technique for the identification of linear system. *IEEE Trans. Autom. Control*, Vol. AC-10, pp.461-464.
- [97] Stoica, P. and T. Söderström (1982). Instrumental-variable methods for identification of Hammerstein systems. *Int. J. Control*, Vol. 35, No. 3, pp. 459-476.
- [98] Stoica, P. and T. Söderström (1981). The Steiglitz-McBride identification algorithm revisited—convergence analysis and accuracy aspects. *IEEE Trans. Autom. Control*, Vol. AC-26, No. 3, pp. 712-717.
- [99] S. Skogestad, "Dynamics and control of distillation columns - A tutorial introduction", *Trans. IChemE*, Vol. 75, Part A, 1997.
- [100] Skogestad, S. MATLAB Distillation Column Model. Online documentation at http://www.chembio.ntnu.no/users/skoge/book/matlab_m/cola/cola.html.
- [101] Stec, P. and Y.C. Zhu (2001). Some study on identification of ill-conditioned processes for control. To be presented in ACC'01, June 25-27, Arlington, USA.
- [102] Stogestad, S. and I. Postlethwaite (1996). *Multivariable Feedback Control*. Wiley, Chichester.
- [103] Tjärnström, F. and L. Ljung (1999). L_2 model reduction and variance reduction. Report No. LiTH-ISY-R-2158, Department of Electrical Engineering, Linköping University, S-581 83 Linköping, Sweden.

- [104] Tulleken, H.J.A.F. (1990). Generalized binary noise test-signal concept for improved identification-experiment design. *Automatica*, Vol. 26, No. 1, pp. 37-49.
- [105] Unbehauen, H. and G.P. Rao (1990). Continuous-time approaches to system identification—a survey. *Automatica*. Vol. 26, pp 23-35.
- [106] van den Boom, T., M. Klompstra and A. Damen (1991). System identification for H_∞ -robust control. *Preprint of the 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, Budapest, July, 1991. pp. 1431-1436.
- [107] van den Hof, P.M.J. (1997). Closed-loop issues in system identification. In Proceedings of the 11th IFAC Symposium on System Identification, Fukuoka, Japan, Vol. 4, pp. 1651-1664.
- [108] Van der Ouderaa, E., J. Schoukens and J. Renneboog (1998). Peak factor minimization using a time-frequency domain swapping algorithm. *IEEE Trans. Instrum. and Meas.*, Vol. 37, pp. 342-351.
- [109] van Overschee, P. and B. de Moor (1996). *Subspace identification: theory, implementation, application*. Kluwer Academic Publishers.
- [110] Verdult, V., M. Verhaegen, C.T. Chou and M. Lovera (1998). Efficient and systematic identification of MIMO bilinear state space models. *Proc. of IEEE CDC Conference*, Tampa, FL, USA
- [111] Verhaegen, M. (1993). Subspace model identification, part 3: analysis of the ordinary output error state space model identification algorithm. *Int. J. Control*, Vol. 58, No.3, pp. 555-586.
- [112] Verhaegen, M. (1994). Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, Vol. 30, No. 1, pp. 61-74.
- [113] Verhaegen, M. (1998). Accurate identification of the temerature-product quality relationship in a multi-component distillation column, *Chemical Engineering Communications*, Vol. 163, pp. 111-132.
- [114] Verhaegen, M. and Dewilde (1992). Subspace model identification, part 1: the output-error state-space model identification class of algorithms. *Int. J. Control*, Vol.56, No.5, pag. 1187-1210.
- [115] Verhaegen, M. and D. Westwick (1996). Identifying MIMO Hammerstein systems in the context of Subspace Model Identification methods," *Int. J. Control*, Vol. 63, No. 2, pp. 331-349.
- [116] Vidyasagar, M. (1985). *Control System Synthesis: A Factorization Approach*. MIT Press, Cambridge, Massachusetts.

- [117] Wahlberg, B. (1989). Model reduction of high-order estimated models: the asymptotic ML approach. *Int. J. Control.*, Vol. 49, No. 1, pp 169-192.
- [118] Wahlberg, B. and L. Ljung (1986). Design variables for bias distribution in transfer function estimation. *IEEE Trans. Autom. Control*, Vol. AC-31, pp. 134-144.
- [119] Walcott, B. L., M.J. Corless and S.H. Zak (1987). Comparative study of non-linear state observation techniques. *International Journal of Control*, 43(6):2109-2132.
- [120] Westwick, D. and M. Verhaegen (1996). Identifying MIMO Wiener systems using subspace model identification, *Signal Processing*, Vol. 52, pp. 235-258.
- [121] Wigren, T. (1993). Recursive prediction error identification using the nonlinear Wiener model. *Automatica*, Vol. 29, No. 4, pp. 1011-1025.
- [122] Yuan, Z.D. and L. Ljung (1985). Unprejudiced optimal open loop input design for identification of transfer functions. *Automatica*, Vol. 21, pp. 697-708.
- [123] Zadeh, L.A. (1962). From circuit theory to system theory. *Proc. IRE*, Vol. 50, pp. 856-865.
- [124] Zames, G. (1981). Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Trans. Autom. Control*, Vol. AC-26, pp. 301-320.
- [125] Zhang, Q. and M. Basseville (1998). Monitoring nonlinear dynamical systems: a combined observer-based and local approach. In *37th IEEE Conference on Decision and Control (CDC'98)*, Tampa, USA.
- [126] Zhang, Q., M. Basseville and A. Benveniste (1994). Early warning of slight changes in systems. *Automatica*, 30(1):95-113. Special Issue on Statistical Methods in Signal Processing and Control.
- [127] Zhang, Q., M. Basseville and A. Benveniste (1998). Fault detection and isolation in nonlinear dynamic systems: A combined input-output and local approach. *Automatica*, 34(11):1359-1373.
- [128] Zhu, Y.C. (1989a). Estimation of transfer functions: asymptotic theory and a bound of model uncertainty. *Int. J. Control.*, Vol. 49, pp 2241-2258.
- [129] Zhu, Y.C. (1989b). Black-box identification of MIMO transfer functions: asymptotic properties of prediction error models. *Int. J. Adaptive Control and Signal Processing*, Vol. 3, pp. 357-373.
- [130] Zhu, Y.C. (1990). *Identification and Control of MIMO Industrial Processes: An Integration Approach*. Dr. Dissertation, Dept. EE, Eindhoven University of Technology, The Netherlands.

- [131] Zhu, Y.C. (1998). Multivariable process identification for MPC: the asymptotic method and its applications. *Journal of Process Control*, Vol. 8, No. 2, pp. 101-115.
- [132] Zhu, Y.C. (1999). Parametric Wiener model identification for control. *Proceedings of IFAC World Congress*, July, 1999, Beijing, Vol. H, pp 37-42.
- [133] Zhu, Y.C. (2000a). On the use of error criteria in identification for control. *Proceedings of 12th IFAC Symposium on System Identification*, 21-23 June, 2000, Santa Barbara, USA
- [134] Zhu, Y.C. (2000b). Tai-Ji ID: Automatic closed-loop identification package for model based process control. *Proceedings of 12th IFAC Symposium on System Identification*, 21-23 June, 2000, Santa Barbara, USA
- [135] Zhu, Y.C. (2000c). Hammerstein model identification for control using ASYM. *International Journal of Control*, Vol. 73, No.18 pp. 1692-1702.
- [136] Zhu, Y.C. (2001). Identification of an N-L-N Hammerstein-Wiener model. Submitted to *Automatica*.
- [137] Zhu, Y.C. and A.C.P.M. Backx (1993). *Identification of Multivariable Industrial Processes: for Simulation, Diagnosis and Control*. Springer-Verlag, London.
- [138] Zhu Y.C., A.C.P.M. Backx and P. Eykhoff (1991). Multivariable process identification based on frequency domain measures. *Proceedings of 30-th IEEE Conference on Decision and Control*, Brighton, December, 1991, pp 303-308.
- [139] Zhu, Y.C. and P.P.J. van den Bosch (2000). Optimal closed-loop identification test design for internal model control. *Automatica*, Vol. 36, pp. 1237-1241.

Index

- Abdelghani, M., 305
Actuator, 9, 32, 39, 43, 218, 241, 253
Adaptive control, 151
Adaptive estimation, 151
Advanced process control (APC), 4, 251
Akaike's Final Prediction Error criterion, 136
Akaike, H., 136
Alias, 58
Anti-aliasing filter, 58
Approximate model, 163
Approximation, 15
ARMAX model, 118
ARX model, 73, 78, 93, 101, 113, 131, 134, 173, 185, 186, 223, 225, 233, 243
Astrom, K.J., xiv, 25, 56–58, 70, 119, 258
Asymptotic distribution, 123
Asymptotic method, 162, 186
Asymptotic theory, 162, 183
Asymptotic variance, 165
Autocorrelation function, 24, 25, 117
Backx, A.C.P.M., 59, 60, 131, 141, 162, 195, 267, 268
Bandwidth, 37, 39, 40, 54, 57
Basseville, M., 294, 297, 305–308, 311, 314, 315, 317
Batch process, 8
Benveniste, A., 297, 305–307, 314
Beyer, J., 218
Bias, 92, 94, 105, 110, 124, 125, 132
Biased, 75, 93, 95, 96
Billings, S.A., 28, 231
Black-box, xiii, 164
Block-oriented model, 28, 217
Bloemen, H.H.J., 267, 268
Bohlin, T., 119
Bonnecase, D., 305
Box-Jenkins model, 120, 126, 134, 142, 158, 175, 179, 184, 189, 214, 218, 232, 242
Box, G.E.P., 120
Boyd, S., 162
Cassar, J.P., 306
Central limit theorem, 166
Chang, F.H.I., 218
Chen, H.-W., 28
Chen, J., 294, 297
Chernyshov, K., 241
Chiens, I.-L., 258
Chou, C.T., 211, 213, 216, 267, 268
Chow, E.Y., 295
Ciccarella, G., 307
Clark, R.N., 294, 295
Clarke, D.W., 114, 115
Closed-loop identification, 133
Closed-loop system, 41, 169
Closed-loop test, 52, 55, 113, 165, 166, 169
Condition number, 264
Consistency, 133
Consistent estimator, 89
Continuous-time model, 15
Control, 4
Controlled variable (CV), 9, 20, 32
Convergence, 105
Corless, M.J., 307
Covariance matrix, 90
Criterion, 2
Cross-correlation function, 24
Cross-criterion approach, 136
Cross-spectrum, 24, 25
Cross-validation, 79, 135
Crude unit, 256, 277

- Cutler, C.R., 251, 253
- Damen, A., 162
- Damen, A.A.H., 59, 60
- Data, 2
- Data acquisition, 58
- Data pretreatment, 31
- de Bruyne, F., 55, 257
- De Moor, B., 211
- Deethanizer, 284
- Delay, 7, 10, 16, 19, 40, 61
- Design variable, 41, 177
- Dewilde, 209
- Discrete Fourier transform (DFT), 23
- Distillation column, 11
- Disturbance, 1
- Disturbance variable (DV), 32
- DMC, 277, 284
- Dorey, P., 305
- Doyle, J.C., 162
- Dynamic, 1
- El Ghaoui, L., 297
- Equation error, 73
- Equation error criterion, 102
- Equation error method, 65, 75
- Equation error model, 93
- Erol, F., 260
- Error criterion, 67
- Eskinat, E., 218, 223
- Estimation, 2
- Estimator, 67
- Expectation, 24
- Experiment, 2, 31
- Extended observability matrix, 202
- Eykhoff, P., xiv, 1, 24, 25, 27, 41, 70, 141, 162
- Fakhouri, S.Y., 28, 231
- Falkner, A.H., 241
- Fault detection and isolation (FDI), 293
- Final output error criterion (FOE), 136, 176, 226
- Finite impulse response (FIR), 68
- Forgetting factor, 153
- Forssell, U., 134, 135, 172
- Fosth, D.C., 295
- Frank, P.M., 294, 295
- Frequency domain, 23, 52, 94, 95, 98, 124, 128, 174, 176
- Frequency domain method, 14
- Frequency response, 19, 40, 43, 162
- Frequency weighting, 95, 98, 125, 174
- Fruehauf, P.S., 258
- Gach-Devauchelle, B., 305
- Gain, 38
- Gallman, P.G., 28, 218, 223, 227
- Gauss-Newton method, 105
- Gaussian, 49
- Gaussian distribution, 123, 164
- Gauthier, J.-P., 307, 308
- Generalized least-squares (GLS) method, 114
- Gens, G., 218
- Germani, A., 307
- Gertler, J.J., 294, 295
- Gevers, M., 55, 141, 162, 257
- Glad, T., 307
- Glass tube drawing process, 10, 142, 191
- Global minimum, 104
- Glover, K., 190, 191
- Golub, G.H., 116, 225, 332
- Goodwin, G.C., 141, 162
- Goursat, M., 305
- Gradient, 106
- Greblicki, W., 231, 232, 234
- Grizzle, J.W., 307
- Guernez, C., 306
- Guo, L., 133
- Gustavsson, I., 257
- Hagglund, T., 258
- Hakvoort, R.G., 141
- Hammerstein model, 28, 51, 218
- Hammerstein-Wiener model, 241
- Hammouri, H., 307, 308
- Hankel matrix, 135, 190
- Hankel singular value, 191

- Haverkamp, B., 214, 216
Hawkins, R.B., 251, 253
High purity cistillation column, 263
High-pass filtering, 60
Hjalmarsson, H., 55, 257
Ho, B.L., 8, 207
Hsia, T.C., 116, 218

Identifiability, 52
Identifiable, 51, 161, 166
Identification criterion, 157
Identification test, 2, 31
Ill-conditioned processes, 262
Impulse response, 16
Informative, 31, 161
Input, 1
Input-output data, 2
Instrumental variable method, 110
Internal model control (IMC), 167
Isermann, R., 294, 305

Jacobsen, E.W., 56, 262, 263, 267, 272
Jansson, M., 209
Jenkins, G.M., 120
Johnson, S.H., 218, 223
Jones, H.L., 295

Kailath, T., 17, 201, 333
Kalman, R.E., 8, 207
Klompstra, M., 162
Kosut, R.L., 162
Koung, C.-W., 56, 262
Kucera, V., 21, 300

Lancaster, 219
Laplace transform, 16
Lau, M., 162
Least-squares, 65
Least-squares method, 67
Likelihood function, 123
Linear model, 26
Linear process, 26
Linear regression, 65
Ljung, L., 1, 3, 25, 41, 42, 52, 53, 55, 56, 61, 91, 105, 121–125, 128, 131, 133–135, 141, 153, 157, 158, 162–164, 166, 168, 171, 172, 176, 186, 211, 223, 225, 257, 307
Local minimum, 82
Lovera, M., 216
Luus, R., 218
Luyben, W.L., 218, 223

MacGregor, J.F., 56, 262
Manipulated variable (MV), 9
Manipulated variable (MV), 20, 32
Marino, R., 307, 308
Massoumnia, M.A., 304
Mathematical model, 2
Mathis, G., 323, 327
Matlab, xi, xiv, 43, 47, 49, 99, 112, 132, 142, 206, 207, 216
Matrix fraction description (MFD), 22
Matrix inversion lemma, 152
Maximum likelihood (ML) method, 123
Maximum likelihood estimate, 123
McBride, L.E., 99
Mean square error, 125, 133, 170
Mean value, 24, 41
Measured disturbances, 1
Measurement noise, 17
Mehra, R.K., 167
Minimum variance estimator, 89
Model, 1
Model parametrization, 126, 199, 218, 231, 242
Model predictive control (MPC), 9
Model reduction, 116, 131, 174, 187, 189, 190, 225, 235
Model set, 2, 26, 129
Model validation, 2, 78, 139, 146, 177, 188
Model-based control, 4
Mora, M.D., 307
Moraal, P.E., 307
Morari, M., 162, 258, 264
Moustakides, G., 314
Multiple sinusoids, 50
Narendra, K.S., 28, 218, 223, 227

- Nikiforov, I., 294, 308, 311, 315
 Nikoukhah, R.N., 297–299, 301, 304, 307, 309
 Ninness, B., 141, 162
 Nonlinear model, 26, 51, 217
 Nonlinearity, 8, 26, 39
 Nonparametric, 55
 Nonparametric model, 43
 Normal distribution, 123
 Nyquist frequency, 57
- O'Reilly, J., 297
 Observer canonical form, 17
 Olagnon, M., 305
 On-line, 151
 Open-loop operation, 38
 Open-loop test, 52, 53
 Operating point, 6
 Optimal input design, 53
 Optimal test, 167
 Order selection, 176, 187
 Orthogonal decomposition, 204
 Othman, S., 307, 308
 Output, 1
 Output error, 79
 Output error criterion, 82, 103
 Output error method, 102, 106
- Pajunen, G., 219
 Parameter estimation, 3, 173, 186
 Parameter vector, 74
 Parameters, 66
 Parametrization, 86
 Patton, R.J., 294, 297
 Pereyra, V., 116, 225
 Periodic signal, 28
 Periodogram, 23
 Persistent excitation, 51
 Physical modeling, 6
 Pintelon, R., 14
 Pole, 49, 56
 Pole placement, 258
 Postlethwaite, I., 9, 188
 Power spectrum, 24
- Prediction error, 114, 118
 Prediction error method, 113, 121
 Prevosto, M., 305
 Pseudo-random-binary-sequence (PRBS), 41
- Rchalet, J., 251
 Recursive estimation, 151
 Renneboog, J., 51
 Residual evaluation, 294
 Residual generation, 294
 Richalet, J., 253
 Rivera, D., 258
 Rolling mill, 10
- Salgado, M.E., 141, 162
 Salkauskas, 219
 Schoukens, J., 14, 51
 Settling time, 20, 38, 53, 54, 57
 Signal slicing, 60
 Skogestad, S., 9, 188, 258, 263, 264
 SMI Toolbox, xiv, 216
 Soderstrom, T., xiv, 41, 42, 52, 91, 107, 112, 113, 116, 129, 136, 141, 153, 157, 158, 169, 218, 227, 257
 Spectral density, 24
 Staroswiecki, M., 306
 State feedback, 9
 State space identification, 200
 Stec, P., 262
 Steiglitz, K., 99
 Steiglitz-McBride method, 99, 100, 106
 Stoica, P., xiv, 41, 42, 52, 91, 107, 112, 113, 116, 129, 141, 158, 169, 218, 227
 Subspace identification, 199, 208
 System, 1
 linear, 2
 System identification, 2
 System Identification Toolbox, xiv, 43, 142
- Test design, 31
 Time constant, 7, 8, 37, 38, 40, 57, 139, 223
 Tjarnstrom, F., 131, 176
 Tomei, P., 307, 308
 Trend correction, 60

- Tulleken, H.J.A.F., 41, 222
Unbiased estimator, 89
Uncorrelated, 89, 111
Unmeasured disturbance, 1
Unstable, 106, 134

van den Boom, T.J.J., 162, 267, 268
van den Bosch, P.P.J., 172
van den Hof, P.M.J., 141, 158
van der Ouderaa, E., 51
van Loan, C., 332
Van Overschee, P., 211
Variance, 26, 39, 49
Variance error, 53, 125, 129, 133
Verdult, V., 216, 267, 268
Vergheese, G.C., 304
Verhaegen, M., 209, 211, 213, 214, 216, 231,
 267, 268
Vidyasagar, M., 162

Wahlberg, B., 56, 131, 174, 209
Walcott, B.L., 307
Walton, V.M., 295
Weighted least-squares, 67
Weighting function, 126
Wernstedt, J., 218
Westwick, D., 216
White noise, 25, 39, 40
Wigren, T., 219
Willsky, A.S., 295, 297, 304
Wittenmark, B., xiv, 25, 56–58, 119

Yuan, Z.-D., 166, 168, 171

z-transform, 19
Zadeh, L.A., 2
Zafirou, E., 162, 264
Zak, S.H., 307
Zames, G., 9, 162
Zhang, Q., 306–308, 314
Zhu, Y.C., 28, 53, 54, 131, 136, 141, 162, 172,
 186, 195, 197, 218, 231, 241, 262, 265