THE UNIVERSITY OF ALBERTA

PRACTICAL NUMERICAL ISSUES IN INDUSTRIAL IDENTIFICATION

by

ANDRE VIEN

©

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA

Spring 1994

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-612-11402-3

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR ANDRE VIEN

TITLE OF THESIS PRACTICAL NUMERICAL ISSUES IN INDUSTRIAL IDENTIFICATION

DEGREE FOR WHICH THESIS WAS PRESENTED DOCTOR OF PHILOSOPHY

YEAR THIS DEGREE GRANTED Spring 1994

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) André Vien

PERMANENT ADDRESS:

934 [illegible] St
Kelowna BC

DATED March 29 1994

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled PRACTICAL NUMERICAL ISSUES IN INDUSTRIAL IDENTIFICATION submitted by ANDRE VIEN in partial fulfilment of the requirements for the degree of DOCTOR OF PHILOSOPHY.

....
Dr. R.K. Wood
Supervisor

....
Dr. S.L. Shah

....
Dr. M. Rao

....
Dr. R.E. Rink

....
Dr. D. Hodouin
External Examiner

Date..MARCH...29,.1994...

To Becky

Abstract

[illegible] thesis is concerned with the practical [illegible] in applying process identification. More [illegible] the methods used for the reduction of the [illegible] number of the P matrix (the inverted matrix [illegible] in most identification algorithms) to improve [illegible]al accuracy are investigated. It is demonstrated that, under mild assumptions, the P matrix in the identification of a first order system can be made well conditioned but that this is not always possible for higher order systems.

Elimination of the constant (bias) parameter in the model is shown to reduce the condition number of the P matrix. Methods such as the running mean, the filtered mean and k-incremental are recommended. Methods such as expected mean, sample mean and fixed equation should be avoided. The k-incremental method noise sensitivity is minimized if k is chosen larger than twice the model order. Different filters for the input and output signals, as suggested in the literature, should not be employed.

Matrix scaling is shown to have no effect on numerical accuracy. Coding (data transformation) producing a zero mean and unity variance is recommended. Normalization is shown to reduce signal to noise ratio. Its effect on numerical accuracy cannot always be predicted and in some cases can be shown to have no effect on numerical accuracy. Analysis of a widely used example shows that normalization is not a necessary condition for stable adaptive control. Some alternatives are suggested.

The condition number of the P matrix is shown to be a poor measure of persistency of excitation.

It is demonstrated that the off-diagonal elements of the P matrix have an approximately exponential effect on the condition number. The matrix factorization methods examined provide similar numerical conditioning benefits.

Application of time series analysis is illustrated for a secondary ore crusher. Ignoring the mean of the signals in the model had a deleterious effect on the results. Comparison of the Box-Jenkins model with least squares and maximum likelihood shows that least squares is not adequate for this example but maximum likelihood would be suitable in the context of identification for control purposes.

Acknowledgements

The author wishes to express his sincere gratitude to Dr. R.K. Wood for his support and encouragement during the course of this research. The Brenda Mines Ltd. data provided by Rob Spring and Robert Edwards through the courtesy of Dr. B.C. Flintoff is gratefully acknowledge.

I would like to thank my fellow graduate students in process control; in particular the second floor students and the members of the Friday Pasta Club with whom I had many discussions.

The author wishes to thank the Department of Chemical Engineering and the Natural Sciences and Engineering Research Council of Canada for their financial support.

Finally, I would like to acknowledge the support of my computer widow, Becky, whose encouragements, especially during the long hours spent in front of a computer terminal, are greatly appreciated.

Table of Contents

List of Figures

Figure Page

List of Tables

Table Page

1. General Introduction

Most common process identification algorithms involve the inversion of a matrix. Although too frequently ignored reducing the condition number of the matrix to be inverted is of prime importance in achieving the successful identification of a suitable dynamic model. In this thesis, the effectiveness of the most common methods of reducing the condition number (scaling, coding i.e. data transformation, matrix factorization, normalization and reducing the rank of the matrix by substitution) are examined to provide guidance to a potential user to make a judicious choice of method(s) for reducing the condition number.

The thesis also includes a summary of other practical issues (outliers, model order, quantization, feedback, choice of input signal, choice of sampling interval, length of the data collection and model validation criteria) that must be addressed when performing a process identification excercise. This summary, combined with the derivation of the most common process identification algorithms and the matrix conditioning analysis, makes this thesis a unique document providing the practitioner with guidance and suggestions in regards to the practical issues in process identification.

During the course of this work a process identification program was developed to "experiment" with the various practical aspects under study. This program proved indispensable in confirming the theoretical results.

The thesis ends with an example using industrial plant data. This allows the reader to see how the various aspects of process identification are used collectively.

Since the focus of this thesis is on the condition number of the matrix to be inverted, this work does not address statistical considerations such as the treatment of initial conditions, (statistically) robust techniques, missing data, diagnostics, etc.

It is worth while to note that some confusion may exist between "numerical aspects" and "computational aspects", for example when an iterative algorithm such as the Newton-Raphson, Cochrane-Orcutt or Gauss-Newton is described. These algorithms are sometimes classified as being concerned with numerical aspects (i.e. in the sense of numerical integration) or with computational aspects (e.g. computation of maximum likelihood estimates) e.g. see Gupta and Mehra, 1974 and Corradi, 1979. Numerical accuracy, as interpreted in this work and as used throughout this thesis relates to the accuracy of the calculations in a finite (computer) word length representation.

1.1 The concept of process identification

Before the condition number of the matrix to be inverted can be examined a brief presentation on the general notion of process identification is warranted.

It is often necessary to gain a better understanding of the dynamic behavior of a process if improvement in operation or control is desired. One approach would be to develop a process model from the dynamic state mass, energy and momentum balances of the process. This is termed modelling from first principles. The advantage of modelling from first principles is that it gives insight into the process, also, the different parameters and variables have a physical interpretation. However the time and effort necessary to develop such a model, for other than a rather simple process, is often not economically justifiable and may not be possible at all due to the complexity of the process. Furthermore, many of the parameters in the mechanistic model would often be unknown and would need to be estimated from process and/or laboratory data.

A useful alternative in many cases is to obtain a dynamic model directly from data collected from the process. The objective is then to find values for the parameters of a postulated model such that the process output can be reliably predicted. It is this approach of empirically identifying dynamic characteristics of a system from input/output data that constitutes the field of process identification.

Process identification can be broken down into four steps:

i. collect input/output data

ii. postulate a model structure (e.g. polynomial, exponential, etc.)

iii. estimate model parameters (including polynomial orders if applicable)

iv. validate the model

In practice, the procedure of process identification i. iterative. The data are used to estimate the unknown parameters for an assumed model. Then, based on the adequacy of the model in describing the system dynamics, the model structure is altered and the estimation of the parameters repeated. This procedure is repeated until a satisfactory model is established or it is concluded that additional data must be collected.

When investigating a process where there is little *a priori* knowledge of system dynamic behavior, it is reasonable to start with some limited testing (e.g. step, pulse, frequency, etc.) to obtain some estimate of the system dynamics. The results can then be used to plan further experiments. This includes the design of the input signal which must be sufficiently "rich" to obtain good results in the desired range of frequencies. An overview of the topic of input signal selection is presented in Appendix A.

When a model has been obtained from experimental data, it is necessary to validate the model to ascertain if the model adequately describes the process. For this purpose it is useful to compare the step and impulse responses of the model to those of the process, if available. It is also possible to establish model residuals and model statistics, as described in Appendix B, to assess the suitability of the model representation.

Models can be classified as discrete or continuous, deterministic or stochastic, and for use in an on-line or off-line application.

Discrete models are generally used since most experimental input/output data are logged by computer and/or measured values of the input and output signals are only available periodically (e.g. gas chromatograph).

If the model does not include modelling of the process noise or if the noise is neglected the model is classified as a deterministic model (Goodwin and Sin, 1984). Obviously deterministic models are not suitable for processes that exhibit low signal to noise ratios since the system response to a change in input signal cannot easily be distinguished from the system response to noise. For such processes stochastic models must be employed.

For off-line identification, input/output data are collected from the system and this data series (collection of data pairs) is subsequently used with a specified identification algorithm to obtain the parameters of the model.

In many processes the system characteristics may be time variant. In this case it is often necessary to modify the model parameters thus providing an adaptive control scheme. This periodic updating of the process model can be done by repeating the off-line identification or the identification can be performed on-line. On-line identification is also termed recursive identification. This involves updating the parameters after each data point rather than accumulating a data set for subsequent calculation of the model parameters.

However, recursive identification algorithms can also be used on a series of data as an alternative to the more conventional off-line technique. The advantage of recursive methods over off-line procedures is that only the relatively recently acquired data are utilized for updating the model parameters so only a small amount of computer memory is necessary. On the other hand, when the recursive identification approach is used on time invariant processes the accuracy of the model obtained is usually not as good as that from an off-line calculation unless several passes through the collected data are used. If this technique is employed the memory advantage of the recursive method is lost.

Due to the rapid industrial growth of model-based process control, there is a need for an increase in ease and reliability of model identification. The practical aspects of process identification are investigated using a computer program for process identification and time series analysis (PITSA) developed as part of this study.

1.2 Thesis overview

Chapter 2 is intended for the reader that is not familiar with process identification or the reader that is interested in a "layman's approach" to justifying the sum of squares as a cost function. A facile argument on the choice of the least squares cost function is presented along with the model structures considered in this thesis.

A brief overview of the pertinent literature is presented in Chapter 3. It covers mostly the period up to 1989 but some references to more recent literature are included. The overview is mostly concerned with numerical accuracy and does not cover statistical computing or algorithmic implementation aspects of process identification.

The most common process identification algorithms are derived in Chapters 4 and 5. These derivations are presented to demonstrate that most common process identification algorithms involve the inversion of a matrix (or its recursive equivalent) in the calculation of the parameter estimates. This emphasizes the importance of the results obtained in the remaining chapters. Chapters 4 and 5 thus provide a consolidated reference to a large collection of algorithms and their derivation.

Chapter 6 briefly outlines some of the practical aspects of process identification. Numerous references are provided. Additional and more detailed material on the pratical aspects in process identification are presented in

Appendices A, B and C to supplement this chapter.

Using the standard approach to the numerical analysis of the solution of a set of equations (which is well known in the literature e.g. Lawson and Hanson, 1974, Forsythe *et al.*, 1977) it can be shown that the numerical accuracy of an identification algorithm is inversely proportional to the condition number of the matrix to invert.

However, this type of analysis does not consider the effect of the magnitude of specific elements of the matrix on its condition number. The effect of the magnitude of individual elements on the condition number is analyzed in Chapter 7.

On the basis of the results in Chapter 7, the implementation issues which involve removing the constant (bias) parameter, scaling and coding are examined in Chapters 8, 9 and 10 respectively. The effect of normalization on least squares identification is examined in Chapter 11 and alternatives to normalization are proposed, based on an example, to maintain stability of the adaptive control scheme.

A typical process identification study using experimental data obtained from a secondary crusher in a mining operation is presented in Chapter 12. It is shown that not accounting for the mean of the input and output data (accounting for the mean can be accomplishe removing the mean from the data or by including a constant, or bias, parameter in the model) can have a serious impact on the reliability of the identified model. Comparison of the Box-Jenkins algorithm with least squares and maximum likelihood algorithms is provided to illustrate the relative performance of the most likely alternatives to the Box-Jenkins algorithm for this data set.

The conclusions from this study of the practical issues in process identification and recommendations for future work are presented in Chapter 13.

2. Modelling Stochastic Processes

2.1 Introduction

In most identification techniques the underlying approach involves the minimization of the sum of squares of the modelling error. Because so many algorithms tend to minimize the sum of squares of errors, known as least squares, insight, from a layman's perspective, into why the least squares technique is so widely used is warranted. The statistical interpretation of the least squares is not examined as this can be found in several textbooks e.g. Draper and Smith, 1981. The reader interested in the geometrical interpretation of the least squares identification technique should consult the book of Norton, 1986.

Following a brief discussion of the least squares identification technique, the models used throughout this thesis are presented with a brief heuristic explanation as to why they were chosen.

2.2 A facile argument for the choice of the least squares family of cost functions

Most identification schemes are based on the minimization of the square of errors (also known as residuals or model prediction errors) between a measured output signal and a predicted output. This minimization of the squares of errors is termed "least squares". The mathematical formulation of the least squares minimization differs for each algorithm depending primarily on the form of the cost function (there are several possible variations for the sum of squares) and/or the manner in which the modelling error is calculated. Even the so-called maximum likelihood parameter identification technique involves the minimization of the sum of squares of errors (by making the proper assumptions) and not the maximization of the likelihood function as implied by the name.

This section provides an explanation as to why the least squares family of algorithms (which includes such algorithms as ordinary least squares, extended least squares, generalized least squares, instrumental variables and the most common form of the maximum likelihood algorithm) are so popular. First, an intuitive approach is presented to provide a simple explanation of why the minimization of the squares of errors is preferred to other minimization criteria. This is followed by an analytical presentation of the same arguments.

2.2.1 Choice of the cost function

When formulating an identification problem, a criterion is introduced as a measure of how well a model fits the experimental data. This criterion is also known as the cost function in the optimization literature.

Before selecting a cost function it is necessary to establish the desired properties of the minimization criterion. The first and most important property is that the modelling error should be minimized. In general terms this means that the difference between the predicted model output and the actual output should be as small as possible.

The second desirable property is that the sum of errors be zero. This is desirable because the modelling error is assumed to have a zero mean. Another advantage of having a zero mean error is that the mean of the model input and output will be the same as the mean of the actual data.

A third desirable property is that the modelling error reduces with increasing time of operation in the case of recursive algorithms. This feature is meaningful only if the process parameters are not time varying (otherwise there is no basis to expect that the error should reduce as the length of time for identification is increased).

A fourth property is desirable if identification is to be used in an adaptive control strategy. That is, the calculations should be as simple and as easy as possible in order to reduce computation time with minimal loss of accuracy.

Therefore the desired properties are minimization of the modelling error, sum of errors tending to zero and simple, easy calculation of the cost function.

Obviously the sum of modelling errors itself cannot be used as a cost function since it has no minimum (it is minus infinity). The two cost functions that are investigated here are the sum of absolute errors and the sum of squares of errors. Other functions will not be investigated because they will be more complex than these two (and therefore not as easily computed) or usual simplifying assumptions reduce them to the sum of squares of errors. This is the case for the Bayesian criterion as well as for the maximum likelihood criterion for which the assumption of a normal distribution of the modelling error reduces the problem to that of a least squares parameter estimation. For a discussion on the equivalence of different criteria see Ljung and Soderstrom, 1983, Chapter 2.

2.2.2 Rational for the choice of the cost function

A very simple example, presented in Figure 2.1, will be used to provide an insight into the rational for the selection of the sum of squares of errors (SSE) over the sum of absolute errors (SAE) as the preferred identification cost function.

Since the time scale is irrelevant for this example, the data can be thought of two groups of three points each that were sampled with a very large time interval between them. For instance, the ordinate could represent the change in pressure in a vessel for three consecutive days and the change in pressure for three days taken a month later. Assuming a linear relationship and knowing that the straight line must pass through the two groups at the same ordinate value (by inspection any other line cannot be minimum) it then follows that the fitted line will be horizontal in this particular case.

Attention can then be focused on one set of three points if we keep only the vertical displacement of the straight line as a variable and force the line to be horizontal.

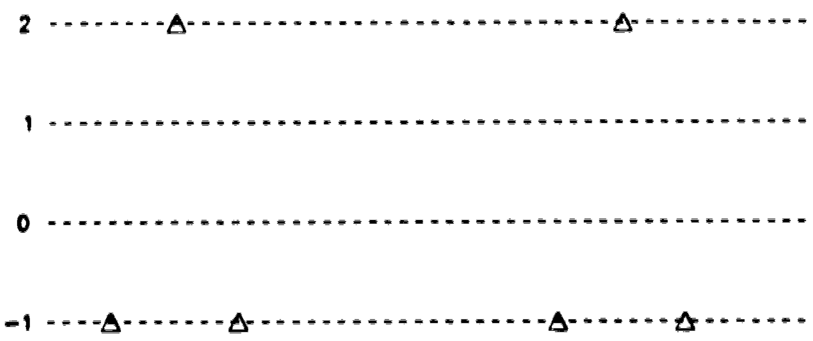


Fig. 2.1 Simple example to compare SAE and SSE

Focusing on only one set of three points, one can see by inspection that the minimum for both cost functions will lie between the extreme points 2 and -1 as any horizontal line outside of these limits would have an associated cost function that is larger than any solution inside the limits. Table 2.1 presents the value of the cost functions for certain vertical levels. As mentioned earlier, these levels could represent the change in pressure in a vessel.

From Table 2.1 it can be observed that the sum of absolute errors is not an acceptable cost function since the sum of errors is not zero at its minimum. For the sum of squares of errors the sum of errors is zero at the minimum and the sum of squares of errors is simple to compute. Consequently it should be the cost function to use for

TABLE 2.1

Cost function values for the simple example

level	COST FUNCTION		
	SAE	SSE	sum of errors
-1	3	9	3
0	4	6	0
1	5	9	-3
2	6	18	-6

process identification.

A more dramatic illustration of the shortcomings of the the sum of absolute errors as a cost function is if a fourth point is added at level 2 (or a point at level -1 is removed). As can be seen from the cost function values in Table 2.2, there is no minimum for the sum of absolute errors.

TABLE 2.2

Cost function values for an additional point at level 2

level	COST FUNCTION		
	SAE	SSE	sum of errors
-1	6	18	6
0	6	10	2
.5	6	9	0
1	6	10	-2
2	6	18	-6

The sum of squares of errors minimum value occurs at the same level as when the sum of errors is zero and therefore is still acceptable for this case.

2.2.3 Analytical approach to the selection of the cost function

Let the model be

$$y_t = \beta + f(x) + \epsilon_t \tag{2.1}$$

where β is a constant term, $f(x)$ is a function of an independent variable x and ϵ_t is the error at time t. The cost function for the sum of absolute errors, for N points, is

$$J = \sum_{i=1}^{N} |\epsilon_i|$$

$$= \sum_{i=1}^{N} |y_i - \beta - f(x)|$$

The derivative of this function does not exist at its minimum ($\min\{\Sigma|y_i - \beta - f(x)|\} = 0$ and the derivative of the absolute value is undefined at the minimum i.e. at zero). So analytically this function is not well suited for a minimization problem and therefore should not be used.

With the sum of squares of errors as the cost function the following can be obtained

$$J = \sum_{i=1}^{N} \epsilon_i^2$$

$$= \sum_{i=1}^{N} (y_i - \beta - f(x))^2$$

taking the derivative with respect to β and setting it to zero

$$\frac{d\ J}{d\ \beta} = -2\sum_{i=1}^{N} (y_i - \beta - f(x)) = 0$$

$$\sum_{i=1}^{N} (y_i - \beta - f(x)) = 0$$

$$\sum_{i=1}^{N} \beta = \sum_{i=1}^{N} (y_i - f(x)) \qquad (2.2)$$

now the sum of errors is

$$\sum_{i=1}^{N} \epsilon_i = \sum_{i=1}^{N} (y_i - \beta - f(x))$$

$$= \sum_{i=1}^{N} (y_i - f(x)) - \sum_{i=1}^{n} \beta$$

so replacing $\sum_{i=1}^{N} \beta$ by its value at the minimum (Equation 2.2) yields

$$\sum_{i=1}^{N} \epsilon_i = \sum_{i=1}^{N} (y_i - f(x)) - \sum_{i=1}^{N} (y_i - f(x))$$

$$= 0$$

therefore we can conclude that if a parameter β is included in the model, the property that the sum of errors is zero is guaranteed regardless of any function of the independent variable.

This conclusion is very interesting from a process identification point of view as it indicates that the mean of the process data must always be considered either explicitly through β or implicitly by ensuring that the mean of $f(x)$ and y_t are zero.

This can be shown by defining the model prediction as

$$y^m_t = \beta + f(x)$$

then

$$y_t = y^m_t + \epsilon_t$$

$$\frac{1}{N}\sum_{i=1}^{N} y_i = \frac{1}{N}\sum_{i=1}^{N} y^m_i + \frac{1}{N}\sum_{i=1}^{N} \epsilon_i$$

also

$$\sum_{i=1}^{N} \epsilon_i = 0$$

and defining the mean as

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

then

$$\bar{y} = \bar{y}^m$$

This indicates that the mean of the data points $\bar{y}$ is also the mean of the model $\bar{y}^m$ when the sum of squares of errors (giving $\sum_{i=1}^{N} \epsilon_i = 0$) is used as the cost function and with a parameter β in the model. This parameter, β, will be referred to as the constant (or bias) parameter later on. Note that if $\bar{y}$ is zero and the mean of $f(x)$ is zero then β is zero. In general $f(x)$ is chosen to be a linear function of x such that if the mean of x is zero then the mean of $f(x)$ is also zero. In the process control and identification literature the mean of the input and output are frequently assumed to be zero in order to remove the parameter β. If the mean of the input and/or the output is not zero then the parameter β must be included in the model.

In conclusion, the cost function should be the sum of squares of errors (with the restriction that a constant parameter is included in the model form or that the mean of the input and output data are zero) because the sum of errors will be zero and it is simple to calculate. This statement may seem too simplistic for theorists but it is nice to know that "down to earth" arguments can be used to justify least squares.

2.2.4 Modification of the cost function for recursive identification

Goodwin and Sin, 1984 have introduced the following cost function to justify the initial condition they use in a recursive algorithm

$$J = \frac{1}{2} \sum_{i=1}^{N} (y_i - \phi^T\theta)^2 + \frac{1}{2}(\theta - \theta_0)P_0^{-1}(\theta - \theta_0)$$

where ϕ^T is a vector containing past output and input values, θ is a vector containing all the parameters, θ_0 is the vector of initial parameter estimates and P_0 is the initial estimate of a matrix proportional to the covariance matrix. The first term on the right hand side of the equation represents the sum of squares of errors and is the least squares cost function (the constant 1/2 does not change the location of the minimum). The second term accounts for the initial conditions (this term becomes negligible as the number of data points increases and it can be ignored when asymptotic properties are analyzed). Unfortunately using this cost function will not give the least squares estimates for a finite number of data points, although the deviation from the true least squares estimates will tend to zero as the number of data points goes to infinity. Heuristically one can see that the least squares estimates should be independent of the initial conditions used for recursive identification and therefore the second term will "pull" the estimates away from the true least squares estimates.

This can also be shown by the following. The cost function can be written in vector form as

$$J = \frac{1}{2}(Y - \Phi\theta)^T(Y - \Phi\theta) + \frac{1}{2}(\theta - \theta_0)^T P_0^{-1}(\theta - \theta_0)$$

where Y is a vector of past outputs and Φ is a matrix containing past ϕ's. Minimizing with respect to θ gives

$$-\Phi^T Y + \Phi^T\Phi\hat{\theta} + P_0^{-1}\hat{\theta} - P_0^{-1}\theta_0 = 0$$

$$(\Phi^T\Phi + P_0^{-1})\hat{\theta} = P_0^{-1}\theta_0 + \Phi^T Y$$

where $\hat{\theta}$ denotes the estimates of θ at the minimum. This result was also obtained by Goodwin and Sin. For the rigorous least squares identification algorithm similar calculations lead to

$$\Phi^T\Phi\theta_{LS} = \Phi^T Y$$

Then $\hat{\theta}$ can be calculated as

$$\begin{aligned}\hat{\theta} &= (\Phi^T\Phi + P_0^{-1})^{-1}(P_0^{-1}\theta_0 + \Phi^T\Phi\theta_{LS})\\ &= (\Phi^T\Phi + P_0^{-1})^{-1}[(\Phi^T\Phi + P_0^{-1})\theta_{LS} + P_0^{-1}(\theta_0 - \theta_{LS})]\end{aligned}$$

$$\hat{\theta} = \theta_{LS} + (\Phi^T\Phi + P_0^{-1})^{-1}\ P_0^{-1}(\theta_0 - \theta_{LS})$$

so it follows that if $P_0^{-1} = 0$ or $\theta_0 = \theta_{LS}$, the second term of the cost function will vanish reducing the proposed cost function to that of the least squares. Since $P_0^{-1} = 0$ cannot be used in the recursive calculations, it is never set to zero. Since $\theta_0 = \theta_{LS}$ implies that the least squares parameters are known, then for any practical purposes $\hat{\theta}$ will not be the same as θ_{LS}. The difference between $\hat{\theta}$ and θ_{LS} (the second term of $\hat{\theta}$) will vanish in time only as $\Psi^T\Psi$ grows to be a matrix with very large elements, much larger than those of P_0^{-1}. Since most applications deal with a finite number of data, $\Psi^T\Psi$ may not grow large enough, thus the cost function proposed by Goodwin and Sin is not totally acceptable. In other words, guessing θ_0 and P_0 is not as good as using the least squares estimates for initialization. This is especially true for short data sequences.

The difference between $\hat{\theta}$ and θ_{LS} could also be reduced by choosing a small P_0^{-1}(large P_0). This choice for P_0 also implies little confidence in initial parameters i.e. a large P_0 will produce large variations of the parameter estimates (mainly in the initial stages). This may be undesirable in some cases. The choice of P_0 should be guided by the fact that on one hand P_0 should be large so that the deviation from true least squares estimates is minimized. On the other hand P_0 should not be too large so that parameters will not exhibit an excessively wide variation during their estimation. The effect of the initial value of the parameter estimates on the estimated model parameters will be stressed again in Chapter 5 where the recursive algorithms are derived.

2.3 Choice of model structure

The model structure is the specific form of $f(x)$ (in Equation 2.1) that is used. In general the model structure is taken to be linear in the parameters. This means that the output variable may be a nonlinear function of the input variable. For example

$$y = a_1 x + a_2 x^2 + a_3 x^3$$

is nonlinear in x but is linear in the a parameters. Some nonlinear functions can be linearized by transformations. For example

$$y = A e^{K x}$$

can be linearized as

$$\ln(y) = \ln(A) + K x$$

This can be written in terms of transformed variables as

$$y' = a_1 + a_2 x'$$

Although in the sequel only models that are linear in both the parameters and variables are used, it must be understood that the technique also applies to nonlinear models that are linear in parameters and to nonlinear models that can be linearized by applying a suitable variable transformation.

In the field of process control the general model structure used is one where the output is a function of past outputs, past inputs and past and present noise. These models are often expressed in a form that uses the backshift operator which allows the model to be written in polynomial form (see Appendix D.3 for a description of the backshift operator). Model structure determination not only deals with the choice of how many polynomials are to be used but also the order of the polynomials for a given system.

First consider the number of polynomials to be used. One obvious model is one that relates the output to past inputs, past and present noise, and a constant term. Present input is not included because the process would have to be instantaneous in order for the present input to affect the output (it can also be regarded as an implied delay of one sample). This model is written as

$$y_t = B\ u_t + C\ \epsilon_t + \beta$$

where

$$B = b_1q^{-1} + b_2q^{-2} + \ldots + b_mq^{-m}$$

$$C = 1 + c_1q^{-1} + c_2q^{-2} + \ldots + c_{mn}q^{-mn}$$

$$\beta = \text{a constant}$$

$$q^{-1}u_t = u_{t-1}$$

In this model the coefficients of the B polynomial are known as the finite impulse response weights (FIR) or Markov parameters. The random noise, ϵ_t, is generally assumed to be "white noise". That is, ϵ_t is uncorrelated to past noise.

Many processes have delays, meaning that the first parameters in B are zero. For example if there is a delay of two samples (beyond the implied delay), the first input influencing y_t will be u_{t-3}. Therefore b_1 and b_2 will be zero. Since keeping parameters in the model that are zero is unnecessary and is a waste of calculations, the model is modified to

$$y_t = B\ u_{t-k} + C\ \epsilon_t + \beta \tag{2.3}$$

where k is the delay. Note that k does not include the first sample that affects the output (i.e. in the example above k=2 and not 3, the third sample affects the output therefore k=3-1=2). Another way of viewing Equation 2.3 is simply that k denotes the number of parameters set to zero to account for the delay.

This model can be enhanced even further. In certain cases, a large number of B and/or C parameters will be required to provide an acceptable fit to the input/output data. A large number of parameters is undesirable because this requires more computation time and some numerical accuracy will likely be lost. It also means that many past values of u_t and ϵ_t are necessary to perform the identification.

One way of avoiding a large number of parameters is to employ a ratio of polynomials. By proper choice of coefficients, a ratio of two polynomials can almost match any polynomial using fewer parameters. The ratio can even replace an infinite polynomial. This is due to the fact that the result of the long division of two polynomials can be an infinite polynomial. Taking this into consideration the model can be written as

$$y_t = \frac{B}{A}\ u_{t-k} + \frac{C}{D}\ \epsilon_t + \beta \tag{2.4}$$

where

$$A = 1 + a_1q^{-1} + a_2q^{-2} + \ldots + a_nq^{-n}$$

$$D = 1 + d_1q^{-1} + d_2q^{-2} + \ldots + d_{nn}q^{-nn}$$

This model is the first model of practical use. It provides an economy of parameters while being applicable to a large number of process identification problems. This model is the most general model used in the thesis. It will be referred to as the Box-Jenkins model (Box and Jenkins, 1976).

Models with both the B and A polynomials on the right hand side of the equation are known as output error models since the noise term represents the misfit between the actual output and modelled output based on inputs and past modelled outputs.

Although this is a general model with an economic number of parameters it has one major drawback: it is nonlinear in the parameters. This means that a nonlinear least squares estimation must be performed to find the parameters. This will increase computing time and convergence of the parameters is no longer assured (if the initial parameters are too far from the optimal values). Nevertheless this general form of model will be retained because of its flexibility but a simplified version of this model will be used whenever possible.

The first simplification to the model is to assume that the A and D polynomials are equal. This model which represents a compromise between nonlinearity and economy of parameters can be written as

$$A\ y_t = B\ u_{t-k} + C\ \epsilon_t + \beta \tag{2.5}$$

This model can be considered nonlinear if both sides are divided by C so that the model has a "white noise" error term ϵ_t instead of the "colored noise" $C\epsilon_t$. This is the approach used in the maximum likelihood algorithm. The model can also be considered linear if estimates of the noise ϵ_t are used in the model. This is the approach employed by the extended least squares algorithm.

The model expressed by Equation 2.5 can be considered as a model based on past outputs as well as past inputs and present and past noise. Models based on past outputs rather than past modelled outputs are known as equation error models since the noise represents the "closure" term of the equation.

If the C parameters are known, the model can be further simplified to

$$A\ y_t^F = B\ u_{t-k}^F + \epsilon_t + \beta^F \tag{2.6}$$

where

$$y_t^F = y_t/C$$
$$u_{t-k}^F = u_{t-k}/C$$
$$\beta^F = \beta/C$$

This approach is that of the generalized least squares technique where the residuals are modeled as if they were a process by themselves and the C parameters thus identified are used to filter the input/output data.

A further assumption that C=1 (i.e. all the C parameters are zero) leads to the simple (but effective) model used for the least squares and weighted least squares identification techniques. This model is

$$A\ y_t = B\ u_{t-k} + \epsilon_t + \beta \qquad (2.7)$$

3. Literature Overview

3.1 Introduction

The material included in this chapter is not intended to be an exhaustive review of the identification literature but rather to provide the reader with some indication of the pertinent publications in the field. Since the emphasis of the thesis is on numerical accuracy, the literature associated with statistical computing is not included since virtually all the contributions tend to relate to detailed algorithmic implementations for the computer and do not deal with the numerical aspects. The numerical methods journals such as Numerische Mathematik and the SIAM Journal on Numerical Analysis cover a wide range of subjects such as solving differential equations and finite element methods. Articles from these journals that pertain to numerical accuracy of a least squares problem use the perturbation analysis approach. No references to these articles have been included as the perturbation analysis is fully summarized in LINPACK (Dongarra *et al.*, 1979).

Selected books, thesis, surveys and comparative articles are noted to illustrate the breadth of the field of process identification and time series analysis. Computer programs that were encountered in the published literature are mentioned to give the reader an idea of the state of programming in the area of process identification. Finally, other references of general interest in the field of process identification are indicated.

3.2 Books and theses

The most often cited book in the process identification literature is without a doubt that of Box and Jenkins, 1976. This book is cited throughout this thesis but more particularly in reference to the Box-Jenkins model (Chapter 4), correlation analysis (Appendix C), and model validation (Appendix B). Many studies are based on the approach proposed by Box and Jenkins, 1976 (e.g. MacGregor, 1984) or use the well known "gas furnace data" (data provided in Appendix J of Box and Jenkins, 1976) with other process identification algorithms (Young, 1984). The book of Nelson, 1973 covers Parts I and II of Box and Jenkins, 1976 in a simplified manner and therefore is easier to understand for a person being exposed to time series analysis for the first time. There are several other books dealing with time series analysis such as Cryer, 1986, Pandit and Wu, 1983 and Priestley, 1981 just to name a few.

The frequently cited book of Ljung and Soderström, 1983 presents a general framework for the analysis of recursive identification algorithms. The approach used in this thesis is the derivation of the recursive algorithms from their nonrecursive counterparts. The advantage of this approach is that the somewhat complex analysis of convergence is not required and insight gained with nonrecursive algorithms can also be applied to recursive algorithms.

The general concepts of recursive identification are covered in the book of Goodwin and Sin, 1984 but the orientation of this book is more towards identification for adaptive control applications. The special topic of forgetting factors is considered by Rogers, 1989.

A good overview of process identification is provided in the book of Eykhoff, 1981 which presents reviews of several process identification techniques. A more recent book on the topic of system identification is that of Soderstrom and Stoica, 1989. Several algorithms are also described by Mozel, 1980. Some special modifications to the least squares algorithm are presented by Hsia, 1977. A recent thesis shows how model structure estimation can be combined with parameter estimation (Niu, 1994). Another thesis addresses the issue of requirements of identification in the context of long range predictive control (Shook, 1991).

A book of more general interest is that of Isermann, 1980b which covers topics such as closed loop identification as well as identification algorithms in the context of digital control. Draper and Smith, 1981 , a good reference for regression analysis, also presents many useful mathematical and statistical concepts.

Numerical accuracy aspects are covered by several books such as Householder, 1964, Forsythe *et al.*, 1977 and Bierman, 1977. Computational aspects are addressed by Knuth, 1973 and Thisted, 1986. General background information on vector and matrices may be found in Dennis and Schnabel, 1983.

3.3 Surveys and comparative articles

Surveys providing a general overview of the field of process identification are presented by Strejc, 1981b and Gustavsson, 1975. A survey of several variants of the recursive least squares algorithms, to maintain alertness of the identification algorithm for use in an adaptive control context, has been presented by Shah and Cluett, 1991. Another paper addressing the problem of identification in the context of adaptive control is that of Malik *et al.*, 1991. Several factorization techniques are described by Sima, 1984 while Soderstrom and Stoica, 1981 present a comparison of some instrumental variable algorithms. Analysis of the convergence of seven recursive parameter estimation methods has been studied by Matko and Schumann, 1982. A conceptual overview has been presented by Ljung (Ljung, 1991 and Ljung, 1993).

A wide variety of studies, based on simulation and experimental data, are summarized in Table 3.1. The following algorithm abbreviations,

LS	Least Squares
STA	Stochastic Approximation
IV	Instrumental Variable
GLS	Generalized Least Squares
ML	Maximum Likelihood

ELS Extended Least Squares

EKF Extended Kalman Filter

UDU U-D factorization of the least squares algorithm

COR Correlation Method

as well as algorithm abbreviations starting with R indicating that the implemented algorithm is in its recursive form are utilized in the table.

TABLE 3.1 Summary of selected review, experimental and simulation identification literature

Reference	Algorithms	Simulation	Experimental	Comments
Astrom and Eykhoff, 1971	-LS/RLS -GLS -ML -IV -Levin's method -tally principle -STA -EKF	$y_t - 1.5y_{t-1} + .7y_{t-2} = u_{t-1} + .5u_{t-2} + \lambda e_t$		λ values of 0, .1, .5, 1 and 5 are used. Simulations performed only for LS. Simulations conducted for model orders from 1 to 7 with a t-test used to verify the significance of the change of model order. Mostly descriptive.
Gentil et al., 1973	-ML -GLS -IV -COR (with parameter from continuous)		paper machine	Results from IV with delayed inputs as instruments were found to not be as satisfactory as other methods. Impulse response from COR used to find continuous model parameters. A PRBS input signal was utilized.
Graupe et al., 1980	-LS -covariance -COR -RLS -RCOR	$y_t + 2.042y_{t-1} + 1.3542y_{t-2} + .292y_{t-3} = e_t$ $y_t + 1.9y_{t-1} + 1.24y_{t-2} + .324y_{t-3} + .029y_{t-4} = e_t$		Models with only A parameters are considered. Square root filtering. Examined effect of finite word length and computation time. For COR both partial autocorrelation and autocorrelation used.
Gustavsson, 1972	-LS -tally principle -ML -COR -spectral analysis	$y_t - 1.5y_{t-1} + .7y_{t-2} = u_{t-1} + .5u_{t-2} + \lambda(e_t - e_{t-1} + .2e_{t-2})$	nuclear reactor distillation column paper machine superheater (power generator)	Tests of model order also presented. Tally principle algorithm found to be unstable in some instances. Noise λ is .1, 1 , 10.. A PRBS input signal used except for a pulse signal used for the paper machine and the superheater.

TABLE 3.1 Summary of selected review, experimental and simulation identification literature (cont'd)

Reference	Algorithms	Simulation	Experimental	Comments
Isermann, 1980a	-LS/RLS -RELS -GLS -IV/RIV -ML/RML -COR-LS/RCOR-LS			Description of the properties of the algorithms but algorithms not presented. COR-LS algorithm calculates LS parameters from correlation functions *instead of input/output data.*
Isermann et al., 1974	-LS/RLS -GLS/RGLS -IV/RIV -STA -COR-LS -Fourier and 3 parameter model	$y_t = \frac{1+.5q^{-1}}{1-1.5q^{-1}+.7q^{-2}} u_{t-1} +$ $\frac{.0118}{1-1.027q^{-1}+.264q^{-2}} e_{t-1}$ $y_t = \frac{-.102+.173q^{-1}}{1-1.626q^{-1}+.696q^{-2}} u_{t-1} +$ $\frac{.0116}{1-1.027q^{-1}+.264q^{-2}} e_{t-1}$ $y_t = \frac{.065+.048q^{-1}-.008q^{-2}}{1-1.5q^{-1}+.705q^{-2}-.1q^{-3}} u_{t-2}$ $+ \frac{.0117}{1-.527q^{-1}+.0695q^{-2}} e_{t-1}$		Comprehensive study. Investigated performance of different algorithms. Noise to signal ratios of .1 and .2 employed. Fourier series used to estimate continuous model parameters.
James et al., 1974	-LS -suboptimal LS -2 types of STA -IV -2 types of hill climbing -GLS	$y_t - 1.5y_{t-1} + .7y_{t-2} = u_{t-1} +$ $.5u_{t-2} + e_t + 1.5e_{t-1} -$ $.7e_{t-2}$		A total of 200 data points used for noise to signal ratios of .115, .23, .46. A Gaussian input signal was utilized.

TABLE 3.1 Summary of selected review, experimental and simulation identification literature (cont'd)

Reference	Algorithms	Simulation	Experimental	Comments
Kurz et al., 1980	-RLS -RML	Simulation by analog computer linked to a digital computer using a special noise generator	air heater	Examined the identifiability condition. Introduced a forgetting factor that approaches unity exponentially. Used DC values. Did not use RIV because biased estimates result for closed loop identification.
Saridis, 1974	-COR/RCOR -2 types of STA -ML/RML -max. a posteriori filter -EKF	$x(k+1) = A\,x(k) + d[\phi(k) + u(k)]$ $y(k) = h^T x(k) + v(k)$		ML update performed every M steps (M chosen by user). Maximum a posteriori filter approach used linear term of series expansion whereas EKF also uses first order term of series expansion. No input used except for COR which had a discrete interval binary noise.
Sinha and Sen, 1975	-RLS -GLS/RGLS -IV -STA -COR-LS	$y_t = \frac{0.079q^{-1} + 0.047q^{-2}}{1+0.975q^{-1} - 0.0223q^{-2}} u_t +$ $\frac{0.7e}{1-0.8q^{-1}} e_t$		The input signal was Gaussian with u_t and e_t selected to be N(0,1). Model parameters was adjusted for noise to signal ratios of 20%, 40% and 100%. The input signal was Gaussian.
Soderstrom et al., 1978	-RLS -RIV -RGLS -RELS -RML	$y_t - .8y_{t-1} = u_{t-1} + e_t + .7e_{t-1}$ $y_t - .8y_{t-1} = u_{t-1} + \frac{1}{1+.7q^{-1}} e_t$ $y_t - 1.5y_{t-1} + .7y_{t-2} = u_{t-1} + .5u_{t-2} + e_t - e_{t-1} + .2e_{t-2}$		Noted that to obtain unbiased estimates that RIV must be employed for open loop identification. Other algorithms may also be used under closed loop conditions. RGLS algorithm used RLS for 50 samples before starting filtering. Signal to noise ratios of 1 and 10 employed using a PRBS input signal.

TABLE 3.1 Summary of selected review, experimental and simulation identification literature (cont'd)

Reference	Algorithms	Simulation	Experimental	Comments
Strejc, 1981a	-LS/RLS -IV/RIV -GLS/RGLS -ELS/RELS	$y_t + .1y_{t-1} - .72y_{t-2} = u_t - 2.3u_{t-1} + .6u_{t-2} + e_t$ $y_t + .1y_{t-1} - .72y_{t-2} = u_t - 2.3u_{t-1} + .6u_{t-2} + e_t + 1.3e_{t-1} + .46e_{t-2}$		The IV algorithm was shown to provide biased estimates in closed loop identification. Square root filtering implemented. Used the IV algorithm with delayed inputs as instruments. Three variations of ELS presented. A Gaussian input was utilized with Gaussian, uniform and Laplace noise to produce signal to noise ratio of unity.
Wong et al., 1983	RLS	$y_t + 1.2191y_{t-1} + .3679y_{t-2} = .3611u_{t-1} + .2587u_{t-2}$		Factorization methods compared in study of square root , Householder and UDU factorization. Simulation performed in closed loop.

3.4 Identification programs

There is a wide variety of identification programs, some of which are now commercially available, that have originated at a university with research groups developing process identification programs tailored to meet their needs (i.e. with a preference for particular identification algorithms). It is impossible to enumerate all of these programs since many have not been cited in the published literature or, like the one offered by McMaster University, are available through advanced process control continuation courses as part of the lecture material. Only those programs that were encountered in the published literature, usually with limited information and/or documentation available, are noted. The libraries of subroutines such as IMSL and NAG are not included since they are a collection of subroutines and are not complete program packages. Several other commercial packages that only provide time series analysis or regression analysis were not included (only the program packages that offered a selection of identification algorithms are included). For the reader interested in those programs Chemical Engineering Progress publishes a comprehensive software directory (Simpson, 1993).

In the following, the name of the program is followed by the symbol U or C indicating that the program is available from a university [U] or is commercially available

[C].

3.4.1 Control system design programs which include identification

In his book, Chalam, 1987 provides a detailed listing of some adaptive control system software packages that have been developed by researchers around the world. Only the aspects of these packages relating to process identification are discussed here. For the detailed references on these packages the reader should consult this book. The programs are

1. ADPAC (ADaptive control PACkage) [U] is a package for the interactive design of SISO adaptive control systems. The process identification algorithms included in ADPAC are the recursive least squares and extended least squares algorithms for ARMAX models. Both algorithms employ Givens orthogonal factorization (square root factorization).
2. KEDDC [U] is a versatile package which makes possible the analysis, identification and design of a variety of adaptive control systems including SISO and MIMO systems. Process identification routines encompass both deterministic and stochastic methods. The recursive identification algorithms implemented are least squares, extended least squares, instrumental variables and maximum likelihood.
3. CYPROS [C] is a package that covers a wide range of

problems: system identification, parameter estimation, simulation, analysis and design. The model representation forms include state-space and transfer function matrices in both continuous and discrete form. Maximum likelihood and the extended Kalman filter are the only two identification techniques available in the CYPROS package.

4. MATRIX$_x$ [C] is a package for control system design and analysis, system identification, data analysis, and simulation. System identification tools include nonparametric frequency-domain techniques, maximum likelihood algorithms (batch and recursive forms) and adaptive algorithms.
5. DESIGN MASTER [U] is composed of five programs: SIMNON, IDPAC, MODPAC, SYNPAC, and POLPAC. The program of interest for process identification is IDPAC which allows data analysis, spectral analysis, correlation analysis, parameter estimation in linear models with MIMO systems, and model validation.
6. ISER-CSD (Interactive System for Education and Research in Control System Design) [C] is a package intended for the simulation of dynamic systems in time or frequency domains, SISO and MIMO systems, continuous-time or discrete-time models, and models in transfer function form or in state-space form. The program also has four methods for the identification of MISO systems.
7. DPASCS-F [U] is a software package for the analysis of

multivariable control systems based on state-space and frequency design methods. This package includes programs for identification of multivariable systems based on the maximum likelihood and generalized least squares techniques.

8. CTRL-C [C] is an interactive computer program for the analysis and design of multivariable control systems, time analysis, frequency analysis, matrix analysis and digital signal processing.
9. TRIP (TRansformation and Identification Program) [U] is an interactive program for the analysis and design of SISO systems. It provides the option of seven different types of models for linear systems (c.f. Equations 2.4 to 2.7).

3.4.2 Other identification programs

Other known programs are

1. MATLAB [C] is an interactive computer language for the analysis and design of multivariable control systems that shares a common origin with CTRL-C but is directed towards personal computers. The system identification toolbox is being continually updated and already includes the maximum likelihood, least squares, Box-Jenkins, instrumental variables and extended least squares identification algorithms in both nonrecursive and recursive forms. The central features of the package are functions that implement the most useful parametric

and nonparametric identification techniques. Details of the system identification toolbox are described in Ljung, 1987b.

2. Young and Hampton, 1984 have described the program CAPTAIN (Computer Aided Program for Time series Analysis and Identification of Noisy systems) [U]. This program was initially based on the recursive instrumental variable algorithm and the recursive extended least squares. The newer features of the program are, refined algorithms (adaptive prefiltering), MISO model identification and continuous time model estimation.

3. SYSID [U] (Denham *et al.*, 1975) is a package that is command driven that can provide correlation analysis and fast Fourier transforms as well as four different identification algorithms: least squares, generalized least squares, instrumental variables and correlation - least squares. The program can also perform dynamic simulations.

4. PIM+ [C] (Systems Technology Inc., 1988 and Landau, 1990) is a program available for the IBM-PC that is menu driven. This package was originally developed by I.D. Landau from the Laboratoire d'Automatique de Grenoble. Identification of up to four types of discrete linear parametric models using any of nine recursive identification methods can be performed. With a suitable interface card, the program can function in real time. Model validation, time and frequency domain analysis and

simulation are available.

5. SATER [U] (Van den Boom and Bollen, 1984) is a package that contains several parameter estimation routines (extended least squares, generalized least squares, instrumental variables and maximum likelihood), several order testing routines, a conversion program to convert discrete models into continuous models as well as simulation and test signal generation modules. Facilities for preprocessing measured signals, as well as modules for analyzing continuous systems based on Nyquist, Bode and root locus diagrams are provided.
6. SIRENA+ [U] (Laporte *et al.*, 1985) is a package that integrates a number of classical identification methods based on either time-domain or frequency domain system representations. The different identification methods are in output-error or an equation-error form, either batch or recursive, with or without noise filtering.
7. IP-DEM [U] (Caralp *et al.*, 1974) is a package composed of three programs. The first program is employed to collect experimental data and perform preliminary data treatment for the purpose of identification. The second program effects identification and selects the best models. The identification algorithms implemented are: instrumental variables, extended least squares and maximum likelihood. The third program allows the estimated model to be validated with a different set of data.

8. ESPION [U] (Haest *et al.*, 1990) is an expert system that performs an intelligent search for the "best" model according to a "quality index".
9. SEXI [U] (Gentil *et al.*, 1990) is also an expert system for process identification that searches for the "best" model.
10. Xmath [C] (Aling *et al.*, 1993) has several identification algorithms such as least squares, maximum likelihood, instrumental variables, etc. It uses QR and SVD decomposition for numerical robustness. The demarking feature of this program is its graphical user interface.

3.5 Other selected principal references concerning identification

The book of Eykhoff, 1974, probably one of the most complete books on the topic of process identification, even though somewhat outdated, remains a valuable reference. Another older book is that of Goodwin and Payne, 1977 which offers a good presentation of input signal design.

A geometrical interpretation of recursive process identification is presented in the book of Norton, 1986. The book of Ljung, 1987b, intended to be used in conjunction with MATLAB, contains not only a coverage of the common identification techniques but numerous references to the identification literature.

Numerical accuracy aspects of linear systems are discussed in the books of Wilkinson (Wilkinson, 1965, Wilkinson, 1963) and in Lawson and Hanson, 1974. Implementation issues relating to the numerical accuracy aspects of linear systems are addressed in LINPACK (Dongarra *et al.*, 1979). Various aspects of the numerical solution of control and estimation problems are presented in a tutorial paper of Laub, 1985.

An IFAC award winning series of papers (Willems, 1986a, Willems, 1986b, Willems, 1987) has provided a general mathematical description that encompasses time series and linear systems. These papers are oriented towards the presentation of linear systems theory rather than practical considerations of linear systems. In Ljung, 1987a a general description of process identification focusing mainly on frequency interpretation is presented.

Many papers have proposed algorithms and proved convergence for recursive algorithms. A landmark paper in this area was that of Ljung, 1977 who presented a new procedure to analyze the convergence of recursive stochastic algorithms. Many recent papers use this approach for the analysis of convergence of recursive parameter algorithms. Correspondence between Panuska and Ljung (Panuska, 1980, Ljung, 1980, Panuska, 1980b) has demonstrated that theoretical convergence of an algorithm is a desirable property but is not necessary nor sufficient to ensure convergence when the algorithm is applied to plant data where numerical accuracy aspects of the algorithm must also be taken into consideration.

Some algorithms (e.g. Carayannis *et al.*, 1983) take advantage of special structures such as the autoregressive model (with only past values of the output signal and present noise value used i.e. no values of the input signal and no past values of the noise) also known as the infinite impulse response (IIR) model. Another special structure is the moving average model, also known as the finite impulse response (FIR) or Markov process, for which values of the input signal and present noise values are used with no consideration of past values of the output signal and past noise values. Because of these implied assumptions, these algorithms are applicable only in special cases and so will not be given further consideration in this work.

Dugard and Landau, 1980, presented output error identification algorithms and showed that their performance compared favorably with the recursive extended least squares and recursive maximum likelihood identification techniques. The model used is the same as the Box-Jenkins model (c.f. Equation 2.4) with the D polynomial set to unity. Rather than use a nonlinear estimation scheme as in Box and Jenkins, 1976, auxiliary outputs are used instead of actual outputs making the use of the least squares algorithm possible. This type of algorithm does not seem to have found wide use, possibly because an auxiliary model for which the parameters are *a priori* unknown is required.

Stochastic identification algorithms exhibit slow convergence and have been found to be too unreliable (Isermann, 1982) for use in analyzing plant data so they will not be investigated in this work.

Modified least squares algorithms have been considered by Hsia, 1977 but it was found that the parameter estimates were not as accurate as the estimated parameters obtained by use of the generalized least squares algorithm.

Although a symmetric version of the instrumental variable method exists, it is seldom used (De Larminat, 1985). Use of the recursive instrumental variable method of identification under closed loop conditions results in biased parameter estimates (Isermann, 1982) unless special instruments (auxiliary variables) are utilized (Soderstrom, 1987).

Alternate versions to the standard maximum likelihood algorithm exist. For example, Gertler and Banyasz, 1974 used a maximum likelihood algorithm derived for the following model

$$Ay_t = Bu_{t-k} + \frac{1}{D}e_t$$

which is rarely used. The identification algorithm using this model is similar to the generalized least squares identification algorithm in many respects.

The use of the autocorrelation and crosscorrelation functions to replace the input and output values is a modification that reduces computation and memory requirements (c.f. COR-LS in Table 3.1). Memory requirements are reduced as the input and output data are now "condensed" to their correlation functions. Moreover, computing time is saved in the case of iterative procedures (extended least squares, instrumental variables, generalized least squares, maximum likelihood) as some of the calculations do not have to be repeated. Use of the correlation functions in conjunction with the generalized least squares identification techniques has been considered by Ahmed, 1984. Their use with the instrumental variable method (Ahmed, 1985) and least squares identification (Isermann *et al.*, 1974) has also been considered. Since the parameter estimates obtained with these algorithms do not tend to be as precise as those obtained using the conventional algorithms (larger variance), this variation of the basic algorithms has not been considered in this work.

Another possible method of enhancing the various algorithms is to "refine" them by prefiltering the input/output data to iteratively remove the noise in a similar manner to that employed with the generalized least squares identification (Young, 1984). To limit the scope of this work, prefiltering of the input/output data will not be investigated except as a tool to code the data (coding is examined in Chapter 10).

The use of recursive methods for off-line identification is addressed by Solbrand *et al.*, 1985. It was shown that several passes through the data were required to obtain similar precision as that obtained by nonrecursive methods.

In their paper, Stoica *et al.*, 1986 present two new model validation selection criteria that can be used to compare results from two identification runs. An extensive bibliography on this topic is also included. These validation criteria were not studied further in this thesis as their behavior was shown by Stoica *et al.*, 1986 to be asymptotically the same as the behavior of Akaike's information criterion and the behavior of the shortest data description criterion which are described in Appendix B. The use of validation criteria is still a topic of research (Veres, 1990a and Veres, 1990b).

The topic of forgetting factors remains not fully resolved. Progress in this area is on-going (Parkum *et al.*, 1992, Yung and Man, 1993, Bittanti and Campi, 1991). A different approach to maintain alertness is by modifying the covariance matrix directly (Sadighi and Pierre, 1990, Ramambason, 1992).

If the results of process identification are to be used in the context of process control then consideration should be given to the combination of identification and control (Shook *et al.*, 1991, Shook *et al.*, 1992, Rivera *et al.*, 1990 and Rivera, 1991).

Although many consider process identification as a mature research subject there are still some important discoveries being made e.g. the work of Niu and co-workers (Niu *et al.*, 1990, Niu and Fisher, 1992 and Niu *et al.*, 1992).

The volume of literature in the field of process identification is so extensive that a complete survey of the literature is not attempted in this work. Several papers, reviewed in enough detail to assess that their content was not directly relevant to the focus of this thesis, are listed in Appendix E.

4. Derivation of the Nonrecursive Algorithms

4.1 Introduction

The derivation of the more common nonrecursive identification algorithms presented in this chapter is used to illustrate the similarity of the various algorithms in using the inverse of a square matrix. This provides a strong argument for the wide applicability of the results of the latter chapters. The description of the algorithms will be useful in Chapter 5 as a basis for the derivation of the recursive algorithms.

This chapter forms a unique reference where a large number of nonrecursive identification algorithms are presented to the extent of including a full derivation of the algorithms.

The derivation of the algorithms that follows is adapted primarily from the presentations of Isermann *et al.*, 1974, Box and Jenkins, 1976 and Strejc, 1980. The nomenclature follows mostly that employed by Isermann *et al.*

4.2 Least squares algorithm

The least squares identification algorithm is the most widely used algorithm due to its simplicity and its effectiveness. The model used for the least squares algorithm is

$$A\ y_t = B\ u_{t-k} + \beta + e_t \qquad (4.1)$$

where

$$A = 1 + a_1q^{-1} + a_2q^{-2} + \ldots + a_nq^{-n}$$

$$B = b_1q^{-1} + b_2q^{-2} + \ldots + b_mq^{-m}$$

$$k = \text{delay}$$

$$\beta = \text{constant parameter (bias)}$$

$$\epsilon_t = \text{model error at time } t$$

the error (also known as the residuals, the white noise) can be expressed as

$$\epsilon_t = y_t + A'\ y_t - B\ u_{t-k} - \beta$$

where

$$A' = a_1q^{-1} + a_2q^{-2} + \ldots + a_nq^{-n}$$

the cost function for the least squares is

$$J = \sum_{i=1}^{N} \epsilon_i^2 \qquad (4.2)$$

where N is the horizon on which the minimization is performed (i.e. the number of points used for identification). This cost function is usually modified so that past values that are not available (i.e. before the initial time, i=1) are not involved. To take this into account a variable, μ, is defined as

$$\mu = (\text{maximum of } n \text{ or } m+k) + 1$$

The cost function can be written in vector form as

$$J = E^TE \qquad (4.3)$$

where

$$E = Y - \Psi\theta \qquad (4.4)$$

and

$$E^T = [\epsilon_\mu, \epsilon_{\mu+1}, \ldots, \epsilon_N]$$

$$Y^T = [y_\mu, y_{\mu+1}, \ldots, y_N]$$

$$\Psi = \begin{bmatrix} -y_{\mu-1} & -y_{\mu-2} & \cdots & -y_{\mu-n} & u_{\mu-k-1} & u_{\mu-k-2} & \cdots & u_{\mu-k-m} & 1 \\ -y_{\mu-1+1} & -y_{\mu-2+1} & \cdots & -y_{\mu-n+1} & u_{\mu-k-1+1} & u_{\mu-k-2+1} & \cdots & u_{\mu-k-m+1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y_{N-1} & -y_{N-2} & \cdots & -y_{N-n} & u_{N-k-1} & u_{N-k-2} & \cdots & u_{N-k-m} & 1 \end{bmatrix}$$

$$\theta^T = [a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m, \beta]$$

from Equations 4.3 and 4.4 the cost function may be written as

$$\begin{aligned} J &= (Y-\Psi\theta)^T(Y-\Psi\partial) \qquad (4.5) \\ &= Y^TY - (\Psi\theta)^TY - Y^T(\Psi\theta) + (\Psi\theta)^T\Psi\theta \\ &= Y^TY - 2\theta^T\Psi^TY + \theta^T\Psi^T\Psi\theta \end{aligned}$$

minimizing J with respect to θ

$$\begin{aligned} \frac{d\ J}{d\ \theta} &= -2\Psi^TY + 2\Psi^T\Psi\theta = 0 \\ \Psi^T\Psi\theta &= \Psi^TY \\ \theta &= (\Psi^T\Psi)^{-1}\Psi^TY \qquad (4.6) \\ \theta &= P\Psi^TY \qquad\qquad P = (\Psi^T\Psi)^{-1} \end{aligned}$$

It should be noted that P^{-1} cannot be singular for P to exist and that P and P^{-1} are symmetric. From P the standard deviations can be found as the covariance matrix is $\Sigma = P\sigma^2$ where σ^2 is the variance of residuals.

It is the calculation of this matrix P that is the focus of the latter chapters of the thesis. The calculation of the matrix P is the source of most of the numerical accuracy difficulties associated with the least squares algorithm.

It must also be noted that if there is colored noise (i.e. there is a C polynomial in the model, c.f. Equation

2.5) then there will be a bias in the parameters since the least squares algorithm does not take the C polynomial into account. This can be seen from the following (in vector form)

$$Y = \Psi\theta + CE$$

where

$$C = 1 + c_1q^{-1} + c_2q^{-2} + \ldots + c_{mn}q^{-mn}$$

then

$$\hat{\theta} = (\Psi^T\Psi)^{-1}\Psi^T Y \text{ (least squares estimate of } \theta, \text{ Equation 4.6)}$$

$$\hat{\theta} = (\Psi^T\Psi)^{-1}\Psi^T(\Psi\theta + CE)$$

$$\hat{\theta} = (\Psi^T\Psi)^{-1}\Psi^T\Psi\theta + (\Psi^T\Psi)^{-1}\Psi^T CE$$

$$E[\hat{\theta}] = \theta + E[(\Psi^T\Psi)^{-1}\Psi^T CE]$$

Ψ^T and $E_{\#-i}$ are uncorrelated for i=0 (the only term if the noise is white) but may correlated if $i \neq 0$ and Ψ^T has output terms or input terms correlated with past noise. Since the C polynomial introduces past E's (i>0) the second term will not vanish. This term is called the parameter estimate bias.

Investigation into the possibility of a bias will be used again in Chapter 8 where the effect of removing the constant term β is examined.

4.3 Weighted least squares algorithm

The difference between least squares and weighted least squares identification is that the cost function is modified to put a weight on the square of error. Generally the weighting is such that old data bares less importance than recent data (usually using exponential weighting).

It should be noted that in regression analysis a more general weighting scheme is used. In this case weights are placed on the off diagonal elements of the covariance matrix as well as on the diagonal elements. The reader that is not familiar with regression analysis may wish to consult a book such as that of Draper and Smith, 1981.

The model used for the weighted least squares is the same as for the least squares namely

$$A\ y_t = B\ u_{t-k} + \beta + \epsilon_t$$

The modified cost function is

$$J = \sum_{i=\mu}^{N} w_i \epsilon_i^2$$

where

$$w_i = \text{weight at time } i$$

or it can be expressed in vector form as

$$J = E^T W E \qquad (4.7)$$

where E is defined as for the least squares and W is a diagonal matrix defined as

$$W = \text{diag}(w_\mu,\ w_{\mu+1},\ \ldots,\ w_N)$$

combining Equations 4.4 and 4.7 gives

$$J = (Y-\Phi\theta)^T W (Y-\Phi\theta)$$

$$J = Y^T W Y - Y^T W \Phi\theta - (\Phi\theta)^T W Y + (\Phi\theta)^T W \Phi\theta$$

$$= Y^T W Y - 2\theta^T \Phi^T W Y + \theta^T \Phi^T W \Phi \theta$$

minimizing J with respect to θ leads to

$$\frac{d\ J}{d\ \theta} = -2\Phi^T W Y + 2\Phi^T W \Phi \theta = 0$$

$$\Phi^T W \Phi \theta = \Phi^T W Y$$

$$\theta = (\Psi^T W \Psi)^{-1} \Psi^T W Y$$

$$\theta = P \Psi^T W Y \qquad P = (\Psi^T W \Psi)^{-1} \qquad (4.8)$$

Again the formation of the matrix P is required. Therefore we can expect to have similar numerical accuracy problems as in the case of least squares.

4.4 Generalized least squares algorithm

This identification algorithm due to Clarke, 1967 is similar to the least squares identification algorithm but with the difference that the input and output signals are filtered to remove colored noise. The model used in this case is

$$A\ y_t = B\ u_{t-k} + \beta + N_t$$

where A and B are defined as before and N_t is colored noise which is modelled by

$$N_t = -f_1 N_{t-1} - f_2 N_{t-2} - \dots - f_\nu N_{t-\nu} + \epsilon_t + \beta_\nu$$

This colored noise can be used in a separate least square identification to provide the required parameters for the filtering of the data (f's). The order, ν, has to be selected appropriately.

The generalized least squares is an iterative procedure in which the filter parameters are updated using the residuals calculated with the most recent model parameters available. It is assumed that the model error (residuals) is a good representation of the actual noise sequence. The procedure can be summarized by the following steps

i. perform a least squares estimation assuming $N_t = \epsilon_t$

(white noise) to find θ

ii. let the residuals N be $N = Y - \Phi\theta$ where $N^T = [N_\mu, N_{\mu+1}, \dots, N_N]$. Then the error in the filter model is

$$\epsilon_t = N_t + f_1 N_{t-1} + f_2 N_{t-2} + \dots + f_\nu N_{t-\nu} - \beta_\nu$$

This is a least squares model with the input u_t set to zero and the output y_t being N_t (c.f. Equation 4.1). Let $\theta^T = f^T = [f_1, f_2, \dots, f_\nu, \beta_\nu]$,

$$\Phi = \Xi = \begin{bmatrix} -N_{\mu-1} & -N_{\mu-2} & \dots & -N_{\mu-\nu} & 1 \\ -N_\mu & -N_{\mu-1} & \dots & -N_{\mu-\nu+1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -N_{N-1} & -N_{N-2} & \dots & -N_{N-\nu} & 1 \end{bmatrix}$$

(It should be noted that μ has the same definition as before i.e. $\mu=\nu+1$) and $Y^T = N^T$ then from Equation 4.6

$$f = (\Xi^T\Xi)^{-1}\Xi^T N$$

iii. filter the input/output data

$$u^F_{t-k} = u_{t-k} + f_1 u_{t-k-1} + f_2 u_{t-k-2} + \dots + f_\nu u_{t-k-\nu} - \beta_\nu$$

$$y^F_t = y_t + f_1 y_{t-1} + f_2 y_{t-2} + \dots + f_\nu y_{t-\nu} - \beta_\nu$$

iv. repeat steps i to iii with the filtered data until the parameters stop changing

Since there are two least squares parameter identification involved in this algorithm one could expect the results of the latter chapters to apply. Unfortunately the correspondence with least squares is weakened by the iterative nature of the algorithm and the internal filtering of the input and output data. Nonetheless the principles developed in the following chapters apply to the generalized least squares algorithm since both the input and output values are passed through the same filter.

4.5 Instrumental variables algorithm

This algorithm pioneered by Young, 1970 differs from least squares identification in that the cost function is modified to include the instrumental matrix. The instrumental matrix acts as a filter to remove colored noise. It is introduced in Equation 4.4 such that

$$H^T E = H^T Y - H^T \Phi\theta \qquad (4.9)$$

where the instrumental matrix H must satisfy

$$E[H^T E] = 0 \text{ (means that H is uncorrelated with E)}$$

$$E[\Phi^T H] \quad \text{is non-singular}$$

the cost function is then defined as

$$J = (H^T E)^T (H^T E) \qquad (4.10)$$

Substituting Equation 4.9 in Equation 4.10 gives

$$\begin{aligned} J &= (H^T Y - H^T\Phi\theta)^T (H^T Y - H^T \Phi\theta) \\ &= (H^T Y)^T (H^T Y) - (H^T Y)^T H^T \Phi\theta - (H^T\Phi\theta)^T H^T Y + \\ &\qquad (H^T\Phi\theta)^T (H^T\Phi\theta) \\ &= Y^T H H^T Y - 2\theta^T\Phi^T H (H^T Y) + \theta^T \Phi^T H H^T \Phi\theta \end{aligned}$$

minimizing with respect to θ gives

$$\frac{d\ J}{d\ \theta} = -2\Phi^T H H^T Y + 2\Phi^T H H^T \Phi\theta$$

$$\Phi^T H H^T \Phi\theta = \Phi^T H H^T Y$$

premultiplying by $(\Phi^T H)^{-1}$ (it is non-singular by definition)

$$H^T\Phi\theta = H^T Y$$

$$\theta = (H^T\Phi)^{-1} H^T Y$$

$$\theta = PH^TY \qquad P=(H^T\Psi)^{-1}$$

The numerical accuracy of the algorithm (in the calculation of P) depends on the choice of the instrumental variable matrix. The closer H is to Ψ the more applicable the results from the latter chapters will be.

It should be noted that the standard deviation of the parameter estimates cannot be found from P. Calculations using P produce "pseudo" standard deviations.

This can be shown by examining the variance of $\hat{\theta}$, the estimate of θ

$$\begin{aligned}\hat{\theta} &= (H^T\Psi)^{-1}H^TY \\ &= (H^T\Psi)^{-1}H^T(\Psi\theta + E) \\ &= \theta + (H^T\Psi)^{-1}H^TE \\ E[\hat{\theta}] &= \theta \qquad \text{since } E[H^TE] = 0 \\ \text{var}(\hat{\theta}) &= E[(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T] \\ &= E[(H^T\Psi)^{-1}H^TEE^TH(H^T\Psi)^{-T}] \\ &= \sigma^2E[(H^T\Psi)^{-1}H^TH(\Psi^TH)^{-1}]\end{aligned}$$

if $E[\Psi^T] \cong H^T$ then

$$\text{var}(\hat{\theta}) \cong (H^T\Psi)^{-1}\sigma^2$$

as can be seen, the condition to have valid estimates of the approximate or pseudo standard deviations is that the instruments are the expected values of the input and output.

4.5.1 Choice of instrumental matrix

The instrumental matrix is not unique. It can be shown that some instrumental matrices are optimal (Young, 1970). One of these optimal matrices is composed of an auxiliary model output and the actual input. This will satisfy the required constraints on the instrumental matrix because the output of the model is not related to the noise of the process. The auxiliary model is

$$A\ h_t = B\ u_{t-k} + \beta$$

where A, B and β are defined as before. The use of the actual parameters for the auxiliary model will require an iterative procedure until the parameters stop changing. If this auxiliary model is used then H is similar to Φ with h_t replacing y_t.

$$H = \begin{bmatrix} -h_{\mu-1} & -h_{\mu-2} & \cdots & -h_{\mu-n} & u_{\mu-k-1} & u_{\mu-k-2} & \cdots & u_{\mu-k-m} & 1 \\ -h_{\mu-1+1} & -h_{\mu-2+1} & \cdots & -h_{\mu-n+1} & u_{\mu-k-1+1} & u_{\mu-k-2+1} & \cdots & u_{\mu-k-m+1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -h_{N-1} & -h_{N-2} & \cdots & -h_{N-n} & u_{N-k-1} & u_{N-k-2} & \cdots & u_{N-k-m} & 1 \end{bmatrix}$$

This choice of instrumental variable matrix ensures that H is close to Φ. The results of the latter chapters will therefore apply to the instrumental variable method with this instrumental variable matrix (once the convergence of the iterative procedure has been reached). Since the instrumental variable method produces a P matrix that is not symmetric several matrix factorization methods described in Chapter 7 will not be applicable.

4.6 Extended least squares algorithm

In this algorithm the model is "extended" to include a noise model. With this model the procedure for least squares is applied. The model used is

$$A\ y_t = B\ u_t + C\ \epsilon_t + \beta$$

where A, B and β are defined as before and

$$C = 1 + c_1 q^{-1} + c_2 q^{-2} + \ldots + c_{mn} q^{-mn}$$

the vector error equation is

$$E = Y - \Psi\theta$$

where E and Y are defined as before and

$$\theta^T = [a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n, c_1, c_2, \ldots, c_{mn}, \beta]$$

$$\Psi = \left[\begin{array}{ccccccc} -y_{\mu-1} & -y_{\mu-2} & \cdots & -y_{\mu-n} & u_{\mu-k-1} & u_{\mu-k-2} & \cdots \\ -y_{\mu-1+1} & -y_{\mu-2+1} & \cdots & -y_{\mu-n+1} & u_{\mu-k-1+1} & u_{\mu-k-2+1} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y_{N-1} & -y_{N-2} & \cdots & -y_{N-n} & u_{N-k-1} & u_{N-k-2} & \cdots \end{array} \right.$$

$$\left. \begin{array}{cccccc} u_{\mu-k-m} & \epsilon_{\mu-1} & \epsilon_{\mu-2} & \cdots & \epsilon_{\mu-mn} & 1 \\ u_{\mu-k-m+1} & \epsilon_{\mu-1+1} & \epsilon_{\mu-2+1} & \cdots & \epsilon_{\mu-mn+1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{N-k-m} & \epsilon_{N-1} & \epsilon_{N-2} & \cdots & \epsilon_{N-mn} & 1 \end{array} \right]$$

as for least squares the equation for θ is

$$\theta = (\Psi^T\Psi)^{-1}\Psi^T Y$$

$$\theta = P\Psi^T Y \qquad P = (\Psi^T\Psi)^{-1}$$

Because estimates of the residuals (ϵ_t) are needed, an iterative procedure is performed using the estimates of the residuals from the model with the most recent parameter estimates.

Unfortunately, the extended Ψ matrix introduces off-diagonal elements (with an expected value other than zero) as well as diagonal elements in $\Psi^T\Psi$. These off-diagonal elements appear from the dependance of y_t on past ϵ_t. Therefore one can expect more numerical accuracy problems from this algorithm than from the least squares algorithm (see Chapter 7 for the effect of off-diagonal elements on the condition number). Most of the results of the latter chapters are still applicable to this algorithm since a good portion of the $\Psi^T\Psi$ matrix is the same as that of the least squares algorithm.

4.7 Maximum likelihood algorithm

Although this algorithm is derived from the likelihood function, the usual assumptions lead to the minimization of the sum of squares. The difference between this algorithm and the other algorithms stems from the fact that a nonlinear least squares minimization is performed in contrast to the linear estimation performed by the previous algorithms. Therefore this algorithm could also be called a nonlinear least squares identification algorithm (Astrom, 1981).

The model used is

$$A\ y_t = B\ u_{t-k} + C\ \epsilon_t + \beta$$

where A, B, C, k and β are defined as before. The likelihood function for this model is of the form $L(y|u,A,B,C,\beta)$. This is the joint conditional likelihood function (akin to the probability density function) of the output given the input and the model parameters. Usually ln(L) is used because the logarithmic function is monotonic and the maxima and minima of ln(L) occur at the same values as the maxima and minima of L. This transformation is useful in linearizing the likelihood function if it has an exponential form.

At the maximum

$$\frac{d\ \ln(L)}{d\ \theta}\bigg|_{\theta=\theta^*} = 0$$

$$\frac{d^2\ \ln(L)}{d\ \theta^2}\bigg|_{\theta=\theta^*} < 0$$

if ϵ_t is Gaussian then the multinormal likelihood function is

$$L(E) = \frac{1}{(2\pi)^{N/2}\ |\Sigma|^{1/2}} \exp(\frac{-(E-\mu)^T\Sigma^{-1}(E-\mu)}{2})$$

where $|\Sigma|$ is the determinant of Σ, μ is the mean of E and Σ is the covariance matrix (see Box and Jenkins, 1976, Appendix A7.1). If ϵ_t for t prior to which the data is available is omitted in the vector E then the resulting algorithm is the (conditional) maximum likelihood algorithm usually encountered in the process identification literature (e.g. Isermann, 1980b). Assuming the mean μ to be zero and $\Sigma = \sigma^2 I$ where I is the identity matrix (i.e. the residuals are independent of each other and have the same variance) then

$$L(E) = \frac{1}{(2\pi)^{N/2}\ (\sigma^2)^{N/2}} \exp(\frac{-E^TE}{2\sigma^2})$$

$$\ln(L(E)) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln(\sigma^2) - \frac{E^TE}{2\sigma^2}$$

σ^2 is unknown but will be minimum at the maximum of L, therefore the minimum of $\frac{N}{2}\ln(\sigma^2) + \frac{E^TE}{\sigma^2}$ occurs for the same parameter values as the minimum of E^TE. Using $J = \frac{1}{2}E^TE$ as a cost function then

$$J = \frac{1}{2}\sum_{t=1}^{N} \epsilon_t^2$$

$$\frac{\partial J}{\partial \theta_i} = \sum_{t=1}^{N} \epsilon_t \frac{\partial \epsilon_t}{\partial \theta_i} \qquad (4.11)$$

$$\frac{\partial^2 J}{\partial \theta_i \, \partial \theta_j} = \sum_{t=1}^{N} \frac{\partial \epsilon_t}{\partial \theta_i} \frac{\partial \epsilon_t}{\partial \theta_j} + \sum_{t=1}^{N} \epsilon_t \frac{\partial^2 \epsilon_t}{\partial \theta_i \, \partial \theta_j} \qquad (4.12)$$

The parameter values at the minimum can be found using the Marquardt-Levenberg (Levenberg, 1944) parameter update

$$\theta_{k+1} = \theta_k - (V + \nu I)^{-1}\nabla J \qquad (4.13)$$

where V is the Hessian with all the second order derivatives of J with respect to θ and ∇J is the vector with all the first derivatives of J with respect to θ (see Appendix D for description of the Marquardt-Levenberg algorithm).

In this case V is the matrix to be inverted. Unfortunately, this iterative approach combined with the fact that the input and output values are internally filtered (see below) weakens the relationship to the P matrix of the least squares. Nonetheless the principles developed in the following chapters apply to the maximum likelihood algorithm since all the values in V are passed through the same filter. It should be noted that the regressor is also "extended" in a similar fashion as in the extended least squares algorithm (see the partial derivatives below). Here too a good portion of the matrix is similar to that of the least squares algorithm (except for filtering) so that most of the results of the latter chapters apply.

To find the first and second derivatives of the residuals with respect to θ the model equation is rewritten as

$$\epsilon_t = (A\ y_t - B\ u_{t-k} - \beta)/C$$

then the first derivatives are

$$\frac{\partial \epsilon_t}{\partial a_i} = y_{t-i}/C$$

$$\frac{\partial \epsilon_t}{\partial b_i} = -u_{t-k-i}/C$$

$$\frac{\partial \epsilon_t}{\partial \beta} = -1/C$$

$$\frac{\partial \epsilon_t}{\partial c_i} = -\epsilon_{t-i}/C$$

It can be seen that the second order derivatives not involving ∂c_i are zero. The remaining second order derivatives are

$$\frac{\partial^2 \epsilon_t}{\partial a_i\ \partial c_j} = -y_{t-i-j}/C$$

$$\frac{\partial^2 \epsilon_t}{\partial b_i \, \partial c_j} = u_{t-k-i-j}/C$$

$$\frac{\partial^2 \epsilon_t}{\partial \beta \, \partial c_j} = 1/C^2$$

$$\frac{\partial^2 \epsilon_t}{\partial c_i \, \partial c_j} = 2\epsilon_{t-i-j}/C^2$$

Note that inverting the order of the partial differentials on the denominator gives the same results (i.e. it is symmetrical). Also note that in Equation 4.12 the second order partial differentials are multiplied by ϵ_t. This product is summed over all the data points. The reader can easily verify that ϵ_t is not correlated with any of the second order partial derivatives and consequently the sum of the product will tend to zero. Neglecting this term in the second order partial differential equation leads to

$$\frac{\partial^2 J}{\partial \theta_i \, \partial \theta_j} = \sum_{t=1}^{N} \frac{\partial \epsilon_t}{\partial \theta_i} \frac{\partial \epsilon_t}{\partial \theta_j} \qquad (4.14)$$

The partial derivatives in Equations 4.11 and 4.14 can be expressed as a function of filtered input and output values using the noise parameters in an autoregressive filter namely

$$y_t^F = y_t/C$$

$$u_{t-k}^F = u_{t-k}/C$$

$$\beta^F = \beta/C = \frac{\beta}{(1+\sum_{i=1}^{mn} c_i)}$$

The filtered modelling error (noise) can be calculated as

$$\epsilon_t^r = A\ y_t^r - B\ u_{t-k}^r - \beta^r$$

The cost function partial derivatives can now be expressed as

$$\frac{\partial J}{\partial a_i} = \sum_{t=1}^{N} \epsilon_t y_{t-i}^r$$

$$\frac{\partial J}{\partial b_i} = \sum_{t=1}^{N} \epsilon_t (-u_{t-k-i}^r)$$

$$\frac{\partial J}{\partial c_i} = \sum_{t=1}^{N} \epsilon_t (-\epsilon_{t-i}^r)$$

$$\frac{\partial J}{\partial \beta} = \sum_{t=1}^{N} \epsilon_t \left(\frac{-1}{1+\sum_{i=1}^{mn} c_i}\right)$$

and the second order partial derivatives are

$$\frac{\partial^2 J}{\partial a_i\ \partial a_j} = \sum_{t=1}^{N} y_{t-i}^r y_{t-j}^r$$

$$\frac{\partial^2 J}{\partial a_i\ \partial b_j} = \sum_{t=1}^{N} y_{t-i}^r (-u_{t-k-j}^r)$$

$$\frac{\partial^2 J}{\partial a_i\ \partial c_j} = \sum_{t=1}^{N} y_{t-i}^r (-\epsilon_{t-j}^r)$$

$$\frac{\partial^2 J}{\partial a_i\ \partial \beta} = \sum_{t=1}^{N} y_{t-i}^r \left(\frac{-1}{1+\sum_{i=1}^{mn} c_i}\right)$$

$$\frac{\partial^2 J}{\partial b_i\ \partial b_j} = \sum_{t=1}^{N} (-u_{t-k-i}^r)(-u_{t-k-j}^r)$$

$$\frac{\partial^2 J}{\partial b_i\ \partial c_j} = \sum_{t=1}^{N} (-u_{t-k-i}^r)(-\epsilon_{t-j}^r)$$

$$\frac{\partial^2 J}{\partial b_i \, \partial \beta} = \sum_{t=1}^{N} (-u^F_{t-k-i}) \left(\frac{-1}{1+\sum_{i=1}^{mn} c_i} \right)$$

$$\frac{\partial^2 J}{\partial c_i \, \partial c_j} = \sum_{t=1}^{N} (-\epsilon^F_{t-i}) (-\epsilon^F_{t-j})$$

$$\frac{\partial^2 J}{\partial c_i \, \partial \beta} = \sum_{t=1}^{N} (-\epsilon^F_{t-i}) \left(\frac{-1}{1+\sum_{i=1}^{mn} c_i} \right)$$

$$\frac{\partial^2 J}{\partial \beta \, \partial \beta} = \sum_{t=1}^{N} \left(\frac{-1}{1+\sum_{i=1}^{mn} c_i} \right) \left(\frac{-1}{1+\sum_{i=1}^{mn} c_i} \right)$$

These derivatives are used to construct V and ∇J for the Marquardt-Levenberg parameter update.

4.7.1 Box-Jenkins modification to the maximum likelihood

Modification of the model to include a denominator term in the noise model has been proposed by Box and Jenkins, 1976.

The expanded model is

$$y_t = \frac{B}{A} u_{t-k} + \frac{C}{D} \epsilon_t + \beta$$

where A, B, C and β are defined as before and

$$D = 1 + d_1 q^{-1} + d_2 q^{-2} + \ldots + d_{nn} q^{-nn}$$

The parameter estimates are calculated using Equation 4.13. Numerical partial derivatives may be used instead of the analytical ones using

$$\frac{\partial \epsilon_t}{\partial \theta_i} = \frac{\epsilon_t(\theta_i + \delta) - \epsilon_t(\theta_i)}{\delta}$$

where δ is a small increment, typically 1% of θ_i.

4.8 Box-Jenkins preliminary parameter estimation algorithm

This algorithm (Box and Jenkins, 1976) does not involve the inversion of a square matrix but is presented here to provide an algorithm which can be used to find initial estimates for the Box-Jenkins algorithm presented in Section 4.7. "his algorithm uses the fact that the impulse response can be obtained from the crosscorrelation if the input is white noise (Appendix C) to find the model parameters. The impulse response model is

$$y_t = V\ u_t + N_t + \beta \tag{4.15}$$

where

$$V = v_0 + v_1 q^{-1} + v_2 q^{-2} + \dots$$

$$N_t = \text{colored noise}$$

The aim is to find the parameters in the model

$$y_t = \frac{B}{A}\ u_{t-k} + N_t + \beta \tag{4.16}$$

Equating Equations 4.15 and 4.16 gives

$$A\ V\ u_t = B\ u_{t-k} \tag{4.17}$$

and equating equal powers of q^{-1} the following preliminary parameter estimates are obtained

i. $v(j) = 0$ for $j \le k$, k the delay

ii. if $n > 0$, find A from powers of q^{-1} beyond the order of B such that the right hand side of Equation 4.17 is 0. this gives rise to the following set of equations

$\alpha\ A = V$

with

$A = [a_1, a_2, \dots, a_n]$

$V = [v_{k+m+1}, v_{k+m+2}, \ldots, v_{k+m+n}]$

$\alpha_{ij} = - v_{k+m+i-j}$ for $j < m+i$

$\alpha_{ij} = 0$ for $j \geq m+i$

for $i,j = 1, \ldots, n$

iii. $b_1 = v_{k+1}$

iv. if $m > 1$, find B from

if $n > 0$ $B = \beta A + V$

where

$B = [b_2, b_3, \ldots, b_m]$

$\beta_{ij} = v_{k+i-j+1}$ for $j \leq i$

$\beta_{ij} = 0$ for $j > i$

$V = [v_{k+2}, v_{k+3}, \ldots, v_{k+m}]$

if $n = 0$, $B = V$

for $i = 1, \ldots, m-1$

and $j = 1, \ldots, n$

To complete the model, the noise model preliminary estimates are obtained in the same manner as for the prewhitening parameter identification (Appendix C) and using Equation 4.15 to find the values of the colored noise. It should be kept in mind that these estimates are statistically inefficient (Box and Jenkins, 1976).

4.9 Conclusion

Except for the Box-Jenkins preliminary estimation algorithm, the algorithms presented use the general form

$$\theta = P \cdot v$$

where θ is the estimated parameter vector, P is the inverse of a matrix and v is a vector. θ, P, v are defined by the algorithm chosen. Thus, the nonrecursive algorithms can be viewed in a unified manner in the spirit of Ljung and Soderstrom, 1983.

Some of the algorithms require an iterative solution in which P and v are functions of the parameters. This dependance of P on θ for some algorithms weakens the relationship with the matrix P of the least squares algorithm but most of the results of the latter chapters still apply (this is especially true for the algorithms with internal filtering i.e. for generalized least squares, maximum likelihood and Box-Jenkins as all the values are affected in the same manner). The applicability of the results of the latter chapters is thus not limited to the least squares algorithm.

In this unique collection of algorithms and their derivation the algorithms were introduced individually to facilitate their examination and that of their recursive counterpart presented in the next chapter.

Some of the properties of these algorithms that are not related to numerical accuracy but are of general interest are summarized in Table 4.1 (see also Isermann, 1980a). The choice of a method over another should be based on statistical as well as numerical considerations. The purpose of modelling is also important. Are parameters sought or is only the prediction important? For the former the polynomials that should be included can be estimated from correlation analysis (Appendix C). This will dictate which algorithms may be used.

TABLE 4.1

Some properties of the nonrecursive algorithms

Parameter Estimation Method	Polynomials Included	Remarks
LS	A, B	Biased estimates for colored noise Small computational expense Good starting method for IV or ML
WLS	A, B	Same as LS Requires *a priori* knowledge of weighting matrix
ELS	A, B, C	Iterative Close to ML
GLS	A, B	Biased estimates possible Iterative Large computational expense Colored noise filtered Requires *a priori* filter order
IV	A, B	Possibly iterative Gond performance for a wide range of noise models Medium to small computational expense Difficulties appear with closed loop If not iterative requires *a priori* knowledge of instrumental matrix
ML	A, B, C	Iterative Good performance for specific noise model Large computational expense Noise model estimated Converges to local minima
BJ	A, B, C, D	Same as ML except for the model

The least squares algorithm is the most widely used algorithm due to its effectiveness and simplicity. If some part of the data is known to be "bad" e.g. outliers then weighted least squares can be used to minimize their effect. There is little difference between extended least squares and (conditional) maximum likelihood. The convergence of maximum likelihood is more robust than that of extended least squares from the use of the Marquardt-Levenberg algorithm. Generalized least squares and instrumental variable algorithms cannot be advantageously used for stochastic model prediction as the effect of the noise is eliminated from the model (in least squares the effect of the noise is included through biased parameters).

If several algorithms can be used or if only the prediction is important then the simpler the algorithm the better (both from a computational and from a numerical aspect). Performance of different models can be compared using one or more of the criteria described in Appendix B.

5. Recursive Identification Methods

5.1 Introduction

In this chapter the recursive version of the more common identification algorithms are presented. The derivation of the algorithms from their nonrecursive counterparts is used to emphasize their relationship with the inverse of a square matrix. Hence, the remarks on the numerical accuracy aspect discussed in Chapter 4 also apply to this chapter.

This chapter forms a unique reference where a large number of recursive identification algorithms are presented to the extent of including a full derivation of the algorithms.

The development presented for each recursive algorithm is based on the appropriate material that has been presented for the corresponding nonrecursive algorithm described in Chapter 4.

5.2 Recursive least squares algorithm

The nonrecursive least squares parameter estimates for N points can be calculated as (c.f. Equation 4.6)

$$\theta_N = (\Psi_N^T \Psi_N)^{-1} \Psi_N^T Y_N$$

where

$$\Psi_N^T = [\phi_\mu, \phi_{\mu+1}, \ldots, \phi_N]$$

$$\phi_i^T = [-y_{i-1}, -y_{i-2}, \ldots, -y_{i-n}, u_{i-k-1}, u_{i-k-2}, \ldots, u_{i-k-m}, 1]$$

$$Y_N^T = [y_\mu, y_{\mu+1}, \ldots, y_N]$$

To establish the recursive least squares algorithm, consider the situation where one more observation is introduced. Using

$$P_N = (\Psi_N^T\Psi_N)^{-1}$$

gives

$$P_{N+1} = \left[\begin{bmatrix} \Psi_N \\ \phi_{N+1}^T \end{bmatrix}^T \begin{bmatrix} \Psi_N \\ \phi_{N+1}^T \end{bmatrix}\right]^{-1}$$

$$P_{N+1} = (\Psi_N^T\Psi_N + \phi_{N+1}\phi_{N+1}^T)^{-1}$$

Use of the matrix inversion lemma (c.f. Appendix D) with $A = \Psi_N^T\Psi_N$, $B = \phi_{N+1}$, $C = I$ (identity matrix) and $D = \phi_{N+1}^T$ produces

$$P_{N+1} = (\Psi_N^T\Psi_N)^{-1} - (\Psi_N^T\Psi_N)^{-1}\phi_{N+1}(I + \phi_{N+1}^T(\Psi_N^T\Psi_N)^{-1}\phi_{N+1})^{-1}\phi_{N+1}^T(\Psi_N^T\Psi_N)^{-1}$$

$$P_{N+1} = P_N - P_N\phi_{N+1}(I + \phi_{N+1}^T P_N\phi_{N+1})^{-1}\phi_{N+1}^T P_N$$

It should be noted that this last equation uses the inverse of a scalar quantity. The error gain, K_{N+1}, is defined as

$$K_{N+1} = \frac{P_N\phi_{N+1}}{1 + \phi_{N+1}^T P_N\phi_{N+1}} \qquad (5.1)$$

so

$$P_{N+1} = P_N - K_{N+1}\phi_{N+1}^T P_N \qquad (5.2)$$

$$= (I + K_{N+1}\phi_{N+1}^T)P_N$$

The parameter estimates for N+1 points may be expressed as

$$\theta_{N+1} = P_{N+1} \begin{bmatrix} \Psi_N \\ \phi_{N+1}^T \end{bmatrix}^T \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix}$$

$$\theta_{N+1} = P_{N+1}\,[\Psi_N^T Y_N + \phi_{N+1}y_{N+1}]$$

substituting for P_{N+1} gives

$$\begin{aligned}\theta_{N+1} &= (P_N - K_{N+1}\phi^T_{N+1}P_N)\ (\Psi^T_N Y_N + \phi_{N+1}y_{N+1}) \\ &= \theta_N + [P_N\phi_{N+1} - K_{N+1}\phi^T_{N+1}P_N\phi_{N+1}]\ y_{N+1} \\ &\quad - K_{N+1}\phi^T_{N+1}\theta_N \\ \theta_{N+1} &= \theta_N + K_{N+1}(y_{N+1} - \phi^T_{N+1}\theta_N) \qquad (5.3)\end{aligned}$$

In summary, the update of the recursive least squares algorithm is Equations 5.1, 5.3 and 5.2 used successively.

Although the use of the matrix inversion lemma simplified the calculation of the inverse to obtain P_{N+1} from P_N the calculation of P is still subject to the same numerical accuracy constraints as the nonrecursive version i.e. the calculation of the inverse has been broken into smaller steps but has not been fundamentally changed.

5.2.1 Choice of initial values

The computation of the recursion equations (Equations 5.1 to 5.3) is based on the knowledge of θ_N and P_N. These can be found using a nonrecursive least squares but this requires a minimum number of points equal to the number of parameters plus delay if the input signal is persistently exciting and more points are required if the input signal does not excite all the modes of the process. It is common practice to give the parameters, θ_N, and the P matrix, P_N, the initial arbitrary values $\theta_N = 0$, $P_N = \alpha I$, α a large number and I the identity matrix. This introduces an error in the algorithm which can be corrected by reducing the effect of the initial values (c.f. Chapter 2 discussion on the effect of arbitrary values on the cost function). The effect of initial values can be minimized by using very long data sequences or by using a weighted least squares to put little weight on initial values. Choosing a large α also minimizes the effect of the initial parameters as it is used for the calculation of K_{N+1}, the gain associated with the error.

5.3 Weighted least squares algorithm

For the least squares algorithm it is assumed that the process and hence the model parameters are fixed but unknown values. However, there are many instances where the process is time varying. For example, there may be a change in catalyst activity in a chemical reactor or changes in production rate may change the dynamics of a distillation column due to the change in tray loading. In these instances, it may be of interest to track these changing parameters. To accomplish this, the recursive weighted least squares estimation algorithm is utilized.

The nonrecursive weighted least squares estimates for N points was given as (c.f. Equation 4.8)

$$\theta_N = (\Phi_N^T W_N \Phi_N)^{-1} \Phi_N^T W_N Y_N$$

$$P_N = (\Phi_N^T W_N \Phi_N)^{-1}$$

Consider one extra observation, then

$$P_{N+1} = \left[\begin{bmatrix} \Phi_N \\ \phi_{N+1}^T \end{bmatrix}^T \begin{bmatrix} W_N & 0 \\ 0 & w_{N+1} \end{bmatrix} \begin{bmatrix} \Phi_N \\ \phi_{N+1}^T \end{bmatrix} \right]^{-1}$$

$$P_{N+1} = (\Psi_N^T W_N \Psi_N + \phi_{N+1} w_{N+1} \phi_{N+1}^T)^{-1}$$

using the matrix inversion lemma with $A = \Psi_N^T W_N \Psi_N$, $B = \phi_{N+1}$, $C = I$, $D = w_{N+1}\phi_{N+1}^T$ gives

$$P_{N+1} = P_N - P_N\phi_{N+1}(I + w_{N+1}\phi_{N+1}^T P_N \phi_{N+1})^{-1}\ w_{N+1}\phi_{N+1}^T P_N$$

defining the error gain, K_{N+1}, as

$$K_{N+1} = \frac{P_N \phi_{N+1} w_{N+1}}{1 + w_{N+1}\phi_{N+1}^T P_N \phi_{N+1}} \tag{5.4}$$

then

$$P_{N+1} = P_N - K_{N+1}\phi_{N+1}^T P_N \tag{5.5}$$

The parameter estimates can be derived as follows

$$\theta_N = P_N \Psi_N^T W_N Y_N$$

$$\theta_{N+1} = P_{N+1} \begin{bmatrix} \Psi_N \\ \phi_{N+1}^T \end{bmatrix}^T \begin{bmatrix} W_N & 0 \\ 0 & w_{N+1} \end{bmatrix} \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix}$$

$$\theta_{N+1} = P_{N+1}[\Psi_N^T W_N Y_N + \phi_{N+1} w_{N+1} y_{N+1}]$$

substituting for P_{N+1} gives

$$\theta_{N+1} = \theta_N + K_{N+1}(y_{N+1} - \phi_{N+1}^T \theta_N) \tag{5.6}$$

In summary, the weighted least squares is updated by Equations 5.4, 5.6 and 5.5. Here too, P_{N+1} is calculated as the inverse of a matrix and is therefore subject to the same numerical accuracy problems as the nonrecursive version.

5.3.1 Choice of weights

For the sake of completeness the choice of weights is now briefly addressed. Many authors have proposed choices for the weights. Weights calculated using a deadband effectively "turn off" the identification algorithm if the error is low. Three examples of weight selection due to Goodwin and Sin, 1984 are presented here. The first weight scheme is

$$w_{N+1} = K_1 \text{ for } \phi_{N+1}^T P_N \phi_{N+1} \geq \epsilon$$

$$= K_2 \text{ for } \phi_{N+1}^T P_N \phi_{N+1} \leq \epsilon$$

with $K_1 >> K_2 > 0$. This weight scheme produces an algorithm which is close to the projection algorithm and is sensitive to noise. The second possibility is to use

$$w_{N+1} = \frac{\phi_{N+1}^T P_N \phi_{N+1}}{\phi_{N+1}^T \phi_{N+1}} \text{ if } \phi_{N+1}^T \phi_{N+1} \neq 0$$

$$= \text{arbitrary otherwise}$$

and the third weight scheme is

$$w_{N+1} = 1 \text{ if } \frac{|y_{N+1} - \phi_N^T \theta_N|^2}{1 + \phi_N^T P_{N-1} \phi_N} > \Delta^2 > 0$$

$$= 0 \text{ otherwise}$$

where Δ^2 is an upper bound on errors (noise).

5.3.2 Exponential weighting

Rather than put weight on the present value it is common to exponentially discount past values ($w_i = \lambda^{N-i}$, $0 \leq \lambda \leq 1$). For this, it is more convenient to write the weighting matrix under a different form namely

$$W = \lambda \cdot \mathrm{diag}(w_1, w_2, \ldots, 1/\lambda)$$

The lower λ is, the more weight there is on present data and less on past data. For $\lambda = 0$, only present data is considered and for $\lambda = 1$ all the data is considered with equal weighting. Since

$$\sum_{i=0}^{\infty} \lambda^i = \frac{1}{1-\lambda} \qquad (\text{for } \lambda < 1, \infty \text{ if } \lambda = 1)$$

the right hand side of this equation can be employed to calculate the effective number of past data points used. For example, if $\lambda = .95$ then the effective number of data points used is 20 i.e. $\lambda = .95$ is approximately the same as a weighting sequence of 20 values of λ equal to one and the remaining values of λ being zero. The effective number of data points used is termed the asymptotical window length.

When using a forgetting factor, there is a trade-off between speed of adaptation (or tractability) and variance (precision) of the estimates. A low λ allows for fast adaptation but also high variance, a higher λ allows for a slower adaptation but a lower variance of the parameters.

The weighted least squares algorithm, with exponential weighting, can be derived as follows. Starting with the nonrecursive weighted least squares algorithm

$$\theta_N = (\Psi_N^T W_N \Psi_N)^{-1} \Psi_N^T W_N Y_N$$
$$P_N = (\Psi_N^T W_N \Psi_N)^{-1}$$

and considering one more observation and letting λ be λ_{N+1} to allow for a variable forgetting factor yields

$$P_{N+1} = \left[\begin{bmatrix} \Psi_N \\ \phi_{N+1}^T \end{bmatrix}^T \lambda_{N+1} \begin{bmatrix} W_N & 0 \\ 0 & \frac{1}{\lambda_{N+1}} \end{bmatrix} \begin{bmatrix} \Psi_N \\ \phi_{N+1}^T \end{bmatrix} \right]^{-1}$$

$$P_{N+1} = (\Psi_N^T W_N \Psi_N + \phi_{N+1} \phi_{N+1}^T \frac{1}{\lambda_{N+1}})^{-1} \frac{1}{\lambda_{N+1}}$$

then using the matrix inversion lemma with $A = \Psi_N^T W_N \Psi_N$, $B = \phi_{N+1}$, $C = I$, $D = \phi_{N+1}^T \frac{1}{\lambda_{N+1}}$ produces

$$P_{N+1} = \frac{1}{\lambda_{N+1}}(P_N - P_N \phi_{N+1} (I + \frac{1}{\lambda_{N+1}} \phi_{N+1}^T P_N \phi_{N+1})^{-1} \times$$

$$\frac{1}{\lambda_{N+1}}\phi^T_{N+1}P_N)$$

defining the error gain, K_{N+1}, as

$$K_{N+1} = \frac{P_N\phi_{N+1}}{\lambda_{N+1} + \phi^T_{N+1}P_N\phi_{N+1}} \tag{5.7}$$

then

$$P_{N+1} = \frac{1}{\lambda_{N+1}}(P_N - K_{N+1}\phi^T_{N+1}P_N) \tag{5.8}$$

The parameter estimates can be derived as follows

$$\theta_N = P_N\Psi^T_N W_N Y_N$$

$$\theta_{N+1} = P_{N+1}\begin{bmatrix}\Psi_N\\ \phi^T_{N+1}\end{bmatrix}^T \lambda_{N+1}\begin{bmatrix}W_N & 0\\ 0 & \frac{1}{\lambda_{N+1}}\end{bmatrix}\begin{bmatrix}Y_N\\ y_{N+1}\end{bmatrix}$$

$$\theta_{N+1} = P_{N+1}(\Psi^T_N W_N Y_N \lambda_{N+1} + \phi_{N+1}\lambda_{N+1}y_{N+1})$$

substituting for P_{N+1} gives

$$\theta_{N+1} = \theta_N + K_{N+1}(y_{N+1} - \phi^T_{N+1}\theta_N) \tag{5.9}$$

In summary, the exponentially weighted least squares is updated by Equations 5.7, 5.9 and 5.8. It is common to have a variable forgetting factor λ. To eliminate the effect of initial guesses, that may be not very reliable or incorrect, a forgetting factor of the following form may be used

$$\lambda_{i+1} = \lambda_i\lambda_0 + (1 - \lambda_0)$$

with typical values λ_0 = .95 and λ_1 = .95 (Isermann, 1980b). This imposes an exponential data weighting during startup only. Other forgetting factors aim at forgetting the data only if a shift in parameters is perceived, usually by monitoring the size of the modelling error. Comparison of different forgetting factors has been carried out by Rogers, 1989.

5.4 Recursive generalized least squares algorithm

The nonrecursive generalized least squares identification was shown in Chapter 4 to be composed of five steps

i. filter input/output data with the noise filter

ii. perform a least squares estimation on filtered data

iii. calculate a new vector of residuals

iv. perform a least squares estimation on the residuals

v. repeat until converged

Since for recursive generalized least squares it is impossible to repeat the procedure (steps i to iv) without loosing all the benefits of the recursion, the recursive generalized least squares procedure is reduced to

i. filter input/output data with noise filter

ii. perform recursive least squares estimation on filtered data

iii. calculate the new residual (only one, not the whole vector)

iv. perform recursive least squares estimation on the residuals

v. repeat the procedure with the next input/output data point

Assuming a new pair of data points y_{N+1}, u_{N+1} has become available, the recursive generalized least squares algorithm can be described as follows (this is the same algorithm as proposed by H[illegible] [illegible] Sage, 1969)

i. gues[illegible] [illegible] vectors θ_N, f_N and matrices P_N, P_N^r [illegible] [illegible]uals)

ii. fi[illegible]

fo[illegible] $\ldots, -y_{N-n+1}, u_{N-k}, \ldots, u_{N-k-m+1}, 1]$

the [illegible] $(f) = f_N\phi_{N+1}^T$, $y_{N+1}(f) = f_N y_{N+1}$ where f_N is the [illegible] with its parameters taken at time N

iii. recursive least squares

$$K_{N+1} = \frac{P_N\phi_{N+1}(f)}{1 + \phi_{N+1}^T(f)P_N\phi_{N+1}(f)}$$

$$\theta_{N+1} = \theta_N + K_{N+1}(y_{N+1}(f) - \phi_{N+1}^T(f)\theta_N)$$

$$P_{N+1} = P_N - K_{N+1}\phi_{N+1}^T(f)P_N$$

iv. calculate residual

$$e_{N+1} = y_{N+1} - \phi_{N-1}^T\theta_{N+1}$$

v. perform recursive least squares on the residuals

$$\phi_{N+1}^T(r) = [-e_N, \ldots, -e_{N-p+1}, 1]$$

$$K_{N+1}^r = \frac{P_N^r\phi_{N+1}(r)}{1 + \phi_{N+1}^T(r)P_N^r\phi_{N+1}(r)}$$

$$f_{N+1} = f_N + K_{N+1}^r(e_{N+1} - \phi_{N+1}^T(r)f_N)$$

$$P_{N+1}^r = P_N^r - K_{N+1}^r\phi_{N+1}^T(r)P_N^r$$

This algorithm will converge to true values of the parameters if the signal to noise ratio is high enough. For a low signal to noise ratio, it is possible that the estimator will converge to false values (Strejc, 1981a).

Although this algorithm is only an approximation to its nonrecursive counterpart it still involves the inversion of 2 matrices so that comments about this algorithm in the previous chapter do apply.

5.5 Recursive instrumental variable algorithm

The nonrecursive instrumental variable estimation was given by

$$\theta_N = (H_N^T\Psi_N)^{-1}H_N^TY_N$$

The recursive solution is established in the same fashion as for the least squares

$$P_N = (H_N^T\Psi_N)^{-1}$$

$$P_{N+1} = \left[\begin{bmatrix} H_N \\ \phi_{N+1}^T(h) \end{bmatrix}^T \begin{bmatrix} \Psi_N \\ \phi_{N+1}^T \end{bmatrix}\right]^{-1}$$

where $\phi_{N+1}^T(h)$ contains the instruments (auxiliary outputs). This can be expressed as

$$P_{N+1} = (H_N^T\Psi_N + \phi_{N+1}(h)\phi_{N+1}^T)^{-1}$$

using the matrix inversion lemma with $A = H_N^T\Psi_N$, $B = \phi_{N+1}(h)$, $C = I$ (identity matrix) and $D = \phi_{N+1}^T$ (Appendix D) gives

$$P_{N+1} = (H_N^T\Psi_N)^{-1} - (H_N^T\Psi_N)^{-1}\phi_{N+1}(h)(I + \phi_{N+1}^T(H_N^T\Psi_N)^{-1}\phi_{N+1}(h))^{-1}\phi_{N+1}^T(H_N^T\Psi_N)^{-1}$$

$$P_{N+1} = P_N - P_N\phi_{N+1}(h)(I + \phi_{N+1}^TP_N\phi_{N+1}(h))^{-1}\phi_{N+1}^TP_N$$

It should be noted that this last equation uses the inverse of a scalar quantity. The error gain, K_{N+1}, is defined as

$$K_{N+1} = \frac{P_N\phi_{N+1}(h)}{1 + \phi_{N+1}^TP_N\phi_{N+1}(h)}$$

then

$$P_{N+1} = P_N - K_{N+1}\phi_{N+1}^TP_N$$

The parameter estimates are

$$\theta_{N+1} = P_{N+1}\begin{bmatrix} H_N \\ \phi_{N+1}^T(h) \end{bmatrix}^T \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix}$$

$$\theta_{N+1} = P_{N+1} [H_N^T Y_N + \varphi_{N+1}(h) y_{N+1}]$$

substituting for P_{N+1} gives

$$\theta_{N+1} = (P_N - K_{N+1} \varphi_{N+1}^T P_N) \; (H_N^T Y_N + \varphi_{N+1}(h) y_{N+1})$$

$$= \theta_N + [P_N \varphi_{N+1}(h) - K_{N+1} \varphi_{N+1}^T P_N \varphi_{N+1}(h)] \; y_{N+1}$$

$$- K_{N+1} \varphi_{N+1}^T \theta_N$$

$$\theta_{N+1} = \theta_N + K_{N+1}(y_{N+1} - \varphi_{N+1}^T \theta_N)$$

The instrumental variable method is composed of the following steps

i. perform least squares estimation

ii. generate auxiliary output

iii. perform the instrumental variable method

iv. if parameters have not converged repeat from step 2

Since this cannot be carried out with the recursive instrumental variable algorithm (without losing the benefits of recursion), the following approximation is used

i. perform recursive least squares estimation for a certain number of recursions

ii. generate auxiliary output

iii. perform recursive instrumental variable

iv. repeat from step (ii) for the next data point

Note that the recursive least squares estimation startup is carried until reasonable instruments can be found. This will make the algorithm more "robust" and will accelerate convergence.

Assuming a new pair of data points y_{N+1}, u_{N+1} has become available the recursive instrumental variable algorithm can be described as

i. guess θ_N and P_N

ii. recursive least squares estimation and auxiliary output generation

$$\phi^T_{N+1} = [-y_N, -y_{N-1}, \ldots, u_{N-k}, u_{N-k-1}, \ldots, 1]$$

$$K_{N+1} = \frac{P_N\phi_{N+1}}{1 + \phi^T_{N+1}P_N\phi_{N+1}}$$

$$\theta_{N+1} = \theta_N + K_{N+1}(y_{N+1} - \phi^T_{N+1}\theta_N)$$

$$P_{N+1} = P_N - K_{N+1}\phi^T_{N+1}P_N$$

auxiliary output startup

$$\phi^T_{N+1}(h) = [-h_N, -h_{N-1}, \ldots, u_{N-k}, u_{N-k-1}, \ldots, 1]$$

$$h_{N+1} = \phi^T_{N+1}(h)\ \theta_{N+1}$$

note that the most recent θ available is used for this update. The number of times step (ii) is executed before the recursive instrumental variable algorithm is employed is defined by the user. This could be a fixed number of recursions or it could be when the relative change in parameter estimates is smaller than a specified level.

iii. recursive instrumental variable

1) ϕ^T_{N+1} and $\phi^T_{N+1}(h)$ are formed as above

2) the core of the recursive instrumental variable is

$$K_{N+1} = \frac{P_N\phi_{N+1}(h)}{1 + \phi^T_{N+1}P_N\phi_{N+1}(h)}$$

$$\theta_{N+1} = \theta_N - K_{N+1}(y_{N+1} - \phi^T_{N+1}\theta_N)$$

$$P_{N+1} = P_N - K_{N+1}\phi^T_{N+1}P_N$$

3) update of the auxiliary output

$$h_{N+1} = \phi^T_{N+1}(h)\theta_{N+1}$$

This algorithm is only an approximation to the nonrecursive instrumental variable algorithm but the inversion of a matrix is still present so the comments of Chapter 4 still apply.

5.6 Recursive extended least squares algorithm

The nonrecursive extended least squares can be described as follows

i. find the residuals $E = Y - \Psi\theta$

ii. introduce the residuals E in Ψ

iii. $\theta = (\Psi^T\Psi)^{-1}\Psi^T Y$

iv. repeat these steps until parameters have converged

The recursive extended least squares must approximate the iterative process to fully take advantage of the recursion. The procedure is reduced to

i. find the residual $\epsilon_N = y_N - \phi_N^T\theta_N$

ii. place ϵ_N in ϕ_N

iii. use recursive least squares estimation

assuming a new pair of data points y_{N+1}, u_{N+1} is available, the recursive extended least squares algorithm can be described as follows

i. guess θ_N and P_N

ii. $\epsilon_N = y_N - \phi_N^T\theta_N$

$$\phi_{N+1}^T = [-y_N, \ldots, u_{N-k}, \ldots, \epsilon_N, \ldots, 1]$$

$$K_{N+1} = \frac{P_N\phi_{N+1}}{1 + \phi_{N+1}^T P_n \phi_{N+1}}$$

$$\theta_{N+1} = \theta_N + K_{N+1}(y_{N+1} - \phi_{N+1}^T\theta_N)$$

$$P_{N+1} = P_N - K_{N+1}\phi_{N+1}^T P_N$$

Even though this is an approximation to its nonrecursive counterpart comments on the calculation of the inverse of P in Chapter 4 do apply (especially when closer to convergence).

5.7 Recursive maximum likelihood algorithm

The nonrecursive maximum likelihood algorithm is an iterative procedure using

$$\theta_{N+1} = \theta_N - (V + \nu I)^{-1}\nabla J$$

where V is the Hessian with the second order partial derivatives of J with respect to θ and ∇J is the vector of first derivatives of J with respect to θ (c.f. Equation 4.13). In vector form the partial derivatives may be expressed as

$$\nabla J = (\frac{d\ E^T}{d\ \theta})E$$

and V can be approximated by

$$V \approx (\frac{d\ E^T}{d\ \theta})\ (\frac{d\ E^T}{d\ \theta})^T \tag{5.10}$$

or using the filtered values

$$\phi^T_{t+1} = [-y_t,\ \ldots,\ u_{t-k},\ \ldots,\ \epsilon_t,\ \ldots,\ 1]$$

$$\zeta^T_{t+1} = \frac{\phi^T_{t+1}}{C}$$

and defining

$$\psi^T = -(\frac{d\ E^T}{d\ \theta})$$

$$\psi^T = [\zeta_1,\ \ldots,\ \zeta_N]$$

then

$$\nabla J = -\psi^T E$$

$$V = \psi^T\psi$$

The update for the N+1st iteration can now be written as

$$\theta_{N+1} = \theta_N + (\psi^T\psi + \nu I)^{-1} \psi^T E$$

Expanding to show the new data explicitly gives

$$\theta_{N+1} = \theta_N + \left[\begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix}^T \begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix} + \nu_{N+1} I \right]^{-1} \begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix}^T \begin{bmatrix} E_N \\ \epsilon_{N+1} \end{bmatrix}$$

with

$$P_{N+1} = \left[\begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix}^T \begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix} + \nu_{N+1} I \right]^{-1}$$

or

$$P_{N+1} = (\psi^T_N \psi_N + \zeta_{N+1}\zeta^T_{N+1} + \nu_{N+1} I)^{-1}$$

Unfortunately the solution requires the inverse of an nxn matrix (n the total number of parameters) and the inversion lemma will not help in this case. Instead of the Marquardt-Levenberg algorithm, the Gauss-Newton algorithm may be used

$$\theta_{N+1} = \theta_N - V^{-1}\nabla J$$

$$\theta_{N+1} = \theta_N + \left[\begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix}^T \begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix} \right]^{-1} \begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix}^T \begin{bmatrix} E_N \\ \epsilon_{N+1} \end{bmatrix}$$

let

$$P_{N+1} = \left[\begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix}^T \begin{bmatrix} \psi_N \\ \zeta^T_{N+1} \end{bmatrix} \right]^{-1}$$

$$P_{N+1} = (\psi^T_N \psi_N + \zeta_{N+1}\zeta^T_{N+1})^{-1}$$

using the matrix inversion lemma with $A = \psi^T_{N+1}\psi_{N+1}$, $B = \zeta_{N+1}$, $C = I$, $D = \zeta^T_{N+1}$

$$P_{N+1} = (\psi^T_N\psi_N)^{-1} - (\psi^T_N\psi_N)^{-1}\zeta_{N+1}(I + \zeta^T_{N+1}(\psi^T_N\psi_N)^{-1}\zeta_{N+1})^{-1}\zeta^T_{N+1}(\psi^T_{N+1}\psi_N)^{-1}$$

$$= P_N - P_N\zeta_{N+1}(I + \zeta^T_{N+1}P_N\zeta_{N+1})^{-1}\zeta^T_{N+1}P_N$$

noting that it involves the inversion of a scalar, let

$$K_{N+1} = \frac{P_N\zeta_{N+1}}{1 + \zeta^T_{N+1}P_N\zeta_{N+1}}$$

so that

$$P_{N+1} = P_N - K_{N+1}\zeta^T_{N+1}P_N$$

now θ_{N+1} can be rewritten as

$$\theta_{N+1} = \theta_N + P_{N+1}(\psi^T_N E_N + \zeta_{N+1}\epsilon_{N+1})$$

substituting for P_{N+1}

$$\theta_{N+1} = \theta_N + (P_N - K_{N+1}\zeta^T_{N+1}P_N)(\psi^T_N E_N + \zeta_{N+1}\epsilon_{N+1})$$

$$\theta_{N+1} = \theta_N + P_N\psi^T_N E_N + P_N\zeta_{N+1}\epsilon_{N+1} - K_{N+1}\zeta^T_{N+1}P_N\psi^T_N E_N - K_{N+1}\zeta^T_{N+1}P_N\zeta_{N+1}\epsilon_{N+1}$$

also

$$\theta_N - \theta_{N-1} = P_N\psi^T_N E_N$$

substituting back

$$\theta_{N+1} = \theta_N + \theta_N - \theta_{N-1} - K_{N+1}\zeta^T_{N+1}(\theta_N - \theta_{N-1}) + K_{N+1}\epsilon_{N+1}$$

near the optimum $\theta_N \cong \theta_{N-1}$ so

$$\theta_{N+1} \cong \theta_N + K_{N+1}\epsilon_{N+1}$$

where

$$\epsilon_{N+1} = y_{N+1} - \phi^T_N \theta_N$$

This approximation gives the recursive maximum likelihood algorithm normally presented in the literature (Ljung and Soderstrom, 1983). The neglected term in the Gauss-Newton algorithm can be thought of as the projection using the gradient of θ with respect to the recursion number (time). In practice this term leads to a large change in parameters in the initial stages and does not improve convergence (Soderstrom, 1973). Convergence may be improved by using a variable forgetting factor to eliminate the effect of starting parameter values. This can also be applied to the noise parameters alone as they are used for filtering (Ljung and Soderstrom, 1983).

Convergence can also be improved by constraining the noise parameters to be in the stable region (Soderstrom, 1973).

If the C polynomial does not have its roots inside the unit circle in the q plane the algorithm may diverge. This is especially true in the early stages where the C parameters are not well known. It has been noted that the C parameters have a slower convergence than the A and B parameters (Isermann, 1980a).

Although an approximation to the nonrecursive maximum likelihood was utilized, the matrix inversion is retained therefore the comments of Chapter 4 are also valid in this case.

From a computational point of view changing from Marquardt-Levenberg to Gauss-Newton means that the convergence is less robust.

5.8 Recursive Box-Jenkins algorithm

Although the nonrecursive Box-Jenkins algorithm may use numerical derivatives, this is impractical for a recursive algorithm. For the recursive case analytical derivatives are used. The Box-Jenkins model is

$$y_t = \frac{B}{A}u_{t-k} + \frac{C}{D}\epsilon_t + \beta$$

The derivatives for this model are

$$\frac{\partial \epsilon_t}{\partial a_i} = \frac{BD}{CA^2}\, u_{t-k-i}$$

$$\frac{\partial \epsilon_t}{\partial b_i} = -\frac{D}{CA}\, u_{t-k-i}$$

$$\frac{\partial \epsilon_t}{\partial c_i} = -\frac{1}{C}\, \epsilon_{t-i}$$

$$\frac{\partial \epsilon_t}{\partial d_i} = \frac{1}{D}\, \epsilon_{t-i}$$

$$\frac{\partial \epsilon_t}{\partial \beta} = -\frac{D}{C}$$

The second derivatives not involving c_i or d_i are zero, the remaining second derivatives are

$$\frac{\partial^2 \epsilon_t}{\partial a_i\ \partial c_j} = -\frac{BD}{C^2A^2}\, u_{t-k-i-j}$$

$$\frac{\partial^2 \epsilon_t}{\partial b_i\ \partial c_j} = \frac{D}{C^2A}\, u_{t-k-i-j}$$

$$\frac{\partial^2 \epsilon_t}{\partial c_i\ \partial c_j} = \frac{1}{C^2}\, \epsilon_{t-i-j}$$

$$\frac{\partial^2 \epsilon_t}{\partial d_i\ \partial c_j} = -\frac{1}{CD}\, \epsilon_{t-i-j}$$

$$\frac{\partial^2 \epsilon_t}{\partial \beta\ \partial c_j} = \frac{D}{C^2}$$

$$\frac{\partial^2 \epsilon_t}{\partial a_i \, \partial d_j} = \frac{B}{CA^2} u_{t-k-i-j}$$

$$\frac{\partial^2 \epsilon_t}{\partial b_i \, \partial d_j} = - \frac{1}{CA} u_{t-k-i-j}$$

$$\frac{\partial^2 \epsilon_t}{\partial c_i \, \partial d_j} = - \frac{1}{CD} \epsilon_{t-i-j}$$

$$\frac{\partial^2 \epsilon_t}{\partial d_i \, \partial d_j} = \frac{1}{D^2} \epsilon_{t-i-j}$$

$$\frac{\partial^2 \epsilon_t}{\partial \beta \, \partial d_j} = - \frac{1}{C}$$

recall Equation 4.12

$$\frac{\partial^2 J}{\partial \theta_i \, \partial \theta_j} = \sum_{t=1}^{N} \frac{\partial \epsilon_t}{\partial \theta_i} \frac{\partial \epsilon_t}{\partial \theta_j} + \sum_{t=1}^{N} \epsilon_t \frac{\partial^2 \epsilon_t}{\partial \theta_i \, \partial \theta_j}$$

ϵ_t is not correlated with the second order derivatives, therefore the second summation will tend to zero. Neglecting this term yields

$$\frac{\partial^2 J}{\partial \theta_i \, \partial \theta_j} \cong \sum_{t=1}^{N} \frac{\partial \epsilon_t}{\partial \theta_i} \frac{\partial \epsilon_t}{\partial \theta_j}$$

As was obtained for the recursive maximum likelihood (c.f. Equation 5.10)

$$V \cong \left(\frac{d\, E^T}{d\, \theta}\right) \left(\frac{d\, E^T}{d\, \theta}\right)^T$$

$$\nabla J = \left(\frac{d\, E^T}{d\, \theta}\right) E$$

defining the following filtered quantities

$$u^f_{t-k} = \frac{D}{CA} u_{t-k}$$

$$u^g_{t-k} = \frac{B}{A} u^f_{t-k}$$

$$\epsilon_t = \frac{D}{C} y_t - \frac{BD}{CA} u_{t-k} - \frac{D}{C} \beta$$

$$= \frac{D}{C} y_t - B\, u^f_{t-k} - \frac{D}{C} \beta$$

$$\epsilon^f_t = \frac{1}{C} \epsilon_t$$

$$\epsilon_t^g = \frac{1}{D}\,\epsilon_t$$

the derivative can then be expressed as function of filtered values

$$\frac{\partial \epsilon_t}{\partial a_i} = u^g_{t-k-i}$$

$$\frac{\partial \epsilon_t}{\partial b_i} = -u^f_{t-k-i}$$

$$\frac{\partial \epsilon_t}{\partial c_i} = -\epsilon^f_{t-i}$$

$$\frac{\partial \epsilon_t}{\partial d_i} = \epsilon^g_{t-i}$$

$$\frac{\partial \epsilon_t}{\partial \beta} = -\frac{1 + \sum_{i=1}^{nn} d_i}{1 + \sum_{i=1}^{mn} c_i}$$

defining

$$\psi_N^T = -\left(\frac{d\ E_N}{d\ \theta}\right)$$

$$\zeta^T_{N+1} = [-u^g_{N-k},\ \ldots,\ u^f_{N-k},\ \ldots,\ \epsilon^f_N,\ \ldots,\ -\epsilon^g_N,\ \ldots,\ \frac{1 + \sum_{i=1}^{nn} d_i}{1 + \sum_{i=1}^{mn} c_i}]$$

then

$$\nabla J = -\ \psi_N^T E$$

$$V = \psi_N^T \psi_N$$

The remainder of the derivation is the same as for the recursive maximum likelihood giving

$$K_{N+1} = \frac{P_N \zeta_{N+1}}{1 + \zeta^T_{N+1} P_N \zeta_{N+1}}$$

$$P_{N+1} = P_N - K_{N+1}\zeta^T_{N+1}P_N$$

$$\theta_{N+1} = \theta_N + K_{N+1}\epsilon_{N+1}$$

with

$$\epsilon_{N+1} = y_{N+1} - \phi^T_{N+1}\theta_N$$

Although the algorithm of Sherif and Liu, 1984 was derived in an extended Kalman filter context, their algorithm is equivalent to the recursive Box-Jenkins algorithm.

5.9 Conclusion

The recursive algorithms can be expressed in the following general recursive form

$$\theta_{n+1} = \theta_n + K_{n+1}\ \epsilon_{n+1}$$

where θ_{n+1} is the new parameter vector, θ_n is the old parameter vector, K_{n+1} is the gain vector which involves the inverse of a matrix and ϵ_{n+1} is the prediction error. These values are defined by the algorithm chosen.

All of the algorithms are close approximations to their nonrecursive counterparts. The approximations are the result of initial conditions and/or combining an iterative step with a recursive step in order to retain the computer memory advantage of the recursion. The recursive algorithms converge to the same values as their nonrecursive equivalent when the effect of initial conditions vanishes. Therefore numerical accuracy is expected to be similar to that of the nonrecursive algorithms unless the initial transients lead to additional numerical accuracy difficulties.

Since the required initial values are usually unknown, the weighted least squares algorithm is generally preferred to the least squares algorithm so that the initial effects may be attenuated by a proper choice of the weighting sequence.

Nonrecursive algorithms usually perform better than recursive algorithms for short sequences since the effect of initial values will remain relatively important for short sequences. With large data records, recursive algorithms are more appealing since their computer memory requirements are smaller than those of the nonrecursive algorithms.

Some characteristic of the different recursive algorithms, not related to numerical accuracy, are presented in Table 5.1. The remarks regarding the nonrecursive algorithms discussed in Chapter 4 also apply to the recursive version of the algorithms.

TABLE 5.1

Some properties of the recursive algorithms

Parameter estimation method	Remarks
RLS	*a priori* factors: initial parameter estimates, initial P matrix Reliable convergence
RWLS	*a priori* factors: initial parameter estimates, initial P matrix, forgetting factor formulation Convergence depends on forgetting factor
RELS	Close to RML but convergence less reliable
RGLS	*a priori* factors: 2 sets of initial parameter estimates, 2 initial P matrices, noise filter order
RIV	*a priori* factors: initial parameter estimates, initial P matrix and possibly the instrumental matrix No reliable convergence therefore start with RLS
RML	*a priori* factors: initial parameter estimates, initial P matrix Slow convergence in starting phase Convergence more reliable than RELS
RBJ	Same as RML except for the model Slower convergence than RML

6. Practical aspects in process identification

6.1 Introduction

Many practical aspects in process identification must be considered if a satisfactory model is to be identified. These include outliers, initial conditions and forgetting factors for recursive algorithms, time delay and model order estimation, quantization and round-off errors, feedback (closed loop), type of input signal, sampling interval, identification time (run length) and validation criteria.

Brief comments on these aspects follow with references. Additional and more detailed information on the practical aspects in process identification are provided in Appendices A, B and C. Only the round-off effects, principally due to the inversion of the P matrix, are investigated in the remainder of the thesis. For further discussion of practical aspects of process identification see Isermann, 1980a.

6.2 Outliers

Outliers can be most easily detected by plots of residuals. Astrom, 1981 suggests that residuals that are larger than three standard deviations be discarded. Other possibilities are to replace the outliers by the maximum permissible values, by the mean, by filtering or by any other scheme that would give an estimate of the expected value of the outlier. It should be noted that these "fixes" may bias the parameter estimates and the only "true" solution is to discard the outliers and not use those data points in the parameter estimation.

6.3 Time delays and model orders

Time delays and model orders can be estimated through correlation analysis (Appendix C, Box and Jenkins, 1976). The adequacy of models of different orders can be compared based on certain statistics (Appendix B). Crosscorrelation of the system input and residuals can be used to detect a faulty time delay estimate (Astrom, 1981).

6.4 Quantization and round-off

Round-off (numerical accuracy) effects can be minimized through proper coding of the variables and numerical enhancement methods such as those studied in the remainder of this thesis. The importance of quantization can be seen by the example of rounding values to the nearest integer which can introduce oscillatory behavior (Astrom and Eykhoff, 1971).

6.5 Feedback or closed loop

Feedback can be detected using the crosscorrelation of the system input and residual (Astrom, 1981). Closed loop systems must meet the identifiability conditions (Isermann, 1980b). An example of a Box-Jenkins model for a system in which an external signal was used is presented in Box and MacGregor, 1974. A more recent discussion of closed loop identification can be found in Aude and Sandoz, 1986. Baur and Unbehauen, 1979 have used the instrumental variable method with an extra signal as the instrument to replace the input. They showed that the instrumental variable method does not work very well even with the extra signal but its performance would be better if the controller were known.

6.6 Choice of input signal

Although optimal input signal designs are available (Goodwin and Sin, 1984), they require the *a priori* knowledge of the model (which is usually not available). Some other input signals are conducive to good identification regardless of the underlying plant model. An introduction to the field of input signal design is presented in Appendix A.

6.7 Sampling interval

If the underlying process is continuous, changing the sampling interval will change the value of the parameters of the discretized model and may help or hinder identification. In certain circumstances, identification may be blamed for controller problems when in fact identification is not at fault (Appendix A).

Another problem linked to the choice of sampling interval is the phenomenon of aliasing. This occurs when a high frequency signal is sampled too slowly and appears as a low frequency signal (Appendix A). This can be avoided by a suitable low pass analog filter before sampling. Also, if not all the sampled data points are used but only every nth point is used then a prefilter based on the original sampled data should be used to remove the high frequencies that the nth sampled system cannot model (change in cut-off frequency).

6.8 Identification time

The length of the identification period depends largely on the input signal and on the signal to noise ratio. The experiment must be long enough so that all pertinent information about the process is collected (Appendix A).

6.9 Validation criteria

The model can be tested by crossvalidation, that is, by testing the estimated model using a different data set. Standard deviations and the parameter correlation matrix can be used to determine if some parameters should be removed. Crosscorrelation between inputs and residuals may indicate a feedback path or a erroneous delay estimate. Comparison between collected data and simulated output can also be used as a validation criterion. The residuals can be tested for whiteness (Appendix C). Finally several statistics are available to measure the adequacy of the model (Appendix B).

6.10 Conclusion

Although only the round-off (numerical accuracy) effects are examined in the remainder of this thesis many other practical aspects in process identification must be considered if a satisfactory model is to be identified. These include outliers, initial conditions and forgetting factors for recursive algorithms, time delay and model order estimation, quantization errors, feedback (closed loop), type of input signal, sampling interval, identification time (run length) and validation criteria.

Brief comments on these aspects were presented. Additional and more detailed information on the practical aspects in process identification are available in Appendices A, B and C and through the references cited.

7. Effect of the Magnitude of the Elements of a Matrix on its Condition Number

7.1 Introduction

In Chapters 4 and 5 it was argued that the numerical accuracy of the various algorithms can generally be linked to the numerical accuracy of the least squares algorithm. In the sequel only the numerical accuracy of the least squares algorithm will be examined in detail.

Using the standard approach to the numerical analysis of the solution of a set of equations (which is well known in the literature e.g. Lawson and Hanson, 1974, Forsythe *et al.*, 1977) it can be shown that the numerical accuracy of least squares is inversely proportional to the condition number of the P matrix. However, this type of analysis, known as perturbation analysis, does not consider the effect of the magnitude of specific elements of the matrix on its condition number.

This chapter extends the results of perturbation analysis by examining the effect of the magnitude of specific elements of the P matrix on its condition number. This analysis will be useful for developing and assessing techniques employed to reduce the condition number of the P matrix. These techniques are examined in the following chapters.

Various matrix factorizations are reviewed and are shown to provide similar numerical conditioning benefits.

Finally, two theorems on the effet of coding (scaling each variable) and the choice of input signal for first and higher order systems are presented. Significant insight into the application of the least squares algorithm is developed from these two theorems.

7.2 Use of the condition number as a measure of numerical accuracy

There are two factors that will influence the condition number of a matrix: the ratio of the diagonal elements and the ratio of the off-diagonal elements of the matrix. The ratio of diagonal elements refers to the range of magnitudes of the diagonal elements of the P matrix while the ratio of the off-diagonal elements relates to their magnitude with respect to diagonal elements. It should be noted that, by construction, the P matrix cannot have an off-diagonal element larger than the largest diagonal element on the same row or column.

The numerical sensitivity of the calculation of the matrix inverse due to the ratio of diagonal elements is problem dependent and generalizations using the condition number are not possible in this case. But, since a vector multiplication follows the matrix inversion in the least squares problem the ratio of diagonal elements should be as close to unity as possible, for better numerical accuracy.

For properly ratioed problems (for the purpose of this work, a properly ratioed matrix is defined as a matrix for which the ratio of the diagonal elements is close to unity) the condition number, κ, provides a quantitative measure of the computational difficulty that can be expected. For example if $\kappa(A)$ is 10^k then the solution of a linear system computed in t-digit (decimal) arithmetic (e.g. 6 digits for single precision) will have no more than t-k accurate figures (Dongarra *et al.*, 1979).

7.3 Effect of the ratio of the off-diagonal elements on the condition number

In this section, the effect of the ratio of the off-diagonal elements on the condition number is examined. The results of this section will be used in latter chapters to stress the importance of selecting an appropriate method to manipulate the off-diagonal elements.

Let A be a symmetric matrix of order n and E be a symmetric matrix of the same order with a single pair of nonzero elements and zeros on the diagonal. Let the value of the nonzero elements be a. The eigenvalues of E are $-a$, a and n-2 eigenvalues of 0. Using the property (e.g. see Golub and Van Loan, 1983)

$\lambda_k(A) + \lambda_n(E) \le \lambda_k(A+E) \le \lambda_k(A) + \lambda_1(E)$

where λ_k is the kth largest eigenvalue then

$\lambda_1(A+E) \le \lambda_1(A) + \lambda_1(E)$

$\lambda_n(A+E) \ge \lambda_n(A) + \lambda_n(E)$

The condition number can be expressed as (Dennis and Schnabel, 1983)

$$\kappa_2(A+E) = \frac{\lambda_1(A+E)}{\lambda_n(A+E)}$$

$$\kappa_2(A+E) \le \frac{\lambda_1(A) + \lambda_1(E)}{\lambda_n(A) + \lambda_n(E)}$$

$$\kappa_2(A+E) \le \frac{\lambda_1(A) + |a|}{\lambda_n(A) - |a|} \qquad (7.1)$$

Unfortunately this provides only an upperbound on the condition number. For the special case of A = I (i.e. for the ideal case)

$$\kappa_2(A+E) = \frac{1 + |a|}{1 - |a|}$$

This ratio can be thought of as a first order Padé approximation to $e^{2|a|}$. This implies that as the off-diagonal terms increase in magnitude, the condition number will increase at an approximately exponential rate. Table 7.1 gives the condition number as a function of a for A being the identity matrix.

TABLE 7.1

Condition number as a function of the magnitude of the off-diagonal element a

a	Condition number	a	Condition number
0.0	1.0	0.6	4.0
0.1	1.22	0.7	5.67
0.2	1.58	0.8	9.0
0.3	1.86	0.9	19.0
0.4	2.33	0.95	39.0
0.5	3.0	0.99	199.0
		1.0	∞

7.4 Combined effect of several off-diagonal elements on the condition number

The results obtained in this section will provide the motivation (in latter chapters) for the methods used to reduce the condition number when off-diagonal elements are present.

The case of several pairs of elements can be addressed by applying Inequality 7.1 to each pair in succession giving

$$\kappa(A + \Sigma E_i) \le \frac{\lambda_1(A) + \Sigma|a_i|}{\lambda_n(A) - \Sigma|a_i|}$$

This implies that the upperbound of $\kappa(A+E)$ follows an approximately multiple exponential rate. Although the upperbound can be much larger than the condit.on number itself it nonetheless points to an exponential trend.

The combined effect of several off-diagonal elements can be exposed to closer scrutiny by using the following 3 x 3 matrix

$$A = \begin{bmatrix} 1 & \alpha & \beta \\ \alpha & 1 & \gamma \\ \beta & \gamma & 1 \end{bmatrix}$$

with

$$|\alpha| \le 1$$

$$|\beta| \le 1$$

$$|\gamma| \le 1$$

Determining the eigenvalues of this matrix analytically is rather involved so to avoid this difficulty the following relationship between 3 x 3 matrix norms is used (Dennis and Schnabel, 1983)

$$3^{-1/2}\|A\|_\infty \le \|A\|_2 \le 3^{1/2}\|A\|_\infty$$

where $\|A\|_\infty$ is the maximum row sum of absolute values. Applying this relationship to the condition number of a 3 x 3 matrix gives

$$\frac{1}{3}\kappa_\infty(A) \le \kappa_2(A) \le 3\kappa_\infty(A)$$

From this it can be concluded that α, β and γ will have a similar effect on $\kappa_\infty(A)$ and on $\kappa_2(A)$. The inverse of the matrix A, needed to calculate $\kappa_\infty(A)$ is given by

$$A^{-1} = \frac{1}{\Delta}\begin{bmatrix} 1-\gamma^2 & \gamma\beta-\alpha & \alpha\gamma-\beta \\ \gamma\beta-\alpha & 1-\beta^2 & \alpha\beta-\gamma \\ \alpha\gamma-\beta & \alpha\beta-\gamma & 1-\alpha^2 \end{bmatrix}$$

where $\Delta = 1 - \alpha^2 - \beta^2 - \gamma^2 + 2\alpha\beta\gamma$. The condition number can be expressed as

$$\kappa_\infty(A) = \frac{1}{|\Delta|}\max\begin{bmatrix} 1+|\alpha|+|\beta| \\ 1+|\alpha|+|\gamma| \\ 1+|\beta|+|\gamma| \end{bmatrix}\max\begin{bmatrix} |1-\gamma^2|+|\gamma\beta-\alpha|+|\alpha\gamma-\beta| \\ |\gamma\beta-\alpha|+|1-\beta^2|+|\alpha\beta-\gamma| \\ |\alpha\gamma-\beta|+|\alpha\beta-\gamma|+|1-\alpha^2| \end{bmatrix} \quad (7.2)$$

The first expression in the first bracket may be assumed to be the maximum value of the first bracket without loss of generality since the result will be symmetric with respect to the variables (i.e. α, β and γ may be interchanged and the condition number will remain unchanged).

Assuming that the first expression of the first bracket is maximum implies that

$$|\beta| \ge |\gamma| \quad (7.3)$$

$$|\alpha| \ge |\gamma| \quad (7.4)$$

The maximum expression of the second bracket is determined by the greater of

$$|1-\gamma^2| + |\alpha\gamma-\beta| + |\gamma\beta-\alpha| \qquad (7.5)$$

$$|1-\beta^2| + |\alpha\beta-\gamma| + |\gamma\beta-\alpha| \qquad (7.6)$$

$$|1-\alpha^2| + |\alpha\gamma-\beta| + |\alpha\beta-\gamma| \qquad (7.7)$$

Expression 7.5 will now be shown to be larger than Expression 7.6. The comparison between Expression 7.5 and Expression 7.7 will produce similar results due to the symmetry between Expression 7.6 and Expression 7.7 and Inequalities 7.3 and 7.4 with respect to the parameters α and β. The comparison can therefore be reduced to comparing

$$|1-\gamma^2| + |\alpha\gamma-\beta| \text{ and } |1-\beta^2| + |\alpha\beta-\gamma|$$

Inequality 7.3 combined with $|\beta| \le 1$ gives

$$|1-\gamma^2| \ge |1-\beta^2|$$

The comparison can therefore be reduced to verifying that

$$|\alpha\gamma-\beta| \ge |\alpha\beta-\gamma|$$

Since both sides of the inequality are positive quantities the inequality can be squared giving

$$(\alpha\gamma-\beta)^2 - (\alpha\beta-\gamma)^2 \ge 0$$

or

$$(\alpha^2-1)(\gamma^2-\beta^2) \ge 0 \qquad (7.8)$$

Since $|\alpha| < 1$, it follows from Inequality 7.3 that Inequality 7.8 holds. Consequently the first row of the second bracket in Expression 7.2 is thus the maximum row sum of the inverse of the matrix A so the condition number is given by

$$\kappa_\infty(A) = \frac{1}{\Delta}\,(1+|\alpha|+|\beta|)\,(|1-\gamma^2|+|\gamma\beta-\alpha|+|\alpha\gamma-\beta|) \qquad (7.9)$$

with the following restrictions

$$|\alpha| < 1$$

$$|\beta| < 1$$

$$|\gamma| \leq |\alpha|$$

$$|\gamma| \leq |\beta|$$

$$\Delta > 0$$

This last inequality states that the determinant is positive (a requirement for positive definite matrices). From the symmetry involved in the calculation of the condition number, α, β and γ can be interchanged to satisfy the inequalities.

Minimum condition number

In this section the condition number as expressed by Equation 7.9 is examined to determine if a minimum can occur by cancelling effects of the off-diagonal elements or if a minimum occurs only when the off-diagonal elements are zero.

To simplify the analysis, the lower bound on the condition number is examined. First, the last two positive terms of the numerator of the condition number are neglected giving

$$\kappa_\infty(A) \geq \frac{(1 + |\alpha| + |\beta|)(|1-\gamma^2|)}{1-\alpha^2-\beta^2-\gamma^2+2\alpha\beta\gamma}$$

Then $|\alpha|$ and $|\beta|$ are replaced by their lower bound, $|\gamma|$, in the positive terms of the numerator and the negative terms of the denominator so that the lower bound on the condition number may be expressed as

$$\kappa_\infty(A) \geq \frac{(1+2|\gamma|)(1-\gamma^2)}{1 - 3\,\gamma^2 + 2\,\alpha\beta\gamma}$$

Closer examination of the denominator reveals that

$$1 - 3\gamma^2 + 2\alpha\beta\gamma \le 1 - 3\gamma^2 + 2|\alpha\beta\gamma|$$

and since $|\alpha| < 1$ and $|\beta| < 1$

$$1 - 3\gamma^2 + 2\alpha\beta\gamma < 1 - 3\gamma^2 + 2|\gamma|$$

the lower bound on the condition number can be written as

$$\kappa_\infty(A) > \frac{(1+2|\gamma|)(1-\gamma^2)}{1 + 2|\gamma| - 3\gamma^2}$$

This bound can also be expressed as

$$\kappa_\infty(A) > \frac{(1+2|\gamma|)(1-|\gamma|)(1+|\gamma|)}{(1+3|\gamma|)(1-|\gamma|)}$$

which can be simplified to

$$\kappa_\infty(A) > \frac{1 + 3|\gamma| + 2\gamma^2}{1 + 3|\gamma|}$$

The minimum of the lower bound on the condition number is unity when γ is zero. For any other value of γ the lower bound on the condition number will be greater than unity and therefore the condition number will not be minimum since the minimum is unity in this case. Since the minimum condition number will occur only for $\gamma=0$ then the condition number at the minimum is reduced to

$$\kappa_\infty(A) = \frac{(1 + |\alpha| + |\beta|)^2}{1 - \alpha^2 - \beta^2}$$

for which the minimum occurs only if α and β are zero. It thus follows that the minimum condition number occurs only when there are no off-diagonal elements and the existence of off-diagonal elements cannot compensate or interact to cancel the effect of each other on the condition number.

Increase of the condition number with several off-diagonal elements

It is instructive to consider the increase in the condition number for the special case of $\gamma = 0$ which allows the condition number to be expressed as

$$\kappa_{\infty}(A) = \frac{(1+|\alpha|+|\beta|)^2}{1-\alpha^2-\beta^2} \tag{7.10}$$

From this expression it follows that any nonzero off-diagonal component (recall that α, β and γ can be interchanged for the condition number) will tend to increase the condition number approximately exponentially as the magnitude of the off-diagonal components increase. Furthermore the combined effect on the condition number of more than one off-diagonal component will become larger than the sum of the individual effects as the magnitude of the off-diagonal elements increases. It can be seen from Equation 7.10 that the condition number increases as $\alpha^2 + \beta^2$ approaches unity which is more restrictive than α^2 or β^2 approaching unity. This latter restriction can be interpreted as a square that circumscribes the circle defined by $\alpha^2 + \beta^2 = 1$. The area enclosed by the square is larger than that enclosed within the circle with the difference between the two areas representing the added restriction due to the "interaction" between the off-diagonal terms. This is depicted in Figure 7.1 where the "interaction" is the portion outside the circle. The smaller area also implies that the condition number increases at a faster rate with several off-diagonal elements as compared to a single off-diagonal element.

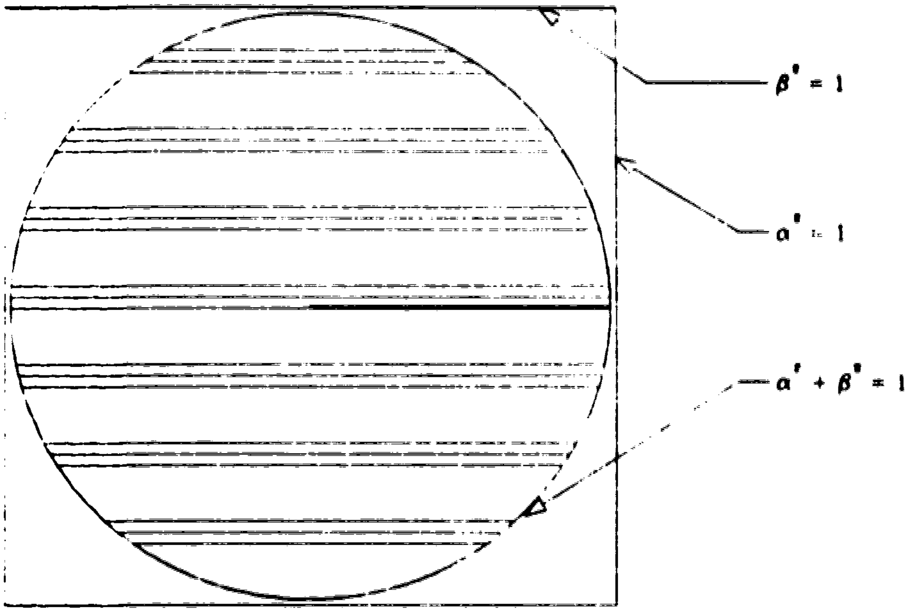


Fig. 7.1 Restricted area produced by the combination of two off-diagonal elements

It should be noted that singularity ($\kappa_\infty = \infty$) can occur for an infinite number of cases (any point on this circle) even when the value of one off-diagonal element (γ) is zero. This is in contrast to the pair of points that can generate singularity when a single off-diagonal element is present. This remark stresses the fact that ill-conditioned matrices are not a rare occurrence and conditions for the existence of singularity are not as strict as commonly perceived.

7.5 Matrix factorizations

In the numerical analysis field the usual method used to enhance conditioning of a matrix is through matrix factorization. In this section the relationship between some of these factorization methods and the condition number is presented for completeness. An extensive analysis of these factorization methods is available elsewhere (e.g. Dongarra *et al.*, 1979, Bierman, 1977). Analysis of the round-off error propagation has been examined by Verhaegen, 1989.

7.5.1 Square root factorization

In this section, the relationship between the condition number of the P matrix and the condition number of its square root is presented. The condition number chosen is κ_2.

$$\begin{aligned}\kappa_2(P) &= \|P\|_2\|P^{-1}\|_2\\ &= [\max \lambda(P)]\ [\max \lambda(P^{-1})]\end{aligned}$$

This last equality is valid since P is a positive definite symmetric matrix. The square root of the matrix P is defined as

$$P = Q^TQ$$

where Q is usually a square nonsymmetric matrix. The factorization is not unique unless a special form of the matrix Q, such as lower triangular, is assumed. The lower triangular factorization is known as the Cholesky decomposition (Bierman, 1977). The norm of a general matrix Q is (Dennis and Schnabel, 1983)

$$\begin{aligned}\|Q\|_2 &= (\max \lambda(Q^TQ))^{1/2}\\ &= (\max \lambda(P))^{1/2}\\ \|Q\|_2 &= (\|P\|_2)^{1/2}\end{aligned}$$

For the norm of Q^{-1} the relationship $\lambda(Q^TQ) = \lambda(QQ^T)$ for Q square and nonsingular is utilized. This relationship can be shown by using the property that similarity transformations do not affect the eigenvalues (Golub and Van Loan, 1983) giving

$$\lambda(Q^TQ) = \lambda(QQ^TQQ^{-1})$$
$$= \lambda(QQ^T)$$

Since the norm of Q^{-1} is

$$\|Q^{-1}\|_2 = (\max \lambda(Q^{-T}Q^{-1}))^{1/2}$$

then

$$\|Q^{-1}\|_2 = (\max \lambda(Q^{-1}Q^{-T}))^{1/2}$$
$$\|Q^{-1}\|_2 = (\max \lambda((Q^TQ)^{-1}))^{1/2}$$
$$\|Q^{-1}\|_2 = (\max \lambda(P^{-1}))^{1/2}$$
$$\|Q^{-1}\|_2 = (\|P^{-1}\|_2)^{1/2}$$

If Q is chosen to be square it will be nonsingular if P is nonsingular and the condition number of Q is thus

$$\kappa_2(Q) = \|Q\|_2\|Q^{-1}\|_2$$
$$\kappa_2(Q) = (\|P\|_2)^{1/2} (\|P^{-1}\|_2)^{1/2}$$
$$\kappa_2(Q) = (\kappa_2(P))^{1/2}$$

So use of Q instead of P will reduce the approximately exponential increase of the condition number, as the ratio of the off-diagonal elements increases, to one half the rate of that of the condition number of P.

7.5.2 U-D factorization

Another form of factorization that is widely used is the U-D factorization proposed by Bierman, 1977. The main advantage of the U-D factorization is that it eliminates the need to calculate square roots.

The U-D factorization of P is

$$P = UDU^T$$

where U is an upper triangular matrix with unity values on its diagonal and D is a diagonal matrix. This can also be written as

$$P = UD^{1/2}D^{1/2}U^T$$

By setting

$$Q^T = UD^{1/2}$$

the U-D factorization can be thought of as a form of square root factorization where Q is chosen to be lower triangular. The norm of Q^T is

$$\begin{aligned}\|Q^T\|_2 &= (\max \lambda(QQ^T))^{1/2}\\ &= (\max \lambda(Q^TQ))^{1/2}\\ &= \|Q\|_2\end{aligned}$$

and since the norm of its inverse is

$$\|Q^{-T}\|_2 = \|Q^{-1}\|_2$$

it follows that the condition number can be expressed as

$$\begin{aligned}\kappa_2(UD^{1/2}) &= \kappa_2(Q)\\ &= (\kappa_2(P))^{1/2}\end{aligned}$$

Unfortunately results concerning U or D individually are not available.

7.5.3 Singular value decomposition

The singular value decomposition (SVD) is a powerful numerical analysis tool that also provides factorization (Klema and Laub, 1980, Dongarra *et al.*, 1979). In the context of process identification, the singular value decomposition is usually applied to the P matrix giving

$$P = U\Sigma V^T$$

where U and V are orthonormal matrices and Σ is a diagonal matrix (Dongarra *et al.*, 1979). Since orthonormal matrices and their inverse do not affect the l_2 norm, the norm of P and P^{-1} can be expressed as

$$\|P\|_2 = \|\Sigma\|_2$$
$$\|P^{-1}\|_2 = \|\Sigma^{-1}\|_2$$

with the condition number expressed as

$$\begin{aligned}\kappa_2(P) &= \|\Sigma\|_2\|\Sigma^{-1}\|_2 \\ &= \kappa_2(\Sigma)\end{aligned}$$

Another decomposition, although seldom employed in the identification literature but preferred in the numerical analysis literature (see Dongarra *et al.*, 1979), is

$$A = U_A\Sigma_A V_A^T$$

The matrix P is expressed as a function of this matrix A (e.g. Φ for least squares) as

$$\begin{aligned}P &= (A^TA)^{-1} \\ &= (V_A\Sigma_A U_A^T U_A\Sigma_A V_A^T)^{-1}\end{aligned}$$

Since U_A is an orthonormal matrix U^TU is equal to the identity matrix, this reduces the expression for the matrix P to

$$\begin{aligned}P &= (V_A\Sigma_A^2 V_A^T)^{-1} \\ &= V_A\Sigma_A^{-2} V_A^T\end{aligned}$$

The norm of P and P^{-1} can be expressed as

$$\|P\|_2 = \|\Sigma_A^{-2}\|_2$$
$$\|P^{-1}\|_2 = \|\Sigma_A^2\|_2$$

Since Σ_A is diagonal, the square can be taken out of the norm and the condition number of P can be expressed as

$$\begin{aligned}\kappa_2(P) &= \|\Sigma_A\|_2^2 \|\Sigma_A^{-1}\|_2^2 \\ &= (\kappa_2(\Sigma_A))^2\end{aligned}$$

The singular value decomposition of A has the same numerical conditioning advantage as the square root decomposition of P with the added benefit of dealing with the condition of a diagonal matrix as opposed to a (normally) triangular matrix.

Unfortunately, SVD is partly iterative. This implies that the computational load is large and that it cannot easily be implemented in a recursive form. Since SVD does not offer better numerical conditioning than square root factorization but simply concentrates the conditioning in a diagonal matrix, the square root factorizations are usually preferred.

7.5.4 QR factorization

In practice, the orthogonal QR factorization is computationally cheaper and frequently is just as reliable as SVD unless one explicitly needs V (Klema and Laub, 1980). The mechanics of the QR factorization are explained in Appendix D. The method corresponds to solving an overdetermined set of linear equations expressed as

$$Ax = b$$

where A is a known matrix (Φ for least squares); x is a vector of unknowns to be determined (θ for least squares);

and b a known vector (Y for least squares). The matrix A can be triangularized by the orthogonal transformation matrix Q giving

$$QAx = Qb$$
$$\begin{bmatrix} Rx \\ 0 \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$$

with the solution for x being $R^{-1}\eta_1$ (usually performed through backsubstitution). It should be noted that this matrix is different from the square root factor matrix Q presented previously, the symbolism used in this section is conform to the symbolism used in the literature. If the matrix P is defined as $(A^TA)^{-1}$, since $Q^TQ = QQ^T = I$, it can be written as

$$\begin{aligned} P &= (A^TQ^TQA)^{-1} \\ &= ((QA)^T(QA))^{-1} \\ &= (R^TR)^{-1} \end{aligned}$$

The QR decomposition can therefore be interpreted as a square root factorization of P^{-1} i.e. the condition number is

$$\begin{aligned} \kappa_2(P^{-1}) &= \kappa_2(P) \\ &= (\kappa_2(R))^2 \end{aligned}$$

The similarity between factorizing P^{-1} through the QR factorization, giving R, and factorizing P through the square root factorization, giving Q, can also be observed from the fact that the recursive versions of the QR factorization and the square root factorization share the same update algorithm (see Dongarra *et al.*, 1979) so $\kappa_2(R)$ is the same as $\kappa_2(Q)$. The recursive update of R has the benefit of being able to keep track of the sum of squares of residuals which is necessary for the determination of confidence bands on the parameters.

For singular systems, some parameters may be set to zero during the backsubstitution. This cannot be achieved with the factorization of P. The recursive QR factorization may be started with R = 0. This implies that the recursive solution for the QR factorization is exactly the same as for the batch solution without any bias ($P_0^{-1}=0$). The square root factorization of P does require some initial nonsingular Q anJ Q would need to be infinity to produce exactly the same results as in the batch case.

It should be noted that since R and the square root factor Q are triangular, singularity may be checked by inspection of the diagonal elements. If one or more elements of the diagonal is zero (near zero in practice) then the system is singular.

7.6 Illustration of the effect of a reconstructed P matrix and U-D factorization on identification using a simulated second order system

The recursive least squares (RLS), reconstructed P matrix recursive least squares (RPRLS) and RLS with U-D factorization (RUD) algorithms are compared to observe the effect of numerical enhancement strategies on the condition number of the parameter correlation matrix (a scaled P matrix in which the main diagonal is unity). In the RPRLS algorithm the matrix P is decomposed using the Cholesky decomposition and then recomposed at each recursion to ensure symmetry (Sripada and Fisher, 1987). For this example the second order system presented in Section 7.8 and sampled at 0.2 time units was used. The initial covariance matrix was set to 1000I. The simulated input and output signals for the second order system are presented in Figures 7.2 and 7.3.

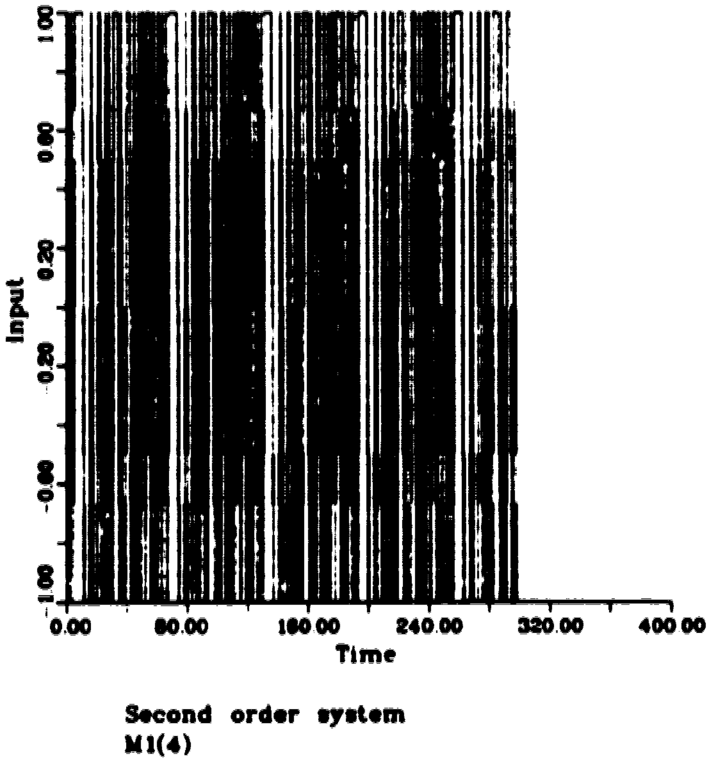


Fig. 7.2 Input signal used for simulation of the second order system

The parameter estimates for T_s=0.2 using RLS are

$$a_1 = - 1.75 \pm 0.05$$

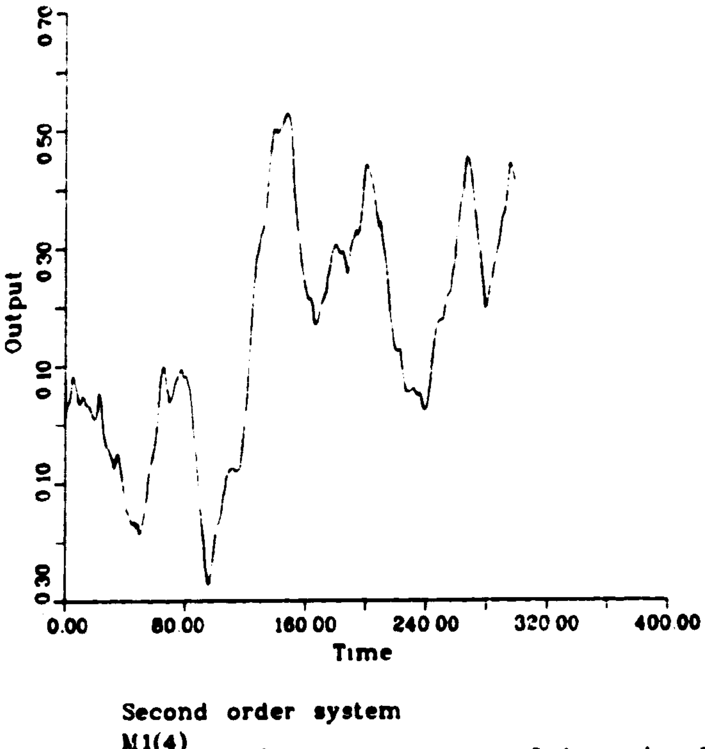


Fig. 7.3 Output signal generated by simulation of the second order system

$a_2 = 0.75 \pm 0.05$

$b_1 = 0.0025 \pm 0.0009$

$b_2 = 0.0025 \pm 0.0009$

The parameter estimates using RPRLS and RUD were the same as those obtained with RLS. The parameter correlation matrix of RPRLS is almost identical to that of RLS with variations in the fourth digit only. There is therefore minimal advantage to ensuring symmetry without propagating the square root. Rounding at three digits, the lower triangular part of the parameter correlation matrix obtained using the RLS and RPRLS algorithms was found to be

a_1	1.000			
a_2	-0.997	1.000		
b_1	0.049	-0.054	1.000	
b_2	0.173	-0.177	0.019	1.000

The condition number for this parameter correlation matrix is 786. Use of the RUD algorithm resulted in the following parameter correlation matrix

a_1	1.000			
a_2	-0.990	1.000		
b_1	0.046	-0.052	1.000	
b_2	0.175	-0.180	0.019	1.000

The condition number of this matrix is 205. The numerical accuracy of this simulation has therefore been significantly improved by the proper use of a factorization method.

7.7 Examination of the condition number of a first order plus fractional time delay system

The results of this section can be summarized by the following theorem:

Theorem 1: Under the following assumptions (1-4), the P matrix of the least squares algorithm for a discretized first order system will exhibit a condition number less than 9.

Assumption 1: Discretization is obtained using a zero order hold

Assumption 2: The mean of the input and output signals is zero and their variance is unity

Assumption 3: The sampling period is less than $\frac{1}{2}\tau$, where τ is the time constant

Assumption 4: u_t is uncorrelated with past u's

Proof: The first order system can be described as

$$\frac{K\ e^{-\{(n+\Delta)Ts\}s}}{\tau s + 1}$$

where Ts is the sampling period, n is the integer part of the time delay (multiple of the sampling period), Δ is the fractional part of the time delay (i.e. a fraction of the sampling period), K is the steady state gain and τ is the time constant. Under Assumption 1, the discretized model is described by

$$y_t = -a_1 y_{t-1} + b_1 u_{t-1-n} + b_2 u_{t-2-n} + \epsilon_t \tag{7.11}$$

where

$$a_1 = -\ e^{-Ts/\tau}$$

$$b_1 = K(1 - e^{-mTs/\tau})$$

$$b_2 = K(e^{-mTs/\tau} - e^{-Ts/\tau})$$

$$m = 1 - \Delta$$

n = discretized time delay

T_s = sampling period

ϵ_t = model residuals (noise)

Squaring both sides of Equation 7.11, using Assumption 4 and taking expectations gives

$$E[y_t^2] = a_1^2E[y_{t-1}^2] + b_1^2E[u_{t-1-n}^2] + b_2^2E[u_{t-2-n}^2] -$$
$$2a_1b_1b_2E[u_{t-2-n}^2] + \sigma^2$$

where σ^2 is the variance of the noise. Use of Assumption 2 leads to

$$1 = a_1^2 + b_1^2 + b_2^2 - 2a_1b_1b_2 + \sigma^2 \qquad (7.12)$$

Noting that the sign of b_1 is the same as the sign of b_2, all terms on the right hand side of Equation 7.12 are positive hence

$$1 > a_1^2 + b_1^2$$

and under Assumption 3

$$|a_1| > 0.6$$

therefore $b_1^2 < 0.64$ or equivalently $|b_1| < 0.8$. For this model the inverse of the P matrix is

$$P^{-1} = \begin{bmatrix} \Sigma y_{t-1}^2 & -\Sigma y_{t-1}u_{t-1-n} & -\Sigma y_{t-1}u_{t-2-n} \\ -\Sigma y_{t-1}u_{t-1-n} & \Sigma u_{t-1-n}^2 & \Sigma u_{t-1-n}u_{t-2-n} \\ -\Sigma y_{t-1}u_{t-2-n} & \Sigma u_{t-1-n}u_{t-2-n} & \Sigma u_{t-2-n}^2 \end{bmatrix}$$

Since u_{t-1-n} and u_{t-2-n} are assumed to be uncorrelated and

$$E[y_{t-1}^2] = E[u_{t-1-n}^2] = E[u_{t-2-n}^2] = 1$$

under Assumption 2, the expected value of P^{-1} is given by

$$E[P^{-1}] = N \begin{bmatrix} 1 & 0 & -b_1 \\ 0 & 1 & 0 \\ -b_1 & 0 & 1 \end{bmatrix}$$

where N is the total number of points. From this expression the condition number is found to be

$$\kappa_2(E[P^{-1}]) = \frac{1 + |b_1|}{1 - |b_1|} \qquad (7.13)$$

The values of the condition number as a function of b_1 are the same as those presented in Table 7.1 considering α as the b_1 value. It follows that for $|b_1| < 0.8$ the condition number will be less than 9. This completes the proof.

Corollary: Including a constant parameter in the discrete model will not significantly increase the condition number since under Assumption 2 the mean of the input and output signal is zero. This can be seen by inspection since introducing the constant parameter will introduce a row and column of zeros except for the diagonal element which has the same expected value as the other diagonal elements (i.e. 1).

Remark 1: The resulting effect of violating Assumption 4 on the maximum condition number is undetermined i.e. case specific for correlations of less than 0.8 between u_{t-1} and u_{t-2} but for values larger than 0.8 the condition number increases approximately exponentially as the correlation increases. The reason for this is that cross-terms must be added to Equation 7.12 which reduces the maximum possible value of b_1. The correlation also appears as extra off-diagonal terms in $E[P^{-1}]$. Until the correlation between u_{t-1} and u_{t-2} is larger than the maximum value of b_1 previously obtained (i.e. 0.8) the net effect on the condition number is uncertain. Beyond this value the cross-correlation term dominates the off-diagonal terms and an approximately exponential increase can be expected.

Remark 2: The implication of Remark 1 is that if u_t is a step (high correlation) then the numerical accuracy will be lower than that of a square wave input (medium correlation) or that of a PRBS (negligible correlation).

Remark 3: If the fractional part of the delay is zero, the b_2 parameter will be zero and the P matrix may be reduced to a 2 x 2 matrix which will have a condition number of unity under Assumption 2. Thecrem 1 is still valid if a 3 x 3 P matrix is retained.

Remark 4: Assumption 2 implies that the input and output signals have been suitably coded i.e. such that their mean is zero and their variance is unity. The effect of violating this assumption limits the applicability of the use of the condition number as a measure of numerical accuracy (effect of diagonal elements preventing generalizations using the condition number) and can decrease the numerical accuracy of the calculation of θ e.g. in the computation of $\Psi^T Y$ (due the possibility of adding small and large numbers).

Remark 5: The integer part of the delay does not influence the condition number of the P matrix for a first order system.

Remark 6: Assumption 3 is not restrictive as data are sampled faster than twice per time constant to satisfy the

Nyquist (Shannon) sampling frequency theorem (In most cases $\frac{\tau}{20} \le T_s < \frac{\tau}{2}$).

Remark 7: In practice, the first order plus time delay model is probably the most widely used model for modelling chemical processes (Stephanopoulos, 1984) as the first order plus time delay model is adequate to approximate the dynamic behavior of most high order systems. This emphasizes the relevance of Theorem 1.

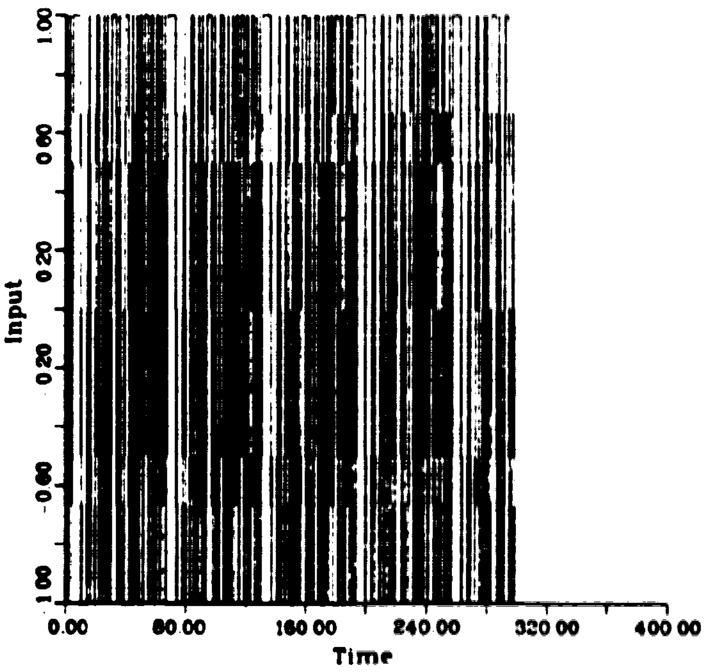
To illustrate the results of Theorem 1, a first order system with

$$m = 0.5$$
$$K = 2.97$$
$$T_s/\tau = 0.25$$

giving the discretized model

$$a_1 = -0.7788$$
$$b_1 = 0.3490$$
$$b_2 = 0.3080$$

was simulated using the PITSA program developed for this thesis (described in Appendix G). A noise generator seed number of 345 was employed to produce a noise variance of 0.01. These particular parameters were chosen so that scaling would not be required and that Equation 7.12 would apply. A PRBS sequence (described in Appendix A) of unity amplitude using a maximum length sequence of order 6 was selected as the input signal and used to generate 300 input/output data points. The input and output signals are depicted in Figures 7.4 and 7.5. The sample mean of the input is 0.0 and the standard deviation is 1.00. The sample mean of the output is 0.05 and the standard deviation is 0.97 so the system is well scaled and coding is not required.



First order system
M1(4)

Fig. 7.4 Input signal used for simulation of the first order system

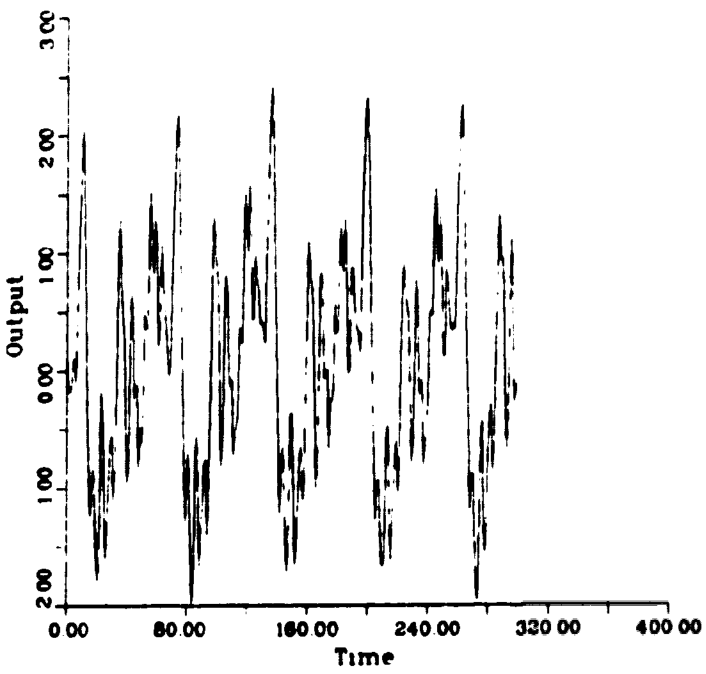
The following parameter estimates were obtained using the nonrecursive least squares algorithm in the PITSA program

$$a_1 = -0.782 \pm 0.006$$

$$b_1 = 0.355 \pm 0.006$$

$$b_2 = 0.314 \pm 0.006$$

The parameter correlation matrix is a scaled form of the P matrix where the scaling is chosen to provide unity diagonal



First order system
M1(4)

Fig. 7.5 Output signal generated by simulation of the first order system

elements. Although the condition number of the parameter correlation matrix is not necessarily the same as the condition number of the P matrix, the scaling can be neglected in this case as the diagonal elements of the P matrix are of the same magnitude through the appropriate choice of model parameters. The lower triangular portion of the parameter correlation matrix provided by the PITSA program was

$$\begin{array}{llll} a_1 & 1.000 & & \\ b_1 & -0.039 & 1.000 & \\ b_2 & 0.331 & -0.007 & 1.000 \end{array}$$

For this example, the theoretical value of the P matrix can be expressed as

$$P' = E[P^{-1}]^{-1} = \frac{1}{N}\begin{bmatrix} \frac{1}{1-b_1^2} & 0 & \frac{b}{1-b_1^2} \\ 0 & 1 & 0 \\ \frac{b_1}{1-b_1^2} & 0 & \frac{1}{1-b_1^2} \end{bmatrix}$$

so the lower triangular part oı the theoretical parameter correlation matrix is

a_1	1.000		
b_1	0.000	1.000	
b_2	0.349	0.000	1.000

It can be seen that the values of the parameter correlation matrix elements found by simulation using the PITSA program are in close agreement with the theoretical values. The condition number of the parameter correlation matrix, obtained by simulation, has the very low value of 2.

7.8 Examination of the condition number of a system of order larger than unity

The results of this section can be summarized by the following theorem and remarks:

Theorem 2: **A discretized system of order larger than unity may exhibit a P matrix with a large condition number even under Assumptions 1-4 (where, for a system of order larger than unity, τ in Assumption 3 may be interpreted as the dominant time constant of the system).**

Proof: **The possible existence of a large condition number** follows from the fact that for high order systems the P^{-1} matrix involves the off-diagonal element $\Sigma\, y_{t-1}y_{t-2}$ which increases as T_s decreases i.e. for higher order systems the condition number is a function of the sampling period. As shown in Section 7.3 an increase in the magnitude of the off-diagonal element will lead to a larger condition number. As shown in Section 7.4 this increase of the condition number cannot be prevented by any means other than reducing the magnitude of the off-diagonal element thus the dependance of the condition number on the sampling period cannot be avoided or controlled.

The increase of the magnitude of the off-diagonal element as the sampling frequency increases can be shown as follows:

The derivative of the output at time t can be approximated by

$$\frac{dy_t}{dt} \approx \frac{y_t - y_{t-1}}{T_s}$$

multiplying both sides by y_t, summing for $t = 1$ to N, and rearanging gives

$$\Sigma\, y_t y_{t-1} \approx \Sigma\, y_t^2 - T_s \Sigma\, y_t \frac{dy_t}{dt}$$

From this equation it can be seen that as T_s decreases $\Sigma\, y_t y_{t-1}$ (or equivalently $\Sigma\, y_{t-1}y_{t-2}$) approaches $\Sigma\, y_t^2$. As $\Sigma\, y_{t-1}y_{t-2}$ approaches $\Sigma\, y_t^2$ the condition number increases at an approximately exponential rate (c.f. Section 7.3). Therefore large condition numbers, even under Asumptions 1-4 are possible. This concludes the proof.

Remark 8: The condition number of a high order system increases at an approximately exponential rate as the sampling period decreases since y_{t-1} and y_{t-2} exhibit stronger correlation ($\Sigma\, y_{t-1}y_{t-2}$ approaches $\Sigma\, y_t^2$).

Remark 9: Theorem 2 stresses the fact that good coding is not enough to cope with ill-conditioned (high order) systems and that numerical methods, such as those presented in Section 7.5 are still required. However numerical methods cannot reduce the condition number as effectively as is possible by adequate coding for ill-conditioning due to an undesirable ratio of the magnitude of the matrix elements so both coding and factorization are required for high order systems.

Remark 10: If $\frac{dy_t}{dt} = 0$ for the entire identification time period then $\Sigma\, y_{t-1}y_{t-2} = \Sigma\, y_t^2$ and the system is singular, as expected. If the input is a step then $\Sigma\, y_t \frac{dy}{dt}$ is relatively small thus $\Sigma\, y_{t-1}y_{t-2}$ is close to $\Sigma\, y_t^2$ and the system is ill-conditioned. This is in addition to the effect observed in Remark 2. The effect of the input signal is further explored in Appendix A where the topic of the choice of input signal is addressed.

Remark 11: The importance of the existence of off-diagonal elements, especially for high order systems, does not appear to have been appreciated in the recent literature. This is typified by the fact that Sripada and Fisher, 1987 suggest that the condition number of the P matrix can be used as a measure of persistency of excitation. This is not the case since the condition number depends on the sampling period for high order systems. For example, identification using a PRBS signal, which is known to be persistently exciting, will produce a large condition number for a second order system with high frequency sampling (c.f. Remark 8) as shown in Table 7.2.

Remark 12: The correlation between parameters of a second order system may also be explained by the following. The continuous second order system has three degrees of freedom (K, τ_1, τ_2) but its discrete representation has four degrees of freedom (a_1, a_2, b_1, b_2). A change in the sampling period will change the values of the parameters of the discrete model representation of the system but obviously does not affect the actual dynamics of the system other than for the filtering effect on the signals that takes place due to the Shannon frequency sampling rule (Appendix A). The sampling period therefore cannot be considered as an extra degree of freedom of the continuous system. The discrepancy in the number of degrees of freedom appears as correlation between parameters.

To illustrate Theorem 2 and Remark 11 the following second order system is used

$$\frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

Discretization with a zero order hold, using the MATLAB function S2Z given in Appendix F, for K=1.0, τ_1=10.0, τ_2=1.0 yields the model parameters shown in Table 7.2. The corresponding condition number of the parameter correlation matrix obtained from the nonrecursive least squares identification using the PITSA program (with coding to give zero mean and unity variance) is also given in Table 7.2. For this simulation the input signal was the same PRBS sequence used for identifying the first order system with a noise variance of 10^{-4}.

TABLE 7.2

Parameters and condition number as a function of the sampling period

	T_s=0.01	T_s=0.1	T_s=0.2	T_s=1.0	T_s=10.0
a_1	-1.9891	-1.8949	-1.7989	-1.2727	-0.3679
a_2	0.9891	0.8958	0.8025	0.3329	0.0
b_1	$0.4982x40^{-5}$	$0.4821x10^{-3}$	0.0019	0.0355	0.5913
b_2	$0.4963x10^{-5}$	$0.4648x10^{-3}$	0.0017	0.0247	0.0408
condition number	66692	3422	521	108	5225

The decrease in the condition number with increasing sampling period is explained by the fact that off-diagonal elements of the parameter correlation matrix are functions of the process parameters. The choice of the sampling period will therefore affect the condition number of the parameter correlation matrix. If the sampling period is too long (T_s=10.0), the condition number increases because dynamical information is lost leading to a degenerate system.

If the sampling period is chosen to be 0.2 (1/5th of the smaller time constant) the lower triangular part of the parameter correlation matrix using the least squares algorithm, with coded input and output values (c.f. Section 10.9), is calculated to be

$$\begin{array}{lrrrr} a_1 & 1.000 & & & \\ a_2 & -0.996 & 1.000 & & \\ b_1 & 0.046 & -0.054 & 1.000 & \\ b_2 & 0.172 & -0.178 & 0.022 & 1.000 \end{array}$$

This would suggest that a_2 be removed and a first order system with partial time delay be estimated. For an overdamped second order system it is therefore numerically preferable to identify a first order system with partial time delay, perhaps using a longer sampling period and removing the unmodelled dynamics through a low pass filter, than estimating four parameters. This confirms the heuristic rule used by practitioners which is to remove "a" parameters rather than "b" parameters when reducing the model order.

7.9 Conclusion

In this chapter two new theorems were presented stating that a first order system with uncorrelated input signal and a sampling period selected to obey the Shannon sampling theorem can always be made well conditioned by ensuring that the input and output mean and variance are zero and unity respectively. This is not necessarily true for a system of order larger than unity as correlation between elements of the regressor is usually present. These correlations depend on the model parameters which themselves depend on the sampling period. The implication is that the condition number of the P matrix will grow approximately exponentially as the sampling period is decreased. This prevents the use of the condition number as a measure of persistency of excitation such as proposed by Sripada and Fisher, 1987.

These results also imply that both factorization and coding are necessary to improve numerical conditioning of high order systems.

It was observed that a step input produces a condition number larger than a square wave or PRBS input if there is more than one b parameter in the model. This is even more applicable to any high (greater than unity) order system since the choice of input signal will affect more off-diagonals terms.

Reconstructing the matrix P in the RPRLS algorithm (through Cholesky decomposition and then multiplication to ensure positiveness and symmetry) did not provide any significant improvements over the recursive least squares (RLS) algorithm for the simulated example. Improvements from reconstruction will occur only in cases of very ill-conditioned systems since the only advantage is that the matrix P is guaranteed to be positive and symmetric. Utilization of the RUD (UD factorization) algorithm did provide a significant reduction in the condition number over RLS and RPRLS. The parameter estimates are similar for the three algorithms used because the condition number of 786 obtained for RLS is not high enough to affect parameter estimates significantly for the low number of input/output data points used.

The condition number of the P matrix was found to have an approximately exponential dependance on the magnitude of the off-diagonal elements relative to magnitude of the diagonal elements. Also, a combination of several off-diagonal elements accelerates the increase in the condition number of a 3x3 matrix due to "interactions" created by combining more than one effect. Unfortunately results for a general matrix is not available but the trend should hold true for higher order matrices based on the examination of the upper bound of the condition number.

As shown, factorizing the P matrix was found useful in reducing the condition number. The square root factorizations were found to provide similar numerical enhancements as the SVD decomposition and require fewer computations. It was also demonstrated that the U-D factorization may be interpreted as a square root factorization in which no square root computations are required.

The results shown do not appear to have been expressly addressed in the published literature due to the focus on the perturbation analysis of the complete matrix rather than focusing on the effect of specific matrix elements.

8. Removing the Constant (Bias) Parameter to Reduce the Condition Number

8.1 Introduction

As shown in Chapters 4 and 5, the more common process identification algorithms involve a matrix inversion to produce the P matrix or its recursive equivalent. In Chapter 7 it was demonstrated that better numerical conditioning can be obtained by reducing the magnitude of the off-diagonal elements of the P matrix. In this chapter, the removal of the constant (bias) parameter to reduce the condition number of the P matrix is examined.

The motivation to remove the constant parameter is partly based on the following relationship for square symmetric matrices (Golub and Van Loan, 1983)

$$\lambda_{r+1}(A_{r+1}) \le \lambda_r(A_r) \le \lambda_r(A_{r+1}) \le \ldots \le \lambda_1(A_r) \le \lambda_1(A_{r+1})$$

where λ_i is the ith largest eigenvalue and A_r is the leading principal submatrix of rank r. If follows from the condition number for square symmetric matrices that

$$\kappa_2(A_r) = \frac{\lambda_1(A_r)}{\lambda_r(A_r)}$$

and

$$\kappa_2(A_{r+1}) = \frac{\lambda_1(A_{r+1})}{\lambda_{r+1}(A_{r+1})}$$

therefore

$$\kappa_2(A_{r+1}) \ge \kappa_2(A_r)$$

i.e. removing a row and a column in A may decrease the condition number.

This can be achieved by removing one parameter from the model. An obvious choice is to remove the constant parameter since it only contains steady state information.

Motivation to properly remove the constant parameter can also be illustrated using the least squares identification algorithm introduced in Chapter 4. The least squares model is (the delay, k, is omitted without loss of generality, c.f. Remark 5)

$$\begin{aligned} y_t = - a_1 y_{t-1} - a_2 y_{t-2} \ldots - a_n y_{t-n} + b_1 u_{t-1} + \\ b_2 u_{t-2} \ldots + b_m u_{t-m} + \beta + \epsilon_t \end{aligned} \qquad (8.1)$$

For N points, the inverse of the P matrix is

$$P^{-1} = \begin{bmatrix} \Sigma y^2_{t-1} & -\Sigma y_{t-1} y_{t-2} & \ldots & -\Sigma y_{t-1} u_{t-1} & -\Sigma y_{t-1} u_{t-2} & \ldots & -\Sigma y_{t-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\Sigma y_{t-1} u_{t-1} & -\Sigma y_{t-2} u_{t-1} & \ldots & \Sigma u^2_{t-1} & \Sigma u_{t-1} u_{t-2} & \ldots & \Sigma u_{t-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\Sigma y_{t-1} & -\Sigma y_{t-2} & \ldots & \Sigma u_{t-1} & \Sigma u_{t-2} & \ldots & \Sigma 1 \end{bmatrix}$$

where the summations are from 1 to N and where $\Sigma 1 = N$.

It was shown in Chapter 7 that off-diagonal elements have an approximately exponential effect on the condition number and that a combination of off-diagonal elements cannot reduce the condition number. To reduce the condition number of P^{-1} therefore requires that the number of nonzero off-diagonal elements be as small as possible. This can be achieved with

$$\Sigma y_{t-1} = \Sigma y_{t-2} = \ldots = \Sigma u_{t-1} = \Sigma u_{t-2} = \ldots = 0$$

which states that the sample mean of the input and output should be as close to zero as possible. To provide good numerical properties also requires that

$$\Sigma y_{t-1}^2 = \Sigma y_{t-2}^2 = \ldots = \Sigma u_{t-1}^2 = \Sigma u_{t-2}^2 = \ldots = \Sigma 1$$

which simply states that the variance of the input/output data should be unity.

Astrom, 1983 stated that "The least squares estimation is poorly conditioned for high signal to noise ratios. It is particularly harmful if estimation is based on signals which have a high superimposed dc level". The first sentence, which seems to be contrary to normal intuition, should be interpreted in the light of the second to mean that identification using the least squares is poorly conditioned if the data is nonzero mean.

Astrom, 1983 also stated that "all estimation methods are poorly conditioned if the models are overparametrized". This is only true for overparametrizations that create correlations between parameters since these correlations appear in the P matrix as off-diagonal elements.

From the preceding discussion it is clear that to numerically enhance computations the mean of the input and output should be removed. Similar results may be obtained by removing the constant term, β, by substitution. This has the added advantage of reducing the rank of the P matrix by one and this reduction will improve the numerical robustness by avoiding computation of quantities that are known to be zero as well as reducing computation time.

It should be noted that in the presentation that follows methods that introduce a significant bias in the parameter estimates are considered to be unacceptable. A bias is also generally obtained if the number of parameters of the model is reduced by simply excluding β from the model. Even though a parameter is removed when substitution is utilized, there is implicitly one more parameter than if β is simply ignored. Therefore the substitution method should be expected to be able to predict the output of the system more accurately than the exclusion method (i.e. ignoring β) although the latter may still provide reasonable predictions in some cases.

Several methods have been proposed to remove the constant term β (also known as the bias or DC value). The most common methods of removing the constant term are (Isermann, 1982)

i. deviation from the expected mean

ii. deviation from the sample mean (averaging)

iii. deviation from a single time equation (e.g. set point and predicted steady state values; or initial values)

iv. deviation from previous values (incremental)

v. deviation from filtered values (high pass filter)

These methods are a form of subtracting an equation or a set of equations from the equation at time t. Each single equation used must be a member of the set of linear equations describing the system at time t i.e. for time varying systems there is the added restriction that the equation or set of equations used must be close in time. Since these methods are used to eliminate β as a parameter, the following conditions must be met

$$\sum_{t=1}^{N} \tilde{y}_t = 0$$

$$\sum_{t=1}^{N} \tilde{u}_t = 0$$

where $\tilde{}$ denotes deviation variables.

8.2 Deviation from the expected mean

This is not a practical choice as the expected mean is unknown and the sample mean may not be the same as the expected mean for short sequences. For example

$$y_t = .9\ y_{t-1} + \epsilon_t \tag{8.2}$$

with $\epsilon_t \approx N(0,1)$, may have long excursions to one side of the expected mean (zero in this case). A simulation (termed a "realization" in the time series literature) of Equation 8.2 for 120 points is shown in Figure 8.1. For this example, $\sum_{t=1}^{N} \tilde{y}_t$ is not zero and the calculated mean for these 120 points is 0.14 so a bias, β, is present. The variance of the mean can be calculated using (Box and Jenkins, 1976)

$$VAR(\bar{y}_t) = \frac{\sigma^2}{n(1-a_1)^2}$$

The variance of the noise was unity so that the variance of the mean is 0.833 or a standard deviation of 0.913. This deviation is large enough to produce a significant bias. It should also be noted that using a forgetting factor will not help as the window is shifted in time so that the bias will oscillate about zero. Since the window usually covers a short time duration the bias will likely be larger than if all the points were used without a forgetting factor.

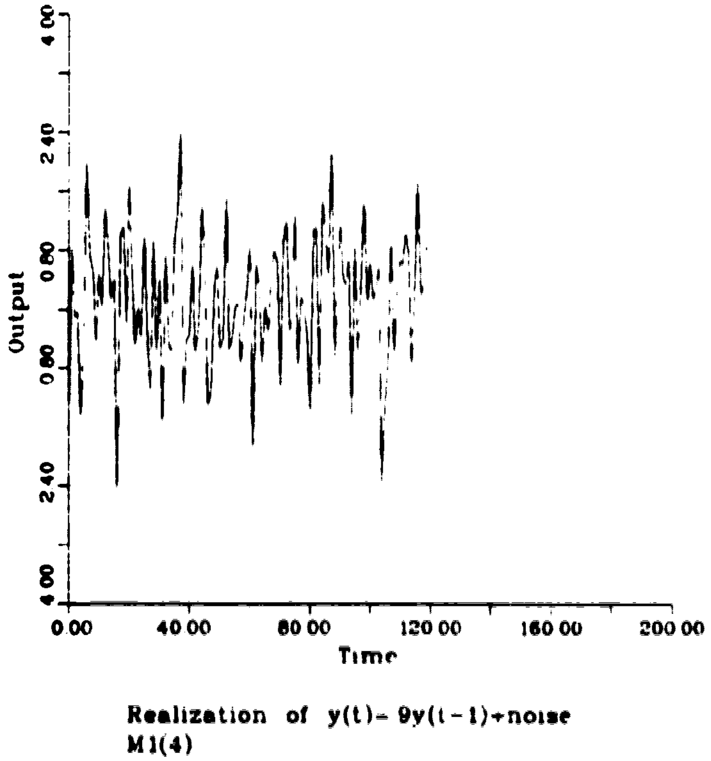


Fig. 8.1 Output y_t versus time for the system $y_t = .9y_{t-1} + \epsilon_t$

8.3 Deviation from the sample mean

Defining the sample mean as

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_{t-i} \qquad (8.3)$$

$$\bar{u} = \frac{1}{N}\sum_{i=1}^{N} u_{t-1-i} \qquad (8.4)$$

where the unity delay in $\bar{u}$ accounts for the fact that the current control action is not available (estimation occurs before control action). Summing Equation 8.1 from t=1 to N and dividing N gives

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N} y_{t-i} = &- a_1\frac{1}{N}\sum_{i=1}^{N} y_{t-1-i} - a_2\frac{1}{N}\sum_{i=1}^{N} y_{t-2-i} \ldots\\ &- a_n\frac{1}{N}\sum_{i=1}^{N} y_{t-n-i} + b_1\frac{1}{N}\sum_{i=1}^{N} u_{t-1-i}\\ &+ b_2\frac{1}{N}\sum_{i=1}^{N} u_{t-2-i} \ldots + b_m\frac{1}{N}\sum_{i=1}^{N} u_{t-m-i}\\ &+ \frac{1}{N}\sum_{i=1}^{N} \epsilon_{t-i} + \beta \qquad (8.5'\end{aligned}$$

Subtracting Equation 8.5 from Equation 8.1 and using

Equations 8.3 and 8.4 gives

$$\begin{aligned} y_t - \bar{y} = & - a_1(y_{t-1} - \bar{y} + \frac{1}{N}(y_{t-1} - y_{t-N-1})) - a_2(y_{t-2} - \bar{y} \\ & + \frac{1}{N}(y_{t-1} + y_{t-2} - y_{t-N-1} - y_{t-N-2})) \ldots - a_n(y_{t-n} - \bar{y} \\ & + \frac{1}{N} \sum_{i=1}^{n} (y_{t-i} - y_{t-N-i})) + b_1(u_{t-1} - \bar{u}) + b_2(u_{t-2} - \bar{u} \\ & + \frac{1}{N}(u_{t-2} - u_{t-N-2})) \ldots + b_m(u_{t-m} - \bar{u} \\ & + \frac{1}{N} \sum_{i=2}^{m} (u_{t-i} - u_{t-N-i})) + \epsilon_t - \frac{1}{N} \sum_{i=1}^{N} \epsilon_{t-i} \end{aligned}$$

or in terms of deviation variables

$$\begin{aligned} \tilde{y}_t = & - a_1\tilde{y}_{t-1} - a_2\tilde{y}_{t-2} \ldots - a_n\tilde{y}_{t-n} + b_1\tilde{u}_{t-1} + b_2\tilde{u}_{t-2} \\ & \ldots + b_m\tilde{u}_{t-m} - \frac{1}{N} \sum_{j=1}^{n} a_j \sum_{i=1}^{j} (y_{t-i} - y_{t-N-i}) \\ & + \frac{1}{N} \sum_{j=2}^{m} b_j \sum_{i=2}^{j} (u_{t-i} - u_{t-N-i}) + \epsilon_t - \frac{1}{N} \sum_{i=1}^{N} \epsilon_{t-i} \end{aligned}$$

A bias still remains due to inappropriate sample mean values (summation terms on the right hand side) and colored noise. In both cases the bias will be reduced as N increases. It should be noted that if a large step change is introduced, y_{t-1} may be very different from y_{t-N-1} and this may lead to a large bias.

It is sufficient to ensure that the input and output signals be in the range of -1 to 1 (so that the summation grows slower than N) to prevent a large bias. This aspect will be examined later when coding is considered.

It should be noted that using

$$\bar{y} = \frac{1}{N+1} \sum_{i=0}^{N} y_{t-i}$$

instead of

$$\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_{t-1}$$

does not remove the bias, it only shifts the values used to calculate the bias. It should also be noted that the bias is seldom observed in simulations due to the cyclic nature of the input signals that are commonly used (PRBS, sine wave, square wave, sawtooth wave, etc.) and the fact that the number of points chosen, N, is usually a multiple of the number of points describing a single cycle. For N chosen in such a manner, the value of y_{t-i} is close or is equal to the value of y_{t-N-i} (the calculated values are in the same position in the cycle). Therefore the bias is usually not observed because of the type of input signal that was employed for the simulation.

Since a bias remains when removing the sample mean, a slight modification seems appropriate to at least remove one source of bias. This can be accomplished by using a running mean, defined as

$$\bar{y}_t = \frac{1}{N} \sum_{i=1}^{N} y_{t-i}$$
$$\bar{u}_t = \frac{1}{N} \sum_{i=1}^{N} u_{t-i}$$

so that

$$\bar{y}_{t-1} = \frac{1}{N} \sum_{i=1}^{N} y_{t-1-i}$$
$$\bar{u}_{t-1} = \frac{1}{N} \sum_{i=1}^{N} u_{t-1-i}$$

Subtracting Equation 8.5 from Equation 8.1 using the running means gives

$$\begin{aligned}(y_t-\bar{y}_t) = &- a_1(y_{t-1}-\bar{y}_{t-1}) - a_2(y_{t-2}-\bar{y}_{t-2}) \ldots \\ &- a_n(y_{t-n}-\bar{y}_{t-n}) + b_1(u_{t-1}-\bar{u}_{t-1}) + b_2(u_{t-2}-\bar{u}_{t-2}) \\ &\ldots + b_m(u_{t-m}-\bar{u}_{t-m}) + \epsilon_t - \frac{1}{N}\sum_{i=1}^{N} \epsilon_{t-i} \qquad (8.6)\end{aligned}$$

One source of bias remains, namely colored noise, but the bias will decrease as N increases with the bias eventually becoming negligible.

8.4 Deviation from a single time equation

Deviation from a single time equation means that an equation at a given reference time (e.g. steady state values or a specific equation fixed in time) is subtracted from the equation at the current time, t, in order to remove the bias term. It should be noted that if the parameters are time variant then the single time equation values cannot be used if those values are "too old" as they will not obey the current set of linear time invariant equations describing the system. (The parameters are assumed to be time invariant for the N points used in the least squares estimation.) Let the reference equation be

$$\begin{aligned} y_{\bullet} = & - a_1 y_{\bullet-1} - a_2 y_{\bullet-2} \ldots - a_n y_{\bullet-n} + b_1 u_{\bullet-1} \\ & + b_2 u_{\bullet-2} \ldots + b_m u_{\bullet-m} + \epsilon_{\bullet} + \beta \end{aligned} \qquad (8.7)$$

Subtracting Equation 8.7 from Equation 8.1 gives

$$\begin{aligned} y_t - y_{\bullet} = & - a_1 (y_{t-1} - y_{\bullet-1}) - a_2 (y_{t-2} - y_{\bullet-2}) \ldots \\ & - a_n (y_{t-n} - y_{\bullet-n}) + b_1 (u_{t-1} - u_{\bullet-1}) + b_2 (u_{t-2} - u_{\bullet-2}) \\ & \ldots + b_m (u_{t-m} - u_{\bullet-m}) + \epsilon_t - \epsilon_{\bullet} \end{aligned}$$

The bias in the parameters due to colored noise was demonstrated tc be (c.f. Chapter 4)

$$\tilde{\theta} = (\Phi^T \Phi)^{-1} \Phi^T E$$

where $\tilde{\theta}$ is the bias in the parameters. In general, for the bias to be zero requires that $\Phi^T E$ be zero. $\Phi^T E$ for deviation variables may be written as

$$\tilde{\Psi}^T\tilde{E} = \begin{bmatrix} \sum_{i=1}^{N} \tilde{y}_{t-1-i}\tilde{\epsilon}_{t-i} \\ \sum_{i=1}^{N} \tilde{y}_{t-2-i}\tilde{\epsilon}_{t-i} \\ \vdots \\ \sum_{i=1}^{N} \tilde{y}_{t-n-i}\tilde{\epsilon}_{t-i} \\ \sum_{i=1}^{N} \tilde{u}_{t-1-i}\tilde{\epsilon}_{t-i} \\ \sum_{i=1}^{N} \tilde{u}_{t-2-i}\tilde{\epsilon}_{t-i} \\ \vdots \\ \sum_{i=1}^{N} \tilde{u}_{t-m-i}\tilde{\epsilon}_{t-i} \end{bmatrix}$$

Assuming $\tilde{u}_t$ is uncorrelated with $\tilde{\epsilon}_t$, y_o is uncorrelated with ϵ_t and using

$$\tilde{y}_{t-1} = y_{t-1} - y_o$$

$$\tilde{\epsilon}_t = \epsilon_t - \epsilon_o$$

the expected value of the first element of $\tilde{\Psi}^T\tilde{E}$ is

$$E[\tilde{y}_{t-1-i}\tilde{\epsilon}_{t-i}] = -E[\tilde{y}_{t-1-i}\epsilon_o]$$

$$E[\tilde{y}_{t-1-i}\epsilon_o] = - a_1E[\tilde{y}_{t-2-i}\epsilon_o] - a_2E[\tilde{y}_{t-3-i}\epsilon_o] \ldots$$

$$- a_nE[\tilde{y}_{t-n-1-i}\epsilon_o] - \sigma_o^2$$

If ϵ_o is an old value, relative to the current time, this substitution can be carried one step further giving

$$E[\tilde{y}_{t-1-i}\epsilon_o] = - \sigma_o^2 + a_1\sigma_o^2 + a_1^2E[\tilde{y}_{t-3-i}\epsilon_o] + \ldots$$

The magnitude of the bias will thus depend on the plant parameters and on the amplitude of the noise. Since plant parameters and noise level are not "controllable", the bias cannot be reduced as for the sample mean.

It is to be noted that setting y_o to the set point and using the predicted steady state value for u_o also corresponds to a deviation from a single time equation. In this case, ϵ_o is still an error term but for future times giving

$$E[\tilde{y}_{t-1-1}\epsilon_e] = -\sigma_e^2$$

It should be noted that the bias is only a function of the noise level in this case.

8.5 Deviation from previous values (incremental)

Although a single time equation is removed, this approach differs from that considered in Section 8.4 in that the equation removed is changed in time rather than be fixed in time. The incremental approach may be generalized to the k-incremental approach where the k th removed equation is used instead of the preceding one. This is slightly more general than the k-incremental approach of Clarke *et al.*, 1983 where k is taken as the process delay. In the variable parameter case, attention should be given to the value of k to ensure that the t-k th equation is not so far removed from the current time as to be part of a different set of linear equations describing the system. The t-k th equation is

$$\begin{aligned} y_{t-k} = &- a_1y_{t-k-1} - a_2y_{t-k-2} \dots - a_ny_{t-k-n} \\ &+ b_1u_{t-k-1} + b_2u_{t-k-2} \dots + b_mu_{t-k-m} + \epsilon_{t-k} + \beta \end{aligned} \tag{8.8}$$

Subtracting Equation 8.8 from Equation 8.1 gives

$$\begin{aligned} y_t - y_{t-k} = &- a_1(y_{t-1} - y_{t-k-1}) - a_2(y_{t-2} - y_{t-k-2}) \dots \\ &- a_n(y_{t-n} - y_{t-k-n}) + b_1(u_{t-1} - u_{t-k-1}) \\ &+ b_2(u_{t-2} - u_{t-k-2}) \dots + b_m(u_{t-m} - u_{t-k-m}) + \epsilon_t \\ &- \epsilon_{t-k} \end{aligned}$$

Examining the first element in the bias vector as previously

$$E[\tilde{y}_{t-1-i}\tilde{\epsilon}_{t-i}] = - E[\tilde{y}_{t-1-i}\epsilon_{t-k-i}]$$

If k=1 then

$$E[\tilde{y}_{t-1-i}\epsilon_{t-1-i}] = \sigma^2$$

This value of k represents the usual incremental case. For this case there is a bias which is directly proportional to the variance of the noise. This explains why the incremental approach is sensitive to process noise (Vermeer, 1987). If k=2 then

$$\begin{aligned} E[\tilde{y}_{t-1-i}\epsilon_{t-2-i}] &= - a_1 E[\tilde{y}_{t-2-i}\epsilon_{t-2-i}] \\ &= - a_1\sigma^2 \end{aligned}$$

For a stable process $E[\tilde{y}_{t-1-i}\epsilon_{t-k-i}]$ will tend to zero as k becomes larger than the asymptotic memory length of the system (i.e. choose k such that the influence of ϵ_{t-k-i} on y_{t-1-i} is minimal). The last term involving the output in the bias vector is $\sum_{i=1}^{N} \tilde{y}_{t-n-i}\tilde{\epsilon}_{t-i}$ which will vanish as k becomes larger than the asymptotic memory length + the model order (i.e. n).

Since in the variable parameter case it is usually assumed that the parameters change much slower than the dynamics of the system using k = n + the asymptotic memory length should not result in using a reference equation that is a member of a different set of linear equations (which would introduce another form of bias).

8.6 Deviation from filtered values

This method is similar to using the running sample mean except that the mean, estimated through low pass filtering, is expressed as

$$\bar{y}_t = \lambda_t\bar{y}_{t-1} + (1-\lambda_t)y_t \tag{8.9}$$

$$\bar{u}_t = \lambda_t\bar{u}_{t-1} + (1-\lambda_t)u_t \tag{8.10}$$

where λ_t is the filter parameter which may be time varying.

The deviation variables obtained using this method can also be thought of as the output of a high pass filter. For example, the output deviation variable is

$$\dot{y}_t = y_t - \bar{y}_t$$

Using Equation 8.9 this can be written as

$$\dot{y}_t = y_t - \lambda_t\bar{y}_{t-1} - (1-\lambda_t)y_t$$

or

$$\dot{y}_t = \lambda_t y_t - \lambda_t\bar{y}_{t-1} \tag{8.11}$$

Using Equation 8.9 to substitute for $\bar{y}_{t-1}$ leads to

$$\dot{y}_t = \lambda_t y_t - \lambda_t(\lambda_{t-1}\bar{y}_{t-2} + (1-\lambda_{t-1})y_{t-1})$$

or

$$\dot{y}_t = \lambda_t y_t - \lambda_t(y_{t-1} - \lambda_{t-1}(y_{t-1}-\bar{y}_{t-2}))$$

Using Equation 8.11 at time t-1 to eliminate $\bar{y}_{t-2}$ gives

$$\dot{y}_t = \lambda_t(y_t - y_{t-1}) + \lambda_t\dot{y}_{t-1}$$

which can be written as

$$(1-\lambda_t q^{-1})\dot{y}_t = \lambda_t(1-q^{-1})y_t$$

This is the same expression as that of a high pass filter (e.g. see Isermann, 1980b). This high pass filter can also be thought of as a low pass filter operating on incremental data (Foley, 1988).

The use of a low pass filter to estimate the mean makes the comparison of the sets of equations tedious but this comparison can be approximated to the running mean method by a suitable choice of N, the sample length, in the running mean method. This approximation can be justified by considering $\bar{y}_t$ as the weighted least squares solution for the mean of y_t. The cost function to minimize for the estimation of the mean at time t based on N past data points is

$$J_t = \sum_{i=t-N}^{t} \prod_{j=i+1}^{t} \lambda_j \, (y_i - \bar{y}_t)^2$$

minimizing J_t with respect to $\bar{y}_t$ gives

$$-2\sum_{i=t-N}^{t} \prod_{j=i+1}^{t} \lambda_j \, (y_i - \bar{y}_t) = 0$$

$$\sum_{i=t-N}^{t} \prod_{j=i+1}^{t} \lambda_j \, \bar{y}_t = \sum_{i=t-N}^{t} \prod_{j=i+1}^{t} \lambda_j y_i$$

also for J_{t-1}

$$\sum_{i=t-N-1}^{t-1} \prod_{j=i+1}^{t-1} \lambda_j \, \bar{y}_t = \sum_{i=t-N-1}^{t-1} \prod_{j=i+1}^{t-1} \lambda_j y_i$$

If $\sum_{i=t-N}^{t} \prod_{j=i+1}^{t} \lambda_j = \sum_{i=t-N-1}^{t-1} \prod_{j=i+1}^{t-1} \lambda_j$ and defining this quantity as γ, then the following can be obtained

$$\bar{y}_t - \lambda_t \bar{y}_{t-1} = \frac{1}{\gamma}\left[y_t - \prod_{j=t-N}^{t} \lambda_j y_{t-N-1}\right]$$

For N sufficiently large, γ is equal to the asymptotic length and the second term in the bracket vanishes giving

$$\bar{y}_t = \lambda_t \bar{y}_{t-1} + \frac{1}{\gamma} y_t$$

Furthermore if $\lambda(\cdot)$ is constant or $\lambda(\cdot)$ has a small range of values the asymptotic length may be approximated by

$$\gamma = \frac{1}{1 - \lambda_t}$$

giving

$$\bar{y}_t = \lambda_t \bar{y}_{t-1} + (1-\lambda_t) y_t$$

which is the same expression as Equation 8.9.

Use of the filtering method can therefore be approximated by a running mean method in which the sample length is equal to the asymptotical length ($N = \gamma$).

Using the asymptotical length to approximate the filter, the filtered values can be expressed as

$$\bar{y}_t = \frac{1}{\gamma} \sum_{i=1}^{\gamma} y_{t-i}$$

$$\bar{u}_t = \frac{1}{\gamma} \sum_{i=1}^{\gamma} u_{t-i}$$

so the result of subtracting Equation 8.5 from Equation 8.1 using the filtered values may be approximated by Equation 8.6 with N replaced by γ.

As for the case where the sample mean is removed, one source of bias remains (colored noise). This bias will decrease as γ increases witn the bias eventually becoming negligible when γ is sufficiently large. This implies that the choice of the filter constant should be limited to values close to unity.

If a different filter constant is used for $\bar{y}_t$ and $\bar{u}_t$ then there will be a bias as the desired equation is now obtained by subtracting Equation 8.5 with $N=\gamma_y$ (the asymptotic length of the output filter) and subtracting Equation 8.5 with $N=\gamma_u$ (asymptotic length of the input filter) from Equation 8.1 giving

$$\begin{aligned}(y_t-\bar{y}_t) = &- a_1(y_{t-1}-\bar{y}_{t-1}) - a_2(y_{t-2}-\bar{y}_{t-2}) \ldots \\ &- a_n(y_{t-n}-\bar{y}_{t-n}) + b_1(u_{t-1}-\bar{u}_{t-1}) + b_2(u_{t-2}-\bar{u}_{t-2}) \\ &\ldots + b_m(u_{t-m}-\bar{u}_{t-m}) + \epsilon_t - \frac{1}{\gamma_y} \sum_{i=1}^{\gamma_y} \epsilon_{t-i} \\ &- \frac{1}{\gamma_u} \sum_{i=1}^{\gamma_u} \epsilon_{t-i} + a_1\bar{y}^{u}_{t-1} + a_2\bar{y}^{u}_{t-2} \ldots + a_n\bar{y}^{u}_{t-n}\end{aligned}$$

$$- b_1\bar{u}^y_{t-1} - b_2\bar{u}^y_{t-2} \ldots - b_m\bar{u}^y_{t-m} \qquad (8.12)$$

where $\bar{y}^u_t$ signifies $\bar{y}_t$ calculated with the input filter and $\bar{u}^y_t$ signifies $\bar{u}_t$ calculated with the output filter. As can be observed from Equation 8.12 use of different filtering constants introduces a bias in the parameter estimates (all the terms after ϵ_t).

If the system parameters change then it will take γ terms before all the values used to calculate the mean no longer obey the "old" equations. Then and only then will the estimation algorithm, with its own asymptotical length, start to "forget" the filtered quantities that are partly or totally calculated with the old equations. It is the sum of the two asymptotical lengths that determines when the effect of the values governed by the old equations on the estimated parameters is negligible. Before the effect of the old equations is totally removed there will be a bias in the parameters as the effective set of equations is incompatible (recall that the set of equations is assumed to have constant parameters).

8.7 Conclusion

To reduce the condition number of the P matrix the number of nonzero off-diagonal elements must be minimized. One method to achieve this is to remove the constant (bias) parameter by substitution of one or more equations. This has the added advantage of reducing the rank of the P matrix by one.

A result that has not been found in the published literature is that the sensitivity to noise of the k-incremental approach can be minimized if k is chosen to be larger than the model order plus the asymptotical memory length of the system.

Analysis of the use of different filters for the input and output signals as proposed by Sripada and Fisher, 1987 revealed that the use of different filters for the input and output signals introduces an unnecessary bias in the estimates and therefore should be avoided.

The use of the expected mean is not recommended as the expected mean can be significantly different from the sample mean. Use of a running mean was found to be better then the usual sample mean. Use of a fixed time equation generally produces a bias larger than the running mean since the effect of the variance cannot be attenuated through a larger number of data points as for the running mean. Deviation from filtered values was shown to be asymptotically equivalent to the running mean.

9. Scaling to Reduce the Condition Number

9.1 Introduction

For the purposes of this chapter, scaling is defined as an operation on a matrix to change the magnitude of its elements. Scaling of individual input and output data points before the matrix is formed will be termed coding to avoid any confusion. Coding is the topic of the next chapter.

It is a misconception that conditioning of the matrix P through the use of a scaling matrix will provide better numerical accuracy in the least squares identification. This is also reported by Dongarra *et al.*, 1979 where they state that "scaling of the form DAD [SPS in the following] will not change to any great extent the accuracy of solutions computed from positive definite routines". This stems from the fact that the system of equations that is solved is

$$\Phi^T\Phi\theta = \Phi^TY \tag{9.1}$$

and not

$$\Phi\theta = Y \tag{9.2}$$

Although Equations 9.1 and 9.2 appear to be of the same form, scaling Equation 9.2 will improve numerical conditioning while scaling Equation 9.1 will not. Equation 9.2 can be suitably scaled by premultiplication by the scaling matrix, S, on each side of the equation. If this operation was applied to Equation 9.1 to scale Φ^T then Φ on the left hand side cannot be scaled. Even postmultiplication would not help as this operation scales θ and not Φ.

Alternatively if the scaling matrix is designed to scale $\Phi^T\Phi$ then it is not suitable for Φ^T on the right hand side of the equation and no numerical advantage is gained. It is to be noted that using SS^{-1} between $\Phi^T\Phi$ and θ will not help either as poor scaling is only shifted and is therefore not adequately handled.

9.2 Detailed analysis of scaling

The fact that Equation 9.1 cannot be numerically improved by scaling can be demonstrated by analyzing the scaling proposed by Sripada and Fisher, 1987. In fact the proposed scaling method of Sripada and Fisher, 1987 leads only to more computations subject to (possibly) more errors. This can be demonstrated by comparing the computations involved in the update of $P_s(t)$ and $\theta_s(t)$ versus the update of $P(t)$ and $\theta(t)$. The notation is that of Sripada and Fisher, 1987 with $\theta_s(t)$ introduced to differentiate the update of the parameter vector with scaling, $\theta_s(t)$, and without scaling, $\theta(t)$.

9.2.1 P update

The scaled covariance matrix and the normalized and scaled regressor vector used for the update of the covariance matrix are (Sripada and Fisher, 1987)

$$P_s(t-1) = S(t-1)P(t-1)S(t-1) \qquad (9.3)$$

$$\phi_{ns}(t) = S(t-1)^{-1}\phi_n(t) \qquad (9.4)$$

Equation 9.4 represents the value of $\phi_{ns}(t)$ used in the update of $P_s(t)$ since $S(t)$ is not available yet. Dropping, for convenience, the time arguments and remembering that matrices P and S are available at time t-1 and λ and ϕ are available at time t allows the expression for updating the covariance matrix to be written as

$$P' = \left[I - \frac{P_s\phi_{ns}\phi_{ns}^T}{\lambda + \phi_{ns}^T P_s \phi_{ns}}\right]\frac{P_s}{\lambda}$$

$$= \frac{P_s}{\lambda} - \frac{1}{\lambda}\left[\frac{P_s\phi_{ns}\phi_{ns}^T P_s}{\lambda + \phi_{ns}^T P_s \phi_{ns}}\right]$$

noting that $(S^{-1})^T = S^{-1}$ and replacing P_s and ϕ_{ns} using Equations 9.3 and 9.4 gives

$$P' = \frac{SPS}{\lambda} - \frac{1}{\lambda}\left(\frac{SPSS^{-1}\phi_n\phi_n^T S^{-1}SPS}{\lambda + \phi_n^T S^{-1}SPSS^{-1}\phi_n}\right)$$

$$P' = S\left[\frac{P}{\lambda} - \frac{1}{\lambda}\left(\frac{P\phi_n\phi_n^T P}{\lambda + \phi_n^T P\phi_n}\right)\right]S$$

$$P' = S(t-1)P(t)S(t-1)$$

This demonstrates that the actual update of P(t) is numerically independent of scaling provided $SPSS^{-1}\phi_n\phi_n^T S^{-1}SPS$ and $\phi_n^T S^{-1}SPSS^{-1}\phi_n$ are *numerically* equivalent to $SP\phi_n\phi_n^T PS$ and $\phi_n^T P\phi_n$ respectively. Examining the ij th element of $SPSS^{-1}\phi_n\phi_n^T S^{-1}SPS$ which is the i th element of $SPSS^{-1}\phi_n$ multiplied by the j th element of $\phi_n^T S^{-1}SPS$ gives

$$\left[\sum_k S_i P_{ik} S_k \frac{1}{S_k}\phi_{n,k}\right] x$$

$$\left[\sum_l \phi_{n,l}\frac{1}{S_l}S_l P_{lj} S_j\right] = S_i \left[\sum_k P_{ik}\phi_{n,k}\right]\left[\sum_l \phi_{n,l}P_{lj}\right] S_j \quad (9.5)$$

It should be noted that numerical accuracy is independent of the order in which the multiplications are performed (except possibly for over and/or underflow) since there is no additive term within the summations on the left hand side. Numerical inaccuracy occurs most frequently when a small number is added to a large number or if several numbers of different magnitudes are added. Multiplication of a small number by a large number does not lead to large numerical inaccuracies, e.g. 1000 x 0.001 x 0.009 is 0.009 in 6-digit precision arithmetic regardless of the order in which the multiplications are performed if floating point arithmetic is used. However, the addition of 1000 + 0.001   0.009 is 1000.01 if added from right to left and 1000.00 if added from left to right.

Since the numerical accuracy on each side of Equation 9.5 is the same, $S^{r}SS^{-1}\phi_n\phi_n^T S^{-1}SPS$ can be considered numerically equivalent to $SP\phi_n\phi_n^T PS$.

Similarly, $\phi_n^T S^{-1}SPSS^{-1}\phi_n$ can be written as

$$\sum_i \sum_j \phi_{n,i}\frac{1}{S_i}S_i P_{ij} S_j \frac{1}{S_j}\phi_{n,j} = \sum_i \sum_j \phi_{n,i} P_{ij} \phi_{n,j}$$

so it follows that $\phi_{ns}^T P_s \phi_{ns}$ is numerically equivalent to $\phi_n^T P \phi_n$ showing that the update of P is not modified by scaling with a diagonal matrix.

9.2.2 $\hat{\theta}$ update

The algorithm given by Sripada and Fisher, 1987 for the update of the parameter estimates, $\hat{\theta}(t)$, can be stated as

$$P_s(t) = S(t)P(t)S(t)$$

$$\phi_{ns}(t) = S(t)^{-1}\phi_n(t)$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + S(t)^{-1}P_s(t)\phi_{ns}(t) \text{ X } \left[y_n(t)-\hat{\theta}(t-1)^T\phi_n(t)\right]$$

which can be rewritten as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \left[S(t)^{-1}S(t)P(t)S(t)S(t)^{-1}\phi_n(t)\right] \text{ X } \left[y_n(t)-\hat{\theta}(t-1)^T\phi_n(t)\right]$$

or as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\phi_n(t)\left[y_n(t)-\hat{\theta}(t-1)^T\phi_n(t)\right]$$

Again in this case scaling does not modify the calculations provided that $S(t)^{-1}P_s(t)\phi_{ns}(t)$ is numerically equivalent to $P(t)\phi_n(t)$. To see this, the calculation for the i th element of the vector $S(t)^{-1}S(t)P(t)S(t)S(t)^{-1}\phi_n(t)$ is

$$\sum_j \frac{1}{S_i} S_i P_{ij} S_j \frac{1}{S_j}\phi_{n,j} = \sum_j P_{ij}\phi_{n,j}$$

Therefore $S(t)^{-1}P_s(t)\phi_{ns}(t)$ is numerically equivalent to $P(t)\phi_n(t)$.

Again, the reason behind this equivalence is that for multiplicative terms, as in this case, the order in which the multiplications are performed is not important except possibly to avoid under and overflow. The order in which the operations are performed becomes important if additive and multiplicative terms are involved e.g. if S was full or triangular instead of being a diagonal matrix.

9.3 Illustration of the effect of scaling on identification

Since in the scaling algorithm (called SRLS in PITSA) the matrix P is decomposed using the Cholesky decomposition and recomposed (Sripada and Fisher, 1987) it cannot be compared directly to RLS. A version of RLS in which the matrix P is reconstructed but not scaled (termed RPRLS) permits a direct comparison with SRLS. It was found that both SRLS and RPRLS gave the same results to the working single precision of the Amdahl 5870 computer for the simulation of the first and second order systems presented in Chapter 7 thus validating the analysis of scaling.

9.4 Conclusion

It was shown that the original system of equations (Equation 9.2) can advantageously be scaled but that the normal equations (Equation 9.1) cannot be adequately be scaled using a scaling matrix. In fact scaling the normal equations only introduced more computations without any numerical benefit. This was demonstrated by a new detailed analysis of the scaling proposed by Sripada and Fisher, 1987. The ineffectiveness of scaling was also observed in simulations.

10. Coding to Reduce the Condition Number

10.1 Introduction

Klema and Laub, 1980 suggest that "columns of A [their regressor matrix] be equilibrated such that the sum of their elements be as nearly equal as possible. Exact powers of 16 for IBM 360/370 machines should be used as scaling factors so that the data is not perturbed in trailing digits. Row scaling will have the effect of introducing weights on the data if one has a least squares problem and therefore should be done at a user's discretion". In the context of process identification, column scaling ensures that the individual input and output values are of the same magnitude. In this work column scaling of the regressor is termed coding to differentiate with matrix scaling discussed in the previous chapter.

Coding is a somewhat arbitrary transformation of the variables performed before the variables are used for process identification. A well known example is that of Series J of Box and Jenkins, 1976 for which the input signal is coded through the transformation $X'=0.6-0.04X$. The purpose of coding is to scale the input and output variables usually in such a way as to reduce the condition number of the P matrix and therefore enhance the accuracy of the computations. Since coding is performed before the P matrix is formed, it is a simple and effective method of scaling the entries of the matrix. The most common types of coding produce dimensionless variables that are also often centered (e.g. with a value such as the mean). A dimensionless variable is obtained by dividing the problem (physical) variable by a characteristic value. Table 10.1 gives examples of characteristic values and the resulting transformations.

TABLE 10.1

Some typical coding characteristic values and their corresponding transformations

Characteristic values	Transformation
median, $z_{.5}$	$z' = \frac{z}{z_{.5}}$
range, centered on reference value, z_o (e.g. steady state or mean)	$z' = \frac{z - z_o}{z_{max} - z_{min}}$
range, one-sided and bounded by k	$z' = \frac{z - z_{min}}{z_{max} - z_{min}} \times k$
range, centered on median, $z_{.5}$	$z' = \frac{z - z_{.5}}{z_{max} - z_{min}} \times k$
standard deviation (statistical normalization)	$z' = \frac{z - \bar{z}}{\sigma}$
(non-zero) reference value, z_o	$z' = \frac{z}{z_o}$

Since these methods do not attempt to remove the bias, the constant parameter should be preserved in the model. The transformation that will provide the lowest condition number is the transformation that produces $\Sigma z'_t$ closest to zero and $\frac{1}{N}\Sigma(z'_t)^2$ closest to unity where z' denotes the coded variable (c.f. Section 8.1).

Once the parameters have been estimated, the model must be used with coded variables. Output prediction is obtained by prediction of the coded output and decoding the coded output using the inverse transformation.

For recursive identification, changing the coding during identification should not be attempted as changing the coding will produce an incompatible set of equations. For example, assume that the regressor is composed of $-y_{t-1}$, u_{t-1} and unity so that there are three parameters, $\theta_1=a_1$, $\theta_2=b_1$ and $\theta_3=\beta$. After a coding change by a factor α, the regressor becomes $-y_{t-1}$, αu_{t-1} and unity. The three estimated parameters based on this set of equations would then be $\theta_1=a_1$, $\theta_2=\frac{1}{\alpha}b_1$ and $\theta_3=\beta$. The estimated parameters are different than in the first case because the underlying equations are incompatible due to a change in coding.

The distinctive features of the different transformations presented in Table 10.1 are now examined.

10.2 Median

This transformation cannot be used for a signal that is symmetric and centered at zero since $z_{.5}=0$ for this case. If the original values are not centered at zero, then $\Sigma z'$ will increase with N, the number of points. This transformation is therefore not recommended.

10.3 Range, centered on a reference value

Assuming that z_{max} and z_{min} are constant and examining the expectation rather than the summation gives

$$E[z'] = \frac{1}{z_{max}-z_{min}}(E[z]-E[z_o])$$

In this case, the better numerical conditioning will occur when

$$E[z] = E[z_o]$$

Choosing z_o as the mean or the steady state value in a controlled environment will yield good results. It should be noted that the steady state value may not be changed during the identification period as this would change the coding (c.f. earlier comments). If a change in reference is desirable (e.g. if a step change in set point occurs) then the parameters should also be modified to reflect the change of transformation.

10.4 Range, one-sided

This case corresponds to the one of Section 10.3 in which z_o is chosen as z_{min}. In this case $E[z] \neq E[z_{min}]$ so that $E[z'] \neq 0$. This implies that $\Sigma z'$ will increase as N increases and therefore is not recommended.

10.5 Range, centered on median

For this transformation

$$E[z'] = \frac{1}{z_{max}-z_{min}}(E[z]- z_{.5})$$

This expectation will only tend to zero for symmetric distributions (e.g. for the normal distribution but not for the Chi-squared distribution) so it must be used with some caution.

10.6 Standard deviation

$$E[z'] = \frac{1}{\sigma}(E[z]-\bar{z})$$

where $\bar{z}$ is the mean of z. This expectation will tend to zero but the difficulty in the use of this transformation is the *a priori* determination of $\bar{z}$ and σ.

10.7 Reference value

The reference value must be non-zero and since it is usually chosen as the steady state or mean value then E[z'] will not be zero. This transformation is therefore not recommended.

10.8 Summary of the methods

Based on the analysis of the mean of z', three transformations are found suitable: range, centered on a reference value; range, centered on the median; and standard deviation. Unfortunately, analysis of the variance is not possible without many simplifying (and somewhat unrealistic) assumptions except in the case of the standard deviation which is known to produce unity variance. A possible modification for the range based transformations involves the use of a multiple of the range as the characteristic value in order to obtain a variance of unity. For instance, if the range is approximately six standard deviations (three on each side of the mean) then using a characteristic value of 1/6th of the range will give good results for the variance of z' provided that the reference value (or median) is close to the mean.

With this type of modification, range coding can be interpreted as using a standard deviation transformation where if the mean is not available it can be estimated by the median for symmetric distributions or the mean can be estimated by a reference value known to be close to the mean such as the set point or steady state value for controlled systems. If the variance is not available, it can be estimated by a multiple of the range of the variable, if known. For instance, if the variable follows the normal distribution and the variable is within the range 99% of the time, the range is equal to 5.2σ so that using 1/5.2 of the range as characteristic value will provide a variance close to unity.

10.9 Illustration of the effect of coding on identification using a simulated second order system

The second order system sampled at 0.2 time units presented in Chapter 7 is used to demonstrate the effect of coding the input/output data on identification.

The sample mean of the input signal is calculated to be 0.00 and the standard deviation is 1.00. The computed sample mean of the output signal is 0.15 and the standard deviation is 0.20. This indicates that coding the output to remove the mean and amplification by a factor of 5 should reduce the condition number of the P matrix.

If coding of the output is introduced by subtracting 0.15 and multiplying by 5, yielding a sample mean of -0.01 and a standard deviation of 0.99, the parameters obtained by using the RLS identification algorithm in PITSA are found to be

$$a_1 = -1.77 \pm 0.05$$

$$a_2 = 0.78 \pm 0.05$$

$$b_1 = 0.012 \pm 0.004$$

$$b_2 = 0.012 \pm 0.004$$

It should be noted that the parameters in Table 5.2 differ from the parameters obtained here since the parameters obtained here are for use with the coded input and output signals. The parameter correlation matrix for RLS with the coded signals is

$$\begin{array}{lrrrr} a_1 & 1.000 & & & \\ a_2 & -0.996 & 1.000 & & \\ b_1 & 0.046 & -0.054 & 1.000 & \\ b_2 & 0.172 & -0.178 & 0.022 & 1.000 \end{array}$$

This matrix has a condition number of 521 which is less than the condition number of 691 that would be obtained with RLS and no coding. For this simulated example coding did significantly reduce the condition number.

10.10 Conclusion

It was shown that in order to reduce the condition number coding of the input and the output should be such that $\Sigma z_t'$ is close to zero and that $\frac{1}{N}\Sigma(z_t')^2$ is close to unity where z_t' is the coded signal. Although this result may seem intuitive it has not appeared in the published literature in the field of process identification and control.

This type of coding is best achieved by statistical normalization but if this is not a viable option it can also be suitably approximated by using a fraction of the range to approximate the variance and the median to approximate the mean (for symmetric distributions). The use of coding to reduce the condition number was illustrated by simulation.

11. Normalization and its Role in Identification

11.1 Introduction

The concept of normalization as employed in system identification and adaptive control applications is a special form of coding where the (possibly time varying) characteristic value ensures that the input and output data are bounded (Ortega *et al.*, 1985, Lozano-Leal and Goodwin, 1985). Normalization of the regressor has also been used by Sripada and Fisher, 1987 in their proof of convergence of the least squares identification algorithm, but, as shown by Ljung and Soderstrom, 1983, normalization is not required for proof of convergence of the least squares identification algorithm.

Use of the term normalization by these authors differs from the meaning of normalization encountered in statistics. In statistics, normalization means transforming a random variable such that its probability distribution follows the normal distribution with mean, μ, and variance, σ^2 (usually the standard normal distribution i.e. $\mu=0$, $\sigma^2=1$). Normalization in the present context signifies coding where the coding factor at time t is chosen as a function of the largest element of the (possibly augmented) regressor at time t (e.g. Cluett *et al.*, 1986). In all cases a lower bound on the coding factor is implemented to ensure boundedness of the normalized values.

The negative effect of normalization on parameter identification will be demonstrated. It will be shown that normalization does not yield better numerical accuracy for constant normalization factors. Through a simulated example, it will be shown that alternatives to normalization can be found for improving the robustness of an adaptive control algorithm. No attempts are made to prove the stability of the alternate methods presented. From this simulated example it is concluded that normalization cannot be considered a "necessary evil". The reader interested in alternatives to normalization is referred to the work of Naik *et al.*, 1992.

11.2 Relationship of normalization to weighted least squares

The effect of normalization on least squares identification is investigated using the normalization factor of Sripada and Fisher, 1987 given as

$$n(t) = \max\{1, \|\phi(t)\|_{\infty}\} \qquad (11.1)$$

For a plant, with time delay d, represented by the model

$$y(t) = \phi(t)^T\theta(t-1) + \epsilon(t)$$

with

$$\theta^T = [a_1, \ldots , a_n, b_1, \ldots, b_m]$$

$$\phi(t)^T = [-y_{t-1}, \ldots , -y_{t-n}, u_{t-d-1} , \ldots , u_{t-d-m}]$$

the normalized model representation becomes

$$y_n(t) = \phi_n(t)^T\theta(t-1) + \epsilon_n(t)$$

where $y_n(t)$, $\epsilon_n(t)$ and $\phi_n(t)$ are respectively $y(t)$, $\epsilon(t)$ and $\phi(t)$ divided by $n(t)$. The least squares cost function for the normalized values is thus

$$
\begin{aligned}
J &= \sum_{t=1}^{N} \epsilon_n(t)^2 \\
&= \sum_{t=1}^{N} \frac{1}{n(t)^2}\epsilon(t)^2 \\
&= \sum_{t=1}^{N} w(t)\epsilon(t)^2 \qquad (11.2)
\end{aligned}
$$

The cost function representation of Equation 11.2 can be interpreted as a weighted least squares with weights of $\frac{1}{n(t)^2}$. The net effect of normalization is to provide more weighting when there is little excitation ($\|\phi(t)\|_2$ smal and less weighting when excitation is large ($\|\phi(t)\|_2$ large). However this approach represents precisely the action that should be avoided for least squares identification. For a constant magnitude of the noise, a large signal is characterized by a high signal to noise ratio, resulting in a more reliable parameter estimation but normalization will negate this favorable situation. Similarly for a small signal, with a lower signal to noise ratio, normalization will amplify the effect of the noise. If the magnitude of the noise is proportional to the signal then it may be appropriate to normalize the data so that large errors are weighted less. In this case the lower bound on the normalization factor should reflect the expected level of the noise for small signals. This aspect is not explored further in this thesis.

For robustness, avoiding large parameter value updates with large signals is desirable so that a sudden large signal does not create large control actions that may jeopardize the stability of the system. Yet for identification large signals are desirable. One approach to overcome this dilemma is to use a deadzone or a P filter as described in a subsequent section. Use of a deadzone puts a lower and an upper limit on the size of the update. In this context, deadzone can be viewed as another form of weighting where very large deviations and very small deviations have less weight than medium deviations.

11.3 Effect of normalization on the least squares algorithm

In theory, both the weighted least squares and ordinary least squares identification algorithms should estimate the same parameter values asymptotically if the noise is white i.e. random, zero mean and uncorrelated with past values. This fact can easily be demonstrated for the batch case for which the weighted least squares estimate is (c.f. Equation 4.8)

$$\hat{\theta}_{WLS} = (\Phi^T W \Phi)^{-1} \Phi^T W Y \tag{11.3}$$

where Φ is a matrix containing present and past ϕ's, W is a diagonal matrix containing the weights w(t) and Y is a vector of past and present y's. For the actual system described by

$$Y = \Phi\theta^* + E$$

where θ^* is the system parameter vector and E is the white noise vector, Equation 11.3 can be expressed as

$$\hat{\theta}_{WLS} = (\Phi^T W \Phi)^{-1}\Phi^T W \Phi\theta^* + (\Phi^T W \Phi)^{-1}\Phi^T W E$$

or

$$\hat{\theta}_{WLS} = \theta^* + (\Phi^T W \Phi)^{-1}\Phi^T W E \tag{11.4}$$

Since E is uncorrelated with Φ and W, the second term on the right hand side of Equation 11.4 will vanish and $\hat{\theta}_{WLS} = \theta^*$. The same is true for ordinary least squares as can be shown by setting W equal to the identity matrix (c.f. Section 4.1). If the noise is structured (i.e. correlated) the parameter estimates from both algorithms will be biased and the estimated parameters will be biased differently by using ordinary least squares versus weighted least squares identification.

The variance of the ordinary least squares parameters is $(\Psi^T\Psi)^{-1}\frac{E^TE}{N-p}$ (e.g. see Draper and Smith, 1981) where N is the number of data points and p is the number of parameters. If E is a vector containing white noise terms with variance σ^2 then the variance of the parameter vector for least squares is

$$VAR(\hat{\theta}_{LS}) = (\Psi^T\Psi)^{-1}\sigma^2$$

and for weighted least squares

$$VAR(\hat{\theta}_{WLS}) = (\Psi^TW\Psi)^{-1}\frac{E^TWE}{N-p} \qquad (11.5)$$

From Equations 11.4 and 11.5 it is concluded that normalization has no effect on the asymptotical convergence of the least squares estimates but since n(t) is a time varying function no such statement can be made about the variance of the parameters.

If the normalization factor, n(t), is constant, say n(t)=n for all times, from Equation 11.3 it follows that

$$\hat{\theta}_{WLS} = n^2(\Psi^T\Psi)^{-1}\frac{\Psi^TY}{n^2}$$

or

$$\hat{\theta}_{WLS} = \hat{\theta}_{LS} \qquad (11.6)$$

The variance of the weighted least squares estimates, Equation 11.5, can be written for a constant normalization factor as

$$VAR(\hat{\theta}_{WLS}) = n^2(\Psi^T\Psi)^{-1}\frac{1}{n^2}\sigma^2$$

so it follows that for a constant normalization factor that the variance of the estimates is identical to the variance of the estimates for ordinary least squares, that is

$$VAR(\hat{\theta}_{WLS}) = VAR(\hat{\theta}_{LS}) \qquad (11.7)$$

This indicates that if n(t) is constant then normalization does not affect the least squares identification algorithm.

11.4 Effect of normalization on numerical accuracy

It has been stated (Sripada and Fisher, 1987) that normalization improves numerical accuracy yet, as has been shown in the previous section, no conclusion concerning the structure of the variance of the parameters (therefore its numerical accuracy) can be stated *a priori* if n(t) is time variant. Furthermore, there is no improvement in numerical accuracy if n(t) is time invariant.

This can be demonstrated as follows. Ortega *et al.*, 1985 have shown that, for a controlled plant, n(t) is bounded and therefore an upper bound on n(t) exists. Consider that the input and output values are divided by this upper bound (or equivalently that the input and output values are coded such that the regressor norm is always less than unity). It follows from Equations 11.6 and 11.7 that the parameter estimates and their variances using constant normalization (n(t)=n=1 at all times for this case) will be the same as for the ordinary least squares i.e. the same calculations will be performed. Since the calculations are not modified by normalization (n=1), the numerical accuracy will not be affected i.e. normalization will not provide any numerical improvement. It should be noted that the constant normalization factor need not be unity but this value was selected to maintain consistency with the normalization factor used by Sripada and Fisher, 1987 (c.f. Equation 11.1).

11.5 Illustration of the effect of normalization on identification using a simulated first order system

It was demonstrated that normalization reduces the signal to noise ratio. To illustrate this fact, simulation was employed to examine the identification of the first order system studied in Chapter 7 using the recursive least squares (RLS) and the normalized recursive least squares (NRLS) algorithms in PITSA. An initial covariance matrix equal to the identity matrix was selected so that convergence would be slow enough to distinguish between the identification performance of the RLS and NRLS algorithms. In practice a larger initial covariance matrix would be used to compensate for the slower convergence of NRLS. This compensation is not possible when used in conjunction with adaptive control for time varying systems. The estimated parameters using RLS for 300 input/output data points were found to be

$$a_1 = -0.779 \pm 0.006$$

$$b_1 = 0.353 \pm 0.006$$

$$b_2 = 0.313 \pm 0.006$$

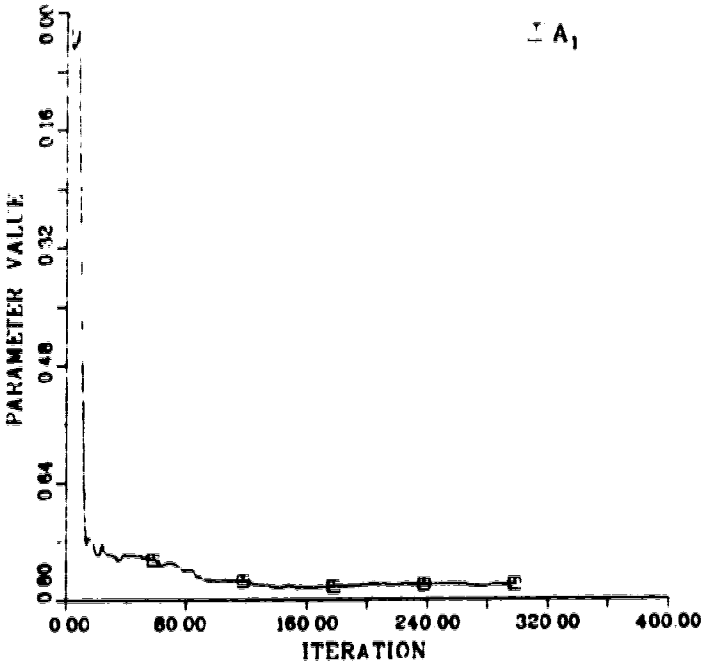
and for NRLS

$$a_1 = -0.77 \pm 0.01$$

$$b_1 = 0.35 \pm 0.01$$

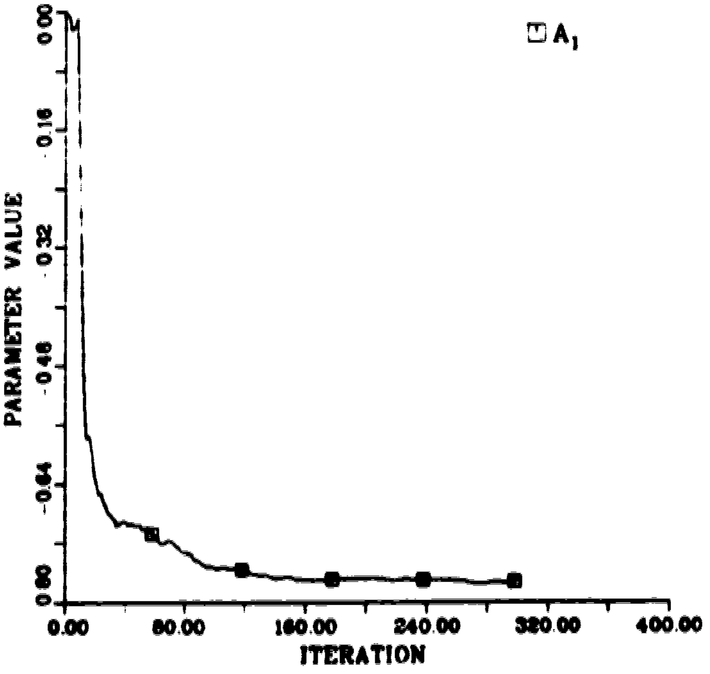
$$b_2 = 0.32 \pm 0.01$$

The slower convergence of the parameter estimates using the NRLS algorithm is evident from the confidence bands for the parameters and can also be observed by comparing Figures 11.1 and 11.2 and comparing Figures 11.3 and 11.4.



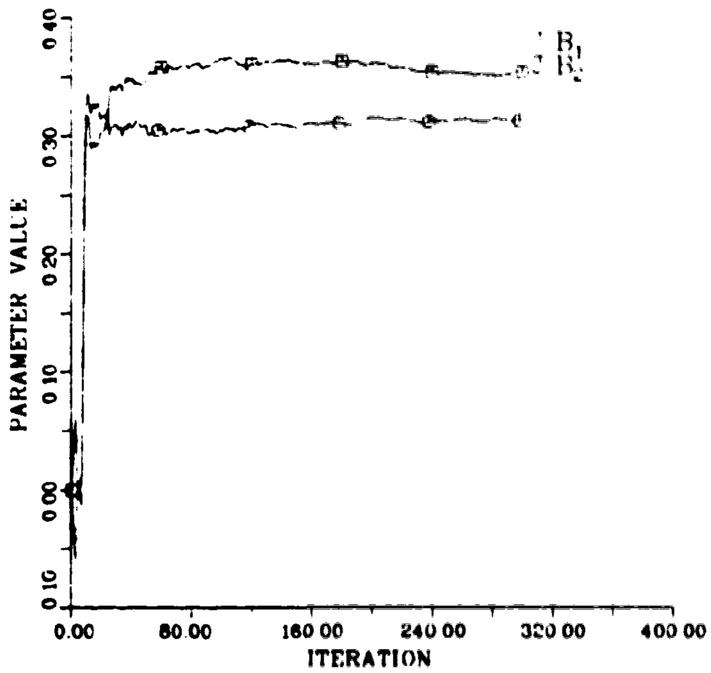
First order system
T(1,2,0)M3(1)

Fig. 11.1 Convergence of the a_1 parameter using the RLS algorithm



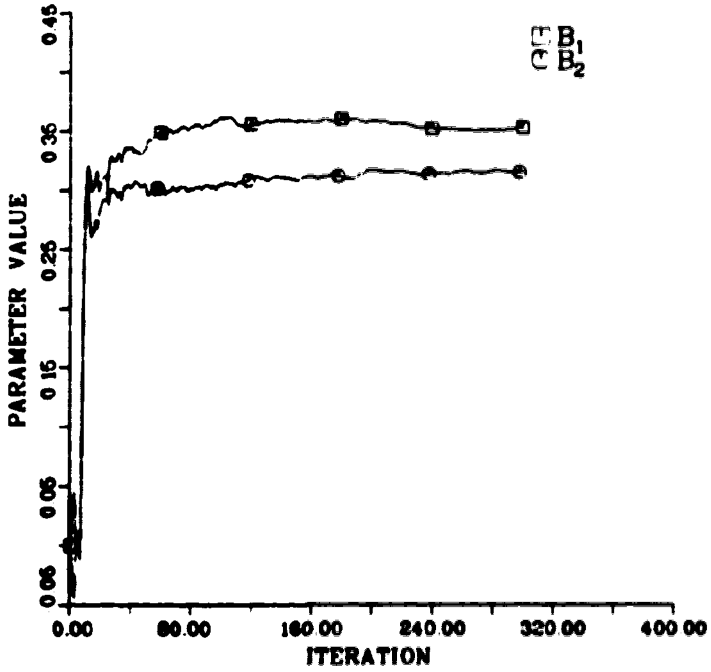
First order system
T(1,2,0)M3(2)

Fig. 11.2 Convergence of the a_1 parameter using the NRLS algorithm



First order system
T(1,2,0)M3(1)

Fig. 11.3 Convergence of the b_1 and b_2 parameters for the RLS algorithm



First order system
T(1,2,0)M3(2)

Fig. 11.4 Convergence of the b_1 and b_2 parameters for the NRLS algorithm

11.6 Alternatives to normalization in adaptive control

Alternatives to normalization were sought as normalization is not suited for use in conjunction with the least squares identification algorithm as previously shown. It is important to show that there are alternatives to normalization and therefore normalization should not be considered a necessary evil.

A heuristic description of the motivation behind the use of normalization is presented in the next subsection along with the rationale for the use of other methods to enhance robustness. Following this, the results of Cluett *et al.*, 1986 are analyzed to ascertain the actual effect of normalization and deadzone on identification and adaptive control. It will be demonstrated, through an example, that neither normalization nor deadzone are required to maintain the stability of the adaptive control scheme that was studied.

11.6.1 Improving robustness

Normalization as used by Ortega *et al.*, 1985 in their proof of the stability of an adaptive pole placement control scheme is based on the key assumption of stabilizability of the system. The stabilizability assumption requires that there exist a controller (or a nonempty set of controllers) which will keep the regressor, ϕ_t, bounded. Their proof of stability reduces to analyzing the stability of the difference between the response of the system using a stabilizing controller and the response of the system using the adaptive controller. If the difference in behavior of the two controllers is stable then the adaptive control scheme is stable.

Assuming that controllers "close enough" to a stabilizing controller are also stabilizing, i.e. stability is a smooth function of the controller parameters, then the fundamental factor in establishing a proof of stability depends on the smoothness of the change of controller parameters to avoid unstable behavior (i.e. the controller parameters cannot move into a region where stabilization of the process is no longer possible). This smoothness of change of controller parameters implies slowing down the adaptation of parameters used in the controller to a level where the controller will be able to stabilize the process at all times i.e. the updated controller parameters are close enough to the previous values so that the updated controller action is not different enough from the action that the previous controller would have taken as to render the system unstable. In other words, normalization is a method to detune adaptive control by slowing down (smoothing) the parameter update thus keeping the system stable.

Use of smoothing to stabilize the controlled process will be demonstrated, by simulation, for the following cases

i. normalization and deadzone,

ii. normalization,

iii. deadzone,

iv. slowing down tne identification algorithm by changing its tuning parameters,

v. filtering of the estimated process parameters before utilization by the controller, and

vi. filtering the input/output regressor.

11.6.2 System model

The system and control algorithm selected to illustrate alternatives to normalization to provide robust adaptive control are identical to those used by Cluett *et al.*, 1986. Although the system was considered as the third order model

$$y_t = 0.820y_{t-1} - 0.0327y_{t-2} + 0.000431y_{t-3} + 0.234u_{t-1} + 0.185u_{t-2} + 0.00479u_{t-3} \qquad (11.8)$$

for control purposes, a first order model using the projection algorithm of Goodwin and Sin, 1984 and a minimum variance (one step ahead) controller were employed. The projection algorithm, although not used elsewhere in this thesis because of its slower convergence compared to the conventional least squares algorithm, is employed in this section to conform to the work of Cluett *et al.*, 1986. The estimated model used by the controller is thus

$$y_t = a_t y_{t-1} + b_t u_{t-1} \qquad (11.9)$$

with parameters a_t and b_t of this model updated using the projection algorithm expressed as

$$\theta_t = \theta_{t-1} + \frac{a\ \phi_t}{c + \phi_t^T\phi_t}(y_t - \phi_t^T\theta_{t-1}) \qquad (11.10)$$

where

$$\theta_t = [a_t, b_t] \qquad (11.11)$$

a, c = projection algorithm tuning parameters

If normalization is not employed then

$$\phi_t^T = [y_{t-1}, u_{t-1}] \qquad (11.12)$$

and for the projection algorithm tuning parameters taken as a=1, c=1. Cluett *et al.*, 1986 have shown that the control action will cause unstable system behavior unless the control algorithm is modified.

If normalization is employed to achieve control action that results in stable operation, the following transformations are performed (Cluett *et al.*, 1986) and identification is carried out with the transformed values

$$y_t^n = y_t/n_t \qquad (11.13)$$

$$y_{t-1}^n = y_{t-1}/n_t \qquad (11.14)$$

$$u_{t-1}^n = u_{t-1}/n_t \qquad (11.15)$$

with $n_t = \max\{1, |y_i|, |u_i|\}$ for $t-4 \leq i \leq t-1$.

If a deadzone is included (Martin-Sanchez *et al.*, 1984), the projection algorithm (Equation 11.10) is modified to the following form

$$\theta_t = \theta_{t-1} + \frac{a\ \psi_t^2\phi_t}{c + \psi_t^2\phi_t^T\phi_t}(y_t - \phi_t^T\theta_{t-1}) \qquad (11.16)$$

by inclusion of the scalar quantity ψ_t^2. The value of ψ_t^2 is determined by a criterion for stopping or continuing parameter adaptation according to

i. if $|y_t - \phi_t^T\theta_{t-1}| \leq \Delta_l$ then $\psi_t^2 = 0$ (11.17)

ii. if $|y_t - \phi_t^T\theta_{t-1}| > \Delta_u$ then $\psi_t^2 = \psi_u^2$ (11.18)

iii. if $\Delta_l < |y_t - \phi_t^T\theta_{t-1}| \leq \Delta_u$ then

$$\psi_t^2 = \frac{2(|y_t - \phi_t^T\theta_{t-1}| - \Delta)}{(2\Delta - |y_t - \phi_t^T\theta_{t-1}|)\phi_t^T\phi_t} \tag{11.19}$$

with

$$\Delta_l = \frac{2(1 + \psi_l^2\phi_t^T\phi_t)\Delta}{2 + \psi_l^2\phi_t^T\phi_t} \tag{11.20}$$

$$\Delta_u = \frac{2(1 + \psi_u^2\phi_t^T\phi_t)\Delta}{2 + \psi_u^2\phi_t^T\phi_t} \tag{11.21}$$

where ψ_l^2 is a lower bound, ψ_u^2 is an upper bound and Δ is an estimate of an upper bound on the absolute value of the modelling error. Following Cluett *et al.*, 1986, the values of ψ_l^2, ψ_u^2 and Δ are taken as 0.1, 1.0 and 0.3759 respectively. The value of Δ represents twice the norm of the unidentified parameter vector (from mismatch) when normalization is performed. The control action is computed as

$$u_t = (y_{sp} - a_t y_t)/b_t \tag{11.22}$$

where y_{sp} is the desired set point.

Simulations were performed with the initial values of the input signal, output signal and a_t set to zero with b_t set to 1.0 to prevent division by zero.

11.6.3 Simulation results

Simulations were conducted using MATLAB (Moler *et al.*, 1986) by programming Equations 11.8, 11.9, 11.11 to 11.22. The results are plotted in Figures 11.5 to 11.8. The results presented in part a) of Figures 11.5 and 11.6 show the behavior reported by Cluett *et al.*, 1986 when employing both a deadzone and normalization.

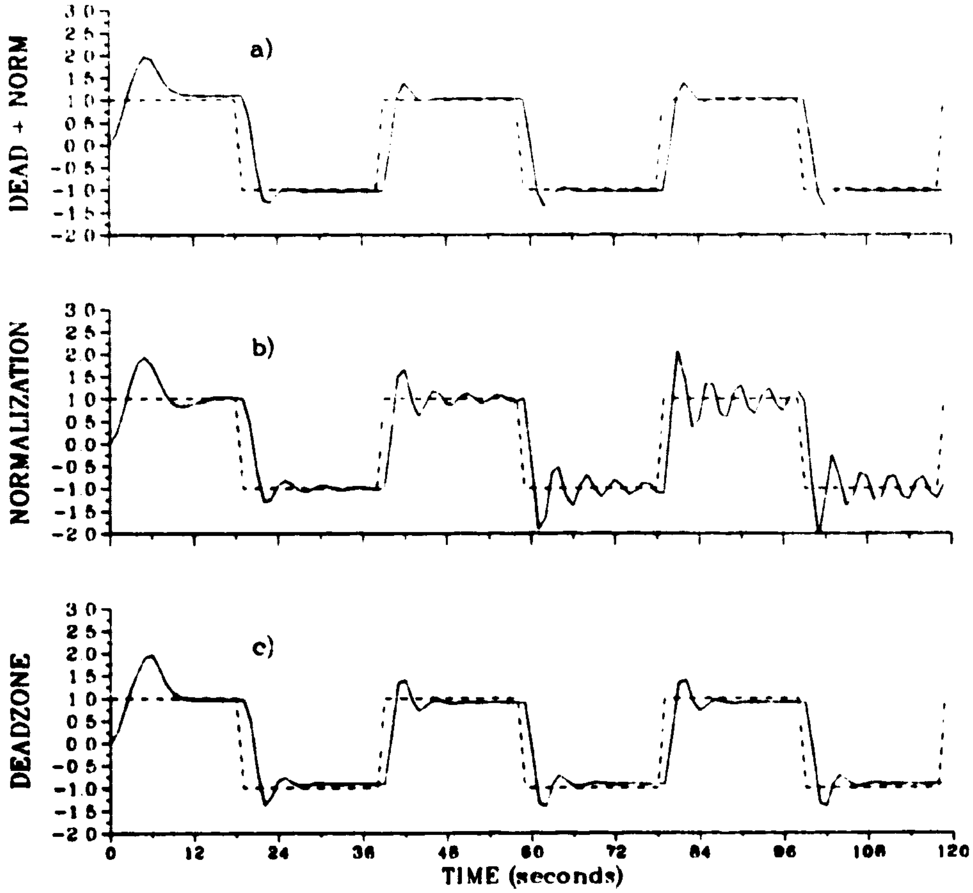


Fig. 11.5 System output (solid line) and set point (dashed line) for a)deadzone and normalization, b)normalization, c)deadzone

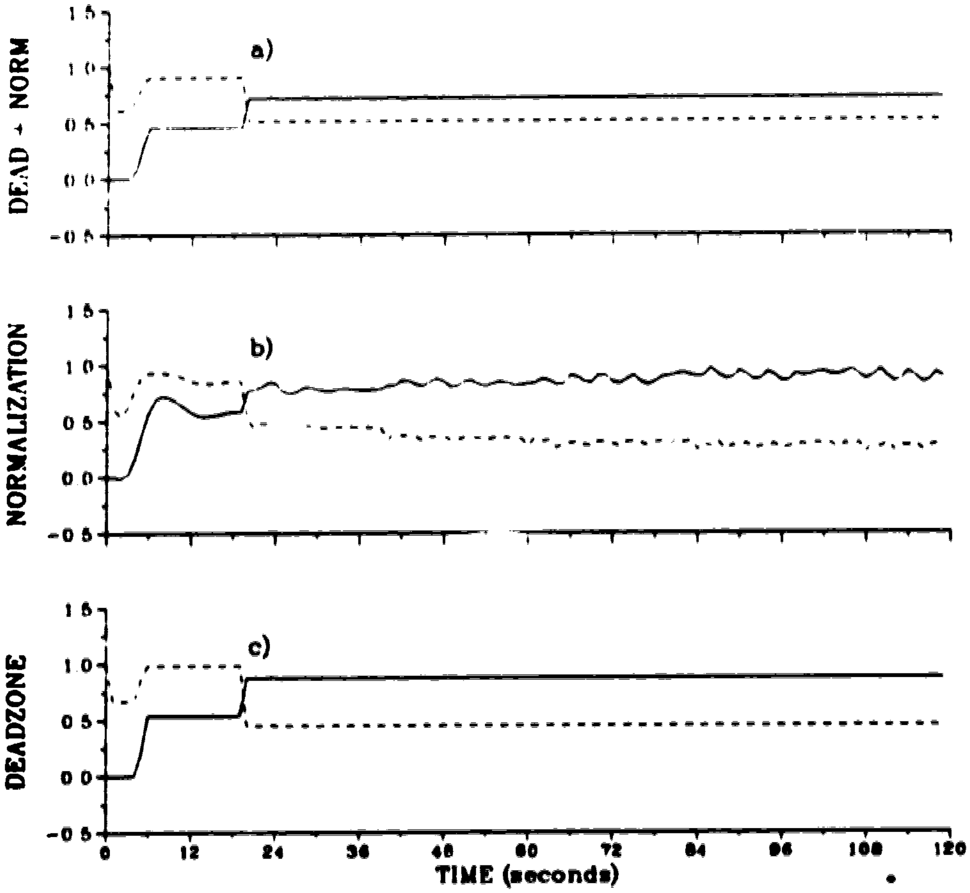


Fig. 11.6 Estimated parameter values: a_t (solid line) and b_t (dashed line) for a)deadzone and normalization, b)normalization, c)deadzone

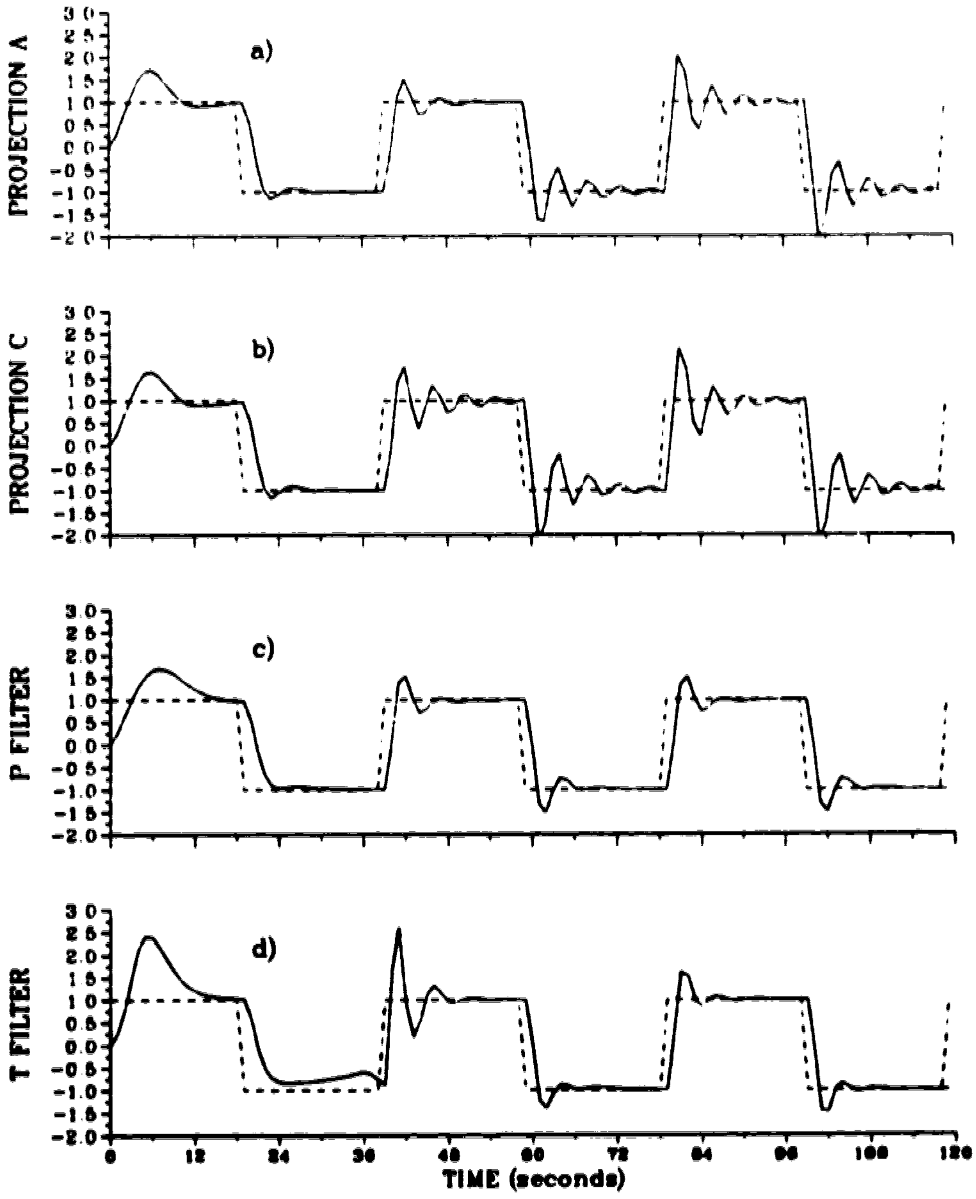


Fig. 11.7 System output (solid line) and set point (dashed line) for a)a=0.5, b)c=4.0, c)a_f=-0.8, d)t_f=-0.95

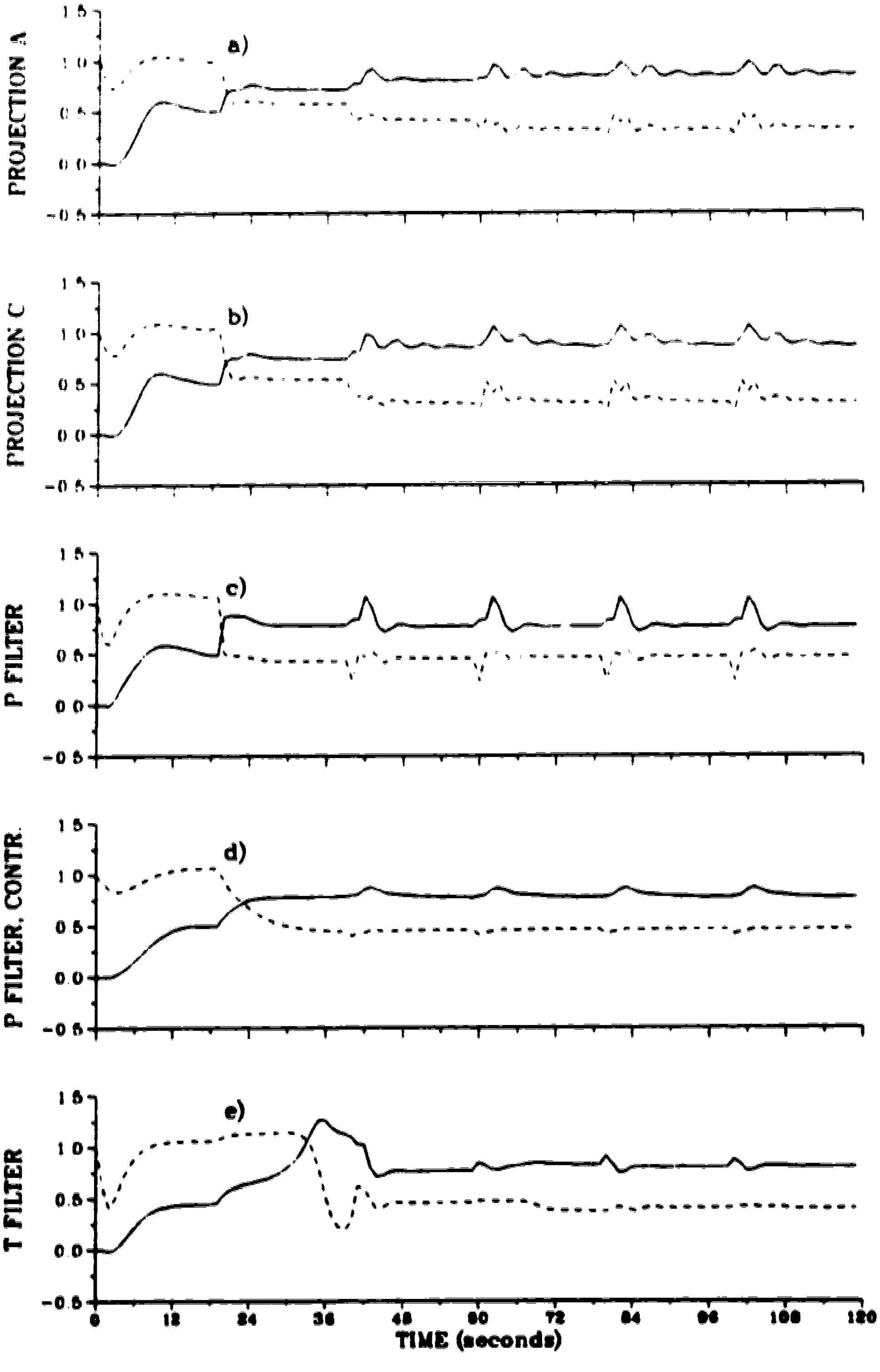


Fig. 11.8 Estimated parameter values: a_t (solid line) and b_t (dashed line) for a)a=0.5, b)c=4.0, c)a_f=-0.8, d)filtered parameters using a_f=-0.8, e)t_f=-0.95

As can be seen from parts b) and c) of these figures, the effect of a deadzone and/or normalization is to slow down the convergence of the identification algorithm. This is readily observed from Figure 11.6 which shows that with a deadzone the parameter values show virtually no adaptation after the second change in set point. Although providing stability, the effect of normalization is not as pronounced as the effect of utilizing a deadzone. To produce a stable algorithm without utilizing normalization, the deadzone was enlarged to 1.6 times the deadzone used for the normalized case. This increase was necessary because the error size is larger if unnormalized. The value of 1.6 reflects the scaling effect of the "average" value of n_t.

To test if the stability of the adaptive control algorithm is linked to slowing down the convergence of the identification algorithm, simulations were performed without a deadzone and without normalization. First, the projection algorithm tuning parameters were changed to slow down the convergence. In part a) of Figures 11.7 and 11.8 the "a" tuning parameter of the projection algorithm, which directly multiplies the update gain of the algorithm (c.f. Equation 11.10), was set to 0.5 instead of the original value of 1.0 used by Cluett *et al.*, 1986. The performance of the projection algorithm using c=4 rather than c=1, which reduces the gain of adaptation thus providing a slower convergence, is displayed in part b) of Figures 11.7 and 11.8.

Another option for obtaining stable control is to employ a low pass filter on the identified parameters before their use in the control algorithm. Use of this filter, called a P filter, has the advantage that the identification is allowed to proceed at "normal speed" i.e. without having to slow down the convergence of the identification algorithm. Use of this approach allows the controller to "keep up" with the identification leading to stable system behavior. Control using a P filter provides control performance similar to that obtained from use of the control algorithm with a deadzone as can be seen from part c) of Figure 11.7. Furthermore, the estimated a_t and b_t parameters change continuously in the P filter case as illustrated in part c) of Figure 11.8. The parameters utilized by the controller as a consequence of use of $\frac{0.2}{1 - 0.8q^{-1}}$ as the P filter are displayed in part d) of Figure 11.8. It should be noted that the numerator value of 0.2 was selected to maintain the unity gain of the filter.

A different filtering approach is to use the T filter. This is a low pass filter used to filter the input/output regressor before it is used in identification. The net effect of this filter is to remove high frequencies which translates into a slower convergence as some of the available information is now discarded. Results using a T filter of the form $\frac{1}{1 - 0.95q^{-1}}$ are presented in part d) of Figure 11.7 and part e) of Figure 11.8. This high filter parameter value of -0.95 was necessary as the high frequency content is low and thus some of the medium frequencies must be removed to slow down the convergence of the algorithm enough to achieve stable behavior (the simulation was performed without adding any noise).

It should be noted that no attempt was made to optimize the tuning parameters of the identification algorithm or the filter parameter values beyond finding values which provide stable control. In all cases sensitivity of the results with respect to the respective tuning parameter was relatively low in the region giving stable control. For the T filter this region was found to be rather small (minimum reasonable choice being -0.95). A second order T filter may have lower parameter values since the mismatch is of second order (third order plant and first order model). In the case of P or T filters, a possible option would be the use of a variable parameter scheme to accelerate the initial phase of the adaptation where reliable parameter values are not available. This aspect was not investigated as the purpose of this work was to demonstrate that alternatives to normalization do exist without establishing the best alternative for this system.

11.7 Conclusion

Normalization (as used in the process identification and control field) employed in conjunction with a least squares identification algorithm has been shown to be equivalent to a weighted least squares identification algorithm in which the weights are adjusted to reduce the signal to noise ratio (for a constant magnitude of the noise). It is believed that this relationship has not been exposed in the published literature. The signal to noise ratio reduction is highly undesirable for identification unless the magnitude of the noise increases with the magnitude of the signal. Therefore normalization is not recommended for use with least squares identification if the magnitude of the noise does not depend on the magnitude of the signal.

Normalization was found to have no effect on numerical accuracy if the normalization factor is constant contrary to the claims of Sripada and Fisher, 1987. The effect on the numerical accuracy of the algorithm using a time variant normalization factor cannot be predicted.

Provided the underlying assumptions are satisfied, using normalization and a deadzone provide sufficient conditions for establishing the stability of adaptive control (Cluett *et al.*, 1986) but these conditions were not found necessary to ensure stability of the simulated adaptive control scheme and therefore another method such as one of the alternatives described could be substituted for normalization.

Simulation results have shown that an alternate mode of stabilization for control purposes can be either the use of a deadzone or P filter (controller parameter filtering). Use of normalization alone and use of the projection algorithm tuning parameters gave oscillatory responses. The first order T filter has undesirable transients that last for two changes in set point compared to a single change in set point using a deadzone or the P filter. The first order T filter constant necessary to obtain a stable response is very high and therefore the choice of the T filter parameter is restricted to a small range of values. This makes the choice of the T filter parameter more critical than the choice of the deadzone or the P filter parameter for this system. Perhaps a second order T filter could be more effective.

12. Time Series Analysis of Brenda Mines Secondary Crusher

12.1 Introduction

In this chapter time series analysis of a secondary ore crusher is utilized to illustrate a typical process identification exercise. The Box-Jenkins approach to process identification (Box and Jenkins, 1976, also described in Appendix C) is employed to find the model structure (polynomial orders) and parameter estimates describing the dynamic behavior of the secondary crusher. Comparison of different approaches to handle nonzero mean data is presented. Results based on least squares and maximum likelihood estimation are also compared.

12.2 Description of the mining process

Brenda Mines Ltd., Peachland B.C., operated an open pit copper and molybdenum sulfide mine processing very low grade ore. The process involved 3 stages of crushing followed by 2 stages of grinding before flotation separation and concentration stages. The run-of-the-mine ore (< 3 feet in diameter) was brought from the pit by dump trucks and dumped directly into a primary gyratory crusher. The crushed rock (< 6 inches) was screened and the undersize (< 3/4 inch) went to the fine ore bins to be fed to the grinding circuit. The oversize was sent to an outdoor coarse ore stockpile.

From the bottom of the coarse ore stockpile, the ore was conveyed to one of two secondary standard crushers. The product was sent to vibrating screens where the undersize (< 1 inch) was separated and sent to the fine ore bins where it joined the primary screens undersize to await grinding. The oversize was returned to a surge bin which fed four tertiary shorthead crushers.

The secondary crusher system consisted of two parallel circuits. The conveyor of a circuit was fed by any or all of four available vibrating feeders underneath the coarse ore stockpile. The selection of feeders on at any given moment was determined by the operator on duty.

Since each feeder discharged ore with different characteristics, which varied with the state of the coarse ore stockpile directly above the feeder, the operator could choose the coarseness of the ore being crushed to some extent. Tonnage was measured by a weightometer located underneath the conveyor belt downstream of the feeders. The transportation lag between the feeders and the weightometer was between 12.2 and 28.7 seconds depending on which feeders were operating. The transportation lag between the weightometer and the crusher was approximately 40 seconds.

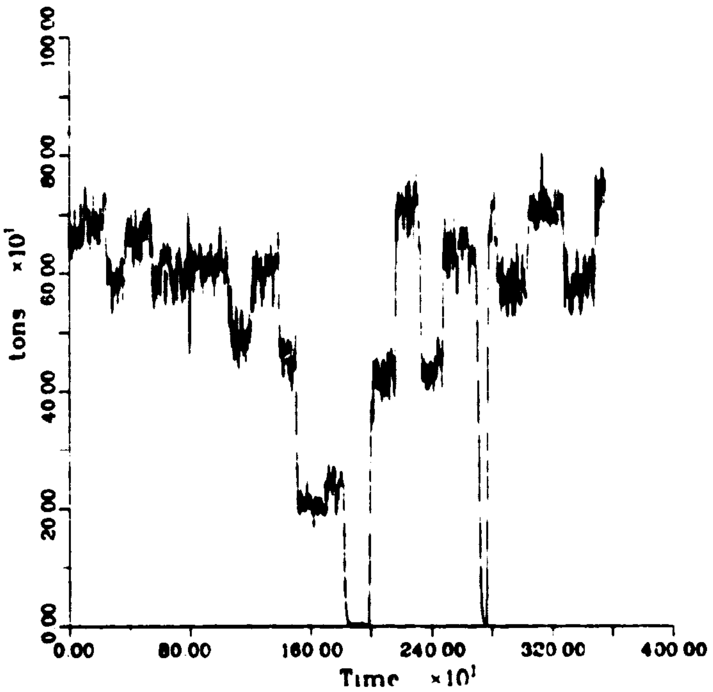
12.3 Purpose of modelling the secondary crusher

The difficulty in controlling this system lied in the transportation lag between the coarse ore feeders and the crushers. The purpose of modelling the secondary crusher power consumption (a measure of the crusher load) as a function of the weight of ore fed to the crusher was to produce a crusher power predictor that could be used for control. Using this predictor, the ore feed rate may be adjusted to maximize ore throughput without overloading the crusher which would result in a shutdown of the crusher.

Since a theoretical model of the crusher must account for ore hardness and size distribution, it is too complex to use on-line. Moreover, hardness and size distribution measurements are not available on-line. Consequently the approach adopted to determine a dynamic model was to employ identification using input/output data.

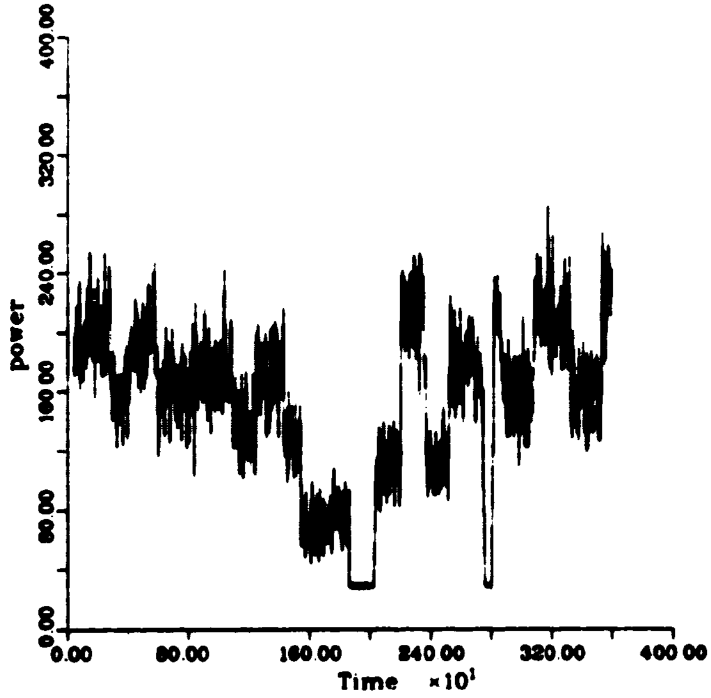
12.4 The data set

The input/output data provided by Brenda Mines was obtained from normal operation but under manual control. A typical one hour record (Edwards, 1986) of the input and output signals is presented in Figures 12.1 and 12.2 respectively. During the experiment, the crusher had to be stopped twice to remove tramp metal. The tramp metal detector is triggered whenever there are broken metallic parts from mining equipment on the conveyor that would block the crusher. Since stopping and starting the crusher is a nonlinear function, only the first 1200 points, corresponding to 20 minutes of operation, were utilized for establishing a suitable dynamic model using the time series analysis features of the PITSA program.



Brenda Mines dec 17

Fig. 12.1 Ore feed rate to the crusher



Brenda Mines dec 17

Fig. 12.2 Crusher power consumption

12.5 Input prewhitening

To permit model order estimation, the input signal to the process must be white noise. The input signal to the crusher used for identification is presented in Figure 12.3. If the process input signal is not representative of white noise, as can be observed in Figures 12.3 to 12.5, then the first step of the identification procedure is to find a prewhitening filter.

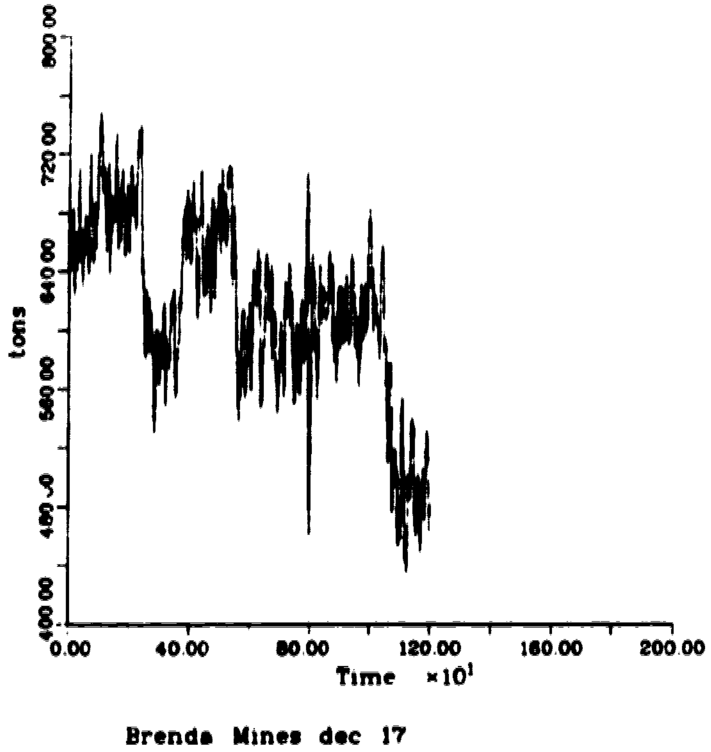
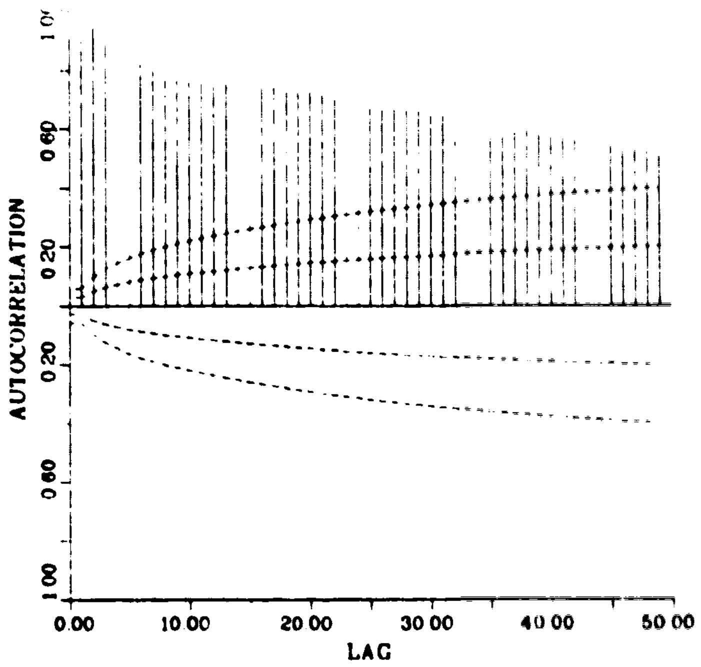


Fig. 12.3 Ore feed rate to the crusher used for estimation

The filter model is

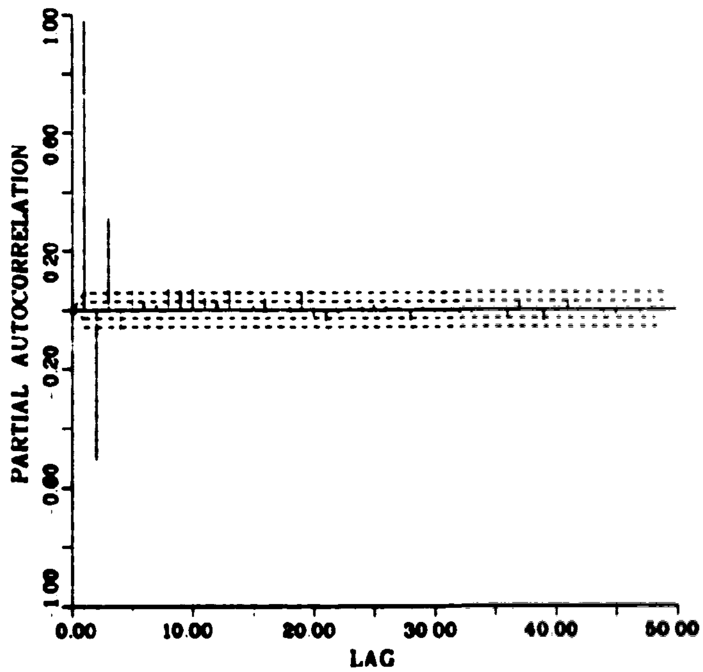
$$\phi u_t^f = \theta \nabla^{df} u_t + \beta$$

where



tons

Fig. 12.4 Autocorrelogram of the ore feed rate



tons

Fig. 12.5 Partial autocorrelogram of the ore feed rate

$$\theta = \theta_0 + \theta_1 q^{-1} + \ldots + \theta_{mf} q^{-mf}$$

$$\phi = 1 + \phi_1 q^{-1} + \ldots + \phi_{nf} q^{-nf}$$

∇^{df} = degree of differencing (df times)

β = constant term (bias)

u_t = input signal

u_t^F = filtered input signal

For the input signal, the autocorrelation function shown in Figure 12.4 does not decrease quickly and the first partial autocorrelation displayed in Figure 12.5 is almost unity suggesting that differencing may be required.

Differencing of the data results in the input shown in Figure 12.6. The autocorrelation and partial autocorrelation function results presented in Figures 12.7 and 12.8 show that the magnitudes of a single autocorrelation and up to four partial autocorrelations are significant indicating that the filter model should have nf=1 and mf up to 4 (but probably less) and further differencing is not required.

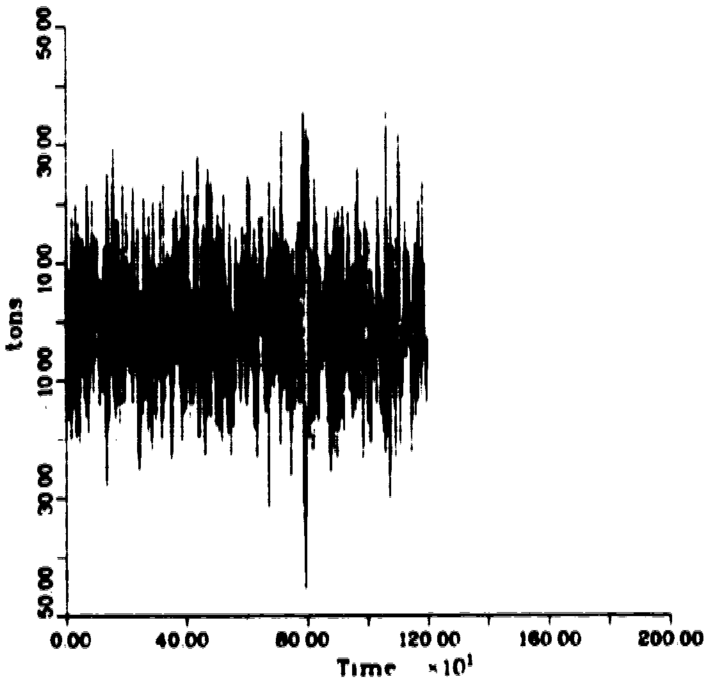
For prewhitening, θ_0 is set to unity as this parameter will only influence the variance of the filtered input. An attempt to estimate filter parameters with nf=1, df=1, mf=3, denoted F(1,1,3) produced the following parameters

$$\phi_1 = 0.573 \pm 0.091$$

$$\theta_1 = -0.285 \pm 0.096$$

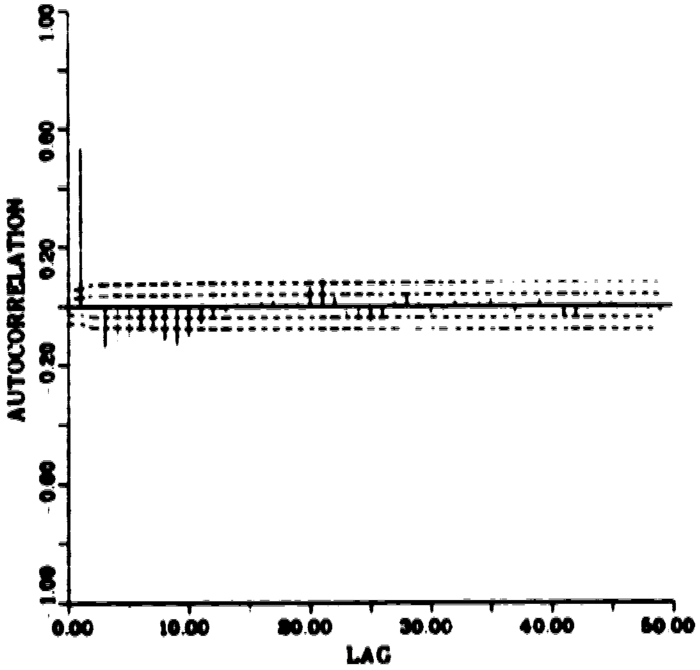
$$\theta_2 = 0.135 \pm 0.081$$

$$\theta_3 = 0.046 \pm 0.057$$



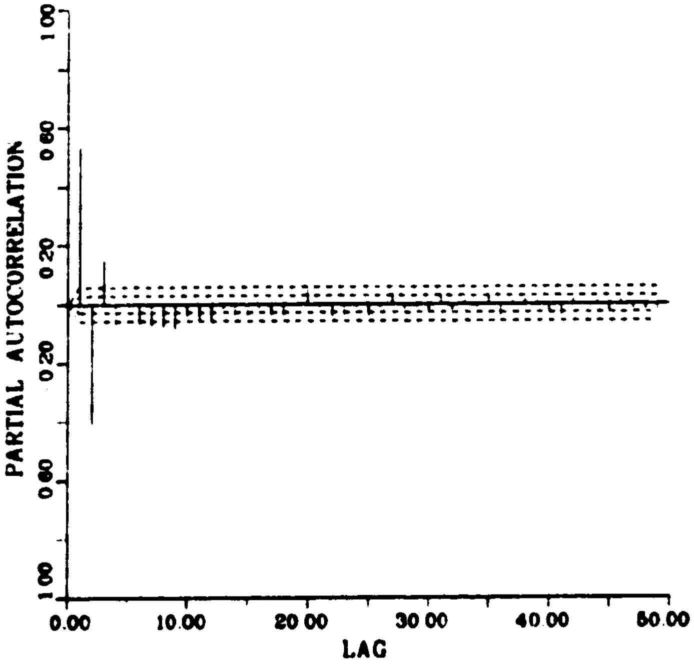
Brenda Mines dec 17

Fig. 12.6 Differenced ore feed rate



tons

Fig. 12.7 Autocorrelogram of differenced ore feed rate



tons

Fig. 12.8 Partial autocorrelogram of differenced ore feed rate

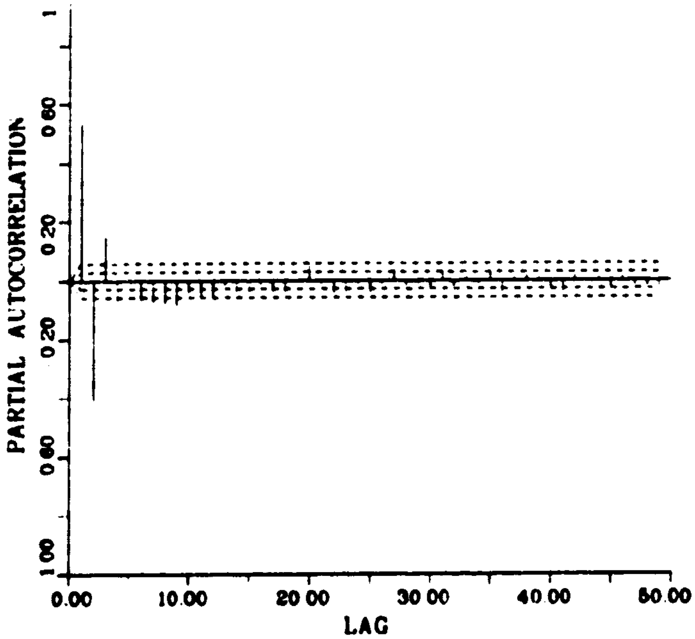
Since θ_3 is not significant, the use of mf=3 instead of mf=4 was justified and the fact that θ_3 is not significant also suggests that F(1,1,2) may lead to an adequate filter. Parameter estimates using F(1,1,2) were found to be

$$\phi_1 = 0.515 \pm 0.048$$

$$\theta_1 = -0.345 \pm 0.052$$

$$\theta_2 = 0.186 \pm 0.042$$

All the parameters are significant, the x^2 statistic (Box and Jenkins, 1976, Appendix B) is 60.54 which is less than the table value at 95% confidence level. Plots presented in Figures 12.9 to 12.17 are examined to observe any irregularity in the convergence of the parameters or any irregularity in the sum of squares of residuals or to ensure



tons

Fig. 12.8 Partial autocorrelogram of differenced ore feed rate

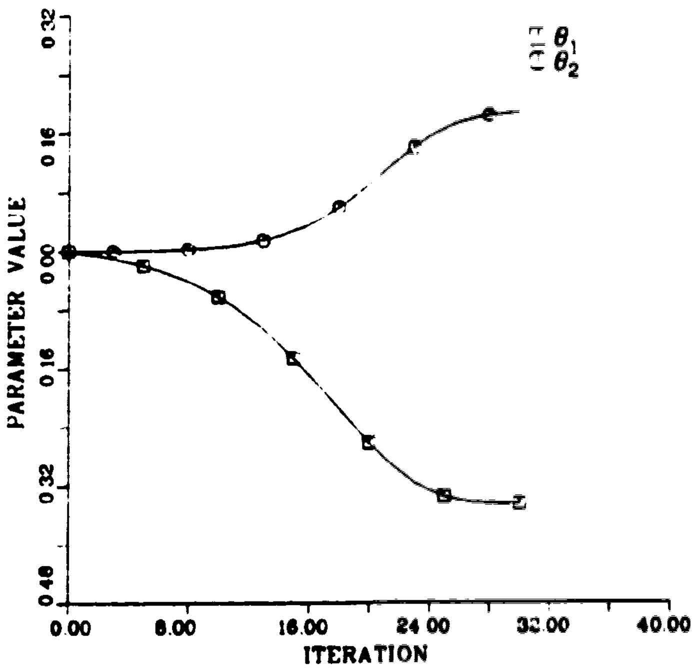
nce θ_3 is not significant, the use of mf=3 instead of mf=4 s justified and the fact that θ_3 is not significant also ggests that F(1,1,2) may lead to an adequate filter. rameter estimates using F(1,1,2) were found to be

$$\phi_1 = 0.515 \pm 0.048$$

$$\theta_1 = -0.345 \pm 0.052$$

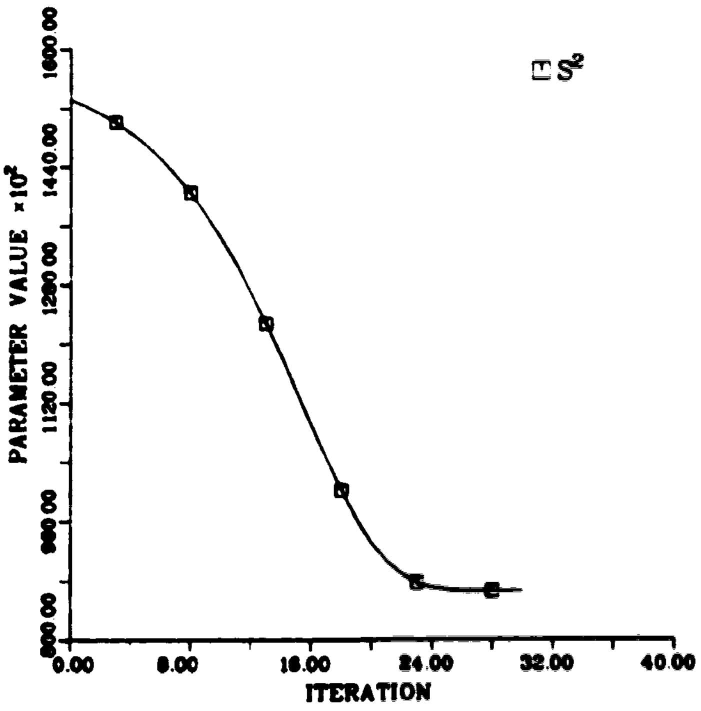
$$\theta_2 = 0.186 \pm 0.042$$

l the parameters are significant, the x^2 statistic (Box d Jenkins, 1976, Appendix B) is 60.54 which is less than e table value at 95% confidence level. Plots presented in gures 12.9 to 12.17 are examined to observe any regularity in the convergence of the parameters or any regularity in the sum of squares of residuals or to ensure



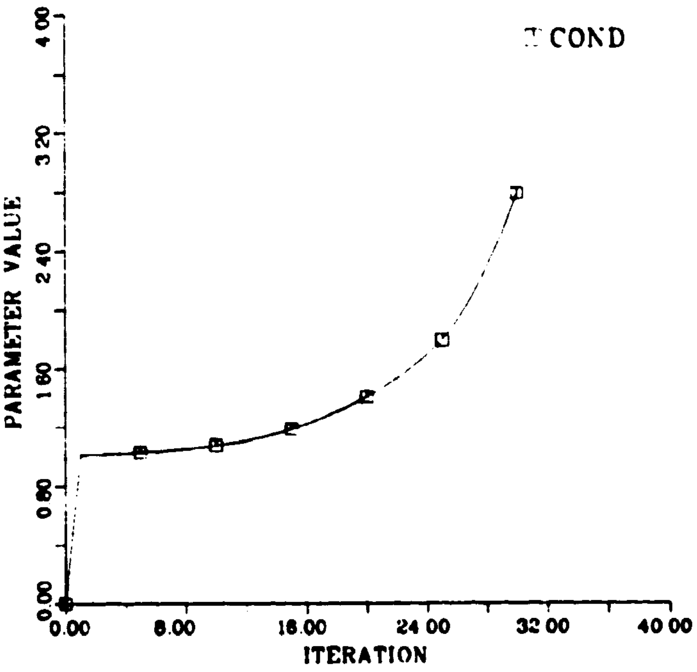
tons

Fig. 12.10 Convergence of θ



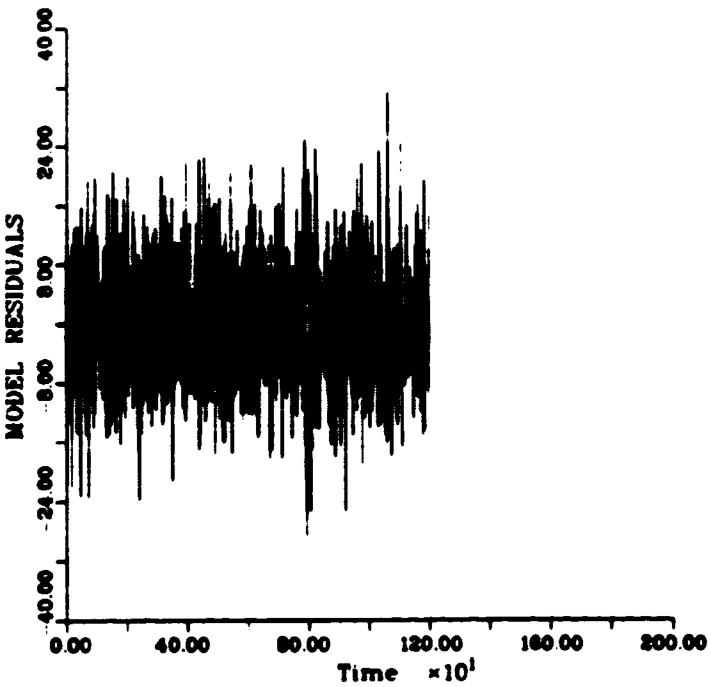
tons

Fig. 12.11 Sum of squares of residuals



tons

Fig. 12.12 Condition number



tons

Fig. 12.13 Filtered ore feed rate

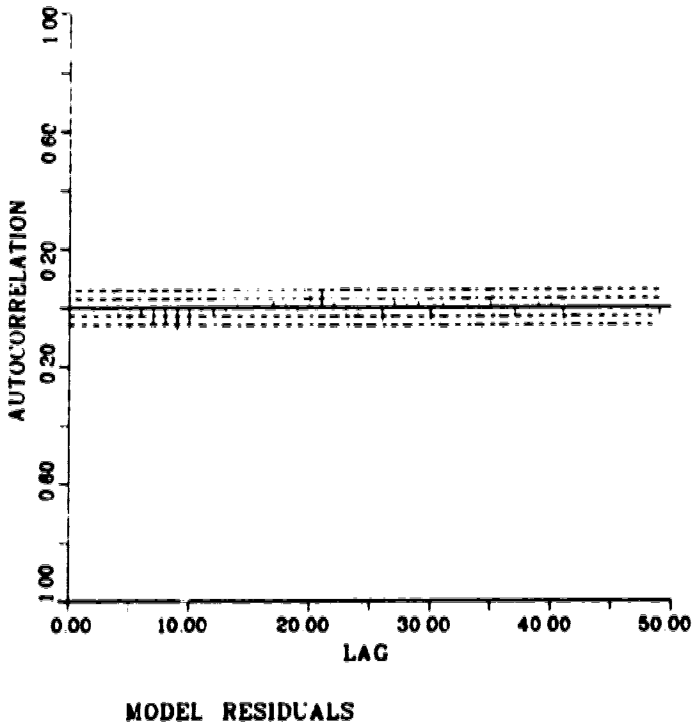


Fig. 12.14 Autocorrelogram of filtered ore feed rate

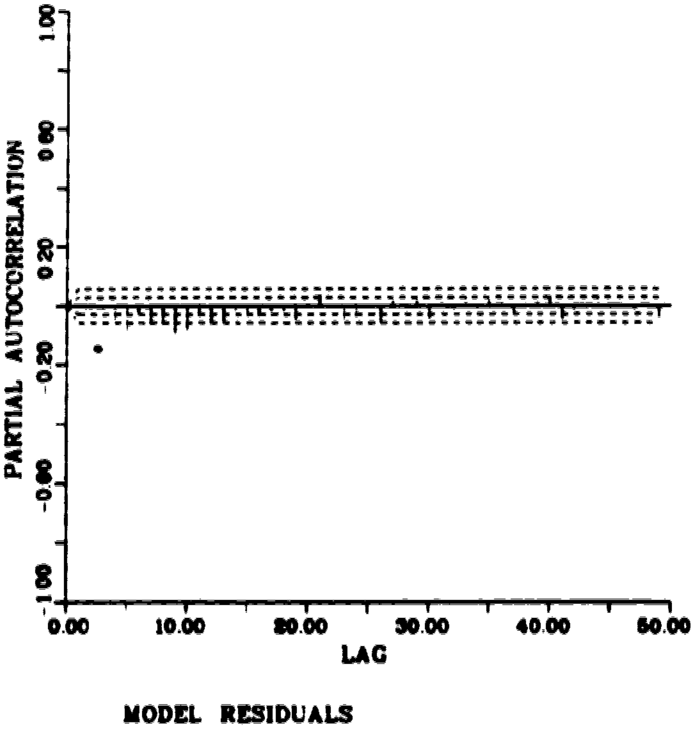
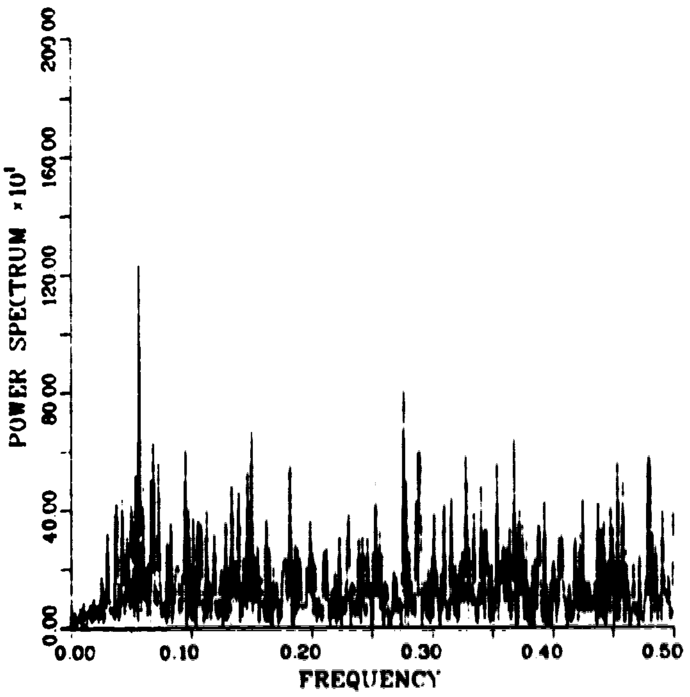
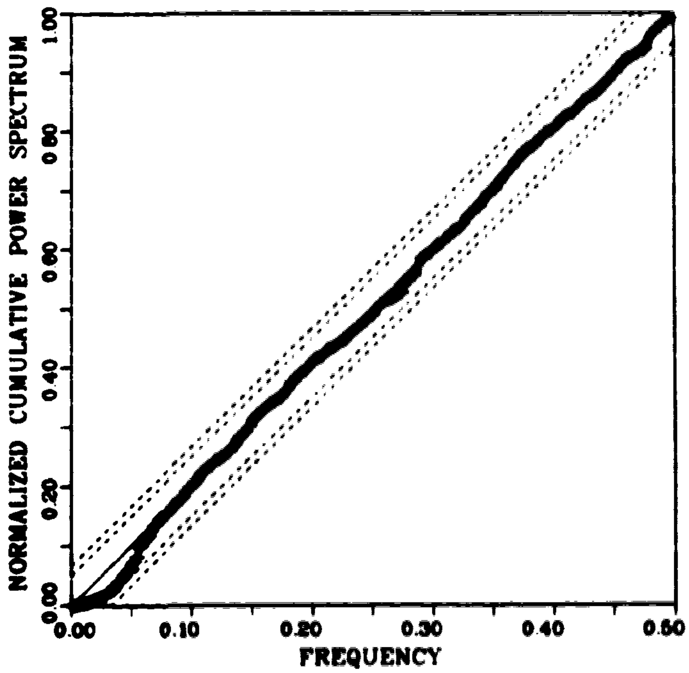


Fig. 12.15 Partial autocorrelogram of filtered ore feed rate



MODEL RESIDUALS

Fig. 12.16 Power spectrum of filtered ore feed rate



MODEL RESIDUALS

Fig. 12.17 Normalized power spectrum of filtered ore feed rate

12.6 Process transfer function and noise model structure estimation

The process model used in the Box-Jenkins approach is of the form (see Appendix C for details)

$$y_t = \frac{B}{A}u_{t-k} + \frac{C}{D}\epsilon_t + \beta$$

where

$$A = 1 + a_1q^{-1} + \ldots + a_nq^{-n}$$
$$B = b_1q^{-1} + \ldots + b_mq^{-m}$$
$$C = 1 + c_1q^{-1} + \ldots + c_{mn}q^{-mn}$$
$$D = 1 + d_1q^{-1} + \ldots + d_{nn}q^{-nn}$$

k = delay

β = constant term (bias)

u_t = input

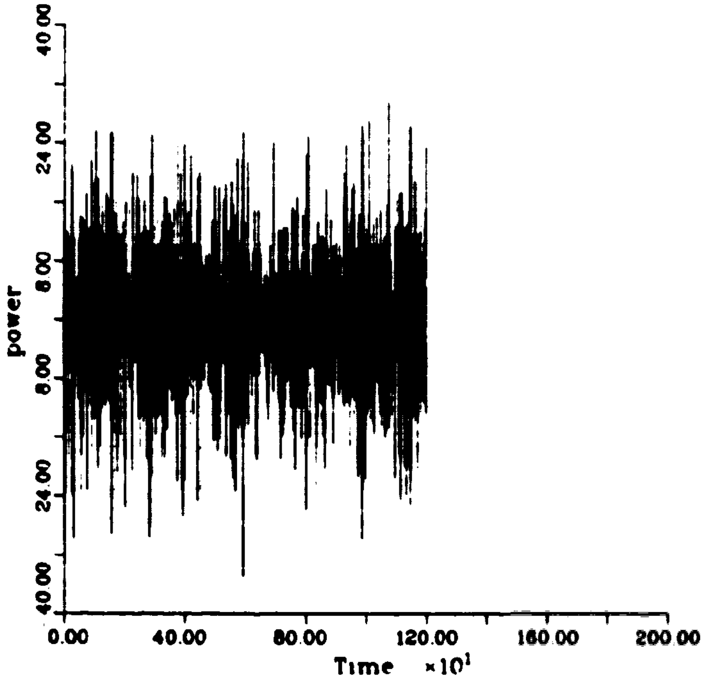
y_t = output

ϵ_t = white noise (residuals)

This model is composed of two distinct parts: the process transfer function (B/A + β) and the noise model (C/D). The process model structure identification is therefore separated into two steps (described in Appendix C) summarized below.

12.6.1 Process transfer function structure identification

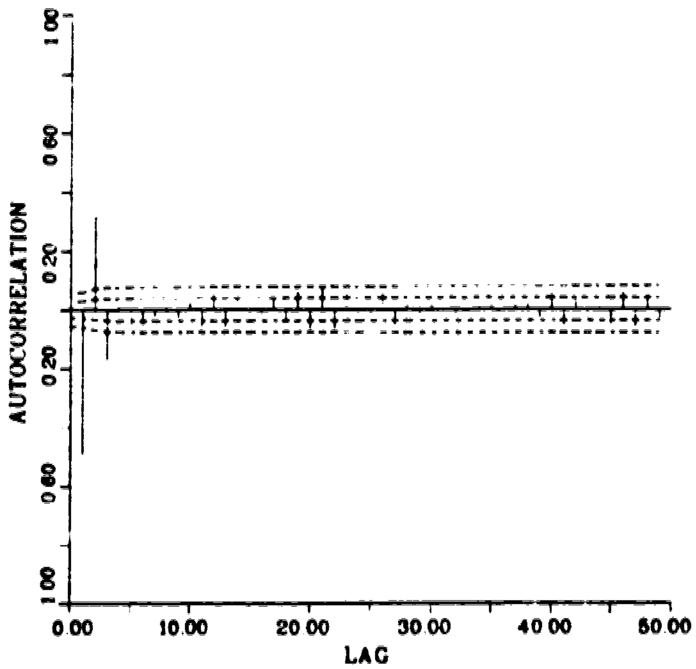
In order to identify the process transfer function structure the input signal must be representative of white noise as described in Section 12.4. To preserve the relationship between the input and output signals, the output signal must also be filtered using the same prewhitening filter as that applied to the input signal. The filtered output characteristics are presented in Figures 12.18 to 12.20.



Brenda Mines dec 17

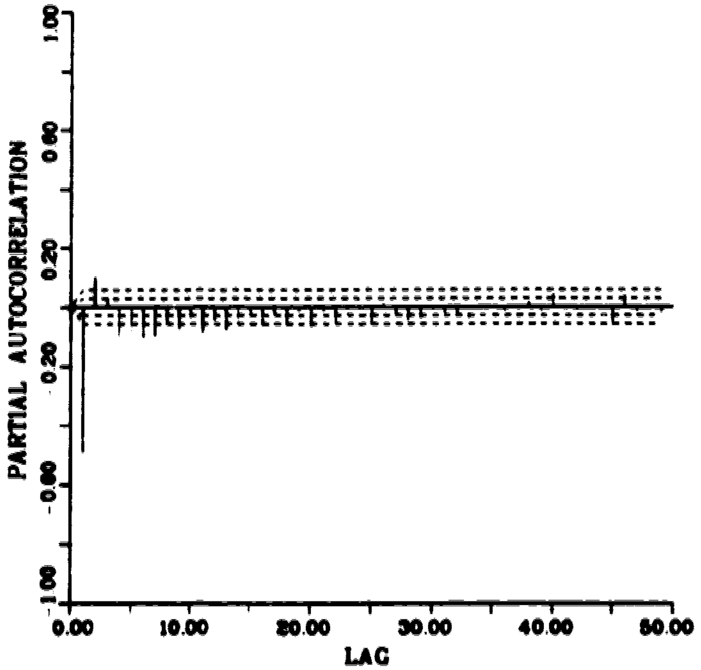
Fig. 12.18 Filtered power consumption

If a high signal to noise ratio is assumed, then the filtered output can be thought of as a time series solely driven by the prewhitened input (now white noise). In this case, the autocorrelation and partial autocorrelation functions of the filtered power consumption suggest the use of 2 or 3 "B" parameters and 1 or 2 "A" parameters. This approximate model order may be confirmed by examining the crosscorrelation function plot, the impulse response and step response of the process presented in Figures 12.21 to 12.23 respectively.



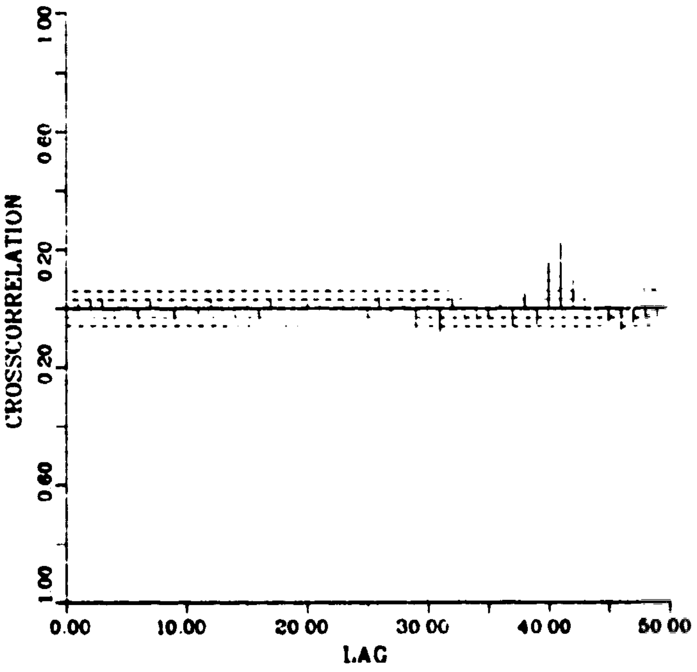
power

Fig. 12.19 Autocorrelogram of filtered power consumption



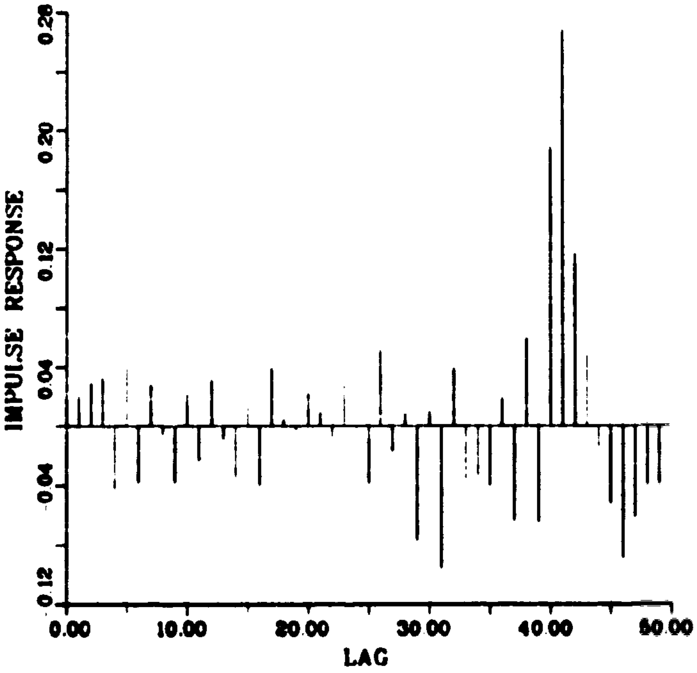
power

Fig. 12.20 Partial autocorrelogram of filtered power consumption



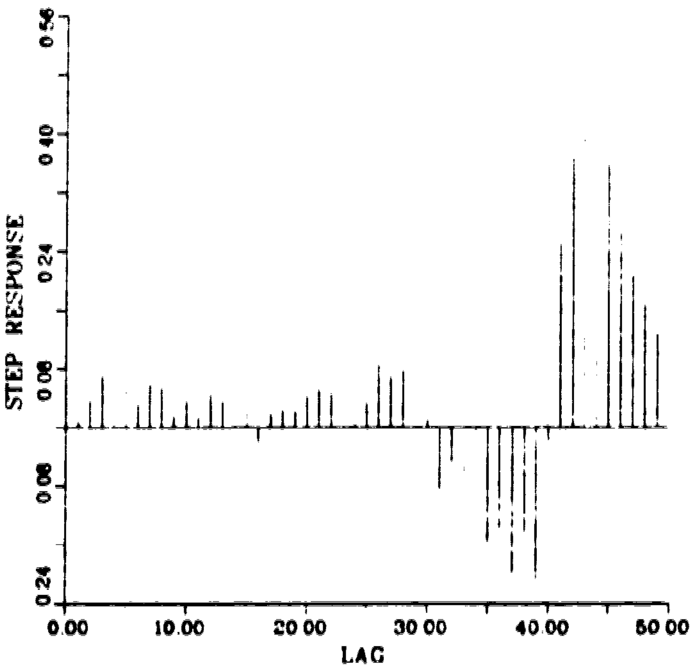
Brenda Mines dec 17

Fig. 12.21 Crusher crosscorrelogram



Brenda Mines dec 17

Fig. 12.22 Crusher impulse response



Brenda Mines dec 17

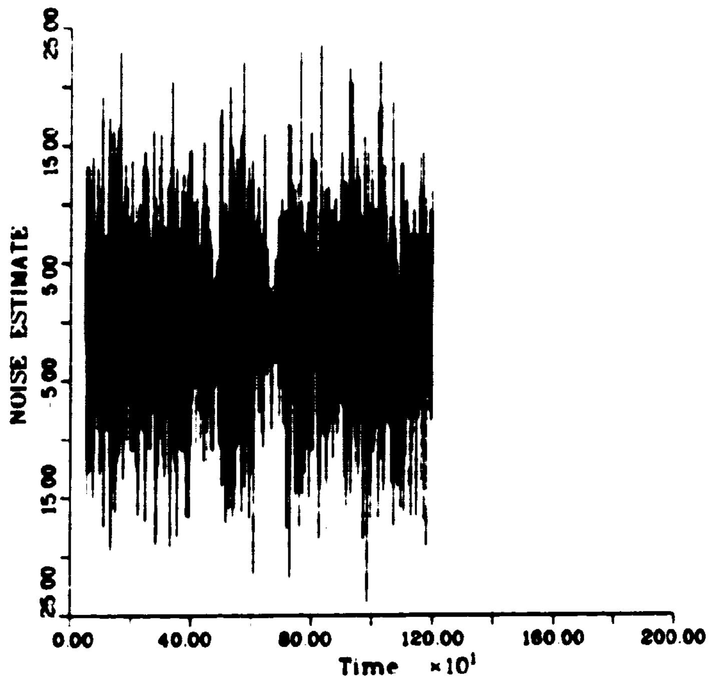
Fig. 12.23 Crusher step response

If the values of the crosscorrelation function at lag 29 and 31 are ignored, as the values are marginal, the delay is estimated to be 39 samples. This is in agreement with the estimated transportation lag between the weightometer and the crusher. From the shape of the crosscorrelation function, the model orders are estimated to be a maximum of 2 "A" parameters, corresponding to a double exponential decay of the crosscorrelation function, and 2 "B" parameters to account for the (possibly two) erratic values before the exponential decay. The step response of the crusher seems to indicate an inverse response (i.e. 2 "B" parameters) but this may be caused by inaccuracies during the reconstruction of the step response from the crosscorrelation function. By using a delay of 39, the inverse response, if any exists, will be avoided. The step response, beyond the delay, indicates an underdamped system (i.e. 2 "A" parameters).

12.6.2 Noise model structure identification

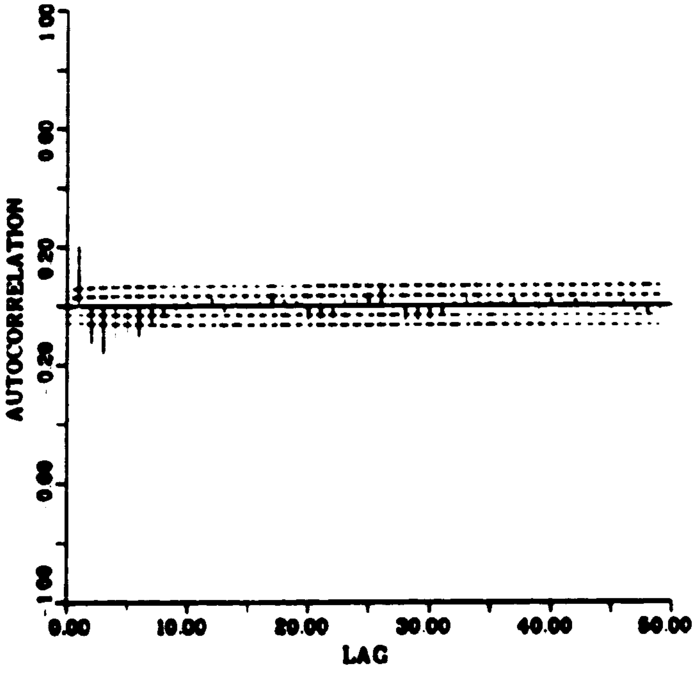
To identify the noise model structure the process transfer function part of the output signal must be removed. The resulting signal is a noise sequence that can be used to identify the noise model structure using standard time series analysis as described in Appendix C. Since the process transfer function parameters are still unknown, the estimated noise sequence is calculated using the impulse response of the crusher instead of the transfer function. Analysis of the noise sequence is presented in Figures 12.24 to 12.28.

The autocorrelation and partial autocorrelation functions of the noise estimate indicate that at least one and possibly two parameters are needed for both noise polynomials, C and D. The spectrum plots provide a clear indication that the noise model should not be ignored.



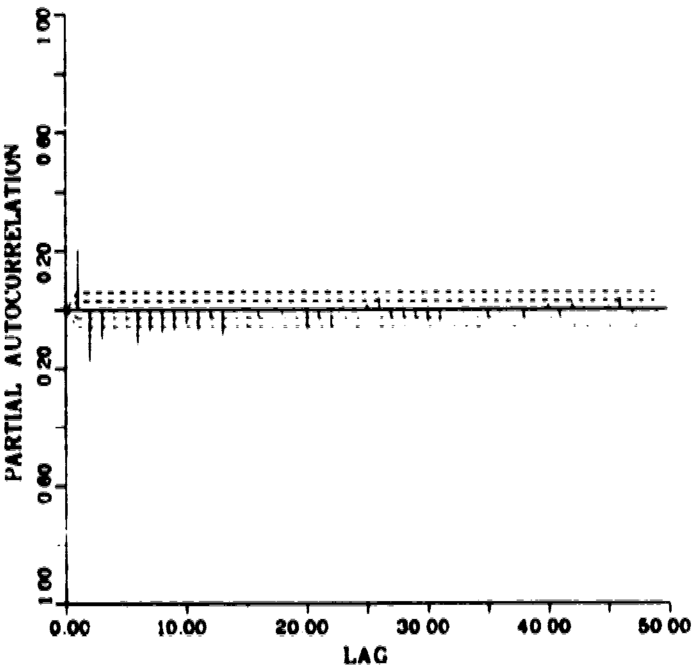
Brenda Mines dec 17

Fig. 12.24 Crusher noise signal estimate



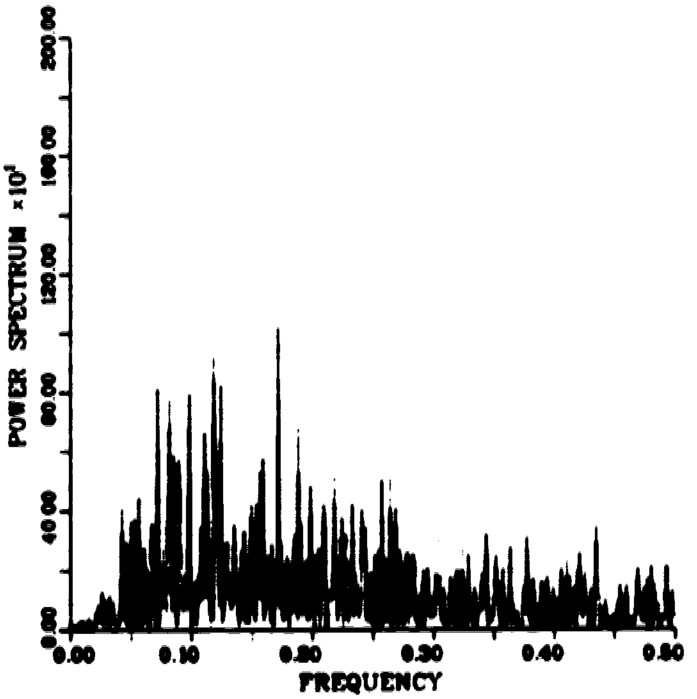
NOISE ESTIMATE

Fig. 12.25 Autocorrelogram of the estimated noise signal



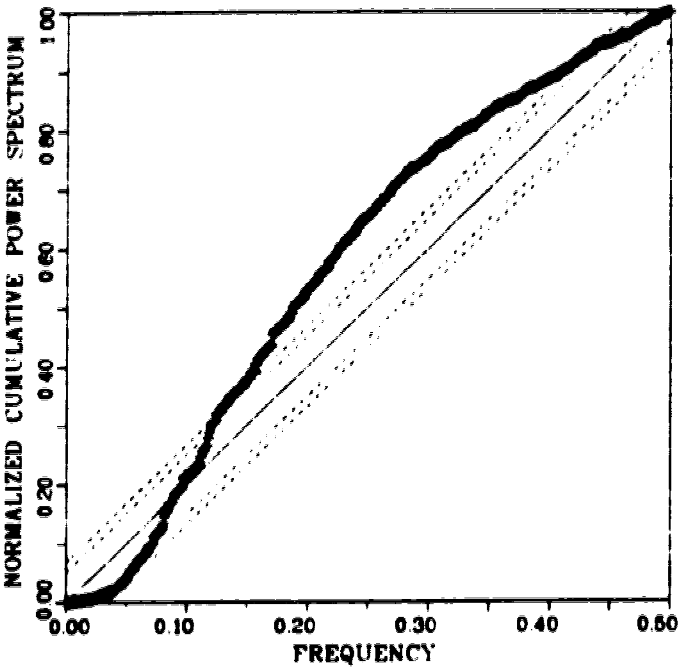
NOISE ESTIMATE

Fig. 12.26 Partial autocorrelogram of the estimated noise signal



NOISE ESTIMATE

Fig. 12.27 Power spectrum of the estimated noise signal



NOISE ESTIMATE

Fig. 12.28 Normalized spectrum of the estimated noise signal

12.7 Model parameter estimation

From the model structures determined in Section 12.5 the first model to be estimated is n=2, m=2, delay=39, nn=2 and mn=2 and is denoted T(n,m,delay)N(nn,dn,mn) or for this specific model T(2,2,39)N(2,0,2) (as described in the PITSA user manual) where 0 indicates that the noise model is not differenced. The only algorithm that can handle such a model is the Box-Jenkins algorithm (c.f. Table 4.1). It should be noted that for numerical accuracy the Box-Jenkins routine in the PITSA program uses the SVD decomposition at each iteration. The routine also checks for numerical singularity. For this model, the parameters are estimated to be

$a_1 = -0.138 \pm 0.214$

$a_2 = 0.078 \pm 0.132$

$b_1 = 0.091 \pm 0.025$

$b_2 = 0.175 \pm 0.046$

$d_1 = -0.711 \pm 0.265$

$d_2 = 0.028 \pm 0.201$

$c_1 = 0.442 \pm 0.263$

$c_2 = 0.137 \pm 0.104$

As can be seen, only three parameters of the eight are found to be statistically significant at the 95% confidence level. This implies that some aspect of the system behavior is left unmodelled or that the data needs some kind of conditioning such as filtering. The most probable cause for this situation is the numerical inaccuracies due to non-zero mean signals i.e. ignoring β to reduce the number of parameters proved to be a costly mistake.

Three methods to deal with the non-zero mean data will be illustrated: differencing (incremental), including a constant parameter and coding.

Differencing is demonstrated first with the model now denoted by F(0,1,0)T(2,2,39)N(2,0,2). The parameters estimated with the PITSA program are now

$a_1 = -0.983 \pm 0.098$

$a_2 = 0.451 \pm 0.055$

$b_1 = 0.165 \pm 0.021$

$b_2 = 0.063 \pm 0.042$

$d_1 = -0.831 \pm 0.072$

$$d_2 = 0.161 \pm 0.056$$

$$c_1 = -0.762 \pm 0.071$$

$$c_2 = -0.213 \pm 0.062$$

As can be observed these results are much improved with the parameter b_2 being the only parameter not statistically significant at a 95% confidence level. Use of the process identification feature of PITSA for a model structure with the b_2 parameter removed, represented as F(0,1,0)T(2,1,39) N(2,0,2), resulted in the following parameter estimates

$$a_1 = -1.172 \pm 0.046$$

$$a_2 = 0.547 \pm 0.036$$

$$b_1 = 0.179 \pm 0.011$$

$$d_1 = -0.856 \pm 0.066$$

$$d_2 = 0.186 \pm 0.050$$

$$c_1 = -0.796 \pm 0.064$$

$$c_2 = -0.182 \pm 0.055$$

and the parameter correlation matrix produced was

a_1	1.00						
a_2	-0.91	1.00					
b_1	0.72	-0.49	1.00				
d_1	-0.01	0.00	-0.00	1.00			
d_2	0.03	-0.03	0.02	-0.89	1.00		
c_1	-0.00	0.00	0.00	0.90	-0.79	1.00	
c_2	0.01	-0.01	0.00	-0.89	0.81	-0.99	1.00

The condition number of this matrix is 494 which, considering the number of parameters, is not that large. Since the parameter c_2 shows a high correlation with the other noise parameters, it might be possible to eliminate this parameter to reduce the model order without affecting the fit of the model. Moreover, eliminating the row and column of C_2 in the above parameter correlation matrix reduces its condition number to 65.

For the model specified as F(0,1,0)T(2,1,39)N(2,0,1), the parameter estimates calculated by the PITSA program were

$$a_1 = -1.163 \pm 0.047$$

$$a_2 = 0.543 \pm 0.037$$

$$b_1 = 0.181 \pm 0.011$$

$$d_1 = -1.027 \pm 0.029$$

$$d_2 = 0.325 \pm 0.028$$

$$c_1 = -0.977 \pm 0.009$$

Examination of the parameter correlation matrix (not presented) did not show any significant correlation between parameters.

Examining the parameter values reveals that the numerator of the noise model is close to differencing. This indicates that differencing was not required and that a different type of conditioning could be suitable. A possible approach would be to include a constant term in the model to account for the nonzero mean data. Although not directly stated in Box and Jenkins, 1976, the use of the β term should always be present unless appropriate action to remove β by subtracting another equation is performed as explained in Chapter 8. If no differencing of the original data is employed, this implies that c_1 will not be required. For no differencing of the original data, the model to be considered would be T(2,1,39)N(2,0,0)Cst (Cst indicates that a constant term, β, is used). For this model, the PITSA program computed the parameter estimates to be

$$a_1 = -1.193 \pm 0.046$$
$$a_2 = 0.584 \pm 0.035$$
$$b_1 = 0.158 \pm 0.008$$
$$d_1 = -1.055 \pm 0.028$$
$$d_2 = 0.316 \pm 0.028$$
$$\beta = -74.961 \pm 0.000$$

This approach leads to a larger parameter correlation matrix condition number (see Table 12.1) which is mostly attributable to correlation of the constant parameter with the a and b parameters. An alternative approach to the use of the constant term would be coding the input and output to obtain zero mean and unity variance data. Keeping β in the coded model does not improve statistical validation criteria and the condition number is slightly increased in this case. If β is ignored then the model is specified as F(0,0,0) T(2,1,39)N(2,0,0) where F(0,0,0) denotes that the data has been coded. The parameter estimates calculated by the PITSA program were

$$a_1 = -1.219 \pm 0.044$$
$$a_2 = 0.602 \pm 0.035$$
$$b_1 = 0.349 \pm 0.021$$
$$d_1 = -1.060 \pm 0.028$$
$$d_2 = 0.309 \pm 0.028$$

A comparison of these three models, using the model validation criteria described in Appendix B, is presented in

Table 12.1. The MAIC validation criteria for the coded model has been converted to an uncoded data equivalent to permit direct comparison. As can be observed, the χ^2 statistic on crosscorrelations are much higher for the nondifferenced models than for the differenced model. An explanation for this may be obtained by examining the crosscorrelation function directly.

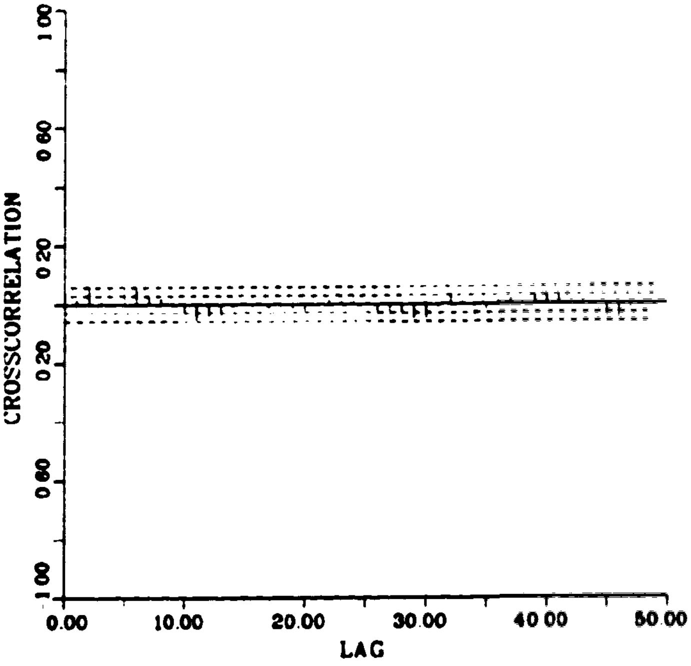
TABLE 12.1

Comparison of various models using validation criteria

Criteria	F(0,1,0)
	T(2,1,39)
	N(2,0,1)
		T(2,1,39)
		N(2,0,0)
		Cst
			F(0,0,0)
			T(2,1,39)
			N(2,0,0)

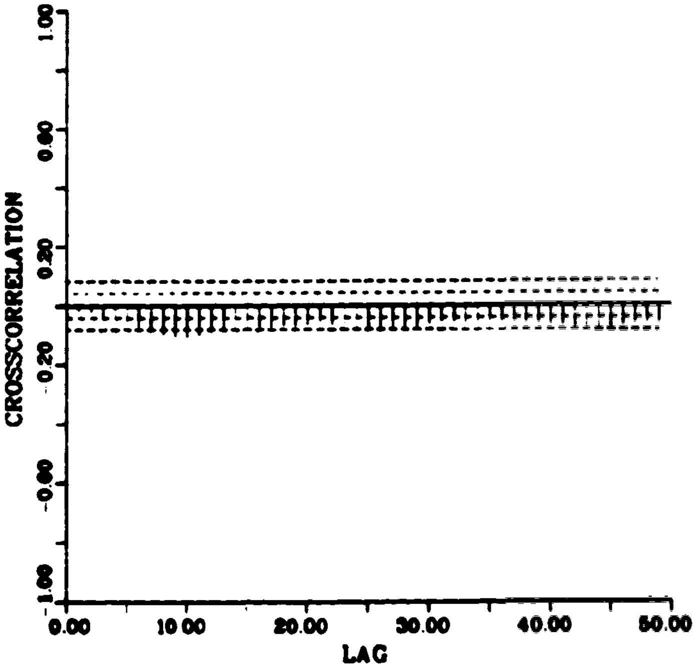
MAIC	4737	4754	4770
χ^2 on autocorrelations	33.8	40.8	38.5
χ^2 on crosscorrelations between input and residuals	48.5	282.4	218.0
Condition number	57	1838	54

The crosscorrelation function between the input and the residuals for the different models generated by the PITSA program are displayed in Figures 12.29 to 12.31. The crosscorrelogram for the model represented by F(0,1,0) T(2,1,39)N(2,0,1) does not show any particularities but the crosscorrelogram of the nondifferenced models do show an effect typically attributed to nonstationary data.



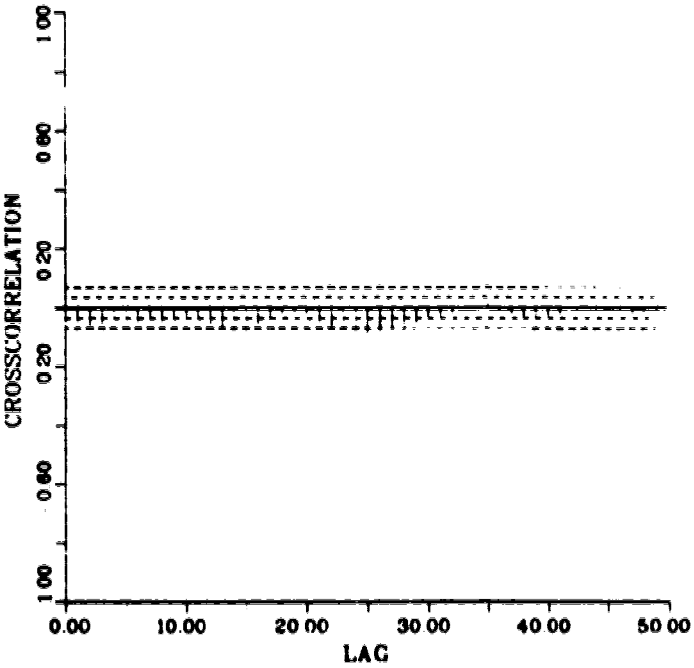
INPUT VS RESIDUALS

Fig. 12.29 Crosscorrelogram for F(0,1,0)T(2,1,39)N(2,0,0)



INPUT VS RESIDUALS

Fig. 12.30 Crosscorrelogram for T(2,1,39)N(2,0,0)Cst



OUTPUT VS RESIDUALS

Fig. 12.31 Crosscorrelogram for F(0,0,0)T(2,1,39)N(2,0,0)

12.8 Comparison with least squares and maximum likelihood

The purpose of this comparison is to see the effect of mismodelling on parameter estimation. Least squares is selected because it is the most commonly used algorithm. Maximum likelihood is also presented since the model it uses is employed in several control schemes such as GPC (Clarke *et al.*, 1987). Differenced data is used as a common ground for comparison (it is also part of the model used in GPC and other control algorithms to provide an integrating function).

The estimated parameters are not directly comparable and therefore will not be presented. For the comparison only the fact that for least squares the model includes 2 a and 1 b parameters and for maximum likelihood the model includes 2 a, 1 b and 1c parameter is important. The validation criteria and condition number for the three models (BJ - Box-jenkins, LS - least squares and ML - maximum likelihood) are presented in Table 12.2. As can be observed the performance of the least squares is much worse (from a statistical point of view). Maximum likelihood performed well considering that the noise model is inappropriate. In all cases the condition number of the parameter correlation matrix is low.

The best method from a statistical point of view is the method with the largest condition number. This is explained by the fact that the models have a different number of parameters (BJ-6, LS-3, ML-4). Therefore the condition number cannot be used to determine the model to select. However it can be used as a validation criteria to ensure that proper care has been taken while calculating the model parameters.

TABLE 12.2

Comparison of various algorithms using validation criteria

Criteria	BJ	LS	ML
MAIC	4737	5060	4757
χ^2 on autocorrelations	33.8	81.3	50.3
χ^2 on crosscorrelations between input and residuals	48.5	244.0	65.1
Condition Number	57	2	22

12.9 Conclusion

The analysis of the Brenda Mines secondary crusher data using the PITSA program has shown that failing to account for the mean of the signals (e.g. through the inclusion of a constant (bias) parameter in the model; or coding of the input and output data; or differencing of the input and output data) can lead to disaster as only 3 out of 8 parameters were found statistically significant. This is in contrast to 7 out of 8 parameters when differencing was used.

Use of differencing or coding drastically reduced the condition number of the parameter correlation matrix when compared to including a constant term in the model.

Eliminating a parameter that was highly correlated with other parameters also drastically reduced the condition number of the parameter correlation matrix without detrimental effect on the fit of the model.

The parameters of three models, which account for the mean of the signals, are

i. F(0,1,0)T(2,1,39)N(2,0,1) (model with differenced data)

$$\nabla y_t = \frac{0.181q^{-1}}{1-1.163q^{-1}+0.543q^{-2}} \nabla u_{t-39} + \frac{1-0.977q^{-1}}{1-1.02q^{-1}+0.325q^{-2}} \epsilon_t$$

ii. T(2,1,39)N(2,0,0)Cst (model with the β parameter)

$$y_t = \frac{0.158q^{-1}}{1-1.193q^{-1}+0.584q^{-2}} u_{t-39} + \frac{1}{1-1.055q^{-1}+0.316q^{-2}} \epsilon_t$$

$$- 74.96$$

iii. F(0,0,0)T(2,1,39)B(2,0,0) (model with coded input and output)

$$(y_t-175.31)/26.97 = \frac{0.349q^{-1}}{1-1.219q^{-1}+0.602q^{-2}}$$

$$x(u_{t-39}-615.55)/61.87$$

$$+ \frac{1}{1-1.060q^{-1}+0.309q^{-2}} \epsilon_t$$

Although most statistical validation criteria were close for the three models, based on the crosscorrelation between the ore feed rate and the model residuals, the model with differenced data is preferred as it removes the nonstationarity character of the data. It also has a much lower condition number than that of the model with a constant term.

Use of the least squares algorithm for this data set (and using differenced data), did not perform well from a statistical point of view. The maximum likelihood algorithm (using differenced data) did perform well in this situation considering that an inappropriate noise model is used. In fact, for this application, the maximum likelihood algorithm is good enough to seriously consider it for use in conjunction with a control algorithm such as GPC for control of the crusher.

13. Conclusion and Recommendations

13.1 Conclusion

Key results from Theorems 1 and 2 are that the P matrix (the inverted matrix in the least squares identification) for a discretized first order system will exhibit a condition number less than 9 but this is not necessarily true for a system of order larger than unity. This is based on the assumption that: (1) the input and output signals are coded (scaled) to provide zero mean and unity variance; (2) the input is uncorrelated with itself in time; (3) data is collected using a period smaller than half the time constant of the system; and (4) the discretization is obtained using a zero order hold.

It is also concluded that both a matrix factorization method and coding (data transformation) are required to obtain a well conditioned P matrix for a system of order larger than unity.

A reduction in the condition number of the P matrix was found to be possible by eliminating the constant (bias) parameter. The following methods of removing the constant parameter were examined to assess their adequacy for use in conjunction with least squares identification.

i. deviation from sample mean. If a fixed sample mean is used then a bias will be present. This bias will diminish as the number of data points increases. In a new result, use of a running mean was shown to produce a smaller bias than that produced by a fixed mean. Therefore the running mean should be preferred over the usual sample mean.

ii. deviation from previous values. The bias was demonstrated to be a function of the noise. An important new result was obtained stating that the bias can be reduced if the deviation is taken from values that are more than the model order plus the asymptotical memory length of the system sample periods in the past.

iii. deviation from filtered values. The use of filtered values as estimates of the mean was demonstrated to be analogous to the use of the running mean. A new analysis of the use of a different filter for the input and output signal as suggested by Sripada and Fisher, 1987 revealed that use of different filters for the input and output signal should not be employed.

iv. deviation from the expected mean. This method will generally produce a bias and therefore is not recommended.

v. deviation from an equation fixed in time. The bias created using this method was found to be a function of the noise and larger than that obtained using the running mean. This method was shown to be not applicable for time varying systems. This method is not recommended except for special circumstances.

A new interpretation of the system of equations solved, in parallel with a new detailed analysis of scaling as proposed by Sripada and Fisher, 1987, demonstrated that scaling of the P matrix does not improve accuracy and only increases the number of computations.

An important conclusion of this thesis is that the condition number of the P matrix cannot be used as a measure of persistency of excitation as proposed by Sripada and Fisher, 1987.

It was concluded that the best coding (data transformation) scheme produces a coded signal mean of zero and variance of unity.

Use of a step input was noticed to produce a higher condition number than a square wave or a PRBS input if there is more than one b parameter in the model. This effect is amplified for a system order larger than unity.

In a new interpretation, use of normalization in conjunction with the least squares identification technique can be viewed as a weighted least squares identification in which the weights are selected in such a manner as to reduce the signal to noise ratio (assuming that the noise magnitude does not vary with the signal amplitude). This is obviously an undesirable situation for identification.

It was concluded that no tangible effect of normalization on the numerical conditioning could be observed contrary to the claims of Sripada and Fisher, 1987.

It was demonstrated that alternatives to normalization can be used to attain stable adaptive control operation of a simulated example. This suggests that normalization is not a "necessary evil" to ensure the stability of an adaptive control system.

Furthermore it is concluded that an alternative such as filtering the parameters before use in the control law should be preferred to normalization for the simulated example.

Several factorization methods that are often used by numerical analysts to improve numerical conditioning were examined. These are: square root, U-D, singular value decomposition and QR. These methods were found to provide comparable numerical enhancements namely the reduction of the condition number to its square root.

Reconstructing the P matrix to ensure positiveness and symmetry (RPRLS algorithm) did not provide any significant improvements over the recursive least squares (RLS) algorithm for a simulated example. Improvements from reconstruction will occur only in cases of very ill-conditioned systems since the only advantage is that the matrix is guaranteed to be positive and symmetric. For this example, utilization of the recursive UD factorization did provide a significant reduction in the condition number over RLS and RPRLS. The parameter estimates were similar for the three algorithms because the condition number obtained for RLS was not high enough to significantly affect the parameter estimates for the number of input/output data points used.

From the perturbation analysis performed by numerical analysts it is known that the numerical sensitivity of the calculations involving a matrix is related to the condition number of the matrix.

Analysis of the effect of the off-diagonal elements of the P matrix, not examined in detail in the published literature, has demonstrated the approximately exponential effect of the magnitude of the off-diagonal elements (relative to the magnitude of the diagonal elements) on the condition number of the P matrix.

It was also established that several off-diagonal elements cannot be combined in such a manner as to reduce the condition number.

A new result stating that the combined effect of several off-diagonal elements of the P matrix on the condition number becomes greater than the sum of the effects of the individual off-diagonal elements as their magnitude increases was obtained.

In the thesis, it was argued that to minimize the effect of the magnitude of the diagonal elements relative to each other (which also prevents generalizations using the condition number) the variance of the input and output data should be of the same magnitude.

Investigation of the most common nonrecursive and recursive algorithms revealed that most algorithms involve the inversion of a matrix or its recursive equivalent pointing to the wide applicability of the results of the thesis.

This thesis forms a unique document in which most common identification algorithms are presented with their derivation. It is also a unique source of information on practical issues in process identification.

Finally, during the analysis of crusher data it was found that ignoring the mean of the signals (i.e. simply neglecting the constant parameter rather than eliminating it from the model by substitution) can lead to disaster as only 3 out of 8 parameters were found statistically significant as opposed to 7 out of 8 parameters when differencing was used. Use of differencing or coding drastically reduced the condition number of the parameter correlation matrix when compared to including a constant term in the model. Eliminating a parameter that was highly correlated with other parameters drastically reduced the condition number of the parameter correlation matrix (normalized P matrix) without detrimental effect on the fit of the model.

Use of the least squares algorithm for this data set (and using differenced data), did not perform well from a statistical point of view. The maximum likelihood algorithm (using differenced data) did perform well in this situation considering that an inappropriate noise model is used. In fact, for this application, the maximum likelihood algorithm is good enough to seriously consider it for use in conjunction with a control algorithm such as GPC for control of the crusher.

13.2 Recommendations

Most of the recommendations for future work, based on this study, can be grouped in the areas of program modifications and enhancements and improved identification techniques for development of dynamic models.

Future projects that would lead to the development of more reliable dynamic models are the following:

i. Exploring alternatives to normalization with a view to prove the stability of the resulting adaptive control system and assuring that they do not hinder the identification step.

ii. Investigation of the relationship between signal to noise ratio and the number of data points required to establish a suitable dynamic model. Some of the questions to be answered are: can an increase in the number of data points compensate for a low signal to noise ratio? If so, can it be quantified? If the number of data points is limited what is the required signal to noise ratio to produce parameters that are satisfactory? Sensitivity analysis of the parameter estimates with respect to signal to noise ratio and the number of data points should be undertaken to establish the minimum requirements to limit this sensitivity.

iii. Effect of the number of parameters. Does an increase in the number of parameters require a corresponding increase in signal to noise ratio or number of data points in order to produce satisfactory parameters? A sensitivity analysis of the parameter estimates with respect to the number of parameters is difficult to establish since the model is changed but perhaps the various validation criteria may be used for this purpose.

iv. Selection of the PRBS sequence. The effect of the clock interval; length of the PRBS sequence and amplitude of the input signal on the estimated parameters and on the validation criteria should be investigated. This could provide valuable guidelines to design a PRBS signal for an identification run. Use of a PRBS sequence based on a random number generator rather than a maximum length sequence should be explored to determine if there are any benefits to using such a scheme.

v. Objective assessment of the acceptability of the identified model. Determining if the estimated model is satisfactory for its intended use is somewhat subjective. Validation criteria should be examined to determine the amount of subjectivity (or bias towards certain types of models against other types of models) inherent in their use. For this purpose several validation criteria are available in the literature (Freeman, 1985, Isermann *et al.*, 1974, Ljung and Soderstrom, 1983, Stoica *et al.*, 1986, Box and Jenkins, 1976) and these should be compared to determine which criteria will be able to discriminate between candidate models in general applications (i.e. with the least amount of subjectivity).

vi. A need exists for a reliable method that can be used to find model orders and transformations required to linearize nonlinear systems. This entails the development of a systematic method to test the type of transformation that is required (e.g. logarithm) and the type of nonlinearity that should be included in the model (e.g. which terms of the Volterra series should be included in the model). This method would dictate the terms to include in the regressor (e.g. $\phi=[y_{t-1}, y_{t-2}^2, y_{t-5}u_{t-3}, u_{t-4}]$) so that a model that is linear in the parameters is obtained. Through proper formulation of the regressor the parameter estimates could be obtained from the usual estimation algorithms.

Once the method has been refined, the development of an expert system to guide its application would be worthwhile.

Although it no longer makes sense to write your own process identification program (it is now much easier to buy an adequate program commercially) the PITSA program is still a usefull tool. The suggestions for enhancements and modifications to the program are as follows:

i. Not all of the algorithms available in the PITSA program have been employed for the systems analyzed in this work. More extensive experimentation with the various algorithms available should be undertaken to examine the effectiveness of each algorithm. A well established procedure similar to that presented by Isermann *et al.*, 1974 should be employed so that the results may be directly compared to those presented in the literature.

ii. The PITSA program should be expanded to include identification of multi-input single-output (MISO) and multi-input multi-output (MIMO) systems as they become more prominent in control applications. This should include extending correlation analysis to permit estimation of model orders in a MISO or MIMO context.

iii. The weighted recursive least squares algorithm in PITSA uses a fixed forgetting factor or a combination of two fixed forgetting factors. Other forgetting factors published in the literature should be incorporated in the program so that the relative merit of the forgetting factor schemes may be established.

iv. Current implementation of the PITSA program does not include factorization of all the algorithms.

The algorithms that are not factorized should be replaced by the more numerically robust form of the algorithm.

v. Use of the PITSA program is currently limited to University of Alberta main frame computer users that have access to a graphics terminal or to a personal computer that supports graphics terminal emulation. The only remaining stumbling block to achieve portability of the program is the graphics support.

Bibliography

Ahmed, M.S. (1984). Fast GLS Algorithm for Parameter Identification for Off Line Estimation. *Automatica*, 20, 231-236.

Ahmed, M.S. (1985). Computationally Efficient IV Algorithm for Structure and Parameter Identification of Linear Systems. *IEE Proc.*, 132 D, 37-44.

Aling, H., U. Milletti, R.L. Kosut, M.P. Mesaros, P. Van Overschee and B. De Moore (1993). Interactive System Identification Module for Xmath (ISID). *Proc. Amer. Contr. Conf.* 3071-3074.

Astrom, K.J. (1981). Maximum Likelihood and Prediction Error Methods. In *Trends and Progress in System Identification* P. Eykhoff ed. Pergamon Press.

Astrom, K.J. (1983). Theory and Applications of Adaptive Control - A Survey. *Automatica*, 19, 471-486.

Astrom, K.J. and P. Eykhoff (1971). System Identification - A Survey. *Automatica*, 7, 123-162.

Astrom, K.J., P. Hangander and J. Sternby (1984). Zeros of Sampled Systems. *Automatica*, 10, 81-103.

Aude, E.P.L. and D.J. Sandoz, (1986). Analysis of the Identification of Closed-Loop Systems Using Least-Squares Methods. *Int. J. Control*, 43, 561-585.

Baur, B. and H. Unbehauen (1979). On-Line Identification of a Heat Exchanger in Closed Loop Using a Modified Instrumental Variable Method. *5th IFAC Ident. and Sys. Par. Estim.* 351-359.

Bierman, G.J. (1977). *Factorization Methods for Discrete Sequential Estimation*. Academic Press.

Bittanti, S. and M. Campi (1991). Tuning the Forgetting Factor in RLS Identification Algorithms. *Proc. IEEE Conf. on Decision and Control*, 2, 1688-1689.

Box, G.E.P. and G.M. Jenkins (1976). *Time Series Analysis: Forcasting and Control*. Revised edition. Holden-day.

Box, G.E.P. and J.F. Mac Gregor (1974). The Analysis of Closed-Loop Dynamic-Stochastic Systems. *Technometrics*, 16, 391-398.

Caralp, L., G. Defaye, G. Durou and P. Jouve (1974). A Statistical Dynamic Identification Package by a Single Output Difference Equation Model Applied to Industrial Processes on Micro-Computers. *IFAC 9th Triennial World Congress, Budapest , Hungary*. 489-494.

Carayannis, G., D.G. Manolakis and N. Kalouptsidis (1983). A Fast Sequencial Algorithm for Least Squares Filtering and Prediction. *IEEE Trans. on Acous., Speech and Signal Proces.*, ASSP-31, 1394-1402.

Chalam, V.V. (1987). *Adaptive Control Systems: Techniques and Applications*. Marcel Dekker.

Clarke, D.W. (1967). Generalized Least Squares Estimation of the Parameters of a Dynamic Model. *IFAC Symp. on Ident. Prague*. paper 3.17.

Clarke, D.W., A.J.F. Hodgson and P.S. Tuffs (1983). Offset Problem and k-Incremental Predictors in Self-Tuning Control. *IEE Proc.*, 130-D(5), 217-225.

Clarke, D.W., C. Mohtadi and P.S. Tuffs (1987). Generalized Predictive Control - part I. The Basic Algorithm. *Automatica*, 23, 137-148.

Cluett, W.R., J.M. Martin-Sanchez, S.L. Shah and D.G. Fisher (1986). Stable Discrete-Time Adaptive Control in the Presence of Unmodelled Dynamics. *Preprints, Amer. Contr. Conf.* 713-720.

Cluett, W.R., S.L. Shah and D.G. Fisher (1987). Robust Design of Adaptive Control Systems Using Conic Sector Theory. *Automatica*, 23, 221-224.

Corradi, C. (1979). A Note on the Computation of Maximum Likelihood Estimates in Linear Regression Models with Autocorrelated Errors. *J. of Econometrics*, 11, 303-317.

Cryer, J.D. (1986). *Time Series Analysis*. Duxbury Press.

Davies, W.D.T. (1966). Generation and Properties of Maximum-Length Sequences part 1, 2 and 3. *Control*, June, July and August, 302-304, 364-365 and 431-433.

De Larminat, P. (1985). Adaptive Control Based on Instrumental Variable Identification. *7th IFAC Ident. and Sys. Par. Estim.* 599-603.

Denham, M.J., B.A. Abaza, D.C.J. Bloomer and L. Rigby (1975). *System Identification Package (SYSID)*. Publication 75/28, Imperial College, London.

Dennis, J.E. Jr. and R.B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall.

Dongarra, J.J., C.B. Moler, J.R. Bunch and G.W. Stewart (1979). *Linpack Users' Guide*. SIAM.

Draper N.R. and H. Smith (1981). *Applied Regression Analysis, 2nd ed.* Wiley.

Dugard, L. and I.D. Landau (1980). Recursive Output Error Identification Algorithms Theory and Evaluation. *Automatica*, 16, 443-462.

Edwards, R. (1986). Personal communication.

Eykhoff, P. (ed.) (1981). *Trends and Progress in System Identification*. Pergamon Press.

Eykhoff, P. (1974). *System Identification: Parameter and State Estimation*. John Wiley & Sons.

Foley, M.W. (1988). *Adaptive Control Using a Kalman Filter*. M.Sc. Thesis. University of Alberta.

Forsythe, G.E., M.A. Malcolm, C.B. Moler (1977). *Computer Methods for Mathematical Computations*. Prentice Hall.

Freeman, T.G. (1985). Selecting the Best Linear Transfer Function Model. *Automatica*, 21, 361-379.

Gentil, S., A.Y. Barraud and K. Szafnick (1990). SEXI: An Expert Identification Package. *Automatica*, 26, 803-809.

Gentil, S., J.P. Sandrez and C. Foulard (1973). Different Methods for Dynamic Identification of an Experimental Paper Machine. *3rd IFAC Symp. on Ident. and Sys. Par. Estim.* PV-1 473-483.

Gertler, J. and C. Banyasz (1974). A Recursive (On-Line) Maximum Likelihood Identification Method. *IEEE Trans. on Aut. Contr.*, AC-19, 816-820.

Golub, G.H. and C.F. Van Loan (1983). *Matrix Computations*. Johns Hopkins.

Goodwin, G.C. and K.S. Sin (1984). *Adaptive Filtering Prediction and Control*. Prentice Hall.

Goodwin, G.C. and R.L. Payne (1977). *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press.

Graupe, D., V.K. Jain and J. Salahi (1980). A Comparative Analysis of Various Least Squares Identification Algorithms. *Automatica*, 16, 663-681.

Gupta, N.K., Mehra, R.K. (1974). Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations. *IEEE Trans. on Aut. Contr.*, AC-19, 774-783.

Gustavsson, I. (1972). Comparison of Different Methods for Identification of Industrial Processes. *Automatica*, 8, 127-142.

Gustavsson, I. (1975). Survey of Applications of Identification in Chemical and Physical Processes. *Automatica*, 11, 3-24.

Haest, M., G. Bastin, M. Gevers and V. Wertz (1990). ESPION. An Expert System for System Identification. *Automatica*, 26, 85-95.

Hastings-James, R. and M.W. Sage (1969). Recursive Generalized Least Squares Procedure for Online Identification of Process Parameters. *IEE Proc.*, 116, 2057-2062.

Householder, A.S. (1964). *The Theory of Matrices in Numerical Analysis*. Ginn (Blaisdell), Waltham, Massachusets.

Hsia, T.C. (1977). *System Identification: Least Squares Methods*. Lexington Books.

Isermann, R. (1980a). Practical Aspects of Process Identification. *Automatica*, 16, 575-587.

Isermann, R. (1980b). *Digital Control Systems*. Springer Verlag.

Isermann, R. (1982). Parameter Adaptive Control Algorithms-A Tutorial. *Automatica*, 18, 513-528.

Isermann, R., U. Baur, W. Bamberger, P. Kneppo and H. Siebert (1974). Comparison of Six On-Line Identification and Parameter Estimation Methods. *Automatica*, 10, 81-103.

James, P.N., P. Souter and D.C. Dixon (1974). A Comparison of Parameter Estimation Algorithms for Discrete Systems. *Chemical Engineering Science*, 29, 539-547.

Klema, V.C. and A.J. Laub (1980). The Singular Value Decomposition: Its Computation and Some Applications. *IEEE Trans. on Aut. Contr.*, AC-25, 164-176.

Knuth, D.E. (1973). *The Art of Computer Programming 2nd ed.* Addison-Wesley.

Kurz, H., R. Isermann and R. Schumann (1980). Experimental Comparison and Application of Various Parameter-Adaptive Control Algorithms. *Automatica*, 16, 117-133.

Landau, I.D. (1990). *System Identification and Control Design Using P.I.M. + Software*. Prentice Hall.

Laporte, P., A. Barraud and S. Gentil (1985). An Interactive Computer Aided Identification Package. *7th IFAC Ident. and Sys. Par. Estim., York U.K.* 581-586.

Laub, A.J. (1985). Numerical Linear Algebra Aspects of Control Design Computations. *IEEE Trans. on Aut. Contr.*, AC-30, 97-108.

Lawson, C.L. and R.J. Hanson (1974). *Solving Least Squares Problems*. Prentice Hall.

Levenberg, K. (1944). A Method for the Solution of Certain Nonlinear Problems in Least Squares. *Quart. Appl. Math.*, 2, 164-168.

Ljung, L. (1977). Analysis of Recursive Stochastic Algorithms. *IEEE Trans. on Aut. Contr.*, AC-31, 551-575.

Ljung, L. (1980). Comments on "Non-Convergence of the Approximate Maximum Likelihood Identification Algorithm". *Automatica*, 16, 219-220.

Ljung, L. (1987a). On the Estimation of Transfer Functions. *IFAC 10th World Congress on Automatic Control*, 10, 271-281.

Ljung, L. (1987b). *System Identification - Theory for the User*. Prentice Hall.

Ljung, L. (1991). Issues in System Identification. *IEEE Control Systems Magazine*, 11, no. 1, 25-29.

Ljung, L. (1993). Perspectives on the Process of Identification. *IFAC 12 th. World Congress*, 5, 197-205.

Ljung, L. and T. Soderstrom (1983). *Theory and Practice of Recursive Identification*. MIT Press.

Lozano-Leal, R. and G.C. Goodwin (1985). A Globally Convergent Adaptive Pole Placement Algorithm Without a Persistency of Exitation Requirement. *IEEE Trans. on Aut. Contr.*, AC-30, 795-798.

MacGregor, M.H. (1984). *User's Manual*. Internal report, University of Alberta.

Malik, O.P., G.S. Hope and S.J. Cheng (1991). Some Issues on the Pratical Use of Recursive Least Squares Identification in Self-Tuning Control. *Int. J. Contr.*, 53, 1021-1033.

Martin-Sanchez, J.M., S.L. Shah and D.G. Fisher (1984). A Stable Adaptive Predictive Control System. *Int. J. Control*, 39, 215-234.

Matko, D. and R. Schumann (1982). Comparative Stochastic Convergence Analysis of Seven Recursive Parameter Estimation Methods. *6th IFAC Ident. and Sys. Par. Estim., Washington, D.C., USA*. 579-584.

Moler, C., J. Little, S. Bangert and S. Kleiman (1986). *PC-MATLAB* version 2.2. The MathWorks Inc.

Mozel, J.C. (1980). Parameter Estimation Using the University of Alberta Identification Program. *M.Sc. Thesis*. University of Alberta.

Naik, S.M., P.R. Kumar and B.E. Ydstie (1992). Robust Continuous-Time Adaptive Control by Parameter Projection. *IEEE Trans. on Aut. Contr.*, AC-37, 182-197.

Nelson, C.R. (1973). *Applied Time Series Analysis for Managerial Forecasting*. Holden-Day.

Niu, S. (1994). Augmented UD Identification for Process Control. *Ph. D. Thesis*. University of Alberta.

Niu, S. and D.G. Fisher (1992). Sequential and Partial Updating of Parameters During Estimation. *Proc. Amer. Contr. Conf.* 3176-3180.

Niu, S., D. Xiao and D.G. Fisher (1990). Recursive Algorithm for Simultaneous Identification of Model Order and Parameters. *IEEE Trans. on Accoustic. Speech. and Signal Processing*, ASSP-38, 884-886.

Niu, S., D.G. Fisher and D. Xiao (1992). An Augmented UD Identification Algorithm. *Int. J. Contr.*, 56, 193-211.

Norton, J.P. (1986). *An Introduction to Identification*. Academic Press.

Ortega, R., L. Praly and I.D. Landau (1985). Robustness of Discrete-Time Direct Adaptive Controllers. *IEEE Trans. on Aut. Contr.*, AC-30, 1179-1187.

Pandit, N.J. and S.M. Wu (1983). *Time Series and Systems Analysis with Applications*. Wiley.

Panuska, V. (1980). Non-Convergence of the Approximate Maximum Likelihood Identification Algorithm. *Automatica*, 16, 215-217.

Panuska, V. (1980b). Author's Reply to Comments on "Non-Convergence of the Approximate Maximum Likelihood Identification Algorithm". *Automatica*, 16, 221.

Parkum, J.E., N.K. Poulsen and J. Holst (1992). Recursive Forgetting Algorithms. *Int J. Contr.*, 55, 109-128.

Priestley, M.B. (1981). *Spectral Analysis and Time Series Analysis*. Academic Press.

Ramambason, O.C., O.D. Crisalle and D. Bonvin. (1992). On Using the Parameter Covariance for Improving the Recursive Least Squares Algorithm. *Proc. Amer. Contr. Conf.* 1431-1435.

Rivera, D.E. (1991). Control-Relevant Parameter Estimation: A Systematic Procedure for Prefilter Design. *Proc. Amer. Contr. Conf.* 237-241.

Rivera, D.E., J.F. Pollard, L.E. Sternman and C.E. Garcia (1990). An Industrial Perspective on Control-Relevant Identification. *Amer. Contr. Conf.*, 3, 2406-2411.

Rogers, M. (1989). Recursive Identification Methods. *M.Sc. Thesis*. University of Alberta.

Sadighi, I. and D.A. Pierre (1990). Corrector Least Squares (CLS) Identification Algorithm for Adaptive Control. *Proc. Amer. Contr. Conf.* 1901-1904.

Saridis, G.N. (1974). Comparison of Six On-Line Identification Algorithms. *Automatica*, 10, 69-79.

Shah, S.L. and W.R. Cluett (1991). Recursive Least Squares Based Estimation Schemes for Self-Tuning Control. *Canadian J. of Chem. Eng.*, 69, 89-96.

Sherif, M.H. and L.-M. Liu (1984). Recursive Parameter Estimation of Transfer Function Models. *Int. J. Control*, 40, 499-518.

Shook, D.S. (1991). Identification Issues in Long Range Predictive Control. *Ph. D. Thesis*. University of Alberta.

Shook, D.S., C. Mohtadi and S.L. Shah (1991). Identification for Long-Range Predictive Control. *IEE Proc.*, 138 D, 75-84.

Shook, D.S., C. Mohtadi and S.L. Shah (1992). A Control-Relevant Identification Strategy for GPC. *IEEE Trans. on Auto. Contr.*, 37, 975-980.

Sima, V. (1984). Factorization Techniques in Discrete Time Adaptive Control. *IFAC 9th Triennial World Congress, Budapest, Hungary*. 1029-1034.

Simpson, K. (ed.) (1993). *1994 CEP Software Directory*. Chemical Engineering Progress.

Sinha, N.K. and A. Sen (1975). Critical Evaluation of Online Identification Methods. *Proc. IEE.*, 122, 1153-1158.

Soderstrom, T. (1973). An On-Line Algorithm for Approximate Maximum Likelihood Identifcation of Linear Dynamic Systems. *Report 7308*. Lund Institute of Technology.

Soderstrom, T. (1977). On Model Structure Testing in System Identification. *Int. J. Control*, 26, 1-18.

Soderstrom, T. (1987). Instrumental Variable Methods for Closed Loop Systems. *IFAC 10th World Cong. on Auto. Control*, 10, 364-369.

Soderstrom, T. and P. Stoica (1981). Comparison of Some Instrumental Variable Methods - Consistency and Accuracy Aspects. *Automatica*, 17, 101-15.

Soderstrom, T. and P. Stoica (1989). *System Identification*. Prentice Hall.

Soderstrom, T., L. Ljung and I. Gustavsson (1978). A Theoretical Analysis of Recursive Identification Methods. *Automatica*, 14, 231-244.

Solbrand, G., A. Ahlen and L. Ljung (1985). Recursive Methods for Off-Line Identification. *Int. J. Control.*, 41, 177-191.

Sripada, N.R. and Fisher, D.G. (1987). Improved Least Squares Identification. *Int. J. Control.*, 46, 1899-1913.

Stephanopoulos, G. (1984). *Chemical Process Control*. Prentice Hall.

Stoica, P., P. Eykhoff, P. Jansen and T. Soderstrom (1986). Model Structure Selection by Cross-Validation. *Int. J. Control.*, 43, 1841-1878.

Strejc, V. (1980). Least Squares Parameter Estimation. *Automatica*, 17, 535-550.

Strejc, V. (1981a). Least Squares and Regression Methods. in *Trends and Progress in System Identification*, P. Eykhoff ed. Pergamon Press.

Strejc, V. (1981b). Trends in Identification. *Automatica*, 17, 7-21.

Stuart, G.W. (1973). *Introduction to Matrix Computations*. Academic Press.

Systems Technology Inc. (1988). *Sales litetarure*.

Thisted, R.A. (1986). *Elements of Statistical Computing: Numerical Considerations*. Chapman and Hall.

Van den Boom, A.J.W. and R. Bollen (1984). The Identification Package Sater. *IFAC 9th Triennial World Congress*. 483-488.

Veres, S.M. (1990a). Relations Between Information Criteria for Model Structure Selection. Part 1: The Role of Baysian Model Order Estimation. *Int. J. Contr.*, 52, 389-408.

Veres, S.M. (1990b). Relations Between Information Criteria for Model Structure Selection. Part 2: Modelling by Shortest Data Description. *Int. J. Contr.*, 52, 409-422.

Verhaegen, M.H. (1989). Round-Off Error Propagation in Four Generally Applicable, recursive, Least Squares Estimation Schemes. *Automatica*, 25, 437-444.

Vermeer, P.J. (1987). *Design and Evaluation of Practical Self-Tuning PID Controllers*. M.Sc. Thesis, University of Alberta.

Wilkinson, J. (1963). *Rounding Errors in Algebraic Processes*. Prentice Hall.

Wilkinson, J. (1965). *The Algebraic Eigenvalue Problem*. Clarendon Press.

Willems, J.C. (1986a). From Time Series to Linear Systems - Part I. Finite Dimensional Linear Time Invariant Systems. *Automatica*, 22, 561-580.

Willems, J.C. (1986b). From Time Series to Linear Systems - Part II. Exact Modelling. *Automatica*, 22, 675-694.

Willems, J.C. (1987). From Time Series to Linear Systems - Part III. Approximate Modelling. *Automatica*, 23, 87-115.

Wong, M.Y., M.M. Bayoumi and S. Nuyan (1983). Comparative Study of Some Estimation Algortihms for the Self-Tuning Control of Time Varying Systems. *ACI 83*. 18.12-18.18.

Young, P.C. (1970). An Instrumental Variable Method for Real-Time Identification of a Noisy Process. *Automatica*, 6, 271-287.

Young, P.C. (1984). *Recursive Estimation and Time Series Analysis - an Introduction*. Springer Verlag.

Young, P.C. and J. Hampton (1984). New CAPTAIN Computer Program Packages for Identification, Estimation and Signal Processing. *IEE colloquium recent advances in identification and signal processing: part I digest 1984/19*. 6.1-6.6.

Yung, W.K. and K.F. Man (1993). Optimal Selected Forgetting Factor for RLS Estimation. *IFAC 12th World Congress*, 9, 1-4.

Appendix A: CHOICE OF INPUT SIGNAL AND SAMPLING PERIOD

A.1 Type of input signal

Process models can be identified in a number of ways. The traditional techniques of step and pulse testing are suitable for processes where the noise component is small relative to the magnitude of the pulse or step. However, in processes where the amount of noise present is significant, the magnitude of the step or pulse necessary to produce usable results can be detrimental to the operation of the process. Large manipulations can also drive the process into regions of nonlinear operation. Statistical methods of process identification avoid these problems by enabling the use of small perturbations to the process (e.g. white noise, PRBS, etc.)

It is difficult to define an optimum input signal *a priori*. If the process model were available, then an optimal input could be designed (Goodwin and Sin, 1984). Unfortunately this is seldom the case. A good identification method should be insensitive to the characteristics of the input signal but some identification methods do require a persistently exiting signal. For a discussion of persistently exiting signals the reader is referred to Isermann, 1980a and Astrom and Eykhoff, 1971.

An example of a poor combination of an identification algorithm and input signal is that of a least squares algorithm combined with a step input. This can be illustrated by choosing a single step as an input signal. The difficulty of using a step with the least squares algorithm stems from the fact that the input does not change (or may change only once if steady state values are used at the beginning of the data collection). The least squares algorithm utilizes the correlation between the input and output to find parameter estimates and no correlation between the input and output can be found when the input does not change. Moreover the condition number of the P matrix will be larger than a square wave or a PRBS if there is more than one b parameter in the model (c.f. Remark 2 and 10 in Chapter 7).

This is well illustrated by the results presented in Table A.1 for nonrecursive least squares identification performed on simulated data, for the discretization of the system described by

$$\frac{y(s)}{u(s)} = \frac{2}{s+1}\,\frac{229}{s^2 + 30s + 229} \qquad \text{(A.1)}$$

using different input signals and noise levels. The different input signals are: a PRBS (pseudo random binary sequence) of length 127 and of unity amplitude; a step that starts at sample 10 (to avoid singularity) and of unity amplitude and finally a reversing square wave (square wave with zero mean) of period of 100 samples and unity amplitude. The sampling period used was 0.12 seconds. The step starts at 1.2 seconds and the square wave period is 12 seconds. The length of each run was set at 500 points which translates into a PRBS that was repeated almost 4 times and a square wave that lasted 5 cycles. The noise was generated using a normal, zero mean, unity variance distribution and was scaled to provide different standard deviations (s.d.). The input/output data for a noise level of .01 s.d. is presented in Figure A.1.

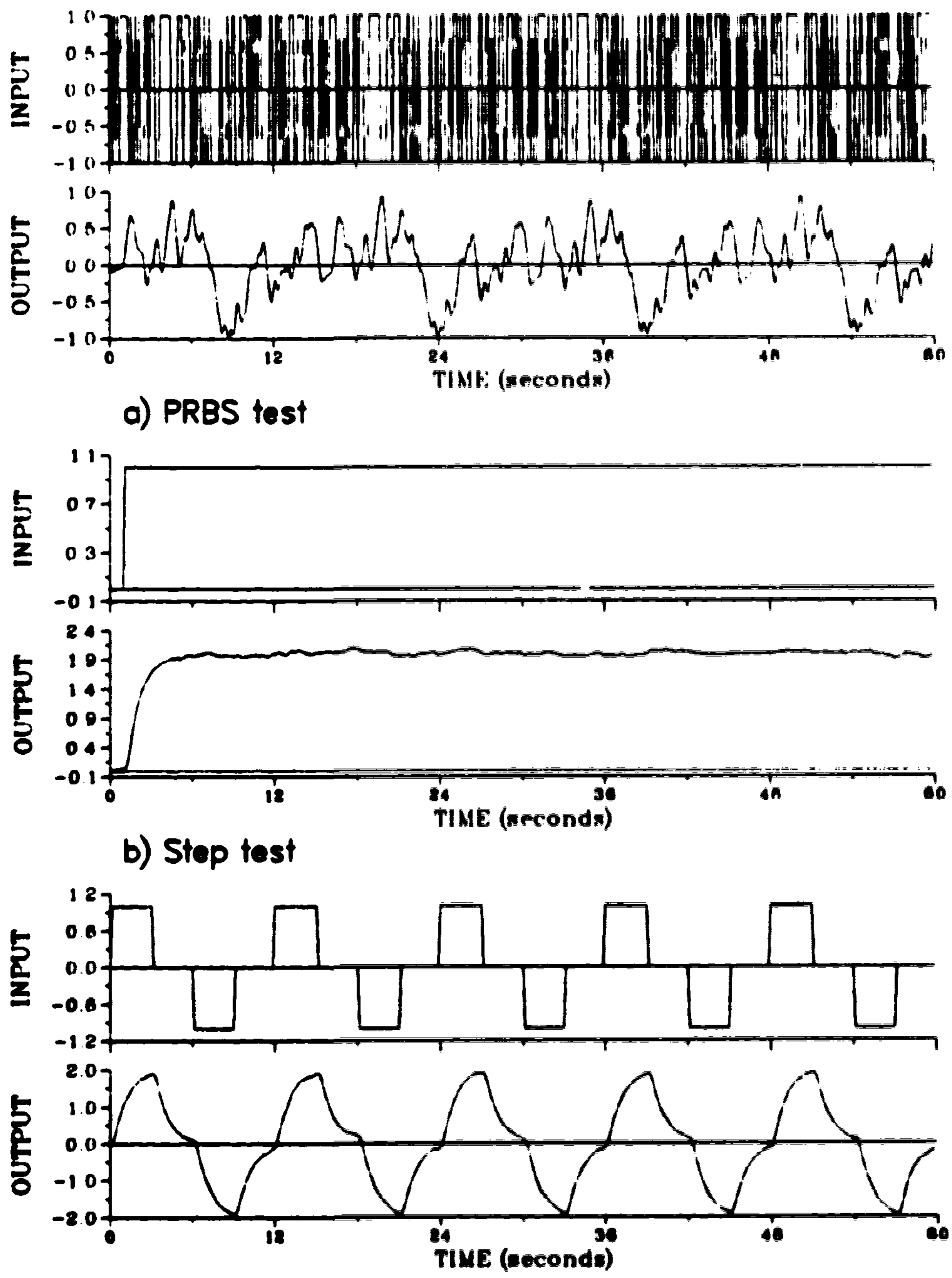
TABLE A.1

Nonrecursive least squares identification results for different input signals and noise of 0.01 and 0.1 standard deviation

	Actual	PRBS		Step		Square Wave	
		.01	.1	.01•	.1	.01	.1
a_1 x 10	-12.080	-12.2±.4	-12.5±.5	-8.±2.	-12.5±.6	-12.4±.5	-12.2±.5
a_2 x 10	3.121	3.3±.5	3.6±.6	-3.6±.4	3.6±.9	3.5±.6	3.6±.7
a_3 x 10	-.242	-.3±.1	-.3±.3	2.±2.	-.5±.6	-.4±.2	-.6±.4
b_1 x 100	5.60	5.59±.04	5.5±.4	4.±9.	-2.±10.	5.9±.3	9.±2.
b_2 x 100	9.50	9.4±.2	8.9±.5	10.±10.	7.±20.	9.3±.5	9.±.3
b_3 x 100	0.87	.7±.4	.9±.6	4.±10.	9.±10.	.6±.6	-1.±2.

• numerical singularity encountered

For the step, the least squares algorithm detected numerical singularity at a noise level of 0.01 but it was not found at the 0.1 level. This means that at the 0.01 level there was not enough noise to keep the system "identifiable" but there was enough noise at the 0.1 level to prevent numerical singularity. This only means that, with the added noise level, the off-diagonal terms in the P



a) PRBS test

b) Step test

c) Square wave test

Fig. A.1 Input and output values for different types of input signal at a 0.12 s. sampling period with $N(0,10^{-4})$ noise.

matrix are different enough to avoid numerical singularity. It does not mean that better parameter estimates were obtained as witnessed by the large standard deviations. Of all the results, the step gave the worst parameter estimates as expected. In fact the parameters for the step at the low noise level give a steady state gain of 4.5 compared to 2 for the actual system. The reversing square wave produced good results at a low noise level but deteriorated significantly at the high noise level. The PRBS produced good results for both noise levels.

This example illustrates that a least squares type of algorithm requires a signal that will "excite" all the modes of the process sufficiently. A signal that does "excite" all the modes of a process is a white noise sequence. Producing a good normal random signal is not an easy task and is a subject of research by itself (Knuth, 1973, Forsythe *et al*., 1977).

Another signal which has properties similar to those of white noise but is easier to produce is a pseudo random binary sequence (PRBS). It is probably the most widely used signal in statistical system testing. This signal can take only two values, ± a, where a is the amplitude. The state is updated every bit interval. This bit interval is an integer multiple of the clock interval (sampling period). The update follows a predetermined pattern or sequence. This pattern depends on the length of the sequence desired before the pattern repeats itself. The length of the sequence should be greater than the settling time of the process otherwise the periodicity of the PRBS may introduce a bias in the parameters.

The PRBS should have properties similar to those of white noise. In general a PRBS pattern has an odd number of bit intervals, which implies that its mean is not zero. To be as close to zero as possible, the number of positive states should not differ from the number of negative states by more than one. Another desirable property is the number of runs. The number of runs of length i is the number of times the state is the same for i consecutive intervals. A PRBS signal should have half of its runs to be of length one, a quarter of length two and so on. Another property is that the autocorrelation function is significant at lag zero only (and at every lag corresponding to the start of recycling the pattern).

The most commonly used PRBS signal is the maximum length sequence (Davies, 1966). This PRBS is generated by an n stage shift register with the first stage determined by the feedback of the appropriate modulo 2 additions (exclusive OR). This will give a sequence of length 2^n-1. The appropriate modulo addition is determined by the number of stages. If an inadequate modulo addition is performed then the sequence will be shorter than the maximum length. Also some inappropriate additions may lead to the zero sequence (the register fills with zeros) which never changes regardless of the modulo 2 addition performed.

A.2 Sampling period and identification length

For a given type of input signal, the sampling period and the number of input/output data points must be chosen. The sampling period will be determined by the highest frequency of interest (Shannon's sampling theorem: only frequencies smaller than half the sampling rate, the inverse of the period, can be identified). To avoid the effects of aliasing, the sampling period should be small enough so that the power of the signal above the Shannon frequency (also known as the Nyquist frequency, half the sampling rate) can be neglected. If the power above this frequency is due to noise, a low pass filter before the sampling equipment may reduce the influence of the high frequency noise. Also if the process exhibits high frequency dynamic components that are not of interest, filtering should be utilized (Gustavsson, 1975).

The number of input/output data points will be determined by the lowest frequency of interest. As a rule of thumb Box and Jenkins, 1976 suggest calculation of the auto-correlation function to a maximum lag of N/4. Applying the same rule here, the number of input/output data points should cover at least four times the lowest frequency of interest (it is also generally known as the settling time).

Although there is no general consensus as to the choice of the sampling period and the number of input/output data points, the following guidelines should provide a window that will be satisfactory for most cases. The sampling period should be 1/6 th to 1/15 th of the 95% settling time (Isermann, 1980a) and the number of input/output data points times the sampling period should be 4 to 10 times the 95% settling time. For a first order system this would imply that between 24 (one 6th for the period and 4 times fo: the number of points) and 150 (one 15th for the period and 10 times for the number of points) points are needed to determine the response curve from initial time to settling time.

Using a very large number of input/output data points has its drawbacks since statistical inferences become meaningless because of the large number of degrees of freedom. Moreover, data collection can be very expensive. Selection of too long a sampling period can lead to aliasing if high frequency noise is present. For example a 60 Hz signal sampled at 14 samples per second appears as a 4 Hz signal.

$$\frac{60}{14} = 4.28571$$

since complete cycles cannot be measured only the remainder appears. 0.28571 (or $\frac{4}{14}$) of a cycle is apparent every $\frac{1}{14}$ seconds or 4 cycles per seconds. It should be noted that using a digital filter after sampling to remove high frequencies will not remove aliasing as the noise has already been shifted to low frequencies and therefore will not be affected by a low pass filter.

To illustrate the choice of the sampling period, T, and the number of input/output data points, N, consider that a frequency resolution of 100 Hz with a maximum frequency of 25 kHz is desired.

$$f_{max} = \frac{1}{2T}$$

so T should be chosen such that

$$\frac{1}{2T} \geq 25 \times 10^{3} \text{ Hz}$$

hence

$$T \leq 2 \times 10^{-5} \text{ s.}$$

so for a frequency resolution of 100 Hz

$$\frac{1}{NT} = 100 \text{ Hz}$$

$$N = \frac{1}{100T} \geq \frac{1}{100(2 \times 10^{-5})} = 500$$

Either N or T can be chosen and the other calculated. In this case the rule for the period can be modified to 1/4 th of the smallest time constant (or equivalently the period is taken as the inverse of 4 times the frequency). Therefore T would be selected as 10^{-5} s. Using the factor of 4 times the largest time constant (or one quarter of the smallest frequency) N is calculated to be 4000.

The choice of the sampling period will affect the value of the parameters. Astrom *et al.*, 1984 investigated the location of zeros (transfer function numerator roots) of sampled systems and have shown that for a pole (transfer function denominator roots) excess of more than two an unstable zero will always result if sampled fast enough, even for a process that has no zeros in the continuous time domain. An unstable zero occurs when one of the roots of the transfer function numerator polynomial (i.e. the B polynomial) is inside the unit circle for the polynomial expressed in terms of q^{-1}. Cluett *et al.*, 1987 have shown that for the transfer function presented in Equation A.1, sampled with a zero order hold, the fastest sampling rate that gives stable zeros is 0.2.

Table A.2 shows the values of the zeros of the discrete form of Equation A.1 for selected sampling periods. For this example, following the guideline of 1/10 to 1/5 of the dominant time constant of interest (Stephanopoulos, 1984) does not always yield stable zeros ($\tau \approx 1.15$, $.115 < T < .230$). This may or may not be significant depending on the application but it should be stressed that if it is in conjunction with a controller, the controller must be able to handle unstable zeros. Isermann, 1980a proposed a sampling period of 1/15 to 1/6 of T_{95} (time for 95% response) yielding sampling periods between 0.22 and 0.55 seconds. This choice results in stable zeros as can be seen in Table A.2.

TABLE A.2

Zeros (in q^{-1}) for selected sampling periods (in seconds)

Sampling period	Zeros (one unstable)		Sampling period	Zeros (stable)	
0.01	-4.02123	-0.29343	0.20	-22.2667	-1.00479
0.04	-5.13584	-0.36196	0.21	-24.6488	-1.06115
0.08	-7.20409	-0.47994	0.24	-33.6587	-1.24228
0.12	-10.2976	-0.62491	0.28	-51.7866	-1.51181
0.16	-15.0015	-0.79929	0.32	-81.0373	-1.81271
0.19	-20.1369	-0.95043	0.36	-128.866	-2.14385
			0.40	-207.900	-2.50367

It should be noted that for the discrete case, an unstable zero does not necessarily mean an inverse response. For the example presented above, predictions from the model containing an unstable zero match the continuous output at the sampling intervals so the predictions do not exhibit an inverse response behavior as can be observed from the step response shown in Figure A.1.

A.3 Final remarks

The ratio of the amplitude of the signal to the amplitude of the oise is also an important variable. The higher the ratio, the more reliable the parameter estimates and the faster the convergence, but the process is subject to more disturbance. As the ratio decreases, the estimation of reliable parameter becomes more difficult, a larger number of input/output data points will be required to compensate for the fact that the disturbance to the process is minimal. For a PRBS signal the sequence may be chosen to cover the sampling period and number of input/output data points discussed above and then the sequence is repeated as long as it is necessary to obtain a good identification.

Appendix B: MODEL VALIDATION

Model validation is used for two purposes: (1) to determine if the model is acceptable (or to be statistically correct if the model is not rejectable) and (2) to compare two estimated models and determine which is more suitable for the data at hand.

Many model validation statistics are available in the literature (Soderstrom, 1977, Freeman, 1985). Only those deemed most important are presented here. Some were selected on the basis on their simplicity and performance with simulated data (Freeman, 1985) while others are from Box and Jenkins, 1976.

B.1 Confidence intervals on the parameters

If the confidence interval on a parameter contains zero, then this indicates that the parameter could be set to zero or that the experimental data is ill-conditioned (e.g. poor excitation or numerical accuracy). In the first case the model should be reduced by eliminating the parameter from the model and the identification procedure repeated. In the second case experimental conditions should be reviewed or the data suitably coded (transformed).

B.2 Parameter correlation matrix

This matrix reveals any dependence between parameters. In the ideal situation the correlation between different parameters should be zero, giving a parameter correlation matrix equal to the identity matrix. Unfortunately this is seldom the case. Denoting a_{ij} as the parameter correlation matrix element in the ith row and jth column, the correlation between parameters i and j increases as the absolute value of a_{ij} approaches unity. The amount of correlation between parameters that can be tolerated is dictated by how "good" the data is (i.e. numerical conditioning, level of excitation, etc.) and if there is a suitable alternate model. Almost invariably, a compromise between reducing the correlation between parameters (through model order reduction and identification of the reduced model) and fitting data (i.e. reducing model residuals) will be necessary.

B.3 Modified Akaike's information criterion (MAIC)

Akaike has proposed two similar statistics, the final prediction error and the information criterion which has become known as Akaike's information criterion (AIC). A modified version of the AIC was found to give better results in most cases (Freeman, 1985). This modified Akaike's information criterion (MAIC) is defined as

$$\text{MAIC} = N \ln(\hat{\sigma}^2) + \alpha p$$

where

N = the number of points

$\hat{\sigma}^2$ = the variance of residuals

p = the number of parameters

α = a weighting factor

If $\alpha = 2$ then the AIC is obtained. In the PITSA program the value of α is set to 4, as generally this value is thought to be the most satisfactory (Freeman, 1985).

The model which results in the lowest MAIC value is retained.

B.4 Shortest data description criterion (SDD)

This criterion, developed as an extension of Akaike's information criterion by Rissanen (see Freeman, 1985 for details), is defined as

$$SDD = N \ln(\hat{\sigma}^2) + (p+1) \ln(N)$$

The model that produces the lowest SDD is retained.

B.5 F test

This is the familiar test of variances in regression analysis. Let V_i^N be the sum of squares for the ith model (i=1,2) using N points. Let p_i denote the number of parameters in the ith model, then the statistic

$$f = \frac{V_1^N - V_2^N}{V_2^N} \frac{N - p_2}{p_2 - p_1} \text{ for } p_1 < p_2$$

is asymptotically $F(p_2-p_1,N-p_2)$ distributed. Model 1 is retained if $f < F$ (at the specified significance level) i.e. the increase in V due to fewer parameters is not significant.

B.6 Chi-square on auto and crosscorrelations

Chi-square statistics linked to the autocorrelation of residuals and the crosscorrelation between residuals and the input signal are useful to determine the whiteness of the residuals and to verify that the input signal and residuals are uncorrelated (Box and Jenkins, 1976).

If the value of the statistic is less than the table value (or given by the program) for the level of significance chosen then the model is not rejected.

B.7 Plots of residuals

Although not a statistic by itself, plots of residuals may reveal possible inadequacy of a model or the presence of unreliable data. These plots include the autocorrelation function, partial autocorrelation function and the power spectrum of the residuals. These are examined to check the whiteness of the residuals. If the residuals do not display the characteristic properties of white noise, as described in Appendix C, then the model should be rejected. Plots of the crosscorrelation between inputs and residuals should not show any relationship, otherwise there is some input/output behavior that is not accounted for by the model.

Appendix C: CORRELATION ANALYSIS

This material is based on the presentation given by Box and Jenkins, 1976. For the reader that has some familiarity with elementary probability and statistics, this Appendix should provide enough background to use correlation analysis to obtain some guidance for selecting an appropriate model structure.

Correlation analysis is based on the analysis of the autocorrelation and crosscorrelation functions of stochastic processes. But before explaining the use of correlation function plots, the general model form, with some related terminology, is presented. The Box-Jenkins model is

$$y_t = \frac{B}{A} u_{t-k} + \frac{C}{D}\epsilon_t + \beta$$

where

$$A = 1 + a_1 q^{-1} + a_2 q^{-2} \ldots + a_n q^{-n}$$

$$B = b_1 q^{-1} + b_2 q^{-2} \ldots + b_m q^{-m}$$

$$C = 1 + c_1 q^{-1} + c_2 q^{-2} \ldots + c_{mn} q^{-mn}$$

$$D = 1 + d_1 q^{-1} + d_2 q^{-2} \ldots + d_{nn} q^{-nn}$$

k = delay

β = constant term (bias)

u_t = input

y_t = output

ϵ_t = white noise, residuals

q^{-1} = backshift operator ($q^{-1}y_t = y_{t-1}$)

The model can be considered to consist of two parts: the deterministic part (transfer function) of the process

($\frac{B}{A}u_{t-k}$ + β) and the stochastic part (noise transfer function) of the process ($\frac{C}{D}\epsilon_t$). Identification of the noise model (noise transfer function) is considered first as several concepts developed for identifying the noise model will also be used for modelling the process transfer function.

Identification of the noise model falls within the general topic of time series analysis. The model

$$N_t = \frac{C}{D}\epsilon_t$$

can be thought of as process (or "colored") noise, N_t, that is generated from white noise, ϵ_t, passing through a linear filter. White noise is assumed to be a normally distributed random variable with zero mean and variance σ^2 ($N(0,\sigma^2)$) uncorrelated with its past values (under the normal hypothesis this also implies independence).

If D=1 and C is a polynomial of order mn then the model is called a moving average (MA(mn)) process. It should be noted that the term "process" to denote a model, used in the time series literature, is retained in this material. If C=1 and D is a polynomial of order nn then the the model is called an autoregressive (AR(nn)) process. If C and D are polynomials then the model is called an autoregressive moving average (ARMA(nn,mn)) process. Note that a finite moving average process can be expressed as an infinite autoregressive process and vice versa.

Correlation analysis is based on the assumption that the series under study is stationary, that is, its distributional properties (mean, variance, etc.) are not time dependent. The roots of the D polynomial must lie outside the unit circle in the q^{-1} plane to provide stationarity. The dual of stationarity is invertibility for which the roots of the C polynomial must lie outside the unit circle in the q^{-1} plane.

Another common form of model (process) is the autoregressive integrated moving average (ARIMA(nn,dn,mn)) process. This form of process arises if the form of the model is

$$\nabla^{dn} N_t = \frac{C}{D}\epsilon_t \qquad (C.1)$$

where

$$\nabla = 1 - q^{-1}$$

The effect of the difference operator, ∇, is to extract dn roots of the characteristic polynomial $\nabla^{dn}D$ that are on the unit circle. Removing the roots on the unit circle will make the process stationary (for instance if the process is a ramp then taking one difference will "level" the process i.e. make it stationary).

C.1 Autocorrelation function

The autocorrelation function is the first information that is used in correlation analysis. The function is related to the concept of variance and correlation coefficient found in elementary statistics. The variance is defined as

$$VAR(n_t) = \sigma^2 = E[(n_t - E[n_t])^2]$$

where

n_t = observed value

$E[n_t]$ = the expected value of n_t (estimated as the mean, $\bar{n}$)

The variance can be estimated by

$$s^2(n_t) = \frac{1}{N} \sum_{i=1}^{N} (n_i - \bar{n})^2$$

The more general autocovariance defined as

$$\gamma(j) = E[(n_t-E[n_t])(n_{t+j}-E[n_t])] \qquad (C.2)$$

is estimated by

$$c(j) = \frac{1}{N} \sum_{i=1}^{N-j} (n_i-\bar{n})\ (n_{i+j}-\bar{n})$$

It is to be noted that $\gamma(0)=\sigma^2$.

The autocorrelation function is obtained from the autocovariance function by the normalization

$$\rho(j) = \gamma(j)/\gamma(0)$$

The autocorrelation function is thus independent of the magnitude of the variance. The autocorrelation function is estimated as

$$r(j) = c(j)/c(0)$$

To see how the autocorrelation function can be used in analyzing the nature of a data series, first consider the moving average process

$$N_t = C\ \epsilon_t$$

Substitution of this process in Equation C.2 gives

$$\gamma(j) = E[(\epsilon_t + c_1\epsilon_{t-1} + c_2\epsilon_{t-2} \ldots + c_{mn}\epsilon_{t-mn})\ (\epsilon_{t+j} + c_1\epsilon_{t+j-1} + c_2\epsilon_{t+j-2} \ldots + c_{mn}\epsilon_{t+j-mn})]$$

and for j=0 we obtain

$$\gamma(0) = (1 + c_1^2 + c_2^2 \ldots + c_{mn}^2)\sigma^2$$

since the independence of ϵ_t gives

$$E[\epsilon_t \epsilon_{t+i}] = \sigma^2 \text{ if } i=0$$

$$= 0 \text{ if } i \neq 0$$

For $j \neq 0$ the autocovariance is

$$\gamma(j) = (c_j + c_1 c_{j+1} + c_2 c_{j+2} \ldots + c_{mn-j} c_{mn})\sigma^2$$

$$\text{for } j=1,\ldots,mn$$

$$= 0 \qquad \text{for } j > mn$$

It thus follows that the autocorrelation function is

$$\rho(j) = 0 \text{ for } j > mn$$

Therefore theoretically the autocorrelation function is zero beyond lag mn for a pure moving average process. Furthermore, since an autoregressive process can be written as an infinite moving average process, the theoretical autocorrelation function of an autoregressive process will be infinite in extent.

Since the analysis involves the estimation of the autocorrelation function from experimental data, a criterion is needed for deciding when the sample autocorrelation function is effectively zero beyond a certain lag. This is done by making use of an expression for the variance of the estimated autocorrelation function of a stationary normal random process given by Bartlett. Using the variance of the estimates r(j) given by Bartlett (see Box and Jenkins for details) for a stationary normal random process and assuming that the autocorrelation function is zero for $j > mn$ (long lag), the variance of the autocorrelation can be approximated by

$$VAR(r(j)) \approx \frac{1}{N}\{1 + 2\sum_{i=1}^{mn} \rho^2(i)\} \text{ for } j > mn$$

substituting r(i) for $\rho(i)$, the standard deviation of the autocorrelation estimates for long lags (j>mn) are obtained.

C.2 Partial autocorrelation function

Since the autocorrelation function reveals the moving average nature of a process, we might expect another tool to reveal the autoregressive nature of the process. This tool is the partial autocorrelation function.

Consider the AR(nn) process

$$DN_t = \epsilon_t$$

changing the notation somewhat, let $d_{nn,j}$ be the j'th coefficient of an AR(nn) process. We now have

$$D = 1 + \sum_{j=1}^{nn} d_{nn,j}q^{-j}$$

If we define the set of autoregressive coefficients $d_{j,j}$ as the partial autocorrelations then

$$d_{j,j} = 0 \text{ for } j > nn$$

for a pure AR(nn) process. Therefore the partial autocorrelation function of a pure autoregressive process is zero for j > nn. Since an MA(mn) can be expressed as an AR(∞), the theoretical partial autocorrelation of a pure moving average process is infinite in extent.

There are two ways of calculating the partial autocorrelation function. The first is based on the Yule-Walker equations.

Consider the AR(i) process

$$N_t = d_{i,1}N_{t-1} + d_{i,2}N_{t-2} \ldots + d_{i,i}N_{t-i} + \epsilon_t$$

noting that the means are zero, multiplying both sides by N_{t-j} and taking expectations

$$\gamma(j) = d_{i,1}\gamma_{j-1} + d_{i,2}\gamma_{j-2} \ldots + d_{i,i}\gamma_{j-i} \text{ for } j > 0$$

$$\rho(j) = d_{i,1}\rho_{j-1} + d_{i,2}\rho_{j-2} \ldots + d_{i,i}\rho_{j-i}$$

so that

$$P_k\Phi_k = \rho_k$$

where

$$P_k = \begin{bmatrix} 1 & \rho_1 & \ldots & \rho_{k-1} \\ \rho_1 & 1 & \ldots & \rho_{k-2} \\ \vdots & \vdots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \ldots & 1 \end{bmatrix}$$

$$\Phi_k^T = [d_{k,1}, d_{k,2}, \ldots, d_{k,k}]$$

$$\rho_k^T = [\rho_1, \rho_2, \ldots, \rho_k]$$

these equations are solved for increasing values of k. The reader is referred to Box and Jenkins for the Durbin recursive algorithm for solving these equations.

The Yule-Walker equations are not well conditioned. A more stable alternative for calculating the partial autocorrelation function is to fit autoregressive models of increasing order with least squares.

To test whether a partial autocorrelation is significant we use the results from Quenouille (as explained in Box and Jenkins). He stated that if the process is AR(nn) then

$$VAR(d_{i,i}) \simeq (1/N) \text{ for } i > nn$$

where N is the number of observations.

C.3 Power spectrum

The power spectrum will help in finding any periodic component of the signal. Although periodic components can be found from the autocorrelation and partial autocorrelation functions, they are shown readily in the power spectrum. The Fourier transform of the autocorrelation function yields the power spectrum. In particular, if the power spectrum reveals a large peak in the low frequency range it may indicate a slowly changing level which can be removed by differencing.

Given an odd number of observations, say N=2*q+1, we obtain the Fourier transform

$$n(k) = \alpha_0 + \sum_{i=1}^{q} \alpha_i \cos(2\pi\frac{ik}{N}) + \beta_i \sin(2\pi\frac{ik}{N}) + e(k)$$

The least squares estimates for α and β are

$$a_0 = \bar{n}$$

$$a_i = (2/N)\sum_{k=1}^{N} n_k \cos(2\pi\frac{ik}{N})$$

$$b_i = (2/N)\sum_{k=1}^{N} n_k \sin(2\pi\frac{ik}{N})$$

Now that the data has been transformed into the frequency domain it has to be presented in a suitable form. Usually the signal strength is plotted against the frequency. This is known as the signal spectrum. The signal strength, or intensity, at frequency f(i) is defined as

$$I(f(i)) = (N/2)(a^2(i) + b^2(i)) \quad i=1,\ldots,q$$

$$f(i) = (i/N)$$

For an even number of samples, N, the procedure is modified slightly. Let N=2q then

$$a(q) = (1/N)\sum_{k=1}^{N} (-1)^k n(k)$$

$$b(q) = 0$$

$$I'f(q)) = Na^2(q)$$

with $a(i)$, $b(i)$ and $I(f(i))$ for $i=1,\ldots,q-1$ defined as before.

For a truly random series

$$n(k) = a_0 + e(k)$$

that is the signal will have only a DC component equal to the signal average, plus some error component causing it to vary about this value. In this case the expected value of $I(f(i))$ is $2\sigma^2(n)$, distributed as $\sigma^2(n)\chi^2(2)$ (the intensity of white noise is uniform at all frequencies and equals $2\sigma^2(n)$).

If there are periodic components in the series the power spectrum will show an increase of intensity in the vicinity of the frequencies of these components.

If the power spectrum is integrated and normalized, deviations from the expected behavior of white noise can be assessed using the Kolmogorov-Smirnov bounds. White noise has a uniform frequency content, therefore the integrated spectrum will be a straight line of slope $2\sigma^2$. Normalizing by σ^2 we obtain a straight line of slope 2. Limit lines can be drawn at distances $\pm K(\lambda)/\sqrt{q}$ above and below the white noise line. For white noise an excursion over the limit line will occur with probability λ. The values of K are approximately 1.63 and 1.22 for λ of .01 and .1 respectively.

C.4 Summary on time series

This summary provides a short reference that can be used jointly with the examples that follow.

C.4.1 Differencing

If the autocorrelation function or partial autocorrelation function does not decay rapidly, consider differencing the data. This can be confirmed by finding a very large power at a frequency of zero on the power spectrum plot (see the ARIMA example below). If differencing was required, examine the differenced series to see if more differencing is needed. It is advisable not to overdifferentiate.

C.4.2 ARMA model orders

Table C.1 can be used as a guide to select model orders.

C.4.3 White noise check

To test for whiteness of the signal, the autocorrelation function and partial autocorrelation function should not show any significant value, the power spectrum should be evenly distributed and the cumulative power spectrum should follow the diagonal line.

TABLE C.1

Model characteristics for order selection

model	autoregressive $DN_t=\epsilon_t$	moving average $N_t=C\epsilon_t$	mixed $DN_t=C\epsilon_t$
autocorrelation	infinite (damped exponentials and/or damped sine waves)	finite	infinite (damped exponentials and/or damped sine waves after mn-nn lags)
	tails off	cuts off after mn lags	tails off after mn-nn lags
partial autocorrelation	finite	infinite (dominated by exponentials and/or sine waves)	infinite (dominated by exponentials and/or sine waves after nn-mn lags)
	cuts off after nn lags	tails off	tails off after nn-mn lags

C.5 Some preliminary examples

A few examples of the use of the autocorrelation function, partial autocorrelation function and power spectrum are presented in this section. These examples were inspired by Nelson, 1973.

C.5.1 Pure autoregressive process

A series is generated from a known autoregressive process (this is termed a "realization" of the process). The data generated will then be analyzed to illustrate the use of the information that can be calculated from input output data. Only graphical information will be used (as it would be the case in dealing with plant process data). The AR(2) process used is

$$(1-q^{-1}+.75q^{-2})N_t = \epsilon_t \qquad \epsilon_t \sim N(0,1)$$

This process has a pair of complex roots at $q^{-1} = \frac{2}{3} \pm \frac{\sqrt{2}}{1.5}j$ (thus stationary). The missing values (N_{-1}, N_{-2}) are set to zero and three hundred points are generated. The realization is presented in Figure C.1.

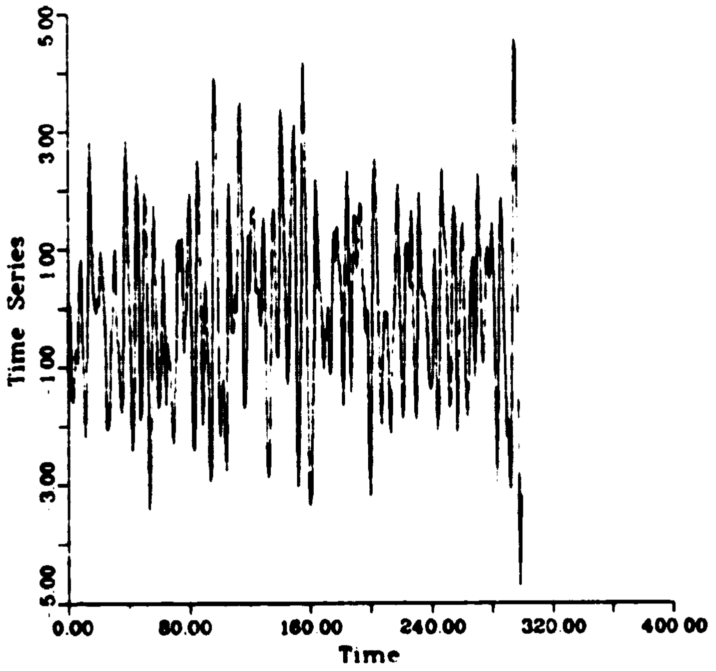
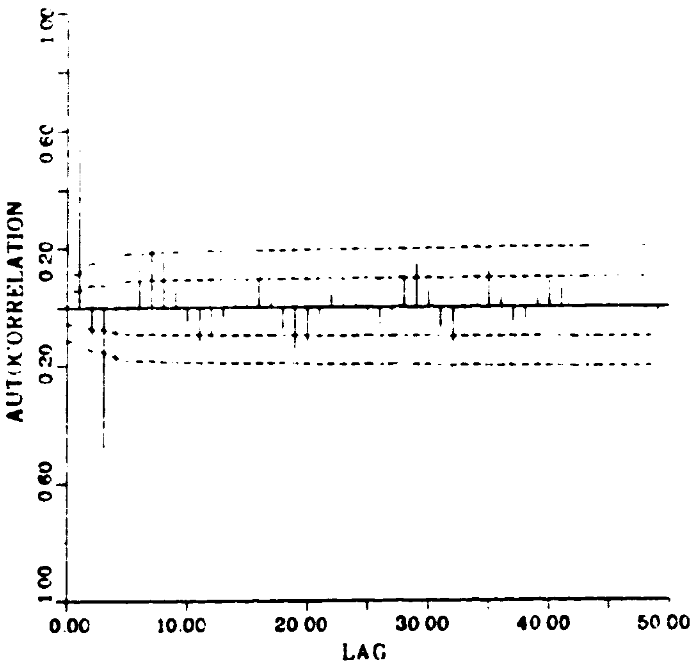


Fig. C.1 Realization of the AR(2) process

The autocorrelation function is shown in Figure C.2. As can be expected the magnitude of the autocorrelation function is significant over several lags and follows a damped sine wave pattern.



Time Series
AR(2) Process
M1(8)

Fig. C.2 Autocorrelogram of the AR(2) process

The partial autocorrelation function is presented in Figure C.3. Here the magnitude of the partial autocorrelation function becomes negligible after lag 2 as it should be for a pure second order autoregressive process.

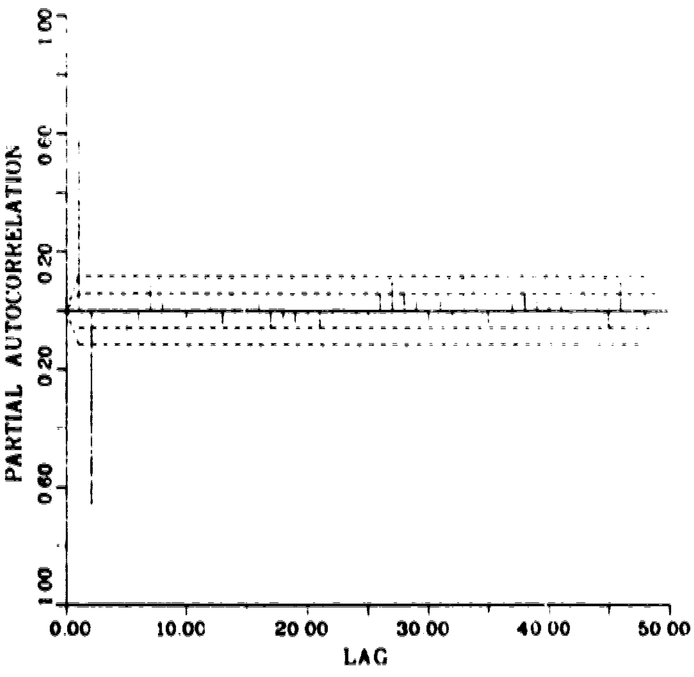
From the form of the autocorrelation and partial autocorrelation functions, the process is tentatively identified as being AR(2). This is only a tentative assessment as the order might be changed later at the model validation step.

C.5.2 Pure moving average process

The MA(2) process used is

$$N_t = (1+.7q^{-1}-.2q^{-2})\epsilon_t \qquad \epsilon_t = N(0,1)$$

which has roots at $q^{-1}=-1.09$ and $q^{-1}=4.59$ (so is an



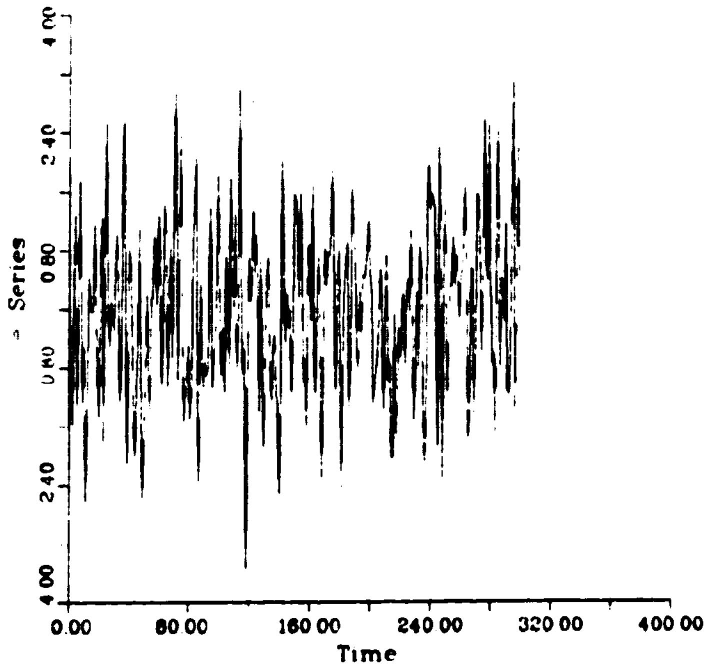
Time Series
AR(2) Process
M1(8)

Fig. C.3 Partial autocorrelogram of the AR(2) process

invertible process, but this property is not required for stationarity). As for the autoregressive process three hundred points are generated yielding the realization presented in Figure C.4.

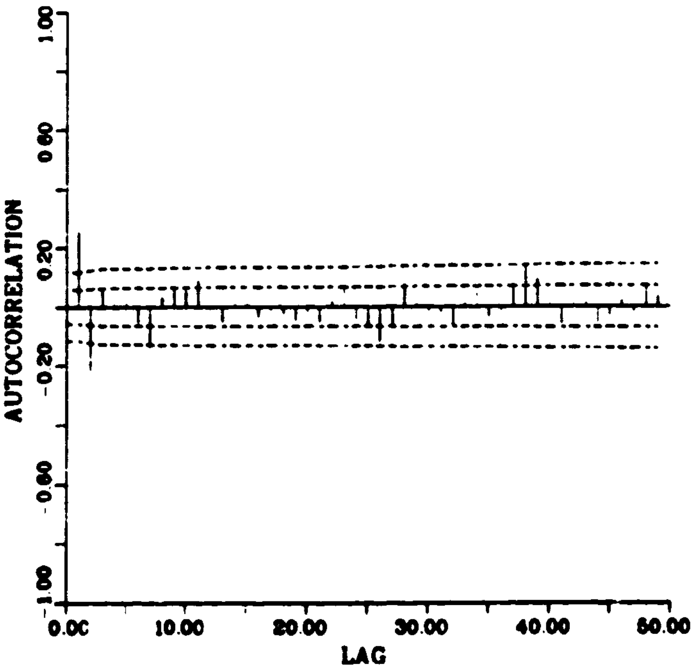
The autocorrelation function is shown in Figure C.5. The magnitude of the autocorrelation function is negligible after lag 2 as expected for a second order moving average process. The partial autocorrelation function presented in Figure C.6 shows that the magnitude of the partial autocorrelation function is significant over several lags.

From the characteristics of the autocorrelation and partial autocorrelation functions the process is tentatively identified as being MA(2).



MA(2) Process
M1(8)

Fig. C.4 Realization of the MA(2) process



Time Series
MA(2) Process
M1(8)

Fig. C.5 Autocorrelogram of the MA(2) process

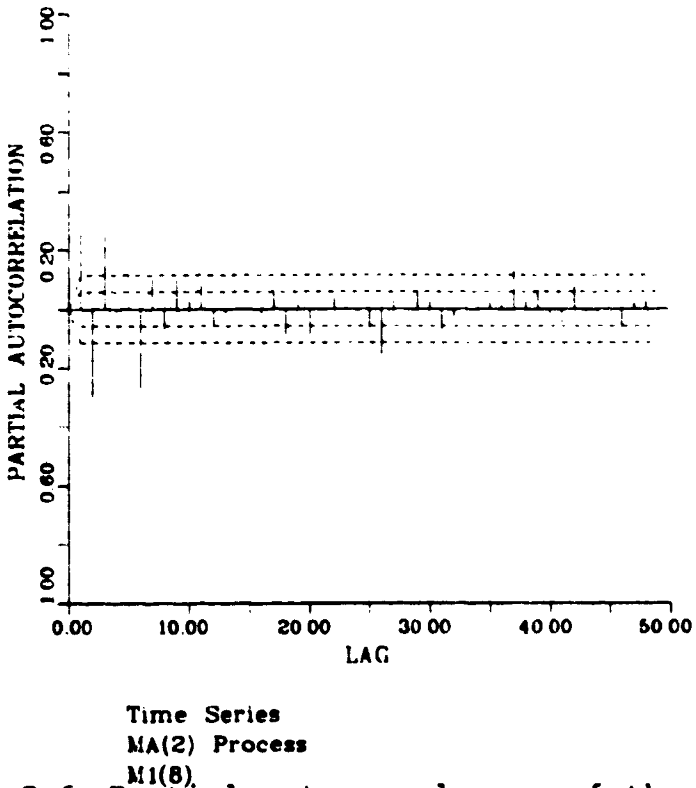


Fig. C.6 Partial autocorrelogram of the MA(2) process

C.5.3 Mixed process

In real applications, mixed (autoregressive and moving average) processes are the most common and the most difficult to identify. The problem is that the magnitude of neither the autocorrelation nor the partial autocorrelation functions will become insignificant. Identification thus requires both judgement and experience. In certain cases it may even be impossible to identify a unique model. Then all the possible models are retained and their parameters estimated (generally resulting in models that are very similar). Model validation is used to choose the best model from all the possible candidate models (e.g. the one with the lowest sum of squares or the one with the fewest parameters, etc).

In this example a nonstationary process is used to show nonstationary behavior. Taking one difference will make this process stationary (in practice a second difference may be required but as a rule of thumb no more than two differences should be taken) and the mixed process is identified. The process chosen is ARIMA(1,1,1)

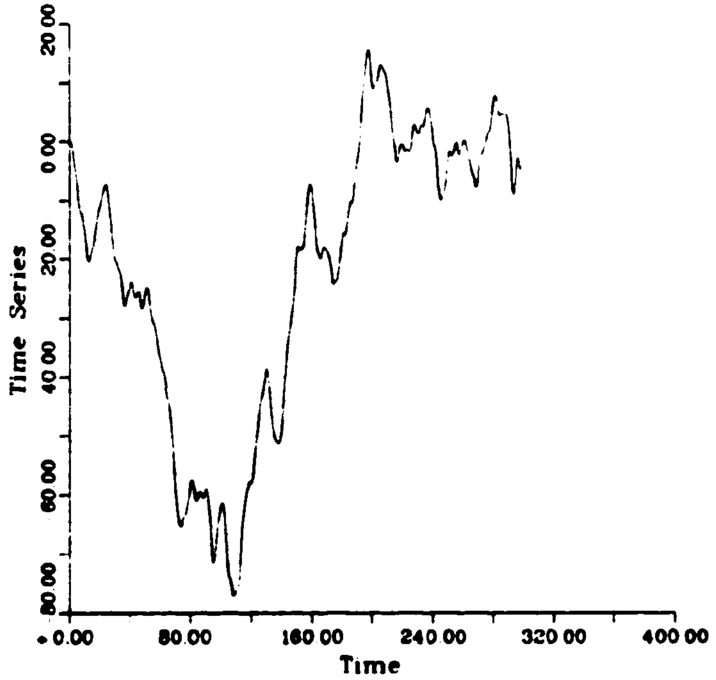
$$(1-0.5q^{-1})\nabla N_t = (1+.9q^{-1})\epsilon_t, \quad \epsilon_t \approx N(0,1)$$

This can be rewritten as

$$N_t = 1.5N_{t-1}-.5N_{t-2}+\epsilon_t+.9\epsilon_{t-1}$$

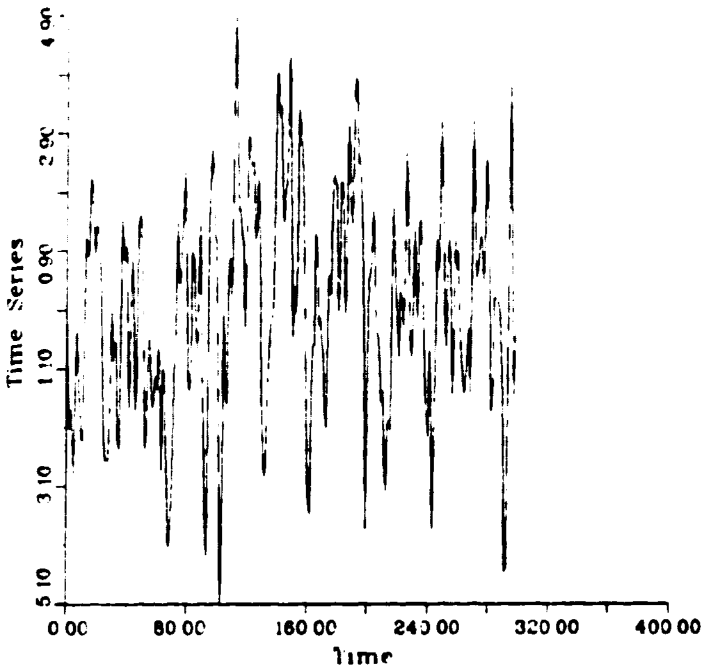
with the roots of the characteristic equation $(1+1.5q^{-1}+.5q^{-2})$ at $q^{-1}=1$ and $q^{-1}=2$. Although this is stationary in the rigorous sense, the zero at $q^{-1}=1$ is on the limit of stationarity. This can create problems. On the other hand, the differenced series characteristic equation is $1-.5q^{-1}$ which has a single root at $q^{-1}=2$. This is well within the stationarity boundary. The two cases (nonstationary and differenced) will be presented in parallel although in practice the autocorrelation and partial autocorrelation functions of the nonstationary (or nearly nonstationary in this case) series would be examined first and then from the behavior of the autocorrelation and partial autocorrelation functions (a slow decrease indicates nonstationarity or near nonstationarity) the choice of differencing or not differencing would be dictated. The nature of the power spectrum can also help in deciding if differencing is needed. If there is a large peak near zero frequency (near DC values) then differencing is required.

In Figures C.7 and C.8 the realization of the two processes are presented. The autocorrelation functions of the two processes are clearly different as can be observed from Figures C.9 and C.10. The slow damping of the autocorrelation function for the ARIMA(1,1,1) process indicates that differencing should be performed. The autocorrelation of the differenced series dampens quickly so a second difference is not necessary.



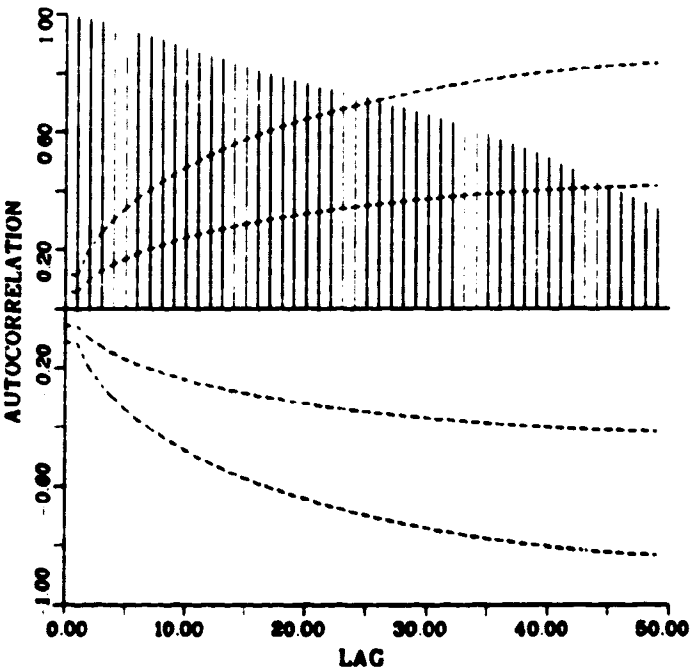
ARIMA(1,1,1) Process
M1(8)

Fig. C.7 Realization of the ARIMA(1,1,1) process



ARIMA(1,1,1) Process
FY(0,1,0)M1(8)

Fig. C.8 Realization of the differenced ARIMA(1,1,1) process



Time Series
ARIMA(1,1,1) Process
M1(8)

Fig. C.9 Autocorrelogram of the ARIMA(1,1,1) process

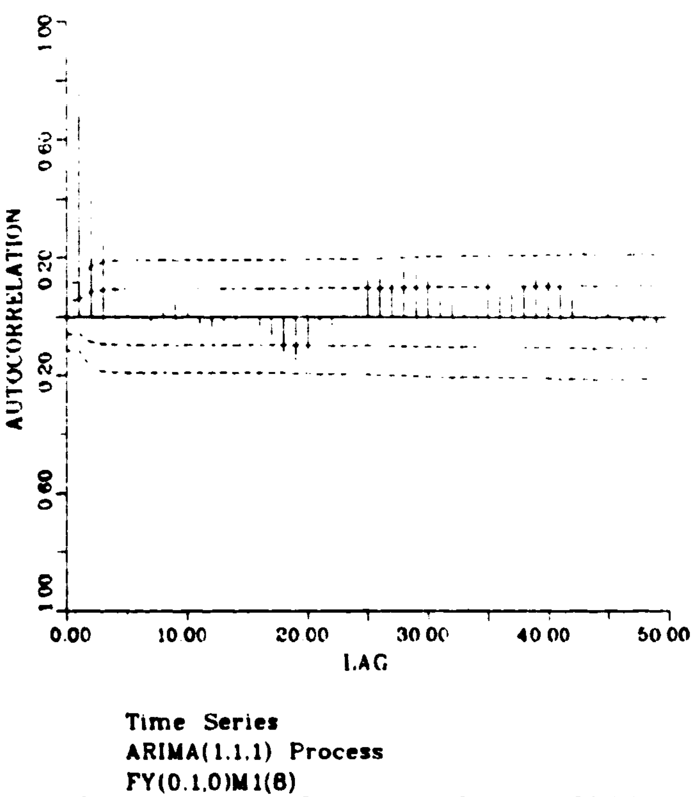
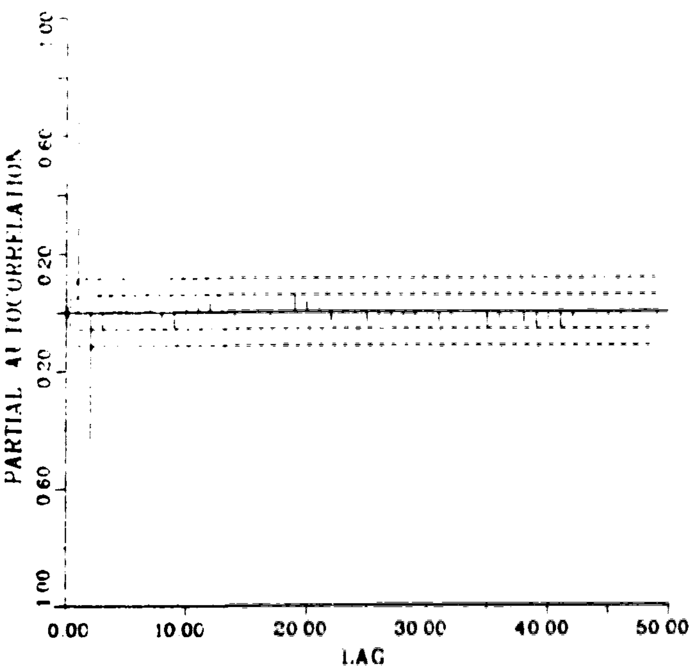


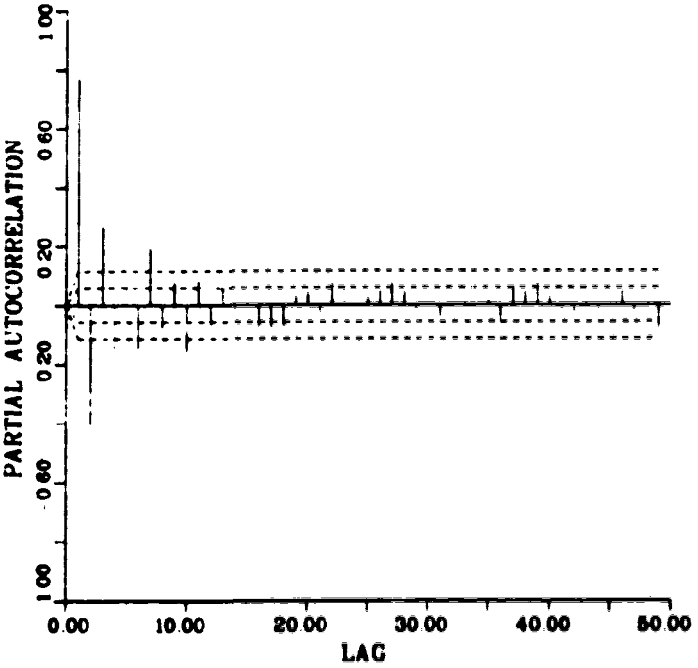
Fig. C.10 Autocorrelogram of the differenced ARIMA(1,1,1) process

The corresponding partial autocorrelation functions are presented in Figures C.11 and C.12. Since the power spectrum can also be used in the choice of differencing, the power spectrum of the two processes is presented in Figures C.13 and C.14. It can be seen from the power spectrum that differencing is required.



Time Series
ARIMA(1,1,1) Process
M1(8)

Fig. C.11 Partial autocorrelogram of the ARIMA(1,1,1) process



Time Series
ARIMA(1,1,1) Process
FY(0,1,0)M1(8)

Fig. C.12 Partial autocorrelogram of the differenced ARIMA(1,1,1) process

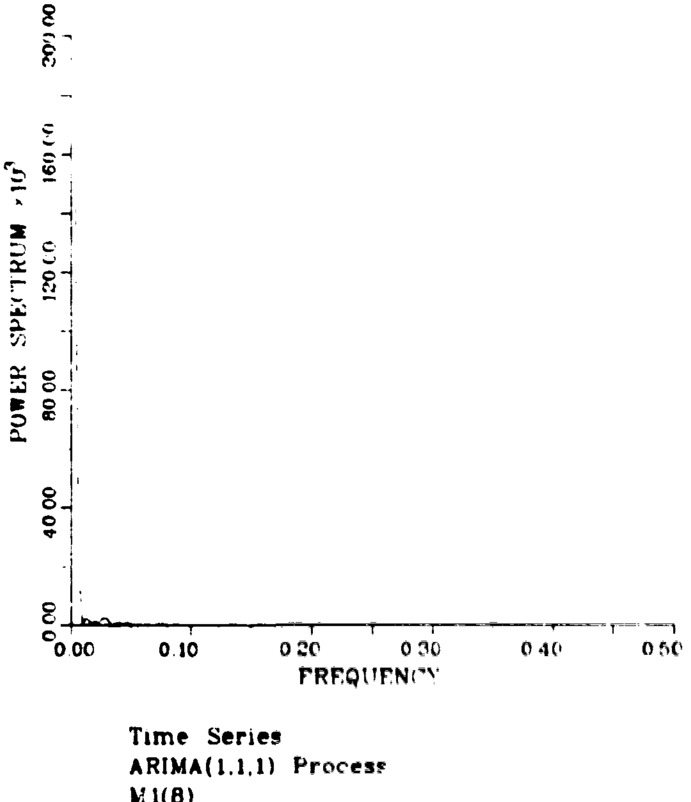


Fig. C.13 Power spectrum of the ARIMA(1,1,1) process

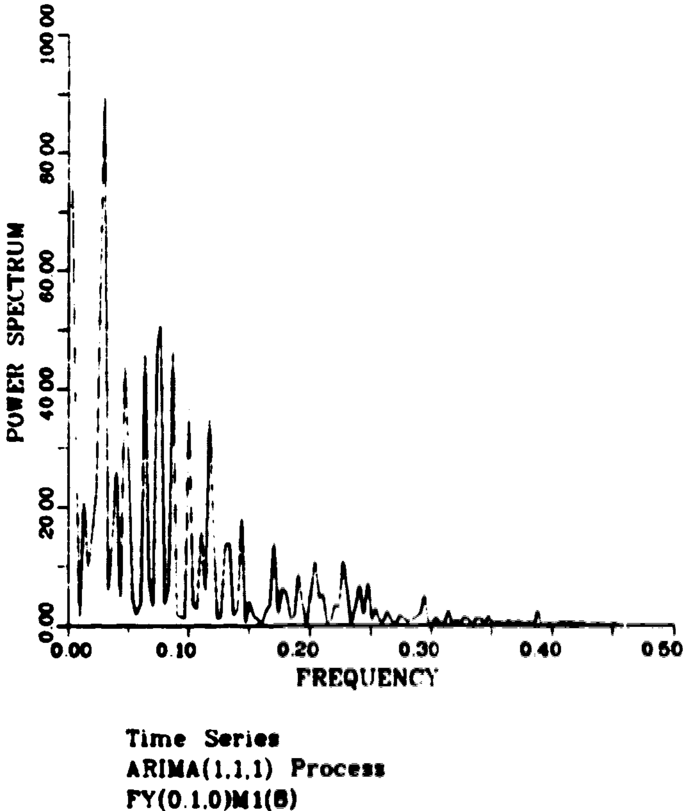


Fig. C.14 Power spectrum of the differenced ARIMA(1,1,1) process

C.6 Crosscorrelation function

The crosscorrelation function is used to ascertain the relationship between u_t and y_t if the model being considered is of the form

$$y_t = \frac{B}{A}u_{t-k} + N_t + \beta$$

The basis for calculating this function is the crossvariance from u_t to y_t defined as

$$\gamma_{uy}(j) = E[(u_t-\bar{u})(y_{t+j}-\bar{y})]$$

It should be noted that the autocovariance of y_t is $\gamma_{yy}(j)$. In general $\gamma_{uy}(j) \neq \gamma_{yu}(j)$ but

$$\begin{aligned} \gamma_{uy}(j) &= E[(u_t-\bar{u})(y_{t+j}-\bar{y})] \\ &= E[(u_{t-j}-\bar{u})(y_t-\bar{y})] \\ &= E[(y_t-\bar{y})(u_{t-j}-\bar{u})] \\ &= \gamma_{yu}(-j) \end{aligned}$$

The crosscorrelation function, defined in a manner analogous to that for the autocorrelation function, is expressed as

$$\rho_{uy}(j) = \frac{\gamma_{uy}(j)}{\sigma_u \sigma_y}$$

The crossvariance is best estimated as

$$\begin{aligned} c_{uy}(j) &= \frac{1}{N}\sum_{k=1}^{N-j} (u_k-\bar{u})\,(y_{k+j}-\bar{y}) \text{ for } j \geq 0 \\ &= \frac{1}{N}\sum_{k=1}^{N+j} (y_k-\bar{y})\,(u_{k-j}-\bar{u}) \text{ for } j<0 \end{aligned}$$

ↄ the crosscorrelation function can be estimated as

$$r_{uy}(j) = \frac{c_{uy}(j)}{\sqrt{c_{uu}(0)}\,\sqrt{c_{yy}(0)}}$$

The criterion for deciding whether a particular crosscorrelation is significant is given by Bartlett (see Box and Jenkins for details, it assumes that u_t and y_t are not crosscorrelated at this lag)

$$\mathrm{VAR}(r_{uy}(j)) \approx \frac{1}{N-j} \sum_{i=-\infty}^{\infty} \rho_{uu}(i)\ \rho_{yy}(i)$$

If u_t is white noise, it follows that

$$\rho_{uu}(i) = 0 \text{ for } i \neq 0$$

$$\rho_{uu}(0) = 1$$

so based on the hypothesis that u_t and y_t are not crosscorrelated and that u_t is white noise, the variance is given by

$$\mathrm{VAR}(r_{uy}(j)) \approx \frac{1}{N-j}$$

C.7 Input prewhitening

If the original data series are already differentiated (if necessary), then the model can be written as

$$y_t = v_0 u_t + v_1 u_{t-1} + v_2 u_{t-2} \ldots + N_t + \beta$$

where v_i are the impulse response weights obtain by the long division of B by A and the weights before the delay has passed are zero. Removing the means by subtracting the equation with the means (i.e. the equation obtained by replacing all the values by their mean) and then multiplying by $(u_{t-j}-\bar{u})$ and taking the expectations gives

$$\gamma_{uy}(j) = v_0\gamma_{uu}(j) + v_1\gamma_{uu}(j-1) + v_2\gamma_{uu}(j-2) \ldots$$
$$+ \gamma_{uN}(j)$$

Assuming that u_{t-j} is uncorrelated with N_t for all j beyond some point K the impulse response of the system will die out. Truncating at that point, a system of equations can be written for j=0 to j=K.

$$\gamma_{uy} = \Gamma_{uu}V$$

where

$$\gamma_{uy} = [\gamma_{uy}(0), \gamma_{uy}(1), \ldots, \gamma_{uy}(K)]$$

noting that $\gamma_{uu}(j)=\gamma_{uu}(-j)$

$$\Gamma_{uu} = \begin{bmatrix} \gamma_{uu}(0) & \gamma_{uu}(1) & \ldots & \gamma_{uu}(K) \\ \gamma_{uu}(1) & \gamma_{uu}(0) & \ldots & \gamma_{uu}(K-1) \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{uu}(K-1) & \gamma_{uu}(K-2) & \ldots & \gamma_{uu}(0) \end{bmatrix}$$

$$V = [v_0, v_1, \ldots, v_K]$$

This system could be solved to obtain the impulse response weights. However the system can be simplified if the input is white noise. Then $\gamma_{uu}(j)=0$ for $j\neq0$ and $\gamma_{uu}(0)=\sigma_u^2$ which gives

$$\gamma_{uy}(j) = v_j\sigma_u^2$$

$$v_j = \frac{\gamma_{uy}(j)}{\sigma_u^2}$$

$$v_j = \frac{\gamma_{uy}(j)}{\sigma_u\sigma_y}\;\frac{\sigma_y}{\sigma_u}$$

$$v_j = \rho_{uy}(j)\frac{\sigma_y}{\sigma_u}$$

This implies that for a white noise input, the impulse response weights are just a scaling of the crosscorrelations. Note that the first k weights (v_0, ..., v_{k-1}) will be zero as the crosscorrelations will be zero until the delay of k has passed.

To make the input white noise, it is assumed that the input can be modelled as a time series

$$\phi u_t = \theta a_t + \kappa \qquad a_t \simeq N(0, \sigma_a^2)$$

where

$$\theta = 1 + \theta_1 q^{-1} + \theta_2 q^{-2} \ldots + \theta_q q^{-q}$$

$$\phi = 1 + \phi_1 q^{-1} + \phi_2 q^{-2} \ldots + \phi_p q^{p}$$

Here κ is introduced to account for the mean of u_t (if it is nonzero). Because this model is similar to the noise model (see Equation B.1), identification of the prewhitening model is performed in the same manner as that employed to determine the noise model.

In order to identify the (unaltered) transfer function the output must also be filtered using the same filter

$$\phi y_t = \theta \beta_t + \kappa$$

then

$$v_j = \rho_{\alpha\beta}(j) \frac{\sigma_\beta}{\sigma_\alpha}$$

The process of fitting a time series model to filter the input series u_t (and filtering the output y_t) is termed prewhitening.

C.7.1 Prewhitening parameter identification

The two prewhitening parameter estimation algorithms that are employed in the in the PITSA program are now described.

Preliminary prewhitening parameter estimation

Assuming that suitable differencing has already been performed, consider the following mixed ARMA(p,q) process

$$\phi N_t = \theta\epsilon_t + \kappa \qquad \epsilon_t \approx N(0,\sigma_\epsilon^2)$$

$$N_t = -\phi_1 N_{t-1} - \phi_2 N_{t-2} \dots -\phi_p N_{t-p} + \epsilon_t + \theta_1\epsilon_{t-1}$$
$$+ \theta_2\epsilon_{t-2} \dots + \theta_q\epsilon_{t-q} + \kappa \qquad (C.3)$$

subtracting the equation with the means from Equation C.3 and then multiplying by $(N_t - \bar{N})$ and taking expectations gives

$$\gamma_{NN}(i) = -\phi_1\gamma_{NN}(i-1) - \phi_2\gamma_{NN}(i-2) \dots - \phi_p\gamma_{NN}(i-p)$$
$$+ \gamma_{N\epsilon}(i) + \theta_1\gamma_{N\epsilon}(i-1) \dots + \theta_q\gamma_{N\epsilon}(i-q)$$

since

$$\gamma_{N\epsilon}(i) = 0 \text{ for } i > 0$$

then

$$\gamma_{NN}(i) = -\phi_1\gamma_{NN}(i-1) - \phi_2\gamma_{NN}(i-2) \dots - \phi_p\gamma_{NN}(i-p)$$
$$\text{for } i \geq q+1$$

or in matrix form this can be expressed as

$$\mathbf{G\Phi} = \mathbf{\Gamma}$$

where

$$\mathbf{\Phi}^T = [-\phi_1, -\phi_2, \dots, -\phi_p]$$
$$\mathbf{\Gamma}^T = [\gamma_{NN}(q+1),\ \gamma_{NN}(q+2),\ \dots,\ \gamma_{NN}(q+p)]$$
$$\mathbf{G} = \begin{bmatrix} \gamma_{NN}(q) & \gamma_{NN}(q-1) & \dots & \gamma_{NN}(q-p+1) \\ \gamma_{NN}(q+1) & \gamma_{NN}(q) & \dots & \gamma_{NN}(q-p+2) \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{NN}(q+p-1) & \gamma_{NN}(q+p-2) & \dots & \gamma_{NN}(q) \end{bmatrix}$$

The initial estimates of the autoregressive parameters (ϕ)

are readily available from these equations, known as the Yule-Walker equations.

Once these estimates are available, Equation C.3 can be written as

$$\phi_0 N_t + \phi_1 N_{t-1} \ldots . + \phi_p N_{t-p}$$
$$= \epsilon_t + \theta_1 \epsilon_{t-1} \ldots + \theta_q \epsilon_{t-q} + \kappa, \quad (\phi_0 = 1)$$

letting

$$w_t = \epsilon_t + \theta_1 \epsilon_{t-1} \ldots + \theta_q \epsilon_{t-q} \qquad \text{(C.4)}$$

allows this equation to be expressed as

$$w_t = \phi_0 N_t + \phi_1 N_{t-1} \ldots + \phi_p N_{t-p} - \kappa$$

subtracting the equation of the means, multiplying by $(w_{t-i} - \bar{w})$ and taking expectations gives

$$\begin{aligned}\gamma_{ww}(i) = {} & \phi_0\phi_0\gamma_{NN}(i) + \phi_0\phi_1\gamma_{NN}(i-1) \ldots \\ & + \phi_0\phi_p\gamma_{NN}(i-p) + \phi_1\phi_0\gamma_{NN}(i+1) + \phi_1\phi_1\gamma_{NN}(i) \\ & \ldots + \phi_1\phi_p\gamma_{NN}(i-p-1) \ldots + \phi_p\phi_0\gamma_{NN}(i+p) \\ & + \phi_p\phi_1\gamma_{NN}(i+p-1) \ldots + \phi_p\phi_p\gamma_{NN}(i)\end{aligned}$$

$$\gamma_{ww}(i) = \sum_{10}^{p} \sum_{k=0}^{p} \phi_l \phi_k \gamma_{NN}(|i+l-k|) \text{ for } i=0,\ldots,q$$

Equation C.4 can be rewritten as

$$w_t = \tau_0 e_t + \tau_1 e_{t-1} \ldots + \tau_q e_{t-q} \quad e_t \approx N(0,1)$$

with

$$\tau_0 = \sigma_e$$
$$\tau_i = \theta_i \tau_0$$

since

$$\gamma_{ee}(i) = 0 \text{ for } i \neq 0$$
$$\gamma_{ee}(0) = 1$$

Noting that the means are zero and multiplying by w_{t-i} and taking expectations yields

$$\gamma_{ww}(i) = \tau_0\tau_i + \tau_1\tau_{i+1} \ldots + \tau_{q-i}\tau_q$$
$$= \sum_{j=0}^{q-i} \tau_j\tau_{i+j} \quad i=1,\ldots,q$$

so the estimates of the moving average parameters can be obtained by iterating on τ (e.g. Newton-Raphson) to solve

$$\sum_{j=0}^{q-i} \tau_j\tau_{i+j} - \gamma_{ww}(i) = 0$$

Nonlinear least squares prewhitening parameter estimation

This algorithm is the same as the maximum likelihood estimation algorithm described in Chapter 4 except for the calculation of the derivatives. Equation C.3 can be rewritten as

$$\epsilon_t = N_t + \phi_1 N_{t-1} \ldots + \phi_p N_{t-p} - \theta_1\epsilon_{t-1} - \theta_2\epsilon_{t-2} \ldots$$
$$- \theta_q\epsilon_{t-q} - \kappa$$

$$\frac{\partial\epsilon_t}{\partial\theta_i} = -\epsilon_{t-i} - \theta_1 \frac{\partial\epsilon_{t-1}}{\partial\theta_i} \ldots - \theta_q \frac{\partial\epsilon_{t-q}}{\partial\theta_i}$$

$$\frac{\partial\epsilon_t}{\partial\phi_i} = N_{t-i} - \theta_1 \frac{\partial\epsilon_{t-1}}{\partial\phi_i} \ldots - \theta_q \frac{\partial\epsilon_{t-q}}{\partial\phi_i}$$

$$\frac{\partial\epsilon_t}{\partial\kappa} = -1 - \theta_1 \frac{\partial\epsilon_{t-1}}{\partial\kappa} \ldots - \theta_q \frac{\partial\epsilon_{t-q}}{\partial\kappa}$$

with the unknown values set to zero.

C.7.2 Transfer function plus noise model order identification

Once the impulse response has been obtained it can be used to estimate the noise series from

$$N_t = y_t - Vu_t - \beta$$

for

$$V = v_0 + v_1q^{-1} + v_2q^{-2} \ldots + v_kq^{-k}$$

Analysis of the autocorrelation function, partial autocorrelation function and power spectrum of the noise series will yield the noise model structure.

For the transfer function recall that

$$y_t = \frac{B}{A}u_{t-k} + N_t + \beta$$

and

$$y_t = Vu_t + N_t + \beta$$

then

$$AVu_t = Bu_{t-k}$$

$$AV = Bq^{-k}$$

equating equal powers of q^{-1}

$$v_i = 0 \text{ for } i=0,..,k$$

$$v_{k+i} = b_i - \sum_{j=1}^{\min(i-1,n)} v_{k+i-j}a_j \text{ for } i=1,..,m$$

$$n = \text{order of } A$$

$$m = \text{order of } B$$

$$v_{k+i} = - \sum_{j=1}^{\min(i-1,n)} v_{k+i-j}a_j \text{ for } i>m$$

This means that the impulse response will be zero up to the lag k. Then the weights will follow an inhomogeneous n'th order difference equation. Finally the weights will follow a homogeneous n'th order difference equation after m terms. The delay, k, is the number of initial crosscorrelation values that are zero. The choice of n and m is less obvious.

Without numerator dynamics (i.e. m=0) the step response can also be used. The step response is the integral of the impulse response or in the discrete case $\text{step}_i = \sum_{j=0}^{i} v_j$. Use of the step response to determine the delay may be misleading if a series of small impulse responses of the same sign appear before the delay has passed. These small values may render the first step response weight negligible through the summation and thus unnecessarily increase the estimated delay. They can also "create" a step weight where there should be none. This would induce an underestimation of the delay. The step response of a zero'th order process is complete at the first step (i.e. it is flat after the first step). A first order process follows a simple exponential curve. At least a second order process will be necessary to explain anything more complex (e.g. overshoot with oscillations, etc.).

The addition of numerator dynamics make the issue more confusing by adding terms to the initial response. The following guidelines should help. Numerator dynamics will cut off after m lags, denominator dynamics will tail off. An exponential decay of the impulse function (may be of alternating signs) indicates that n=1, a damped sine wave and/or damped exponential indicates that n=2 or higher. A process with both numerator and denominator dynamics will have m lags without any obvious patterns and then an exponential decay (n=1) or damped sine wave and/or damped exponential (n=2). Examples of impulse (crosscorrelation) responses for various model orders are presented in Figure 10.6 in Box and Jenkins, 1976.

Assuming a low order model initially and then adding (an) extra parameter(s) only if the model is inadequate will generally produce satisfactory results. It should be noted that the numerator must have at least one parameter otherwise the output is not related to the input.

If the noise level is low, an approximate solution may be used. If the input is white noise and the noise is low, the output can be thought of as a time series driven by only one source of noise. In this case the autocorrelation function, partial autocorrelation function and power spectrum of the output (prewhitened if necessary) can be used to find the transfer function model orders as it would be done for the noise model or for the input prewhitening. This is only an approximation but if the signal to noise ratio is high it will yield good results that are easier to interpret than the use of the crosscorrelation or impulse response.

C.8 Summary of model order identification

To identify model orders, plots of the autocorrelation function, partial autocorrelation function and crosscorrelation function are used. But in order for these plots to be meaningful the input to the transfer function model must be white noise. If the input is not white, a filter must be applied to make the input white. In order that the correct transfer function is identified, the same filter must be applied to the output signal. This filtering of the input and output signals to produce a white input is termed prewhitening. Therefore in the general case the first step in time series analysis is the determination of a prewhitening filter.

C.8.1 Prewhitening filter identification

The first step of prewhitening order identification is to find if differencing the data is required. This is achieved by examining the autocorrelation and partial autocorrelation function plots of the input data. The data should be differenced once if the magnitude of the autocorrelation and partial autocorrelation do not decrease rapidly (indicating a nonstationary data set). If differencing was required, the autocorrelation and partial autocorrelation function plots of the differenced series must be examined to assess if further differencing is required. If so, difference the data again but it is advisable to not difference the data more than twice. A model for which the data had to be differenced is termed an integrated model.

Once the degree of differencing has been established Table C.1 can be used as a guide to select the orders of the filter.

The model found is known as an AutoRegressive Integrated Moving Average (ARIMA) model. The model is denoted by ARIMA(p,d,q) where p is the autoregressive order, d the number of times the data is differenced and q is the moving average order.

The transfer function model does not have the same structure as the prewhitening transfer function therefore the estimation of the prewhitening parameters must be performed by one of the prewhitening identification algorithms as described in Section C.7.

C.8.2 Prewhitening parameter estimation validation

The residuals of the least squares prewhitening estimation should be white noise. To check if the residuals are white noise, the autocorrelation and partial autocorrelation function plots should be examined. If their values are significant (outside the confidence limits) then the residuals are not white. The power spectrum should exhibit an even distribution across all frequencies. The normalized cumulative power spectrum plot should follow the diagonal within the set confidence limits.

If the magnitude of the highest order autoregressive or moving average parameter is close to zero, consideration should be given to reducing the model order with a minimum loss of accuracy. If more than one parameter is close to zero, eliminate only one parameter at a time (select the parameter nearest zero or the one with the overall strongest correlation in the parameter correlation matrix). If the reduced model identification gives another parameter close to zero eliminate one parameter and check the reduced model again.

C.8.3 Transfer function order identification

Once the prewhitening filter has been identified the transfer function model order can be found. First check if the filter produces a white noise input (check as for the residuals above).

Two different methods can be used to find the transfer function orders. For the first method (approximate) the model order can be found in the same manner as for the prewhitening order (i.e. assuming low noise level, the transfer function can be thought of as a prewhitening filter since the input is now white noise). The second method (more exact) uses the fact that the crosscorrelation function plot is similar to the impulse response plot for a white noise input (so the crosscorrelation or the impulse response can be used in the following).

The plot of the crosscorrelation function is examined first to determine if there is any delay. If there is a delay, the crosscorrelation function will have k steps equal to zero i.e. within the confidence limits.

Not including the delay, the crosscorrelation function plot of a moving average process cuts off (sharp difference between the last significant and the first non significant crosscorrelation) after m lags while an autoregressive process tails off. An exponential decay (may be alternating sign) indicates that n=1, a damped sine wave and/or damped exponential indicates that n=2 or higher. A mixed process has m lags without any pattern and then an exponential decay (n=1) or a damped sine wave and/or damped exponential (n=2 or higher). For examples of impulse (crosscorrelation) responses of various model orders see Figure 10.6 in Box and Jenkins, 1976.

Note that there should always be at least one moving average parameter, otherwise the right hand side of the transfer function equation is zero i.e. the output is not related to the input but only to white noise.

Estimation of the noise model order is done in the same manner as for prewhitening. The noise estimates are generated using the impulse response as a model of the process since the transfer function parameters are not yet available.

Appendix D: MATHEMATICAL RELATIONSHIPS AND DEFINITIONS

D.1 Vector algebra

Let Y and U be vectors and Ψ and Ξ be matrices and lower case letters denote elements of a matrix or a vector. The following relationships are given without proof. For more details see Stuart, 1973 or Golub and Van Loan, 1983.

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix}$$

$$Y^T = [y_1, y_2, \ldots]$$

$$\Psi = \begin{bmatrix} \psi_{11} & \psi_{12} & \cdots \\ \psi_{21} & \psi_{22} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$\Psi^T = \begin{bmatrix} \psi_{11} & \psi_{21} & \cdots \\ \psi_{12} & \psi_{22} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$(Y^T\Psi)^T = \Psi^T Y$

$(Y^T\Psi U)^T = Y^T\Psi U = U^T\Psi^T Y = (\Psi U)^T Y$ ($Y^T\Psi U$ is scalar)

$(Y+\Psi U)^T = Y^T+(\Psi U)^T$

$$\frac{d\ Y^T}{d\ U} = \begin{bmatrix} \frac{d\ Y_1}{d\ U_1} & \frac{d\ Y_2}{d\ U_1} & \cdots \\ \frac{d\ Y_1}{d\ U_2} & \frac{d\ Y_2}{d\ U_2} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$\frac{d\ Y^TU}{d\ U} = Y$ (if Y is not a function of U)

if $\Psi_{i,j}=\Psi_{j,i}$ Ψ is said to be symmetric

$\Psi^T\Psi$ is symmetric

$\frac{d\ U^T\Psi U}{d\ U} = 2\Psi U$ if Ψ is symmetric

$\Psi^T = \Psi$ if Ψ is symmetric

$(\Psi^{-1})^T = (\Psi^T)^{-1}$

$\Psi^T\Xi\Psi$ is symmetric if Ξ is symmetric

$\Psi\Psi^T$ is symmetric

Ψ is singular if it is not full rank (determinant=0)

if Ψ is nonsingular Ψ^{-1} exist

$[\Psi^T\Psi]^{-1}\Psi$ is called the pseudo inverse of Ψ

let Ψ and Ξ be two matrices of appropriate dimensions, then

$E[\Psi+\Xi] = E[\Psi] + E[\Xi]$

$E[\Psi\Xi] \leq E[\Psi]\ E[\Xi]$

where E[ ] denotes the expected values

D.2 Norms and condition numbers

This is a condensed form of the material presented by Dennis and Schnabel, 1983. The condition number of a matrix A is defined as

$$\kappa_p(A) = \|A\|_p\ \|A^{-1}\|_p$$

where $\|A\|_p$ is a suitable norm for the matrix A. A suitable matrix norm is the induced norm defined as

$$\|A\|_p = \max_{v\in R^n, v\neq 0} \left\{\frac{\|Av\|_p}{\|v\|_p}\right\}$$

and the class of vector norms is

$$\|v\|_p = \left[\sum_{i=1}^{n} |v_i|^p\right]^{1/p}$$

The most commonly used matrix norms are

$$\|A\|_1 = \max_{1\le j\le n} \|a_{.j}\|_1 \text{ (max. column sum)}$$

$$\|A\|_2 = (\text{max eigenvalue of } A^TA)^{1/2}$$

$$\|A\|_\infty = \max_{1\le i\le n} \|a_{i.}\|_1 \text{ (max. row sum)}$$

$$\|A\|_F = (\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2)^{1/2} \text{ (Frobenius norm)}$$

Some useful properties are

$$\|AB\| \le \|A\|\cdot\|B\|$$

if A is symmetric

$$\|A\|_2 = \max_{1\le i\le n} |\lambda_i|$$

$$\lambda_i = \text{ith eigenvalue of } A$$

for c, a constant,

$$\|cA\| = c\|A\|$$

$$\kappa(cA) = \kappa(A)$$

for A of dimension nxn (square)

$$n^{-1/2}\|A\|_\infty \le \|A\|_2 \le n^{1/2}\|A\|_\infty$$

D.3 Linear operators

Define the back shift operator q^{-1} as

$$q^{-1}y_t = y_{t-1}$$

which leads to

$$q^{-2}y_t = q^{-1}(q^{-1}y_t) = y_{t-2}$$

or in polynomial form

where $\|A\|_p$ is a suitable norm for the matrix A. A suitable matrix norm is the induced norm defined as

$$\|A\|_p = \max_{v \in R^n, v \neq 0} \left[\frac{\|Av\|_p}{\|v\|_p} \right]$$

and the class of vector norms is

$$\|v\|_p = \left[\sum_{i=1}^{n} |v_i|^p \right]^{1/p}$$

The most commonly used matrix norms are

$$\|A\|_1 = \max_{1 \leq j \leq n} \|a_{\cdot j}\|_1 \text{ (max. column sum)}$$

$$\|A\|_2 = (\text{max eigenvalue of } A^T A)^{1/2}$$

$$\|A\|_\infty = \max_{1 \leq i \leq n} \|a_{i \cdot}\|_1 \text{ (max. row sum)}$$

$$\|A\|_F = (\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2)^{1/2} \text{ (Frobenius norm)}$$

Some useful properties are

$$\|AB\| \leq \|A\| \cdot \|B\|$$

if A is symmetric

$$\|A\|_2 = \max_{1 \leq i \leq n} |\lambda_i|$$

$$\lambda_i = \text{ith eigenvalue of } A$$

for c, a constant,

$$\|cA\| = c\|A\|$$

$$\kappa(cA) = \kappa(A)$$

for A of dimension nxn (square)

$$n^{-1/2}\|A\|_\infty \leq \|A\|_2 \leq n^{1/2}\|A\|_\infty$$

D.3 Linear operators

Define the back shift operator q^{-1} as

$$q^{-1}y_t = y_{t-1}$$

which leads to

$$q^{-2}y_t = q^{-1}(q^{-1}y_t) = y_{t-2}$$

or in polynomial form

where $\|A\|_p$ is a suitable norm for

matrix norm is the induced norm d

$$\|A\|_p = \max_{v \in R^n, v \neq 0} \left\{ \frac{\|Av\|_p}{\|v\|_p} \right.$$

and the class of vector norms is

$$\|v\|_p = \left[\sum_{i=1}^{n} |v_i|^p \right]^{1/p}$$

The most commonly used matrix nor

$$\|A\|_1 = \max_{1 \le j \le n} \|a_{\cdot j}\|_1 \text{ (ma}$$

$$\|A\|_2 = \text{(max eigenvalu}$$

$$\|A\|_\infty = \max_{1 \le i \le n} \|a_{i \cdot}\|_1 \text{ (ma}$$

$$\|A\|_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2}$$

Some useful properties are

$$\|AB\| \le \|A\| \cdot \|B\|$$

if A is symmetric

$$\|A\|_2 = \max_{1 \le i \le n} |\lambda_i|$$

λ_i = ith eigenvalue

$$V=\nabla^2 J=\begin{bmatrix} \frac{\partial^2 J}{\partial\theta_1\,\partial\theta_1} & \frac{\partial^2 J}{\partial\theta_1\,\partial\theta_2} & \cdots \\ \frac{\partial^2 J}{\partial\theta_2\,\partial\theta_1} & \frac{\partial^2 J}{\partial\theta_2\,\partial\theta_2} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$\theta_{k+1} = \theta_k - V^{-1}\nabla J_k$$

This is known as the Gauss-Newton algorithm. To ensure that V is invertible, it is sufficient that V be positive definite. For this, V is modified to

$$V = V + \nu I$$

where ν is a positive constant such that $\nu > -\min\{\lambda_i\}$ where λ_i are the eigenvalues of V in the Gauss-Newton alogrithm. That is, ν guarantees invertibility by moving the eigenvalues of V into the right half plane. The algorithm becomes

$$\theta_{k+1} = \theta_k - [V+\nu I]^{-1}\nabla J_k$$

This is known as the Marquardt-Levenberg algorithm and ν is known as the Marquardt-Levenberg correction factor. Note that when ν is very large, the algorithm behaves like a steepest descent algorithm and when ν is small it reduces to the Gauss-Newton algorithm. The choice of a large initial ν will ensure a rapid convergence when far from the optimum and reducing ν will ensure a rapid convergence near the optimum. The algorithm was modified by Levenberg to include scaling of the Hessian and the derivative vector. This helps the algorithm by ensuring that ν has a similar relative effect on all the diagonal elements of the Hessian. The

D.5 Matrix inversion lemma

Let A, B, C and D be matrices of appropriate dimensions, with A square and nonsingular, then

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

The reader can easily verify this by premultiplying both sides by [A +BCD] and obtaining the identity matrix on both sides.

D.6 Householder transformation

Consider a Householder transformation matrix defined as

$$Q = I - 2\ uu^T$$

where u is any unit length vector (i.e. its Eucledian norm is unity). The transformation matrix Q has the following properties

1. Q is symmetric i.e. $Q^T = Q$
2. Q is orthogonal i.e. $Q^TQ = I$
3. the product of Householder transformations is orthogonal i.e. if $Q = Q_p \ldots Q_1$ where Q_i are Householder transformations then $Q^TQ = I$
4. given any vector x there exists a Householder transformation Q such that

$$Qx = \begin{bmatrix} \lambda \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where $\lambda = (x^Tx)^{1/2}$. It can be shown that

$$Q = I - 2\ uu^T$$

with

$$u_1 = [\frac{1}{2}(1 - x_1/\lambda)]^{1/2}$$

$$u_i = -x_i/(2u_1\lambda)$$

satisfies the above equation.

5. given any N x p matrix X (N > p) there exists an orthogonal matrix Q such that

$$QX = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

where R is a p x p upper triangular matrix and Q is the product of p Householder transformations. A transformation Q_1 can be found such that

$$Q_1X = \begin{bmatrix} \lambda_1 & x'_{12} & x'_{13} & \dots & x'_{1p} \\ 0 & x'_{22} & x'_{23} & \dots & x'_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & x'_{N2} & x'_{N3} & \dots & x'_{Np} \end{bmatrix}$$

similarly Q_2 is defined by $I - 2u'u'^T$ where the first element of u' is zero. Then Q_2 can be chosen such that

$$Q_2Q_1X = \begin{bmatrix} \lambda_1 & x'_{12} & x'_{13} & \dots & x'_{1p} \\ 0 & \lambda_2 & x''_{23} & \dots & x''_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & x''_{N3} & \dots & x''_{Np} \end{bmatrix}$$

continuing this process yields

$$Q = Q_p \dots Q_1$$

D.7 QR decomposition

As an example, take the least squares estimates that can be shown to be

$$\theta = (\Psi^T\Psi)^{-1}\Psi^T Y$$

Unfortunately $\Psi^T\Psi$ is often ill-conditioned, leading to an inaccurate inverse. In this circumstance special numerical procedures are required. The QR decomposition is applied before the formation of $\Psi^T\Psi$ to prevent numerical inaccuracies that could occur due to the squaring of the Ψ matrix. The least squares problem can be written as

$$\Psi\theta = Y$$

Applying a series of Householder transformations $Q = Q_p \ldots Q_1$, where p is the total number of parameters, gives

$$Q\Psi\theta = QY$$

with

$$Q\Psi = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

$$QY = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$$

which gives

$$R\theta = \eta_1$$

This can readily be solved by back substitution as R is an upper triangular matrix. The sum of squares of residuals (cost function) is

$$J = (Y - \Psi\theta)^T(Y - \Psi\theta)$$

using the orthogonal property $Q^TQ = I$

$$J = (Y - \Psi\theta)^T Q^T Q(Y - \Psi\theta)$$

$$J = (Y^T Q^T - \theta^T \Psi^T Q^T)(QY - Q\Psi\theta)$$

where

$$Y^T Q^T = (QY)^T = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}^T$$

$$\Psi^T Q^T = (Q\Psi)^T = \begin{bmatrix} R \\ 0 \end{bmatrix}^T$$

$$J = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}^T \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} - \theta^T \begin{bmatrix} R \\ 0 \end{bmatrix}^T \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$$

$$- \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}^T \begin{bmatrix} R \\ 0 \end{bmatrix} \theta + \theta^T \begin{bmatrix} R \\ 0 \end{bmatrix}^T \begin{bmatrix} R \\ 0 \end{bmatrix} \theta$$

$$J = \eta_1^T \eta_1 + \eta_2^T \eta_2 - \theta^T R^T \eta_1 - \eta_1^T R\theta + \theta^T R^T R\theta$$

$$J = (\eta_1 - R\theta)^T (\eta_1 - R\theta) + \eta_2^T \eta_2$$

since the least squares solution is

$$R\theta = \eta_1$$

the first term is zero and the sum of squares of residuals becomes

$$S_R^2 = \eta_2^T \eta_2$$

D.7.1 Recursive QR factorization

After factorization the solution to the least square problem is

$$R_{N+1}\theta_{N+1} = \eta_{N+1}$$

where R_{N+1} is a pxp upper triangular matrix and η_{N+1} is a vector of length p, p the total number of parameters to identify. This set of equations can be solved by back substitution. The remaining N-p equations are of the form

$$0 = \nu_{N+1}$$

therefore $\nu_{N+1}^T\nu_{N+1}$ is the sum of squares of residuals, S_{N+1}^2.

D.7.2 Update of R_{N+1} and η_{N+1}

Consider the situation when N observations have been collected. Assume that the Householder transformation procedure for the QR factorization has been used. Then

$$J_N = (\eta_N - R_N\theta_N)^T(\eta_N - R_N\theta_N) + S_N^2$$

$$J_{N+1} = (\eta_N - R_N\theta_{N+1})^T(\eta_N - R_N\theta_{N+1}) + S_N^2$$
$$+ (y_{N+1} - \phi_{N+1}^T\theta_{N+1})^2$$

$$J_{N+1} = \left[\begin{bmatrix} \eta_N \\ \\ y_{N+1} \end{bmatrix} - \begin{bmatrix} R_N \\ \\ \phi_{N+1}^T \end{bmatrix} \theta_{N+1}\right]^T \left[\begin{bmatrix} \eta_N \\ \\ y_{N+1} \end{bmatrix} - \begin{bmatrix} R_N \\ \\ \phi_{N+1}^T \end{bmatrix} \theta_{N+1}\right] + S_N^2$$

An orthogonal transformation Q_N can be found such that

$$Q_N \begin{bmatrix} R_N \\ \phi_{N+1}^T \end{bmatrix} = \begin{bmatrix} R_{N+1} \\ 0 \end{bmatrix}$$

with Q_N expressed as $Q_p \ldots Q_1$ where Q_i are Givens orthogonal transformations of the form

$$Q_i = \left[\begin{array}{c:c:c:c}
I_{i-1} & \multicolumn{3}{c}{0} \\
\hdashline
 & C_i & 0 & S_i \\
\cdashline{2-4}
0 & 0 & I_{p-i} & 0 \\
\cdashline{2-4}
 & -S_i & 0 & C_i
\end{array}\right]$$

where

$$C_i = R_{N,ii}/\lambda_i$$

$$S_i = \phi_{N+1,i}/\lambda_i$$

$$\lambda_i = [R^2_{N,ii} + \phi^2_{N+1,i}]^{1/2}$$

now define η_{N+1} and ν_{N+1} from

$$Q_N \begin{bmatrix} \eta_N \\ y_{N+1} \end{bmatrix} = \begin{bmatrix} \eta_{N+1} \\ \nu_{N+1} \end{bmatrix}$$

recall that since Q_N is orthogonal $Q_N^T Q_N = I$, then

$$J_{N+1} = \left[\begin{bmatrix} \eta_N \\ y_{N+1} \end{bmatrix} - \begin{bmatrix} R_N \\ \phi^T_{N+1} \end{bmatrix} \theta_N \right]^T Q_N^T Q_N \left[\begin{bmatrix} \eta_N \\ y_{N+1} \end{bmatrix} - \begin{bmatrix} R_N \\ \phi^T_{N+1} \end{bmatrix} \theta_N \right] + S_N^2$$

$$J_{N+1} = \begin{bmatrix} \eta_{N+1} \\ \nu_{N+1} \end{bmatrix}^T \begin{bmatrix} \eta_{N+1} \\ \nu_{N+1} \end{bmatrix} - \theta^T_{N+1} \begin{bmatrix} R_{N+1} \\ 0 \end{bmatrix}^T \begin{bmatrix} \eta_{N+1} \\ \nu_{N+1} \end{bmatrix}$$

$$- \begin{bmatrix} \eta_{N+1} \\ \nu_{N+1} \end{bmatrix}^T \begin{bmatrix} R_{N+1} \\ 0 \end{bmatrix} \theta_{N+1} + \theta^T_{N+1} \begin{bmatrix} R_{N+1} \\ 0 \end{bmatrix}^T \begin{bmatrix} R_{N+1} \\ 0 \end{bmatrix} \theta_{N+1} + S_N^2$$

$$J_{N+1} = \eta^T_{N+1}\eta_{N+1} + \nu^2_{N+1} - \theta^T_{N+1}R^T_{N+1}\eta_{N+1} - \eta^T_{N+1}R_{N+1}\theta_{N+1}$$

$$+ \theta^T_{N+1}R^T_{N+1}R_{N+1}\theta_{N+1} + S_N^2$$

$$J_{N+1} = (\eta_{N+1} - R_{N+1}\theta_{N+1})^T(\eta_{N+1} - R_{N+1}\theta_{N+1}) + \nu^2_{N+1} + S^2_N$$

$$S^2_{N+1} = S^2_N + \nu^2_{N+1}$$

The QR decomposition is also known as a square root decomposition since

$$P_N = (\Psi_N^T\Psi_N)^{-1}$$

$$= (\Psi_N^T Q^T Q \Psi_N)^{-1}$$

$$P_N = \left[\begin{bmatrix} R_N \\ 0 \end{bmatrix}^T \begin{bmatrix} R_N \\ 0 \end{bmatrix}\right]^{-1}$$

$$P_N = (R_N^T R_N)^{-1}$$

from this R_N can be thought of as the square root of the inverse of P_N.

Appendix E: SELECTED IDENTIFICATION BIBLIOGRAPHY

In this appendix, several papers that were considered but are not referenced in the thesis are enumerated. These papers were not deemed relevant for the focus of the thesis or their content could be found in an alternate reference.

1. Abaza, B.A. **Multivariable Parameter Estimation Using a 2-step Least-Squares Estimator.** *Electronic Letters* 12(13) pp.331 332 **1976.**

2. Abul-Haggag, O., Soliman, M.A. **A new Approach for the Identification of Stochastic Models.** *IFAC 7th Conf. on Digital Computer Applications to Process Control (Vienna)* pp.651 659 **1985.**

3. Ahlen, A., Solbrand, G. **Repeated recursive identification for off line estimation.** *ACI 83* pp.24.9 24.13 **1983.**

4. **Ahmed, M.S.,** Shrindhar, **M. New Gain Algorithms for System Identification by Stochastic Approximation.** *Control of Industrial Processes* Milan pp.181 201 **1981.**

5. **Akaike, H. A New Look at the Statistical Model Identification.** *IEEE Trans.* AC-19(6) pp.716 723 **1974.**

6. **Akaike, H. On Entropy Maximization Principle.** *Applications of Statistics. Krishnaiah.* **pp.27 41 1977.**

7. Amemiya, **T. The Nonlinear Two-Stage Least-Squares Estimator.** *J. of Econometrics* 2 pp.105 110 **1974.**

8. Amemiya, **T. The Maximum Likelihood and the Nonlinear Three-Stage Least Squares Estimator in the General Nonlinear Simultaneous Equation Model.** *Econometrica* **45(4) pp.955 968 1977.**

9. **Andersen, N. On the Calculation of Filter Coefficients for Maximum Entropy Spectral Analysis.** *Geophysics* 39(1) **pp.69 72 1974.**

10. **Anderson, B.D.O. Adaptive Systems, Lack of Persistency of Excitation and Bursting Phenomena** *Automatica* 21(3) **pp.247 258 1985.**

11. Anderson, B.D.O., Chung Tsoi, Ah. **Connective Forward and Backward Autoregressive Models.** *IEEE Trans.* AC-29(10) pp.917 925 1984.

12. Anderson, B.D.O., Johnson, C.R. jr. **Exponential Convergence of Adaptive Identification and Control Algorithms.** *Automatica* 18(1) pp.1 13 1982.

13. Andersson, P. **Adaptive forgetting in recursive identification through multiple models.** *Int. J. Control* 42(5) pp. 1175 1193 1985.

14. Aroian, L.A. **Time Series in M-Dimensions Definition, Problems and Prospects.** *Commun. Statist.-Simulla. Computa.* B9(5) pp.453 465 1980.

15. Astrom, K.J. **Numerical identification of linear dynamic systems from normal operating records.** *IFAC Proc. Theory of self-adaptive control systems (Teddington)* pp. 96 111 1965.

16. Baram, Y., Shaked U. **Minimal Order Estimation of Multivariable continuous time stochastic linear systems** *IEEE Trans.* AC-30(5) pp.483 1985.

17. Barrett, J.F., Moir, T.J. **Spectrum Analysis Using Kepstrum Coefficients.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part II. digest 1984/21* 1984.

18. Bastin, G., Dochain D. **On-line estimation of microbial specific growth rates.** *Automatica* 22(6) pp.705 709 1986.

19. Baur, U., Isermann, R. **On-Line Identification of a Heat Exchanger with a Process Computer-a Case Study.** *Automatica* 13 pp.487 496 1977.

20. Beach, C.M., MacKinnon, J.G. **A Maximum Likelihood Procedure for regression with Autocorrelated Errors.** *Econometrica* 46(1) pp.51 58 1978.

21. Beck, M.B., Cook, S.C., Jowitt, P.W. **Environmental Systems Identification: Problems for the Development of Recursive Estimation Algorithms.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part I digest 1984/19* 1984.

22. Bekey, G.A., Karuza, S. **Observations on Some Commonly Used Methods for Identification of Parameters in Linear Systems.** *Simulation* sept. pp.69 75 1974.

23. Belzer, M.R., Bierman, G.J., Vandergraft, J.S. **Maximum Likelihood Estimation Using Square Root Information Filters.** *10th World Cong. on Auto. Control*. 10 pp.300 305 1987.

24. Bierman, G.J. **Sequential Square Root Filtering and Smoothing of Discrete Linear Systems.** *Automatica* 10 pp. 147 158 1974.

25. Bierman, G.J. **Measurement Updating using the U-D Factorization.** *Automatica* 12 pp.375 382 1976.

26. Bierman, G.J. **Efficient Time Propagation of U-D Covariance Factors.** *IEEE trans.* AC-26(4) pp. 890 894 1981.

27. Bierman, G.J., Thornton, C.L. **Numerical Comparison of Kalman Filter Algorithms: Orbit Determination case study.** *Automatica* 13 p..23 35 1977.

28. Billings, S.A. **Parameter Estimation, Structure Detection and Model Validity Tests for Nonlinear Systems.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part I digest 1984/19* 1984.

29. Billings, S.A., Voon, W.S.F. **Correlation based model validity tests for non-linear models** *Int. J. Control* 44(1) pp.235 244 1986.

30. Bohlin, T. **Maximum-Power Validation of Models Without Higher-Order Fitting.** *Automatica* 14 pp.137 146 1978.

31. Box, G.E.P., MacGregor, J.F. **Parameter Estimation with Closed-Loop Operating Data.** *Technometrics* 18(4) pp.371 380 1976.

32. Briggs, P.A.N., Godfrey, K.R. **Pseudorandom Signals for the Dynamic Analysis of Multivariable Systems.** *Proc. IEE* 113(7) pp.1259 1267 1966.

33. Bunch, J.R., Nielsen, C.P. **Updating the Singular Value Decomposition.** *Numer. Math.* [illegible] pp.111 129 1978.

34. Bunch, J.R., Nielsen, C.P., Sorensen, D.C. **Rank-One Modification of the Symmetric Eigenproblem.** *Numer. Math.* 31 pp.31 48 1978.

35. Busigner, P., Golub, G.H. **Linear least squares solutions by Householder transformations** *Num. Math.* 7 pp.269 276 1965.

36. Candy, J.V., Zicker, J.E. **Electromagnetic Signal Processing: An Estimation/Identification Application** *Automatica* 23(2) pp.175 187 1987.

37. Chen, H.F. **Recursive System Identification and Adaptive control by use of the Modified Least Squares Algorithm.** *SIAM J. Control and Optimization* 22(5) pp.758 776 1984. paper 3.17 1967.

38. Clarke, D.W. **Offsets and K-Incremental Predictors in Self-Tuning Control.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part II digest 1984/21* 1984.

39. Clarke, D.W., Briggs, P.A.N. **Errors in Weighting Sequence Estimation I. A Matrix Formulation.** *Int. J. Control* 11(1) pp.49 56 1970.

40. Clarke, D.W., Briggs, P.A.N. **Errors in Weighting Sequence Estimation II. The Effects of Autocorrelated Noise.** *Int. J. Control* 11(1) pp.57 65 1970.

41. Cluett, W.R., Martin-Sanchez, J.M., Shah, S.L., Fisher, D.G. **Stable Discrete-Time adaptive control in the presence** *IEEE trans. Aut. Contr.* 33(4) pp.410 414 1988.

42. Cluett, W.R., Shah, S.L., Fisher, D.G. **stable robust adaptive controller** *Int. J. Control* 45(4) pp.1265 1273 1987.

43. Correa, G.O., Glover, K. **On the Choice of Parametrization for Identification** *IEEE Trans.* AC-31(1) pp.8 15 1986.

44. Dasgusta, S., Huang, Y.-F. **Asymptotically Convergent Modified Recursive Least-Squares with Data-Dependent Updating and Forgetting Factor for Systems with Bounded Noise.** *IEEE trans.* IT-33(3) pp.383 392 1987.

45. Davison, E.J., Kwong, R.H., Zhang, Y.-L. **On the identification of linear time-invariant systems using finite plant records.** *IFAC 9th triennial world congress* pp.637 642 1984.

46. Defaye, G., Caralp, L. **Identification dynamique d'un dé propaniseur.** *C.A.C.E. '79 European Federation of Chemical Engineering* pp.184 197.

47. Defaye, G., Caralp, L. **Essai de modélisation d'une distillation industrielle par identification dynamique. Etude d'un dépropaniseur.** *The Chemical Engineering Journal* 18 pp.131 141 1979.

48. Defaye, G., Caralp, L., Jouve, Ph., Baucou, R. **Modélisation d'un dépropaniseur. Comparaison de différentes méthodes statistiques d'identification dynamique.** *Automatisme* 6/7 pp.177 189 1977.

49. De Groat, R.D., Roberts, R.A. **SVD Update Algorithms and Spectral Estimation Applications.** *19th Asilomar Conference on Circuits, Systems and Computers* pp.601 605 1985.

50. DeKeyser, R.M.C., D'Hulster, F.M., Heyse, J.G., Van Cauwenberghe, A.R. **Experiments With a Software Package for Process Identification and Control.** *IFAC Computer Aided Design of Control Systems Cuenod, M.A. (ed.) pp.473 478* 1980.

51. De Larmimat, P. **An Exact Recursive 2SLS Version of the Instrumental Variable Estimator. Stability Analysis.** *American Control Conference at San Diego.* pp.1105 1106 1984.

52. Desrochers, A.A., Al-Jar, R.Y. **Nonlinear Model Simplification in Flight Control System Design.** *j. of Guidance* 7(6) pp.684 689 1984.

53. Diekmann, K. **Application of Parameter Estimation Methods to Time Variant Systems.** *IFAC 7th Conf. on Digital Applications to Process Control (Vienna)* pp.613 618 1985.

54. Diekmann, K., Unbehauen, H. **Recursive Identification of Multi-Input, Multi-Output Systems.** *IFAC Ident. and Par. Estim. (Darmstadt)* pp.423 428 1979.

55. Doncardi, C., Canon, D. **Sequential Detection of Abrupt Changes in ARMA Models.** *IFAC Adaptive Systems in Control and Signal Processing (San Francisco)* pp.135 136 1983.

56. Dugard, L., Goodwin, G.C., De Souza, C.E. **Prior Knowledge in Model Reference Adaptive Control of Multiinput Multioutput Systems.** *IEEE Trans.* AC-29(8) pp.761 764 1984.

57. Dunyak, J.P., Junkins, J.L., Watson, L.T. **Robust Nonlinear Least Squares Estimation Using the Chow-Yorke Homotopy Method.** *J. of Guidance* 7(6) pp.752 755 1984.

58. Dupont, A. **La nouvelle génération de commande numérique.** *Le Nouvel Automatisme* 49(10) pp.49 56 1984.

59. Edmunds, J.M. Identifying sampled systems using difference operator models. *Control Systems Centre report Non 601.*

60. Eklund, K., Gustavsson, I. Identification of Drum Boiler Dynamics. *IFAC Ident. and Par. Estim. (Hague)* pp.87 108 1973.

61. El-Sherief, H. Multivariable System Structure and Parameter Identification Using the Correlation Method. *Automatica* 17(5) pp.541 544 1981.

62. El-Sherief, H., Abdel-Magid, Y.L. An Efficient On-Line Load-Modeling Algorithm for Short-Term Forecasting of Interconnected Power Systems. *IEEE Trans.* AC-29(2) pp.190 192 1984.

63. Everett, D. Periodic digital sequences with pseudonoise properties *GEC Journal* 33(3) pp.115 1966.

64. Feng, C.-B., Li, H. Design of Discrete-Time Adaptive Systems Based on Nonlinear Programming. *IFAC Adaptive System in Control and Signal Processing (San Francisco)* pp.289 294 1983.

65. Fletcher, R., Powell, M.J.D. On the modification of LDL factorizations *Math. of Comp.* 28(128) pp.1067 1087 1974.

66. Flower, J.O., Windett, G.P. Dynamic Measurement of a Large Diesel Engine Using P.R.B.S. Techniques. part I. Development of Theory for Closed-loop Sampled Systems. *Int. J. of Control* 24(3) pp.379 392 1976.

67. Flower, J.O., Windett, G.P. Dynamic Measurement of a Large Diesel Engine Using P.R.B.S. Techniques. part II. Instrumentation, experimental Techniques and Results. *Int. J. of Control* 24(3) pp.393 404 1976.

68. Fogel, E., Huang, Y. F. On the Value of Information in System Identification - Bounded noise case *Automatica* 18(2) pp.229 238 1982.

69. François, A., Ragot, J., Richard, A. Identification assistée par ordinateur: la méthode des moments. *Le Nouvel Automatisme* 48(9) pp.67 72 1984.

70. Friedlander, B., Stoica, P., Soderstrom, T. Instrumental Variable Methods for ARMA Parameter Estimation. *IFAC Ident. and Sys. Par. Estim. (York)* pp.29 36 1985.

71. Futura, K., Hatakeyama, S., Kominami, H. **Structural Identification and Software Package for Linear Multivariable Systems.** *Automatica* 17(5) pp.755 762 1981.

72. Gabr, M.M. **A Recursive (On-Line) Identification of Bilinear Systems** *Int. j. Control* 44(4) pp.911 917 1986.

73. Gallant, A.R., Jorgenson, D.W. **Statistical Inference for a System of Simultaneous, Non-Linear, Implicit Equations in the Context of Instrumental Variable Estimation.** *J. of Econometrics* 11 pp.275 302 1979.

74. Garner, J.P., Englar, T.S. jr. **A Survey of Some Recently Developed Techniques in Estimation and Identification for Dynamical Systems.** *???????* pp. 185 192 >1984.

75. Gawthrop, P.J. **Parametric Identification of Transient Response Data.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part I digest 1984/19* 1984.

76. Gentleman, W.M. **Least squares computations by Givens transformations without square roots** *J. Inst. Maths Applics* 12 pp.329 336 1973.

77. Gevers, M., Chung Tsoi, Ah. **Structural Identification of Linear Multivariable Systems Using Overlapping Forms: a New Parametrization.** *Int. J. of Control* 40(5) pp.971 987 1984.

78. Gevers, M. Wertz, V. **Uniquely Identifiable State-Space and ARMA Parametrizations for Multivariable Linear Systems.** *Automatica* 20(3) pp.333 347 1984.

79. Gill, P.E., Golub, G.H., Murray, W., Saunders, M.A. **Methods for modifying matrix factorizations** *Math. of Comp.* 28(126) pp.505 535 1974.

80. Godfrey, K.R. **The Identifiability of Linear Compartmental Models.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part II digest 1984/21* 1984.

81. Godfrey, K.R., Moore, D.J. **Identification of processes having direction-dependent responses, with gas-turbine engine applications.** *Automatica* 10 pp.469 481 1974.

82. Golub, G. **Numerical methods for solving linear least squares problems** *Num. Math.* 7 pp.206 216 1965.

83. Golub, G.H., Reinsch, C. **Singular value decomposition and least squares solution** *Num. Math.* 14 pp.402 420 1970.

84. Goodwin, G.C. **Some observations on robust estimation and control** *IFAC Ident. and Sys. Par. Estim. (York)* pp.851 859 1985.

85. Goodwin, G.C., Hill D.J., Palaniswami, M. **A perspective of convergence of adaptive control algorithms** *Automatica* 20 pp. 519 531 1984.

86. Goodwin, G.C., Lozano Leal, R., Mayne, D.Q., Middleton, R.H. **Rapprochement between continuous and discrete model reference adaptive control.** *Automatica* 22(2) pp.199 207 1986.

87. Gotanda, H., Sagara, S., Wada, K. **Noise order determination under unknown process based on covariance of composite noise and output.** *IFAC 9th triennial world congress* pp.711 716 .

88. Gourishankar, V.G., Trybus, L., Rink, R.E. **An adaptive filter for dynamic positioning** *Optimal control applications and methods* 7 pp.271 287 1986.

89. Gustavsson, I., Ljung, L., Soderstrom, T. **Identification of Processes in Closed Loop. -Identifiability and Accuracy Aspects.** *Automatica* 13 pp.59 75 1977.

90. Hagglund, T. **The Problem of Forgetting Old Data in Recursive Estimation.** *IFAC Adaptive System in Control and Signal Processing (San Francisco)* pp.213 214 1983.

91. Hagglund, T. **Recursive Estimation of Slowly Time-Varying Parameters.** *IFAC Ident. and Sys. Par. Estim. (York)* pp.1137 1142 1985.

92. Hanson, R.J., Lawson, C.L. **Extensions and applications of the Householder algorithm for solving least squares problems** *Math. of Comp.* 23 pp.787 812 1969.

93. Hasiewicz, Z., Stankiewicz, A. **On Input-Dependent System Identification by Monte-Carlo Approach.** *IEEE Trans.* AC-30(5) pp.480 483 1985.

94. Hewit, J.R., Wallis, N.A.M. **A sequential steepest descent method for linear system identification.** *Int. J. Systems Sci.* 1(2) pp. 123 131 1970.

95. Hwang, C., Chen, M.-Y. **Analysis and Parameter Identification of Time-Delay Systems via Shifted Legendre Polynomials.** *Int J. Control.* 41(2) pp.403 415 1985.

96. Hwang., C., Guo, T.-Y. **Identification of Lumped Linear Time-Varying Systems via Block-Pulse Functions.** *Int J. Control* 40(3) pp.571 583 1984.

97. Isermann, R., Bauer,U. **Two-Step Process Identification With Correlation Analysis and Least-Squares Parameter Estimation.** *J. of Dynamic Systems, Measurement and Control* dec. 74 pp. 426 432 1974.

98. Jacobs, O.L.R. **Nonlinear Estimation by Online Computation of Bayesian Statistics.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part II digest 1984/21* 1984.

99. Jakeman, A., Young, P. **Refined Instrumental Variable Methods of Recursive Time-Series Part II. Multivariable Systems.** *Int. J. Control* 29(4) pp.621 644 1979.

100. Jakeman, A.J., Young, P.C. **Advanced methods of recursive time-series analysis.** *Int. J. Control* 37(6) pp.1291 1310 1983.

101. Janakiraman, P.A., Renganathan, S. **A New On-Line Identification Method for Discrete Multivariable Linear Systems.** *Automatica* 17(4) pp.653 655 1981.

102. Janiszowsi, K. **Evaluation of the Effects of Identification Errors on the Dynamical Properties of a Process Model.** *Int J. Control* 41(2) pp.445 459 1985.

103. Jumarie, G. **Identification of Stochastic Systems by Using the Maximum Entropy Principle.** *IFAC Ident. and Sys. Par. Estim. (York)* pp.975 979 1985.

104. Kalouptsidis, N.,Carayannis, G., Manolakis, D. **A Fast Covariance Type Algorithm for Sequential Least-Squares Filtering and Prediction.** *IEEE Trans.* AC-29(8) pp.752 755 1984.

105. Kaminski, P.G., Bryson, A.E.jr., Schmidt, S.F. **Discrete square root filtering: a survey of current techniques** *IEEE Trans.* AC-16(6) pp.727 735 1971.

106. Kashyap, R.L. **Maximum likelihood identification of stochastic linear systems.** *IEEE Trans.* AC-15(1) pp. 25 34 1970.

. **Hwang, C., Chen, M.-Y. Analysis and Parameter Identification of Time-Delay Systems via Shifted Legendre Polynomials.** *Int J. Control.* 41(2) pp.403 415 1985.

. **Hwang., C., Guo, T.-Y. Identification of Lumped Linear Time-Varying Systems via Block-Pulse Functions.** *Int J. Control* 40(3) pp.571 583 1984.

. **Isermann, R., Bauer,U. Two-Step Process Identification With Correlation Analysis and Least-Squares Parameter Estimation.** *J. of Dynamic Systems, Measurement and Control* dec. 74 pp. 426 432 1974.

. **Jacobs, O.L.R. Nonlinear Estimation by Online Computation of Bayesian Statistics.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part II digest 1984/21* 1984.

. **Jakeman, A., Young, P. Refined Instrumental Variable Methods of Recursive Time-Series Part II. Multivariable Systems.** *Int. J. Control* 29(4) pp.621 644 1979.

0. **Jakeman, A.J., Young, P.C. Advanced methods of recursive time-series analysis.** *Int. J. Control* 37(6) pp.1291 1310 1983.

1. **Janakiraman, P.A., Renganathan, S. A New On-Line Identification Method for Discrete Multivariable Linear Systems.** *Automatica* 17(4) pp.653 655 1981.

2. **Janiszowsi, K. Evaluation of the Effects of Identification Errors on the Dynamical Properties of a Process Model.** *Int J. Control* 41(2) pp.445 459 1985.

3. **Jumarie, G. Identification of Stochastic Systems by Using the Maximum Entropy Principle.** *IFAC Ident. and Sys. Par. Estim. (York)* pp.975 979 1985.

4. **Kalouptsidis, N.,Carayannis, G., Manolakis, D. A Fast Covariance Type Algorithm for Sequential Least-Squares Filtering and Prediction.** *IEEE Trans.* AC-29(8) pp.752 755 1984.

5. **Kaminski, P.G., Bryson, A.E.jr., Schmidt, S.F. Discrete square root filtering: a survey of current techniques** *IEEE Trans.* AC-16(6) pp.727 735 1971.

6. **Kashyap, R.L. Maximum likelihood identification of stochastic linear systems.** *IEEE Trans.* AC-15(1) pp. 25 34 1970.

120. Landau, I.D., M'Saad, M., M'Sirdi, N.K. **Real Time Process Identification using a Personal Computer.** *IFAC 7th Conf. on Dig. Comp. Applic. to Proc. Cont.* pp.655 659 1985.

121. Landau, I.D., M'Saad, M., M'Sirdi, N.K., Canudas, C. **Identification d'un échangeur de chaleur à l'aide d'un Apple II.** *Revue Gènèrale de thermique.* no 280 pp.387 394 1985.

122. Lawrence, D.A., Johnson, C.R. jr. **Recursive Parameter Identification Algorithm Stability Analysis Via Pi-Sharing.** *IEEE Trans.* AC-31(1) pp.16 24 1986.

123. Leontaritis, I.J., Billings, S.A. **Input-Output Parametric Models for Non-Linear Systems Part I: Deterministic non-Linear Systems.** *Int J. Control* 41(2) pp.303 328 1985.

124. Leontaritis, I.J., Billings,S.A. **Input-Output Parametric Models for Non-Linear Systems Part II: Stochastic Non-Linear Systems.** *Int J. Control* 41(2) pp.329 344 1985.

125. Li, S., Dickinson, B.W. **An efficient method to compute consistent estimates of the AR parameters of an ARMA model** *IEEE Trans.* AC-31(3) pp.275 278 1986.

126. Ljung, L. **Convergence Analysis of Parametric Identification Methods.** *IEEE Trans.* AC-23(5) pp.770 783 1978.

127. Ljung, L. **Analysis of a General Recursive Prediction Error Identification Algorithm.** *Automatica* 17(1) pp.89 99 1981.

128. Ljung, L. **On the Estimation of Transfer Functions.** *Automatica* 21(6) pp. 677 696 1985.

129. Ljung, L., Gustavsson, I., Soderstrom, T. **Identification of Linear Multivariable Systems Operating Under Linear Feedback Control.** *IEEE Trans.* AC-19(6) pp.836 840 1974.

130. Ljung, L., Glover, K. **Frequency Domain Versus Time Domain Methods in System Identification.** *Automatica* 17(1) pp.71 86 1981.

131. Ljung, L., Morf, M., Falconer, D. **Fast Calculation of Gain Matrices for Recursive Estimation Schemes.** *Int J. of Control* 27(1) pp.1 19 1978.

132. Ljung, S., Ljung, L. **Error Propagation Properties of Recursive Least-Squares Adaptation Algorithms.** *Automatica* 21(2) pp.157 167 1985.

133. Lozano L., R. **Convergence Analysis of Recursive Identification Algorithms with Forgetting Factor.** *Automatica* 19(1) pp.95 97 1983.

134. Lozano-Leal, R., Goodwin, G.C. **A Globally Convergent Adaptive Pole Placement Algorithm Without a Persistency of Excitation Requirement** *IEEE 23rd Conf. on Decision and Control*. pp.669 674 1984.

135. Lozano-Leal, R., Ortega, R. **Reformulation of the Parameter Identification Problem for Systems with Bounded Disturbances.** *Automatica* 23(2) pp.247 251 1987.

136. Luman, R.R. **Practical Kalman Filter Software Performance Testing & Validation.** *IEEE Trans.* R-33(3) pp.219 226 1984.

137. MacGregor, J.F. **Optimal choice of the sampling interval for discrete process control** *Technometrics* 18(2) pp.151 160 1976.

138. Maciejowski, J.M. **Approximate Gauss-Markov Realization of Multivariate Stochastic Processes.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part II digest 1984/21* 1984.

139. MacQueen, J., Marshak, J. **Partial Knowledge, Entropy, and Estimation.** *Porc. Nat. Acad. Sci USA* 72(10) pp.3819 3824 1975.

140. Marquardt, D.M. **An Algorithm for Least-Squares Estimation of Nonlinear Parameters.** *J. Soc. Indust. Appl. Math.* 11(2) pp.431 441 1963.

141. Mayne, D.Q., Astrom, K.J., Clark, J.M.C. **A Recursive Algorithm for ARMA Processes.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part I digest 1984/19* 1984.

142. Mayne, D.Q., Astrom, K.J., Clark, J.M.C. **A New Algorithm for Recursive Estimation of Parameters in Controlled ARMA Processes.** *Automatica* 20(6) pp.751 760 1984.

143. Mehra, R.K. **On-Line Identification of Linear Dynamic Systems with Applications to Kalman Filtering.** *IEEE Trans.* AC-16(1) pp.12 21 1971.

144. Mendez Martinez, A. Identification de la réponse impulsionnelle d'un système par intercorrelation au moyen de séquences binaires pseudo-aléatoires. *Thèse docteur-ingénieur faculté des sciences de l'université de Grenoble* 1970.

145. Merrill, W. Identification of Multivariable High-Performance Turbofan Engine Dynamics from Closed-Loop Data. *J. of Guidance* 7(6) pp.667 683 1984.

146. Moir, T.J., Grimble, M.J. Optimal Self-Tuning Filtering, Prediction, and smoothing for Discrete Multivariable Processes. *IEEE Trans.* AC-29(2) pp.128 137 1984.

147. Moore, C. Application of singular value decomposition to the design, analysis, and control of industrial processes *American control conference* pp.643 650 1986.

148. Moore, J.B. On Strong Consistency of Least Squares Identification Algorithms. *Automatica* 14 pp.505 509 1978.

149. Moore, J.B. Stochastic Adaptive Control Via Consistent Parameter Estimation. *IFAC Ident. and Sys. Par. Estim. (York)* pp.611 616 1985.

150. Moore, J.B., Boel, R.K. Asymptotically optimum recursive prediction error methods in adaptive estimation and control. *Automatica* 22(2) pp.237 240 1986.

151. Munack, A., Thoma, M. Application of decomposition/ coordination methods to parameter identification problems in interconnected distributed parameter systems *Automatica* 22(1) pp.111 116 1986.

152. Ng, T.S., Qureshi, Z.H., Cheah, Y.C. Optimal Input Design for an AR Model with Output Constraints. *Automatica* 20(3) pp.359 363 1984.

153. Niederlinski, A. A New Look at Least-Squares Dynamic System Identification. *Int. J. Control* 40(3) pp.467 478 1984.

154. Niederlinski, A. Convergence of Least-Squares Dynamic System Identification with Finite-Accuracy Data. *Int. J. Control* 40(3) pp.479 485 1984.

155. Niederlinski, A. Measurement errors in system identification and the practical universality of low-order LS and AR models. IFAC 9th triennial world congress pp.627 631 1984.

156. Nieman, R.E., Fisher, D.G., Seborg, D.E. **A Review of Process Identification and Parameter Estimation Techniques.** *Int J. Control* 13(2) pp.209 264 1971.

157. Norton, J.P. **Non-Statistical Identification of Models from Records with Bounded Noise.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part II digest 1984/21* 1984.

158. Norton, J.P. **identification of parameter bounds for ARMAX models from records with bounded noise** *Int. J. Control* 45(2) pp.375 390 1987.

159. Nuske, D.J., Wellstead, P.E. **Identification of an Automotive Gas Turbine. part 2. Parameter Fitting.** *Int J. of Cont.* 24(3) pp.311 324 1976.

160. Olcer, S., Egardt, B., Morf, M. **Convergence analysis of ladder algorithms for AR and ARMA models.** *Automatica* 22 pp.345 354 1986.

161. Oprian, C., Taneja, V., Voss D., Aroian, L.A. **General Considerations and Interrelationships Between MA and AR Models, Time Series in N Dimensions, the ARMA Model.** *Commun. Statist.-Simula. Computa.* B9(5) pp.515 532 1980.

162. Ortega, R., Yu, T. **theoretical results on robustness of direct adaptive controllers: a survey** *IFAC Proc.* vol 10 pp.1 15 1987.

163. Oshman, Y., Bar-Itzhack, I.Y. **Square root filtering via covariance and information eigenfactors** *Automatica* 22 pp.599 604 1986.

164. Owens, D.H., Chotai, A. **On Eigenvalues, eigenvectors and singular values in Robust Stability Analysis.** *Int J. of Cont.* 40(2) pp.285 296 1984.

165. Pandrya, R.N., Pagurek, B. **Two Stage Least Squares Estimators and Their Recursive Approximations.** *IFAC Ident. and Sys. Par. Estim. (Hague)* pp.701 710 1973.

166. Panuska, V. **Derivation of the recursive maximum likelihood identification algorithm.** *Electronic letters* 13(1) pp.19 20 1977.

167. Perriot-Mathonna, D.M. **Improvements in the Application of Stochastic Estimation Algorithms-Parameter Jump Detection.** *IEE Trans. on Aut. Cont.* AC-29(11) pp.962 969 1984.

168. Peterka, V. **Subjective Probability Approach to Real-Time Identification.** *Identification and System Parameter Estimation, Rajbman (ed.) pp.1441 1456* 1978.

169. Peterka, V. **A Square Root Filter for Real Time Multivariate Regression.** *Kybernetika* 11(1) pp.53 67 1975.

170. Pfeifer, P.E. **Independence and Sphericity Tests for the Residuals of Space-Time ARMA Models.** *Commun. Statist.-Simula. Computa.* B9(5) pp.533 549 1980.

171. Pfiefer, P.E. **Stationarity and Invertibility Regions for Low Order STARMA Models.** *Commun. Statist.-Simula. Computa.* B9(5) pp.551 562 1980.

172. Porat, B. **Computation of Additive Decomposition of ARMA Spectra.** *IEEE Trans.* AC-29(10) pp.949 951 1984.

173. Puthenpura, S., Sinha, N.K. **Modified maximum likelihood method for the robust estimation of system parameters from very noisy data.** *Automatica* 22(2) pp.231 235 1986.

174. Rana, J.L., Srisailam, M.C. **A General Gradient Estimation Based Identification Algorithm.** *Int. J. Control* 39(6) pp.1309 1321 1984.

175. Reboulet, C., Champelier, C. **An new method for linearizing non-linear systems: the pseudolinearization.** *Int. J. Control* 40(4) pp.631 638 1984.

176. Rissanen, J. **Modeling by Shortest Data Description.** *Automatica* 14 pp. 465 471 1978.

177. Rogers, A.E., Steiglitz, K. **Maximum likelihood estimation of rational transfer function parameters.** *IEEE trans.* AC-12(8) pp.594 597 1967.

178. Romberg T.M., Rees, N.W. **Parameter Adaptive Modelling of a 500 Megawatt Drum Boiler.** *IFAC Ident. and Sys. Par. Estim. (York)* pp.939 944 1985.

179. Rootzen, H., Sternby, J. **Consistency in Least-Squares Estimation: A Bayesian Approach.** *Automatica* 20(4) pp.471 475 1984.

180. Rubin, A.I., Driban, S., Miessner, W.W. **Regression Analysis and Parameter Identification.** *Simulation* 7 pp.39 47 1967.

181. Sakai, H., Tokumaru, H., Futamura A. **ARMA modelling using circular lattice filter.** *IFAC 9th triennial world congress* pp.671 675 1984.

182. Salgado, M.E., Goodwin, G.C., Middleton, R.H. **modified least squares algorithm incorporating exponential resetting and forgetting** *Int. J. Control* 47(2) pp.477 491 1988.

183. Saridis, G.N., Stein, G. **Stochastic Approximation Algorithms for Linear Discrete-Time System Identification.** *IEEE Trans.* AC-13(5) pp.515 523 1968.

184. Saridis, G.N., Stein, G. **A new Algorithm for Linear System Identification.** *IEEE Trans.* AC-13(5) pp.592 594 1968.

185. Savin, N.E., White, K.J. **Testing for Autocorrelation with Missing Observations.** *Econometrica* 46(1) pp.59 67 1978.

186. Seborg, D.E., Edgar, T.F., Shah, S.L. **Adaptive Control Strategies for Process Control: A Survey.** *AIChE Journal* 32(6) pp.881 913 1986.

187. Sen, A., Sinha, N.K. **A Generalized Pseudoinverse Algorithm for Unbiased Parameter Estimation.** *Int. J. Systems Sci.* 6(12) pp.1103 1109 1975.

188. Sen, S., Yakowitz, S.J. **A Quasi-Newton Differential Dynamic Programming Algorithm for Discrete-time Optimal Control.** *Automatica 23(6)* pp.749 752 1987.

189. Sharp, W.E. **Estimation of Semivariograms by the Maximum Entropy Method.** *Mathematical Geology* 14(5) pp.457 474 1982.

190. Shinnaka, S., Tanaka, K., Suzuki, T. **A Simple Order-Determination Method Based on a Generalized Adaptive Law.** *Int. J. of Control* 41(4) pp.1037 1050 1985.

191. Soderstrom, T. **On the Uniqueness of Maximum Likelihood Identification for Different Structures.** *Report 7307 Lund Institute of Technology* 1973.

192. Soderstrom, T. **On a Method for Model Structure Selection in System Identification.** *Automatica* 17(2) pp.387 388 1981.

193. Soderstrom, T. **Identification of Stochastic Linear Systems in Presence of Input Noise.** *Automatica* 17(5) pp.713 725 1981.

194. Soderstrom, T., Ljung, L., Gustavsson, I. **On the Accuracy Problem in Identification.** *Proc. IFAC 6th World Congress. part 2. (Boston)* paper 18.1 **1975.**

195. Soderstrom, T., Stoica, P., Trulsson, E. **Instrumental Variable Methods for Closed Loop Systems.** *IFAC 10th World Cong. on Auto. Control* 10 pp.364 369 **1987.**

196. Solo, V. **The Convergence of an Instrumental-Variable-like Recursion.** *Automatica* 17(5) pp.545 547 1981.

197. Song, T.L., Speyer, J.L. **The modified gain extended Kalman filter and parameter identification in linear systems** *Automatica* 22(1) pp.59 75 **1986.**

198. Stevens, B., Marshall, R., Lewis, F. **Identification of nonprime systems** *American control conference* pp.1267 1269 **1986.**

199. Stoica, P., Friedlander, B., Soderstrom, T. **The Parsimony Principle for a Class of Model Structures.** *IEEE* AC-30(6) pp.597 600 **1985.**

200. Stoica, P., Soderstrom, T., Friedlander, B. **Optimal Instrumental Variable Estimates of the AR Parameters of an ARMA Process.** *IEEE* AC-30(11) pp.1066 1074 **1985.**

201. Taneja, V.S., Aroian, L.A. **Time Series in M Dimensions: Autoregressive Models.** *Commun. Statist-Simula. Computa.* B9(5) pp.491 513 **1980.**

202. **Thornton, C.L., Bierman, G.J. Filtering and Error Analysis via the UDU Covariance Factorization.** *IEEE Trans.* AC-23(5) pp.901 907 **1978.**

203. **Thornton, C., Bierman, G.J. Gram-Schmidt Algorithms for Covariance Propagation.** *Int. J. Control* 25(2) pp.243 260 **1977.**

204. **Unbehauen, H. Theory and Application of Adaptive Control.** *IFAC 7th Conf. on Digital Computer Applications to Process Control (Vienna)* pp.3 19 **1985.**

205. **Unbehauen, H., Gohring, B. Tests for Determining model order in parameter estimation** *Automatica* 10 pp.233 244 **1974.**

206. **Unton, F.Z. Recursive Estimator of the Solutions of Linear Equation Sequence.** *IEEE Trans.* AC-29(2) pp.177 179 **1984.**

207. Unton, F.Z. **Instrumental Variable Method for Systems with Filtered White Noise Input.** *Automatica* 21(2) pp.199 202 1985.

208. Unton, F.Z. **A New Recursive Identification Technique.** *IFAC Ident. and Sys. Par. Estim. (York)* pp.873 878 1985.

209. Van Den Boom, A.J.W., Van Den Enden, A.W.M. **The Determination of the Orders of Process and Noise Dynamics.** *Automatica* 10 pp. 245 256 1974.

210. Vandewalle, J., Staar, J., De Moor, B., Lauwers, J. **An Adaptive Singular Value Decomposition Algorithm and its Application to Adaptive Realization.** pp. 32 47

211. Van Overbeek, A.J.M., Ljung, L. **On-Line Structure Selection for Multivariable State-Space Models.** *Automatica* 18(5) pp.529 543 1982.

212. Van Zee, G.A., Schinkel, W.M.M., Bosgra, O.H. **Estimation of the Transfer Function, Time Moments, and Model Parameters of a Flow Process.** *AIChE Journal* 33(2) pp.341 346 1987.

213. Venot, A., Pronzato, L., Walter, E., Lebruchec, J.-F. **A distributtion-free criterion for robust identification, with applications in system modelling and image processing** *Automatica* 22(1) pp.105 109 1986.

214. Verhaegen, M., Van Dooren, P. **Numerical Aspects of Different Kalman Filter Implementations.** *IEEE trans.* AC-31(10) pp.907 917 1986.

215. Verhaegen, M.H., Vandewalle, J., Bril, F., Cumps, M. **A Robust and Efficient Least Squares Estimation Algorithm.** *IFAC Ident. and Sys. Par. Estim. (York)* pp.861 866 1985.

216. Vermeer, P.J., Roffel, B., Chin, P.A. **An Industrial Application of an Adaptive Algorithm to Overhead Composition Control.** *ACC* 1987.

217. Voss, D.A., Oprian, C.A. **Moving Average Models-Time Series in N-Dimensions.** *Commun. Statist.-Simula. Computa.* B9(5) pp.467 489 1980.

218. Wahlberg, B., Ljung., L. **Design Variables for Bias Distribution in Transfer Function Estimation.** *IEEE Trans.* AC-31(2) pp.134 144 1986.

219. Wang, X. **Recursive Algorithms for Linear LMSE Estimators Under Uncertain Observations.** *IEEE Trans.* AC-29(9) pp.853 857 1984.

220. Wellstead, P.E. **Signal Aliasing in System Identification.** *Int. J. of Control* 22(3) pp.363 375 1975.

221. Wellstead, P.E. **Model order identification using an auxiliary system** *proc. IEE* 123(12) pp.1373 1379 1976.

222. Wellstead, P. **Non-Parametric Methods of System Identification.** *Automatica* 17(1) pp.55 69 1981.

223. Wellstead, P.E., Nuske, D.J. **Identification of an Automotive Gas Turbine. part 1. Frequency Response Estimation.** *Int. J. of Control* 24(3) pp.297 309 1976.

224. Wellstead, P.E., Caldas Pinto, J.R., Sym, N. **Self-Tuning Filters.** *IEE Colloquium Recent Advances in Identification and Signal Processing: part I digest 1984/19* 1984.

225. Wellstead, P.E., Thiruarooran, C., Winterbone, D.E. **Identification of a Turbo-Charged Diesel Engine.** *IFAC Ident. and Sys. Par. Estim. (Helsinki) pp.361 367* 1978.

226. Wright, J.D., Bacon, D.W. **Statistical Identification of the Dynamic Behavior of a Dual Input Heat Exchanger Network.** *The Canadian Journal of Chemical Engineering* 52(6) pp. 405 414 1974.

227. Wu, S.-C. **A Theorem of Equivalence on the Methods of Least-Squares Estimation.** *IEEE Trans.* AC-30(2) pp.172 175 1985.

228. Ydstie, B.E., Sargent, W.H. **Convergence and Stability Properties of an Adaptive Regulator with Variable Forgetting Factor.** *Automatica* pp. 749 751 1983??.

229. Young, P. **Recursive Approaches to Time Series Analysis.** *The Inst. of Math. and its Appl.* May/June pp.209 224 1974.

230. Young, P. **Some Observation on instrumental variable methods of time-series analysis.** *Int. J. Control* 23(5) pp.593 612 1976.

231. Young, P. **The Instrumental Variable Method: A Practical Approach to Identification and System Parameter Estimation.** *IFAC Ident. and Sys. Par. Estim. (York)* pp.1 15 1985.

232. Young, P.C. **Comments on "On-Line Identification of Linear Dynamic Systems with Applications to Kalman Filtering".** *IEEE Trans.* AC-17 pp.269 271 1972.

233. Young, P., Jakeman, A. **Refined Instrumental Variable Methods of Recursive Time-Series Analysis. Part I. Single Input, Single Output Systems.** *Int. J. of Control* 29(1) pp.1 30 1979.

234. Young, P., Jakeman, A. **Refined Instrumental Variable Methods of Recursive Time-Series Analysis. Part III. Extensions.** *Int. J. of Control* 31(4) pp.741 764 1980.

235. Young, P., Jakeman, A., McMurtrie, R. **An Instrumental Variable Method for Model Order Identification.** *Automatica* 16 pp. 281 294 1980.

236. Yuan, Z.-D., Ljung, L. **Black-Box Identification of Multivariable Transfer Functions-Asymptotic Properties and Optimal Input Design.** *Int. J. of Control* 40(2) pp.233 256 1984.

237. Zaghlool, S.A. **System Identification via Families of Digital Algorithms.** *Int J. Control* 34(4) pp.685 712 1981.

238. Zhang, L., Mossaheb, S., Henry, R.M. **Proof of the Convergence of an Algorithm for Closed-Loop Identification Using Phase Shifts of Square Waves.** *Int J. Control* 41(1) pp.231 243 1985.

239. Zhang, X.D., Takeda, H. **An Order Recursive Generalized Least Squares Algorithm for System Identification.** *IEEE Trans.* AC-30(12) pp. 1224 1229 1985.

240. Zwingelstein, G. **Panorama des méthodes d'identification de processus. 1. Les méthodes non paramétriques.** *Le Nouvel Automatisme* 49(10) pp.65 71 1984.

Appendix F: MATLAB PROGRAMS

F.1 Control with reduced order model using normalization

```
%Rohr's example: control with reduced order model
%fac is scaling factor on the deadzone normally calculated
%    according to rules in Cluett's Ph.D. thesis
clear
tmout=input('# iterations? ');
fac=input('scale factor? ');
%initialization
u(5)=0;
y(5)=0;
th(2)=1.;
phl=.1;
phu=1.;
db=.37596*fac;
itime=0;
%main loop
while 1
itime=itime+1;
%generate system output
y(1)=.820*y(2)-.0327*y(3)+.000431*y(4)+.234*u(2)+.185*u(3)
     +.00479*u(4);
%form regressor
v=[y(2) u(2)];
%calculate normalization factor
vc=[abs(y(2:5)) abs(u(2:5))];
n=max(vc);
n=max([n 1]);
%calculate normalized regressor
vn=v/n;
%calculate normalized error
en=y(1)/n-th*vn';
%calculate phi
vtv=vn*vn';
dbl=2*(1+phl*vtv)*db/(2+phl*vtv);
dbu=2*(1+phu*vtv)*db/(2+phu*vtv);
if abs(en)<=dbl
phi=0;
elseif abs(en)>dbu
phi=phu;
else
phi=2*(abs(en)-db)/((2*db-abs(en))*vtv);
end
%update parameters
th=th+phi*en*vn/(1+phi*vtv);
%calculate control action
ysp=1-(fix(rem(itime,40)/19.9)*2);
u(1)=(ysp-th(1)*y(1))/th(2);
yu(itime,1:2)=[y(1) ysp];
```

```
thu(itime,1:3)=[th(1:2) en];
%shift vectors
y(5)=y(4);
y(4)=y(3);
y(3)=y(2);
y(2)=y(1);
u(5)=u(4);
u(4)=u(3);
u(3)=u(2);
u(2)=u(1);
if itime==tmout;break;end
end
axis([0 tmout -2 3])
plot(yu)
xlabel('solid output, dash setpoint')
pause
axis([0 tmout -.5 1.5])
plot(thu)
xlabel('solid a, dash b')
```

F.2 Control with reduced order model using modified projection parameters

```
%Rohr's example: control with reduced order model but no
%       normalization
%fac is scaling factor on the deadzone normally calculated
%    according to rules in Cluett's Ph.D. thesis
%also allows to modify the projection algorithm parameters
%from their normal values of unity
clear
tmout=input('# iterations? ');
fac=input('scale factor? ');
a=input('a? ');
c=input('c? ');
%initialization
u(5)=0;
y(5)=0;
th(2)=1.;
phl=.1;
phu=1.;
db=.37596*fac;
itime=0;
%main loop
while 1
itime=itime+1;
%generate system output
y(1)=.820*y(2)-.0327*y(3)+.000431*y(4)+.234*u(2)+.185*u(3)
     +.00479*u(4);
%form regressor
v=[y(2) u(2)];
%calculate normalization factor as before but set it to
%unity after these (useless) extra calculations are
%performed only to make the comparison with the
%normalized case easier
vc=[abs(y(2:5)) abs(u(2:5))];
n=max(vc);
n=max([n 1]);
n=1;
%calculate normalized regressor
vn=v/n;
%calculate normalized error
en=y(1)/n-th*vn';
%calculate phi
vtv=vn*vn';
dbl=2*(1+phl*vtv)*db/(2+phl*vtv);
dbu=2*(1+phu*vtv)*db/(2+phu*vtv);
if abs(en)<=dbl
phi=0;
elseif abs(en)>dbu
phi=phu;
else
phi=2*(abs(en)-db)/((2*db-abs(en))*vtv);
```

```
end
%update parameters
th=th+a*phi*en*vn/(c+phi*vtv);
%calculate control action
ysp=1-(fix(rem(itime,40)/19.9)*2);
u(1)=(ysp-th(1)*y(1))/th(2);
yu(itime,1:2)=[y(1) ysp];
thu(itime,1:3)=[th(1:2) en];
%shift vectors
y(5)=y(4);
y(4)=y(3);
y(3)=y(2);
y(2)=y(1);
u(5)=u(4);
u(4)=u(3);
u(3)=u(2);
u(2)=u(1);
if itime==tmout;break;end
end
axis([0 tmout -2 3])
plot(yu)
xlabel('solid output, dash setpoint')
pause
axis([0 tmout -.5 1.5])
plot(thu)
xlabel('solid a, dash b')
```

F.3 Control with reduced order model using a parameter filter

```
%Rohr's example: control with reduced order model using
%       filtered parameters
%fac is scaling factor on the deadzone normally calculated
%    according to rules in Cluett's Ph.D. thesis
%    (normally set to zero)
%afilt is the parameter filter constant
clear
tmout=input('# iterations? ');
fac=input('scale factor? ');
afilt=input('parameter filter constant: ');
%initialization
u(5)=0;
y(5)=0;
th(2)=1.;
thf(2)=1.;
phl=.1;
phu=1.;
db=.37596*fac;
itime=0;
%main loop
while 1
itime=itime+1;
%generate system output
y(1)=.820*y(2)-.0327*y(3)+.000431*y(4)+.234*u(2)+.185*u(3)
     +.00479*u(4);
%form regressor
v=[y(2) u(2)];
%calculate normalization factor as before but set it to
%unity after these (useless) extra calculations are
%performed only to make the comparison with the
%normalized case easier
vc=[abs(y(2:5)) abs(u(2:5))];
n=max(vc);
n=max([n 1]);
n=1;
%calculate normalized regressor
vn=v/n;
%calculate normalized error
en=y(1)/n-th*vn';
%calculate phi
vtv=vn*vn';
dbl=2*(1+phl*vtv)*db/(2+phl*vtv);
dbu=2*(1+phu*vtv)*db/(2+phu*vtv);
if abs(en)<=dbl
phi=0;
elseif abs(en)>dbu
phi=phu;
else
phi=2*(abs(en)-db)/((2*db-abs(en))*vtv);
```

```
end
%update parameters
th=th+phi*en*vn/(1+phi*vtv);
%filter parameters before use in controller
%note that this does not affect the parameter identification
%but only the controller
thf=thf*afilt+(1-afilt)*th;
%calculate control action
ysp=1-(fix(rem(itime,40)/19.9)*2);
u(1)=(ysp-thf(1)*y(1))/thf(2);
yu(itime,1:2)=[y(1) ysp];
thu(itime,1:3)=[th(1:2) en];
thuf(itime,1:3)=[thf(1:2) en];
%shift vectors
y(5)=y(4);
y(4)=y(3);
y(3)=y(2);
y(2)=y(1);
u(5)=u(4);
u(4)=u(3);
u(3)=u(2);
u(2)=u(1);
if itime==tmout;break;end
end
axis([0 tmout -2 3])
plot(yu)
xlabel('solid output, dash setpoint')
pause
axis([0 tmout -.5 1.5])
plot(thu)
xlabel('solid a, dash b')
pause
plot(thuf)
xlabel('filtered: solid a, dash b')
```

F.4 Control with reduced order model using a T filter

```
%Rohr's example: control with reduced order model using
%       filtered data
%fac is scaling factor on the deadzone normally calculated
%    according to rules in Cluett's Ph.D. thesis
%    (normally set to zero)
%afilt is the data filter constant
clear
tmout=input('# iterations? ');
afilt=input('data filter constant: ');
%initialization
u(5)=0;
y(5)=0;
th(2)=1.;
vf(2)=0;
yf=0;
phl=.1;
phu=1.;
db=0;
itime=0;
%main loop
while 1
itime=itime+1;
%generate system output
y(1)=.820*y(2)-.0327*y(3)+.000431*y(4)+.234*u(2)+.185*u(3)
     +.00479*u(4);
%form regressor
v=[y(2) u(2)];
%filter regressor and output
vf=v-afilt*vf;
yf=y(1)-afilt*yf;
%calculate normalized regressor (since normalization is not
%employed this is a simple assignment statement)
vn=vf;
%calculate normalized error
en=yf-th*vn';
%calculate phi
vtv=vn*vn';
dbl=2*(1+phl*vtv)*db/(2+phl*vtv);
dbu=2*(1+phu*vtv)*db/(2+phu*vtv);
if abs(en)<=dbl
phi=0;
elseif abs(en)>dbu
phi=phu;
else
phi=2*(abs(en)-db)/((2*db-abs(en))*vtv);
end
%update parameters
th=th+phi*en*vn/(1+phi*vtv);
%calculate control action
ysp=1-(fix(rem(itime,40)/19.9)*2);
u(1)=(ysp-th(1)*y(1))/th(2);
```

```
yu(itime,1:2)=[y(1) ysp];
thu(itime,1:3)=[th(1:2) en];
%shift vectors
y(5)=y(4);
y(4)=y(3);
y(3)=y(2);
y(2)=y(1);
u(5)=u(4);
u(4)=u(3);
u(3)=u(2);
u(2)=u(1);
if itime==tmout;break;end
end
axis([0 tmout -2 3])
plot(yu)
xlabel('solid output, dash setpoint')
pause
axis([0 tmout -.5 1.5])
plot(thu)
xlabel('solid a, dash b')
```

F.5 Conversion from Laplace to discrete transfer function

```
 function [dnum,dden] = s2z(num,den,ts)

% *****************************************************
%
% s2z    Continuous transfer function to discrete
%        transfer function.
%
% *****************************************************
%
%        [dnum,dden] = s2z (num,den,ts)
%
%        Definitions:
%
%        Discrete                dnum(z)
%        Transfer  :    G(z) = ---------
%        Function                dden(z)
%                       (descending order in z)
%
%        Continuous              num(s)
%        Transfer  :    G(s) = --------
%        Function                den(s)
%                       (descending order in s)
%
%        Sampling Time :     Ts
%
% *****************************************************

 [a,b,c,d]=tf2ss(num,den);
 [alpha,beta]=c2d(a,b,ts);
 [dnum,dden]=ss2tf(alpha,beta,c,d,1);
```

Appendix G: PITSA USER MANUAL

This Appendix contains the user manual of the PITSA (Process Identification and Time Series Analysis) program developed during the course of this thesis. All the simulations and experimental results were obtained using this program except for the simulations of the closed loop system of Chapter 11 which were carried out using MATLAB.

The user manual is self-contained with its own table of contents and bibliography. To be used as a stand alone, Appendix B and C should be appended to the user manual.

Table of Contents

List of Figures

List of Tables

Process Identification and Time Series Analysis (PITSA v.3.5)

User Manual

André Vien

INTRODUCTION

This material describes use of the PITSA program but it does not explain the underlying theory involved in performing a time series analysis or process identification. Users not familiar with the area of process identification should consult the Ph. D. thesis of Vien, 1994 and/or other appropriate references.

Two classes of models are used in this program. The class of models used with the Box-Jenkins algorithms is (Box and Jenkins, 1976):

$$Y_t = B/A\ U_{t-\delta} + C/(D\ \nabla^{dn})\ \epsilon_t + \beta \qquad (1)$$

while for all the other algorithms the class of models is:

$$A\ Y_t = B\ U_{t-\delta} + C\ \epsilon_t + \beta \qquad (2)$$

where:

$A = 1 + A_1 q^{-1} + \ldots + A_n q^{-n}$

$B = B_1 q^{-1} + \ldots + B_m q^{-m}$

δ = delay expressed as a number of sample intervals

β = constant term (bias)

$C = 1 + C_1 q^{-1} + \ldots + C_{mn} q^{-mn}$

$D = 1 + D_1 q^{-1} + \ldots + D_{nn} q^{-nn}$

∇^{dn} = degree of differencing (dn times, $\nabla = (1-q^{-1})$)

ϵ_t = white noise

U_t = input

Y_t = output

$$q^{-1} Y_t = Y_{t-1}$$

if filtering is required the filter model is expressed as:

$$\phi Z_t^F = \theta \nabla^{df} Z_t + \kappa \qquad (3)$$

where:

$\theta = \theta_0 + \theta_1 q^{-1} + \ldots + \theta_{mf} q^{-mf}$

$\phi = 1 + \phi_1 q^{-1} + \ldots + \phi_{nf} q^{-nf}$

∇^{df} = degree of differencing (df times)

κ = constant term (bias)

Z_t = input to the filter (input (U_t) or output (Y_t))

Z_t^F = output of the filter (filtered input or output)

It should be noted that two different filters are available: one for the input data and the other for the output data. If prewhitening is desired, in performing a time series analysis, both filters would be the same and would be based on prewhitening of the input as described in Appendix B (also see below). This filter also allows for scaling (setting θ_0) and mean removal (setting κ to the negative of the mean) since θ_0 and κ are still operative even if the orders nf, df and mf are zero.

As can be seen from the filter model as implemented in the program (cf. Equation 3), if no filtering action is desired then θ_0 should be set to unity and κ to zero.

The short notation representation used in the program for the transfer function is T(N,M,DELAY)CST or T(N,M,DELAY) if the constant term β is omitted. For the noise it is N(NN,DN,MN) and filtering by FU(NFU,DFU,MFU) and FY(NFY,DFY,MFY). For the filters (0,0,0) signifies that some scaling was done without any dynamic filtering.

A model structure must be assumed before applying any parametric identification method, so if the exact model structure is not known a priori, some logical approach to establishing a suitable structure is necessary. For this, it is appropriate to use a nonparametric method. The method adopted and implemented in the program is correlation analysis. Correlation analysis helps the user make an initial determination of suitable model orders (structure) and then a parametric identification methcd can be applied. The analysis of the input-output data by correlation analysis will not necessarily yield the best structure, but should limit the search to a few possible structures.

A suitable model structure is established by examining auto-correlation, partial autocorrelation and crosscorrelation plots. However in order to correctly interpret these plots, the input signal (data) must be white noise. If the input data is not representative of white noise the data must be filtered so, in general, the first step in time series analysis will be determination of a prewhitening filter. Filtering of the input data so that the input signal closely approximates white noise is termed prewhitening. Once the prewhitening filter has been determined, this same filter must also be applied to the output data before there is any attempt to determine a suitable process model. If this is not done, the identified process model will not be representative of the original input-output data. The determination of the prewhitening filter, as explained in Appendix B, is based on correlation analysis information (plots).

It should be noted that the filter form used in the program is the inverse of the form used in Appendix B. This can be readily observed by examining Equations 3 and B.1. The correspondance between the two equations is as follows: ϕ=C; $Z_t^f=\epsilon_t$; θ=D; $\nabla^{df}=\nabla^{dn}$; $Z_t=N_t$ and κ=0, θ_0=1. The reason for this change of model representation is that the filter form has its input on the right hand side and thus is similar in form to the transfer function model.

I. PROGRAM OPERATION

A. Hardware requirements

This program expects a terminal that supports basic ANSI commands. If the terminal does not support ANSI commands, the commands will appear as question marks and letters. Although annoying, it is still possible to use the program.

The program also expects the terminal to have a graphics mode. If the terminal does not support graphics, the program can still be used. In this case the user has to answer no when asked if plots are to be shown. The plots can then be saved and examined at a later time.

B. Reading data

The program can read data from any data file, provided the appropriate user defined read function is supplied. The user read function is a FORTRAN subroutine, named DATIN, that replaces the default DATIN subroutine included with the program. The required format is as follows:

```
      SUBROUTINE DATIN(X,U,Y,NDAT,START,SKIP,END,
     &                 XCOL,UCOL,YCOL)
      INTEGER NDAT,START,SKIP,END,XCOL,UCOL,YCOL
      REAL X(*),U(*),Y(*)
```

where:

X	time (abscissa)
U	input
Y	output
NDAT	number of points read

These variables are passed by the subroutine to the main program (so they cannot be used to control the subroutine).

When using the DATIN subroutine supplied with the program the data file can have a header before the data itself. The default DATIN subroutine can read up to 10 columns, from which time (abscissa), input and output values are read. If time values are not contained in the data file, the subroutine will prompt the user for the initial (start) time and sampling interval. This action will allow the subroutine to establish a time scale for the data.

The variable START denotes the first data line to be used in the data file; SKIP allows the user to not use all the data points by specifying the number of lines to be skipped between each data points and END denotes the last line of data to be used. If END is set larger than the number of points in the file all the data points (from START on) are used. The program is currently limited to 4000 data points.

The default subroutine DATIN causes the number of points and the last point read to be printed on the screen so that the user can verify that the data values were read correctly by the subroutine.

In summary, the following parameters are used to control the default DATIN subroutine for reading data from the data file:

START	first line of data
SKIP	number of lines to skip between points
END	last line of data

XCOL	column of time (abscissa)
UCOL	column of input
YCOL	column of output

C. Starting the program

To activate the program type:

```
$SO VIEN:PITSA
```

This will set the MTS parameters required so that the program can function properly. The commands executed are:

```
R *PROFILE
FECP %PAGE=OFF
FECP %TCC=OFF
FECP %RMAR=255
FECP %LEN=255
FECP %PTOP=ON
END
R VIEN:PITSA SERCOM=-ERROR
CONTINUE WITH *MSOURCE*
```

To rerun the program within the same computer session it is only necessary to type the following:

```
R VIEN:PITSA SERCOM=-ERROR
```

If you have created a user defined read subroutine written in FORTRAN 66 or 77, and have compiled the subroutine into a file, say USER.OBJ, then USER.OBJ must be typed before VIEN:... in the run command, that is R USER.OBJ+VIEN:....

D. Modifying parameters

For each option, the relevant specifications for that option are shown for possible modification. When asked if the user wants to change any of these parameters (the query at the bottom of the screen) the user can reply Y (or y for yes), N (or n for no), Q (or q for quit this part) or ? (for help).

If the user enters Y the cursor will move to the first parameter that the user may set. If a parameter is not to be changed, press the RETURN key to move to the next parameter. Changes can be made simply by entering the new value and pressing the RETURN key. If the value is not acceptable a message at the bottom of the screen will appear and the old value will be displayed again. If the value is acceptable the new value will be displayed and can be further modified. If ? is entered then a short description of the current parameter will be shown at the bottom of the screen. If Q or q is entered then the full screen is updated and the cursor is back at the query at the bottom of the screen. Also, after the last parameter the cursor moves back to the query.

If the user enters N then the program will proceed with the execution of the program step.

If the user enters Q the program will return to the main menu.

If the user enters ? a short description of the relevant parameters is presented on the screen and the parameters are shown again for possible modification.

E. Program plots

The plots that are produced depend upon the particular menu option that has been selected. If confidence limits are applicable, they should be used to determine the significance of the plotted values.

On the autoregressive, crosscorrelation and partial autocorrelation plots the confidence limits shown are the one and two standard deviation bounds as calculated by the Bartlett long lag bounds (Box and Jenkins, 1976). The confidence limits for the cumulative power spectrum are the 1% (outside) and 10% (inside) Kolmogorov-Smirnov bounds (Box and Jenkins, 1976).

In certain options plots of parameters vs iterations are generated. The symbols used in these plots are consistent with the model representation (e.g. A_1 is the first parameter of A). β and κ are represented by CNT (constant). Additional plots are given with diagnostic parameters. For these plots S^2 is the sum of squares and COND is the condition number of the matrix used to update the parameters.

If the user is operating from a graphics terminal and requests to see the plots, the first plot relative to the current menu option will appear with a cursor. This cursor indicates that the program is waiting for plotsee commands. Documentation (no. R169) on the plotsee program is available from Computing Services at the University of Alberta. All the plotsee commands can be used. The most useful plotsee commands are: Next, SKip n (where n is the number of plots to skip), STop and ? (for help). The capital letters are the permissible abbreviations. Help will also give the other commands not described here.

F. Program Menus

The different menus available are the following

PITSA MAIN MENU

1. Generate data
2. Retrieve data from data file
3. Save results or data in a file
4. Minimum, maximum, mean and variance of input and output and plots of input and output
5. Prewhitening order identification
6. Box-Jenkins preliminary prewhitening estimation
7. Box-Jenkins least squares prewhitening estimation
8. Transfer function order identification
9. Non-recursive identification
10. Recursive identification
11. Change one or more of any parameter of any option (or change title(s))
12. Stop

NON-RECURSIVE IDENTIFICATION

1. Box-Jenkins prelim. trans. func. estimation
2. Box-jenkins least squares trans. func. estim.
3. Least squares
4. Weighted least squares
5. Instrumental variables
6. Generalized least squares
7. Maximum likelihood
8. Extended least squares
9. Return to main menu

RECURSIVE IDENTIFICATION

1. Least squares (RLS)
2. Normalized least squares (NRLS)
3. Least squares with scaling (SRLS)
4. Least squares with reconstructed P matrix (RPRLS)
5. Least squares with U-D factorization (RUD)
6. Weighted least squares (RWLS)
7. Generalized least squares (RGLS)
8. Instrumental variables (RIV)
9. Extended least squares (RELS)
10. Maximum likelihood (RML)
11. Box-Jenkins (RBJ)
12. Return to main menu

SAVING RESULTS/DATA

1. Save all the parameters
2. Save input-output data
3. Do (1) and (2) above
4. Save filtered data
5. Do (1) and (4) above
6. Save the current plot file
7. Save the current report file
8. Return to main menu

G. Program Structure

The general organization of the program relates to options of the different menus. The normal sequence of options of the main menu a user would use is presented in Figure 1.

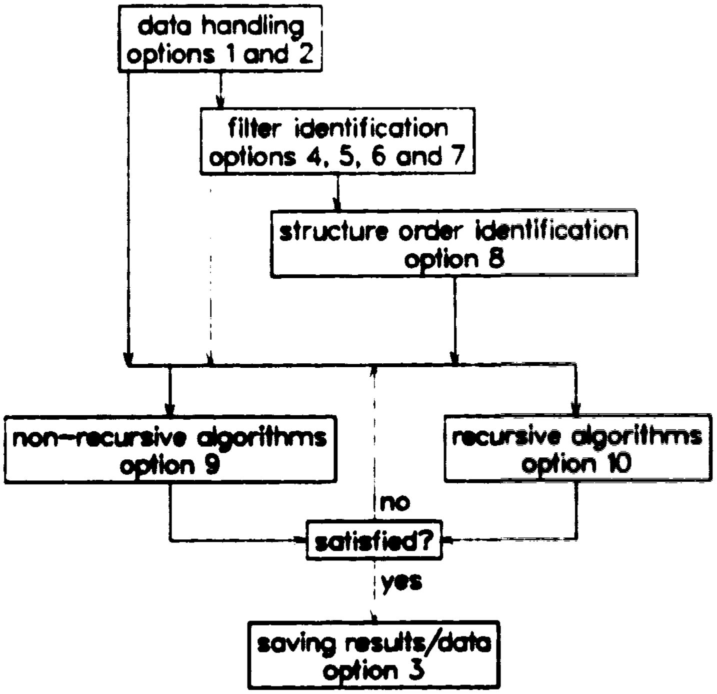


Fig. 1 Flow chart for the normal option sequence for the main menu

The first step involves providing input/output data which can be done by generating data (option 1) or by reading data from a file (option 2). Once the data has been supplied, the user can use the identification algorithms directly if the model structure and input and output filters are known (or have been preselected).

By selecting option 4 the user will be provided with the calculated mean and variance of the inout and output data along with the minimum and maximum values of each data series. This allows the user to design a filter that will normalize, scale and/or remove the mean. Options 5, 6 and 7 are used to design a filter that will produce white noise when applied to the input signal. This is necessary for option 8 and for identification algorithms that require a white noise input such as option 1 of the non-recursive algorithms. Option 5 is used to find the structure of the filter, option 6 to find preliminary estimates of the filter parameters and option 7 to find final estimates fo the filter parameters.

Option 8 provides plots for model structure identification using correlation analysis (see Appendix B for details).

Many algorithms are available either in their non-recursive or recursive version. Results of the identification run can then be used to determine if the identified model is acceptable. If the results are not satisfactory then the structure or the filters may be modified and a new identification, using the same or a different algorithm (recursive or non-recursive), is performed. The new results are examined and the procedure is repeated until a satisfactory identified model is found. Although results of an identification run (actually from any option) may be saved in a file at any time using option 3, it is common practice to save generated or filtered data and only the successful identification runs (usually the last one) in order to reduce disk storage requirements.

II. DESCRIPTION OF PROGRAM OPTIONS

A description of the features and use of each of the options of the different menus is provided in this section.

A. Main menu options

1. *Generate data*

This option allows the user the generate output data for a specified model. The input signal may be generated (the available types of input signals are: PRBS, uniform, sinusoidal, Gaussian, step, reversing (changing sign) square wave and reversing saw tooth wave) or the input signal may be read from a file. The specified model is of the form of Equation 1. Specification of the model of the form of Equation 2 is possible by setting the D polynomial equal to the A polynomial in Equation 1. Once the user has selected this option, the parameters and orders for the model to be used for the data generation will be shown for possible modification along with the titles (including labels for axis). The program will prompt the user for the length of the series to be generated and choice of type of input signal (including from a file).

Depending on the input signal selected, the user will also be prompted for such information as amplitude of input signal, seed number, etc. or the file name and in which column is the input in the case of a file (similar to the default read subroutine). The user will be asked to enter the mean level of the input, the noise model seed number for the random generation, the standard deviation and mean. A sampling time is entered to provide a time scale for the abscissa of the plots.

2. *Retrieve data from data file*

If the user has specified option "2", the program will display the read control parameters for possible modification along with the titles (including labels for axis).

If the program is using the default read subroutine, DATIN, the user will be asked to enter the data file name. The user will be asked if there is a column for the time or if it has to be generated. The user will be prompted to enter the starting value and the sample interval if the time scale is generated.

If the data file only contains a single time series that is to be analyzed, as in time series forecasting (e.g. economics), where only the prewhitening (filtering) section of this program will be used, enter the same values for the output as for the input (i.e. output title same as input title, YCOL same as UCOL). See Appendix A of this manual for a description of the program variables such as YCOL, UCOL, etc.

3. *Saving results/data*

If the user enters option 3 of the main menu, the program will display the following menu

```
1.  Save all the parameters
2.  Save input-output data
3.  Do (1) and (2) above
4.  Save filtered data
5.  Do (1) and (4) above
6.  Save the current plot file
7.  Save the current report file
8.  Return to main menu
```

which shows the user the options that are available for storing data or results.

The program can write input/output data (original or filtered) and/or all user specifiable parameters to a file. The file will contain the title of the run, all the program parameters, if they are desired, followed by the titles for the abscissa (time), input and output and their values in 3 columns (free format). All the plots generated from the last option can be saved (only the plots from the last option used can be saved). The program can also save the report file that contains the numerical results of the last option.

The only user input required (beside the saving data menu option chosen) is the file name. If filtered values are saved the last filter parameters are used. If the file already exists the user will be asked if the file can be replaced.

4. *Minimum, maximum, mean, variance and plots of input and output*

The user can specify filter parameters before the minimum, maximum, mean, variance of the input and output are calculated. Plots of the input vs time, output vs time and output (possibly delayed) vs input are presented to show trends, any unusual aspects of the data and if the relationship between input and output is linear or not. If there is anything unusual or if the relationship in nonlinear proper manipulation of the data (e.g. linearization, removal of outliers) should be performed *before* this program is used. Discussion of these topics can be found in Draper and Smith, 1981. If prewhitening is not required the results obtained from this step can be used to design a filter (e.g. for least squares). Filtering can be used to improve numerical conditioning of the data, thus resulting in an enhanced model identification. A good example of this would be removal of the mean and scaling the input and output data so that their relative magnitude is about the same.

5. *Prewhitening order identification*

Given a time series (i.e. the input series) and the degree of differencing the program will produce the following plots for the differenced series: variable vs time, autocorrelation, partial autocorrelation, power spectrum and cumulative power spectrum.

The table of Appendix B used for model order identification is reproduced in Table 1 but for the filter form so that the user does not have to transpose model orders.

6. *Box-Jenkins preliminary prewhitening estimation*

Given the ARIMA model orders, which are predetermined by the user (e.g. throi.gh the prewhitening order identification), the program will calculate the preliminary estimates of the filter parameters for the prewhitening model. The residual variance is also given.

7. *Box-Jenkins least squares prewhitening estimation*

Given the ARIMA model order and initial estimates of the parameters, which are predetermined by the user (e.g. through the preliminary prewhitening estimation), the program determines the nonlinear least squares estimates of the parameters of the ARIMA model and their standard deviations. The program also provides the parameter correlation matrix, the Akaike's information criterion (Freeman, 1985), the shortest data description criterion (Freeman, 1985), the variance of residuals and CHI-square

TABLE 1

Model characteristics for order selection

	autoregressive	moving average	mixed
model	$Z_t^r = \theta \ Z_t$	$\phi Z_t^r = Z_t$	$\phi Z_t^r = \theta \ Z_t$
autocorrelation	infinite (damped exponentials and/or damped sine waves)	finite	infinite (damped exponentials and/or damped sine waves after nf-mf lags)
	tails off	cuts off after nf lags	tails off after nf-mf lags
partial autocorrelation	finite	infinite (dominated by exponentials and/or sine waves)	infinite (dominated by exponentials and/or sine waves after mf-nf lags)
	cuts off after mf lags	tails off	tails off after mf-nf lags

statistic on the residuals autocorrelations (Box and Jenkins, 1976) for the fitted model.

Because the least squares prewhitening estimation option involves the use of a nonlinear least squares algorithm, special attention must be given to the convergence parameters LAMBDA and NU. The following suggestions should help the user in the choice of LAMBDA and NU. If the initial values are far from the expected values then start with a large LAMBDA (say 5 to 40). If the initial values are close to the expected values then start with a small LAMBDA (say 1). If the condition number is initially near 1.0 consider decreasing LAMBDA and if the condition number decreases quickly tc nearly 1.0 consider decreasing NU. LAMBDA should approach zero near the optimum. If an ill-conditioned or singular matrix is encountered try increasing LAMBDA and/or NU. If the algorithm diverges try changing the initial guesses for the parameters.

The parameters EPSPAR and EPSUM are used as convergence criteria, the former as the relative change in parameters and the latter as the relative change in th sum of squares of residuals. Also the parameter MAXIT is provided to limit the number of iterations in case of nonconvergence of the parameters or of the sum of squares of residuals. The parameter NBAD is also provided to limit the number of iterations if for NBAD successive iterations the sum of squares was not reduced the iterations are stopped as no improvement of the sum of squares can be found.

Plot of the change in parameter vs iterations and variable vs time, autocorrelation, partial autocorrelation, power spectrum and cumulative power spectrum plots of the residuals are produced.

8. *Transfer function order identification*

Given the prewhitening ARIMA model order and parameters, previously determined by the user (e.g. through Box-Jenkins least squares prewhitening estimation), the program can perform a prewhitening transformation on the input and output (if not required set NFU, DFU and MFU to 0, TETAFU=1,0 and similarly for NFY, DFY, MFY and TETAFY) and display the variable vs time, autocorrelation, partial autocorrelation, power spectrum and cumulative power spectrum plots for the prewhitened input and output data. Display of these plots is followed by plots of the crosscorrelation between the prewhitened input and output, the impulse and step responses as well as the variable vs time, autocorrelation, partial autocorrelation, power spectrum and cumulative power spectrum plots for the estimated noise.

9. *Move to the nonrecursive identification menu*

This leads to the submenu which presents all the nonrecursive identification algorithms available. The different items in the menu are described in the next section. Note that the user should know which model orders and type of filtering (if required) including the filter parameters to use before going to this submenu.

10. *Move to the recursive identification menu*

This leads to the submenu which presents the recursive identification algorithms available. The different items in the menu are described in a subsequant section. Note that the user should know which model orders and type of filtering (if required) including the filter parameters to use before going to this submenu.

11. Change the value of any parameter or title

One or more parameter(s) of any option can be changed by selecting this step of the main menu instead of at the time the option is selected. Also other variables not directly attached to a specific option can be changed. This option also allows the user to change the different titles and the length of the axis for plotting purposes. Note that the number of data points can only be changed by reading a new data set. This is a security precaution. Also note that all the variables changed here will take effect only the next time they are needed (e.g. changing the column number will not change the variable used until a new set of data points are read in).

12. Exit the program

Upon exit from the program, the last plot file is not destroyed and is called -PLOT. This file can then be saved in a permanent file for plotting purposes (the plot file can also be saved earlier through the data storage option). Similarly the last report from the identification is in the file -REPORT which can be saved in a permanent file (can also be saved earlier through the data storage option).

B. Items from the nonrecursive identification menu

For all of these items, the filter (prewhitening) orders and parameters must be specified. The results generated are estimates of the parameters and their standard deviation, the parameter correlation matrix, the modified Akaike's information criterion (Freeman, 1985), the shortest data description criterion (Freeman, 1985), the variance of residuals, the CHI-square statistic on the residuals autocorrelations (Box and Jenkins, 1976), the CHI-square statistic for the crosscorrelation between the input and the residuals (Box and Jenkins, 1976). For the residuals, plots of the variable vs time, autocorrelation, partial autocorrelation, power spectrum and cumulative power spectrum are also produced.

1. *Box-Jenkins preliminary transfer function estimation*

Once the transfer function model order and the noise ARIMA model order have been specified (previously determined by the user, e.g. through the transfer function order identification) preliminary estimates of the transfer function and noise model parameters is obtained. The characteristics of this algorithm are such that standard deviations and parameter correlation matrix are not available.

2. *Box-Jenkins least squares transfer function estimation*

If the transfer function model order and its initial parameters, noise model order and its initial parameters and the prewhitening ARIMA model order and parameters are specified (previously determined by the user, e.g. through the Box-Jenkins preliminary transfer function estimation), the program will perform a nonlinear least squares fit of the transfer function and noise model parameters.

Note that because this algorithm is a nonlinear least squares, sometimes it is better to have guesses (say 0) for all parameters rather than estimates from another method. Unfortunately no general rules can be formulated as to whether guesses or previous estimates from another method yields the best results. Because the least squares prewhitening estimation option involves the use of a nonlinear least squares algorithm, special attention must be given to the convergence parameters LAMBDA and NU. The following suggestions should help the user in the choice of LAMBDA and NU. If the initial values are far from the expected values then start with a large LAMBDA (say 5 to 40). If the initial values are close to the expected values then start with a small LAMBDA (say 1). If the condition number is initially near 1.0 consider decreasing LAMBDA and if the condition number decreases quickly to nearly 1.0 consider decreasing NU. LAMBDA should approach zero near the optimum. If an ill-conditioned or singular matrix is encountered try increasing LAMBDA and/or NU. If the algorithm diverges try changing the initial guesses for the parameters.

The parameters EPSPAR and EPSUM are used as convergence criteria, the former as the relative change in parameters the latter as the relative change in the sum of squares of residuals. Also the parameter MAXIT is provided to limit the number of iterations in case of nonconvergence of the parameters or of the sum of squares of residuals. The parameter NBAD is also provided to limit the number of iterations if for NBAD successive iterations the sum of squares was not reduced the iterations are stopped as no improvement of the sum of squares can be found.

3. *Least squares*

A nonrecursive least squares estimation of the reduced model with white noise (i.e. NN=0, DN=0 and MN=0) is performed (Isermann and coworkers, 1974).

To improve numerical conditioning, the mean can be removed through filtering. Another possibility is differencing of the input and output data to obtain an incremental data set.

d least squares

his technique is a modification of the least s that allows a weighting on past values of the series.

The user will be asked to enter the name of the file that contains the desired weights. The user must specify the column in the file which contains the weights as well as the first line containing a weight value.

Weights as read apply from the first to the last data point. If not enough weights can be read in then the number of data points is reduced to the number of weights available (i.e. the most recent data points in time are not used).

If the user presses the RETURN key when asked for a file name then by default an exponential decay weighting is performed. The most recent (last) data point will be given a weight of 1.0 and the other data points will have the weight of the preceding data point reduced by the factor requested.

5. *Instrumental variables*

A nonrecursive instrumental variable estimation of the reduced model with white noise (i.e. NN=0, DN=0 and MN=0) is performed (Isermann and coworkers, 1974).

The model predictions of the output are used as instruments. An iterative procedure is required to update the predictions with the latest parameter values obtained. The least squares estimation is used to provide initial guesses. The convergence criterion is the relative change in parameter values between two iterations (variable EPSPAR). To prevent an infinite loop a maximum number of iterations is provided (variable MAXIT). The relaxation factor, LAMBDA, is available tc promote convergence by slowing down (if LAMBDA is less than one) the auxiliary model parameters. The lower LAMBDA is the slower the adaptation will be.

Note that the standard deviations given are actually "pseudo standard deviations" since the standard deviations cannot be obtained from this algorithm (Vien, 1994).

6. *Generalized least squares*

A nonrecursive generalized least squares estimation of the reduced model with coloured noise is performed (the coloured noise is not part of the model as such) (Isermann and coworkers, 1974).

In this method the first step is to perform a least squares estimation of the transfer function parameters. Then the residuals (noise) are modelled with an autoregressive filter of order NN. The noise model thus obtained is used to filter the input and output before a new least squares estimation of the transfer function is performed. The iterative solution is continued until the noise is white. The relative change in the transfer function parameters, EPSPAR, is the stopping criterion. To prevent an infinite loop a maximum number of iterations is provided (variable MAXIT).

7. *Maximum likelihood estimation*

A nonlinear least squares algorithm is used to obtain the estimates of the parameters of the A, B and C polynomials and of β. This is similar to the Box-Jenkins algorithm except for the model used (Eykhoff, 1981, Vien, 1994).

8. *Extended least squares*

An extended least squares estimation of the reduced model with only the numerator noise polynomial is performed (i.e. NN=0, DN=0 and MN is defined by the user).

In this method the data vector for the least squares is augmented with the estimated noise. It is an iterative procedure which uses the relative change in the transfer function plus noise parameters as a stopping criterion (variable EPSPAR). To prevent an infinite loop a maximum number of iterations is provided (variable MAXIT). Also to help convergence a relaxation factor (variable LAMBDA) is available. Initial guesses are provided by the least squares estimation.

C. Items from the recursive identification menu

For all of these items, the filter (prewhitening) orders and parameters must be specified along with initial values of the parameters and of the covariance matrix (called P). The results generated are the estimates of the parameters and their standard deviation, the parameter correlation matrix, the modified Akaike's information criterion (Freeman, 1985), the shortest data description criterion (Freeman, 1985), the variance of residuals, the CHI-square statistic on the residuals autocorrelations (Box and Jenkins, 1976), the CHI-square statistic for the crosscorrelation between the input and the residuals (Box and Jenkins, 1976). For the residuals, plots of the variable vs time, autocorrelation, partial autocorrelation, power spectrum and cumulative power spectrum are also produced. These are given for the last estimates only for comparison with their nonrecursive counterparts. Also shown are the parameter values plotted against the recursion number (same as data point number).

1. *Least squares*

This is the standard recursive least squares without any numerical enhancement.

2. *Normalized least squares*

In this algorithm the data is first normalized using the l_2 norm of the regressor as a normalization factor. A lower bound of unity is implemented to avoid division by a small number. A description of the normalization factor may be found in Sripada and Fisher, 1987.

3. *Least squares with scaling*

It is claimed that this algorithm provides better numerical properties than the least squares due to scaling (Sripada and Fisher, 1987). It is not so, and this algorithm is included for comparison purposes only. The following algorithm will provide exactly the same estimates with fewer computations.

4. *Least squares with reconstructed P matrix*

In this algorithm, the updated P matrix is decomposed in its Cholesky (square root) factor and then reconstructed. This will ensure that the P matrix is positive definite if P is nonsingular. The new parameter estimates are calculated using this reconstructed P matrix.

5. *Least squares with U-D factorization*

This is the standard U-D factorization of Bierman, 1977.

6. *Weighted least squares*

Currently two types of forgetting factors are available: a constant forgetting factor and and a forgetting factor obeying the following law

$$\lambda_t = \lambda_{t-1}\lambda_0 + (1-\lambda_0)$$

where λ_0 is a second forgetting factor. This scheme will allow to discount initial data rapidly when poor initial estimates are used but will retain all data points after the initial period. Typical values for λ_0 and λ_1 are 0.95 or 0.99. The user will be asked which forgetting factor to use and what are the initial values to use.

7. *Generalized least squares*

For the generalized least squares the parameters and the covariance matrix for the residual model must also be given. If unknown, the parameters can be set to zero and the covariance matrix to a multiple of the identity matrix as would be done for an ordinary least squares.

8. *Instrumental variables*

The instrumental variable method is started with an ordinary least squares to provide initial parameters with which to calculate the instruments (auxiliary output). The number of times the recursive least squares is performed before the recursive instrumental variable method is engaged is controlled by MAXIT, the maximum number of time the recursive least squares can be executed, and EPSPAR, the relative change in parameter estimates. In the later case it is assumed that if the parameter change is not too great then the recursive instrumental variable will perform well as the instruments will be relatively "stable". To avoid correlation between the residuals and the instruments, a relaxation factor, LAMBDA, is available. If LAMBDA is unity then there is no relaxation and if LAMBDA is zero then the auxiliary parameters are not updated. Any value of LAMBDA between zero and unity will give an intermediate result. This relaxation can also be seen as a low pass filter on the auxiliary parameters.

9. *Extended least squares*

Here the noise initial parameters have to be provided and the covariance matrix includes the noise parameters as well as the transfer function parameters. For this algorithm NN is set to zero (only noise numerator dynamics allowed).

10. *Maximum likelihood*

Initial guesses also comprise noise parameters (numerator only) and the covariance matrix does include the noise parameters.

11. *Box-Jenkins*

This is simliar to the recursive maximum likelihood with the model of Box-jenkins (Box and Jenkins, 1976).

III. Examples

Many examples of time series analysis (and process identification) are available in the literature. An example of particular interest is data set J of Box and Jenkins. This example is fully detailed in Box and Jenkins, 1976 and by others (MacGregor, 1984, Young, 1984). The data set from Box and Jenkins is available under the name VIEN:DAT.SERIES.J.

A. Sample terminal session

This sample run of the PITSA program generates 300 data points for a least squares model with a_1=-0.7788, b_1=0.349 and b_2=0.308. The noise generated has a variance of 0.01. The PRBS input utilized a maximum length sequence of 6 and was updated every sampling period..

Note that the user input is in **bold** characters and that comments are *italicized*.

```
#so pitsa
#   $.00,  $16.41T
#R *PROFILE
#11:38:21
#11:38:21  T=0.009  RC=0
#   $.01,  $16.42T
#EM -ERROR
 File "-ERROR" has been emptied.
#   $.01,  $16.43T
#R VIEN:OBJ.PITSA.2 SERCOM=-ERROR
#11:38:21
new page generated here


 ******* ******************************************
 *  Process Identifcation and Time Series Analysis  *
 *            (P.I.T.S.A. version 3.5)              *
 *                                                  *
 *  written by: Andre Vien                          *
 *              Chemical engineering dept.          *
 *              University of Alberta               *
 *              Edmonton, Alberta                   *
*              Canada                            *
*                                                *
*  Any comments or questions about this program  *
*  should be forwarded to Andre Vien (MTS id VIEN) *
*  Constructive criticism would be appreciated.  *
*                                                *
*  reminder: the (Y/N/Q/?) prompt means answer   *
*  y for yes, n for no, q for quit and return to *
*  main menu and ? for help (short description of *
*  the parameters for this step)                 *
**************************************************

 Press RETURN to continue
```

(RETURN key pressed)

new page generated here

```
****************************************************
*  WARNING: THIS PROGRAM WORKS ON THE GIGO         *
*           PRINCIPLE. SPECIAL CARE WAS TAKEN TO   *
*           ENSURE THAT IF WRONG OR UNSUITABLE     *
*           PARAMETERS ARE GIVEN THEN UNPREDIC-    *
*           TABLE ANSWERS WILL BE PRODUCED.        *
*                                                  *
*  THIS VERSION WORKS ON ANSI TYPE TERMINALS       *
*  SET %TCC=OFF, %RMAR=255, %PAGE=OFF              *
 *      %LEN=255, %PTOP=ON                         *
*  ON THE RUN COMMAND ADD: SERCOM=-ERR             *
*  YOU CAN ALSO USE: SO VIEN:PITSA                 *
*  THIS WILL DO THE ABOVE AND START THE PROGRAM    *
*                                                  *
*  documentation is available under VIEN:TXTF.PITSA *
*  send this file to the printer to have a copy.   *
****************************************************

 Press RETURN to continue
```

(RETURN key pressed)

new page generated here

```
 *** PITSA program main menu ***


 1  Generate data
 2  Retrieve data from data file
 3  Put data into data file
 4  Minimum, maximum, mean and variance of input and output,
    plot of output vs input
 5  Prewhitening order identification
 6  Box-Jenkins preliminary prewhitening estimation
 7  Box-Jenkins least squares prewhitening estimation
 8  Transfer function order identification
 9  Non-recursive identification
 10 Recursive identification
 11 Change one or more of any parameter of any
```

```
    option (or change title(s))
 12 Stop

 Enter a number from the above:
 1
```

new page generated here

```
 Generate data
```

The changes presented here are actually performed on a screen template that could not be reproduced here.

```
 N      =     0
 NN     =     0
 CST    =     1

 M      =     1
 DN     =     0

 DELAY  =     0
 MN     =     0

 THETA  =  0.0 ,  0.0 ,  0.0 ,  0.0 ,  0.0 ,
           0.0 ,  0.0 ,  0.0 ,  0.0 ,  0.0
 THETAN=   0.0 ,  0.0 ,  0.0 ,  0.0 ,  0.0 ,
           0.0 ,  0.0 ,  0.0 ,  0.0 ,  0.0

 PTITLE=
 XTITLE= Time
 UTITL = Input
 YTITL = Output
 Do you want to change any of these (Y/N/Q/?)
 Y
```

By answering yes the cursor will move in sequence to each value so that it can be modified. If a value does not need modification the RETURN key is pressed. Only the modified values are presented below.

```
 N      =   1
 M      =   2
 NN     =   1
```

Even though this example generates a least squares model a noise denominator term equal to the A polynomial is required because of the model form assumed by the program (the Box-Jenkins model form).

```
 CST    =   0
```

CST=0 specifies that the model should not include a constant term (β).

The parameters for THETA are

```
   -.7788
 .349
 .308
```

followed by RETURN for all the other parameters of THETA.

The parameters for THETAN are

```
 -.7788
```

followed by RETURN for all the other parameters of THETAN.

At the bottom of the template the title is entered and the axe labels are not modified (i.e. RETURN key pressed).

```
 Simulation of the first order system
```

The program permits further modification.

```
Do you want to change any of these (Y/N/Q/?)
 n
```

new page generated here

```
 enter length of series (int):
 300

 Input signal type

 1 PRBS
 2 Uniform
 3 Sinusoidal
 4 Gaussian
 5 Step
 6 Reversing square wave
 7 Saw tooth
 8 From file

 Enter input signal type code number:
 1
 Enter N (length=2**N-1),amplitude,bit length(int,real,int):
 6,1.,1
 Enter input mean (real):
 0.
 Enter seed, sigma, mean (int,real,real) for the noise.
 (if seed <= 0 then no noise):
 345,.1,0.
 sig^al/noise R.M.S. ratio = 0.6114E+01
 enter sampling time (positive real):
 1.

 Press RETURN to continue
```

(RETURN key p'essed)
new page generated here

```
 *** PITSA program main menu ***


 1  Generate data
 2  Retrieve data from data file
 3  Put data into data file
 4  Minimum, maximum, mean and variance of input and output,
    plot of output vs input
 5  Prewhitening order identification
 6  Box-Jenkins preliminary prewhitening estimation
 7  Box-Jenkins least squares prewhitening estimation
 8  Transfer function order identification
 9  Non-recursive identification
 10 Recursive identification
 11 Change one or more of any parameter of any
    option (or change title(s))
 12 Stop

 Enter a number from the above:
 9
```

new page generated here

```
 *** Non-Recursive Identification ***


 1  Box-Jenkins prelim. trans. func. estimation
 2  Box-jenkins least squares trans. func. estim.
 3  Least squares
 4  Weighted least squares
 5  Instrumental variables
 6  Generalized least squares
 7  Maximum likelihood
 8  Extended least squares
 9  Return to main menu

 Enter a number from the above:
 3
```

new page generated here
The template for the least squares is shown here, but since there are no modifications required for this example the question is answered negatively.

```
Do you want to change any of these (Y/N/Q/?)
 n

 Press RETURN to continue
```

(RETURN key pressed)
new page generated here
The results of the identification run are now produced

```
 Estimates for      Simulation of. the first order system
 using LS
 The transfer function model is of order ( 1, 2, 0)
 Transfer function denominator parameters :

  -0.78191E+00+/- 0.63275E-02
 Transfer function numerator parameters :

   0.35470E+00+/- 0.58230E-02   0.31351E+00+/- 0.61660E-02

 Press RETURN to continue
```

(RETURN key pressed)
new page generated here

```
 Parameter correlation matrix

 A(1)   0.10000E+01
```

```
  B(1)  -0.39331E-01   0.10000E+01
  B(2)   0.33097E+00  -0.66906E-02   0.10000E+01

  Press RETURN to continue
 (RETURN key pressed)
 new page generated here

Modified Akaike's information criterion(alpha=4)=-0.137E+04
Shortest data description criterion = -0.13561E+04
Variance of residuals =  0.10088E-01
CHI-Squared statistic =  0.27680E+02
For 49 degrees of freedom, approximate Chi-Square values are:

  P(Chi-Square <  0.74933E+02)  =  0.99
  P(Chi-Square <  0.66336E+02)  =  0.95
  P(Chi-Square <  0.62035E+02)  =  0.90
  P(Chi-Square <  0.57076E+02)  =  0.80

  Press RETURN to continue
 (RETURN key pressed)
 new page generated here

CHI-Squared statistic for cross-correlation
between filtered input and residuals =  0.64094E+02
between filtered output and residuals =  0.53012E+02
For 46 degrees of freedom, approximate Chi-Square values are:

  P(Chi-Square <  0.71215E+02)  =  0.99
  P(Chi-Square <  0.62827E+02)  =  0.95
  P(Chi-Square <  0.58637E+02)  =  0.90
  P(Chi-Square <  0.53814E+02)  =  0.80
  Do you want to see the plots (Y/N)?
  n

  Press RETURN to continue
 (RETURN key pressed)
 new page generated here

  *** PITSA program main menu ***


  1  Generate data
  2  Retrieve data from data file
  3  Put data into data file
  4  Minimum, maximum, mean and variance of input and output,
     plot of output vs input
  5  Prewhitening order identification
  6  Box-Jenkins preliminary prewhitening estimation
  7  Box-Jenkins least squares prewhitening estimation
  8  Transfer function order identification
  9  Non-recursive identification
  10 Recursive identification
  11 Change one or more of any parameter of any
     option (or change title(s))
 12 Stop

 Enter a number from the above:
 12
screen is cleared
#11:42:28  T=4.477  RC=0
#  $2.60,  $19.03T
#
```

Appendix A: PROGRAM VARIABLES

A summary of the variable names used in the program, along with a short description of their function, are presented in this Appendix. These descriptions are also available "online" by typing ? whenever the prompt (Y/N/Q/?) appears or when changing the value of a parameter. The Y/N/Q/? prompt appears when the program allows the user to change one or more parameters. Only the parameters relevant to the particular menu option are displayed and can be changed.

Variables

CST	flag for presence of constant parameter β (0=no,1=yes)
DELAY	delay expressed as integer multiple of sampling time (δ)
DFU	input filter differencing order (∇^{dfu})
DFY	output filter differencing order (∇^{dfy})
DN	noise differencing order (∇^{dn})
END	last line of data to be read from data file
EPSPAR	relative convergence criterion on parameters
EPSUM	relative convergence criterion on sum of squares
LAMBDA	relative step size change in parameters for next iteration (relaxation factor)
M	transfer function numerator order (B)
MAXIT	maximum number of iterations
MFU	input filter numerator order (θ_u)
MFY	output filter numerator order (θ_y)
MN	noise numerator order (C)

N transfer function denominator order (A)

NAUTO number of points in the autocorrelation

NBAD max. number of consecutive unsuccessful iterations

NCROSS number of points in the crosscorrelation

NFU input filter denominator order (ϕ_u)

NFY output filter denominator order (ϕ_y)

NIMPUL number of points in the impulse response

NN noise denominator order (D)

NPAUTO number of points in the partial autocorrelation

NU multyplier for change in LAMBDA after each iteration

PTITLE run title

SKIP number of lines to skip between each data points

START first line of data to be read from data file

TETAFU input filtering parameters ($\phi_1^u,\ldots,\phi_{NFU}^u,\theta_0^u,\theta_1^u,\ldots,\theta_{MFU}^u,\kappa^u$)

TETAFY output filtering parameters ($\phi_1^y,\ldots,\phi_{NFY}^y,\theta_0^y,\theta_1^y,\ldots,\theta_{MFY}^y,\kappa^y$)

THETA transfer function parameters ($A_1,\ldots,A_N,B_1,\ldots,B_M,\beta$)

THETAN noise parameters ($D_1,\ldots,D_{NN},C_1,\ldots,C_{MN}$)

UCOL column number for U (input)

UTITL input title

XCOL column number for time (abscissa)

XLEN length of X axis on plot (centimetres)

XTITLE abscissa title

YCOL column number for Y (output)

YLEN length of Y axis on plot (centimetres)

YTITL output title

Note that TETAFU, TETAFY, THETA and THETAN are currently limited to 10 parameters. In the filter parameters (TETAFU, TETAFY), θ_0

and κ are always present (even if all the orders are zero). In the transfer function (THETA), β is present only if CST is 1 and there should be always at least one B parameter. The noise does not have any "permanent" parameter. Beyond these restrictions the number of parameters is optional (e.g. for the input filter any combination of up to eight ϕ and/or θ parameters is acceptable).

NAUTO, NPAUTO, NCROSS and NIMPUL are currently limited to one hundred points (including lag zero). All titles are currently limited to thirty five characters.

All variables are integer variables except for EPSUM, EPSPAR, LAMBDA, NU, TETAFU, TETAFY, THETA, THETAN, XLEN and YLEN which are real and PTITLE, UTITL, XTITL and YTITL which are of type character.

Bibliography

Bierman, G.J. (1977). *Factorization Methods for Discrete Sequential Estimation*. Academic Press.

Box, G.E.P. and G.M. Jenkins (1976). *Time Series Analysis: Forecasting and Control*, revised edition. Holden-Day.

Draper N.R. and H. Smith (1981). *Applied Regression Analysis, 2nd ed*. Wiley.

Eykhoff, P. (Ed.) (1981). *Trends and Progress in System Identification*. Pergamon Press.

Freeman, T.G. (1985). Selecting the Best Linear Transfer Function Model. *Automatica*, 21, 361-370.

Isermann, R., U. Baur, W. Bamberger, P. Kneppo and H. Siebert (1974). Comparison of Six On-Line Identification and Parameter Estimation Methods. *Automatica*, 10, 81-103.

MacGregor, M.H. (1984). *User's Manual*. Internal report, University of Alberta.

Nelson, C.R. (1973). *Applied Time Series Analysis for Managerial Forecasting*. Holden-Day.

Sripada, N.R. and Fisher, D.G. (1987). Improved Least Squares Identification. *Int. J. Control.*, 46, 1899-1913.

Vien, A. (1994). *Practical Numerical Issues in Industrial Identification*. Ph.D. Thesis, University of Alberta.

Young, P.C. (1984). *Recursive Estimation and Time Series Analysis - an Introduction*. Springer Verlag.