INSTITUTE
OF
HYDROLOGY

COMPUTER AIDED PROCEDURE FOR TIME-SERIES ANALYSIS AND IDENTIFICATION OF NOISY PROCESSES (CAPTAIN)

by

M W VENN and B DAY

USER MANUAL

REPORT NO 39

July 1977

PREFACE

The original CAPTAIN package consisted of a suite of ALGOL programs developed by Professor Peter Young* and Dr Stuart Shellswell at the Department of Control Engineering, University of Cambridge (Young, Shellswell and Neethling, 1971; Shellswell, 1972) on the basis of core programs Professor Young had originated at the Naval Weapons Centre, California and at Cambridge during the period 1964-70. The original package was improved by Dr Paul Whitehead and Professor Young at Cambridge in the period 1973-74 and was acquired by the Water Resources Board in 1974. A contract between this department and the Water Resources Board (latterly the Water Research Centre) resulted in a more flexible suite of FORTRAN programs for hydrological and water resource applications (Moore and Whitehead, 1975, WRC Int. Rept.) This suite has been rewritten and extended to form the integrated, interactive package described in this report.

We gratefully acknowledge the contributions of these organisations and their staff to this version of the package.

*presently Head, Applied Systems, Centre for Resource and Environmental Studies, AWU, Australia.

CONTENTS

FIGURES

1. INTRODUCTION

This report is intended as a manual for use with the CAPTAIN package (IH version). The package is concerned with the modelling of time series data, and consists of a comprehensive set of routines to assist the time series analyst at all stages of model building.

The package has facilities for model identification, parameter estimation, and model verification. The method of model identification follows the techniques given by Box and Jenkins (1970). These make extensive use of the autocorrelation, partial autocorrelation, and impulse response functions. Parameter estimation is performed by the instrumental variable (IV) and approximate maximum likelihood (AML) modifications to a recursive least squares algorithm, as described by Young, Shellswell and Neethling (1971), and Moore (1977). Models can be verified by correlation techniques, and by forecasting.

In any package for the analysis of time series, facilities should be available that allow the user to display results graphically. The CAPTAIN package features extensive graphics facilites, and these have been designed for both the Calcomp 936 plotter, and the Tektronix 4010 VDU. The Calcomp 936 plotter is a high resolution digital offline plotter, capable of providing good quality graphs for inclusion in papers, reports, etc, for which purpose the graphs are drawn A4 size. The Tektronix 4010 VDU on the other hand, operates as an online terminal, and although the quality of the graphs produced is not very high, graphs are available for immediate inspection. The unit is used interactively, allowing a model to be built in one session at a terminal.

The main aim in the design of the package has been ease of use for the non-programming scientist. This has been attempted by:

(i) directing program flow through the use of 'English-like' instructions.

(ii) allowing the system to run interactively.

(iii) performing all data handling internally.

The control language has been designed to follow English as far as possible. Analyses are initiated by instructions entered in a completely free format. It is considered that instructions resembling English are easier to learn and remember, thus reducing time spent in recalling a command and helping to avoid mistakes in presenting it. Punctuation in the form of commas and full stops may be used, subject to certain restrictions.

Although the package may be run in batch mode from a card deck, more facilities are available to the user if it is run interactively. One of the main advantages with the latter mode of operation is, of course, that results from one analysis can be examined before another analysis is initiated. The package may be used interactively either from an ordinary online terminal, or from a Tektronix VDU, in which case graphical output can be directed to the unit.

The task of model building is made much simpler for the analyst if data management is taken over by the computer program in use. The CAPTAIN package performs all data handling internally, using the 'series' as the basic organisational unit of data. When a series is initially presented to the system, it is given a name by the user. Thereafter, the series may be referenced as often as required, merely by giving its name. Series that have been presented to the system are kept on a disc file, and only brought into main memory when needed. This means that there are no restrictions on the possible length of series (in practical terms), and that since main memory is not being used to hold unnecessary data, the program will run faster. Included in the package is a transformation section, which allows a wide range of editing and arithmetic operations to be performed on the data.

1.1 The Manual

This manual describes the use and operation of the package, with statistical theory kept to a minimum. The theory of the various methods is treated in depth in the companion report by Moore (1977), which also gives examples of applications of the package.

Although this manual is a large document, it is not necessary for the user to read it all before presenting a job to the system. Chapters 2 to 7 and Appendix 3 should be read, but thereafter, only those chapters that are relevant to the job in hand need be referred to. The chapters have been made as independent as possible, even though this has often involved a certain amount of repetition. However, the user should follow up the cross references that are given.

1.2 Acknowledgements

The authors would like to express their thanks to R. J. Moore for his assistance and advice at all stages of the project.

2. STATEMENT OF THE PROBLEM

Time series analysis has been the subject of extensive research in the past few years. Many new methods and techniques have been developed for the solution of a wide range of problems.

One approach to the modelling of time series data is to attempt to analyse the internal mechanics that govern the operation of a stochastic system. This may be termed the mechanistic approach. An alternative approach, and the one adopted here, is to represent the overall characteristics of a system with a simple input-output or 'black-box' model. In this case the internal descriptions are avoided and the overall input-output relationship is inferred directly from the data. While this may not give as much information about the system as the mechanistic approach, there is a certain simplicity about it, and estimates of model parameters are much more readily obtained.

The types of model that can be examined by the package are stochastic models and transfer functions with superimposed noise, both of which are discussed in depth by Box and Jenkins (1970). The two types are displayed schematically in Figures 2.1 and 2.2 respectively. In the stochastic model case, a sequence of observations, z_t, is assumed to be generated by an autoregressive-moving average (ARMA) model of the form

$$\phi(B) z_t = \theta(B) a_t \qquad (2.1)$$

where $\phi(B) = 1 + \phi_1 B + \phi_2 B^2 + \phi_3 B^3 + \ldots + \phi_p B^p$

and $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \theta_3 B^3 + \ldots + \theta_q B^q$

and B is the backward shift operator,

ie: $B^b z_t = z_{t-b}$

a_t is an independent and identically distributed sequence of random variables (ie: white noise), with zero mean and variance σ^2.

It will be seen from Figure 2.2 that the general transfer function model consists of two parts, a process model, and a noise model. The observed output y_t is taken to result from an observed input u_{t-b} which causes most of the output variation, and a stochastic

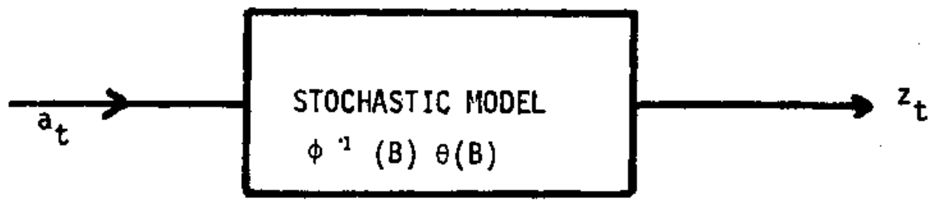

FIGURE 2.1 Representation of a stochastic model

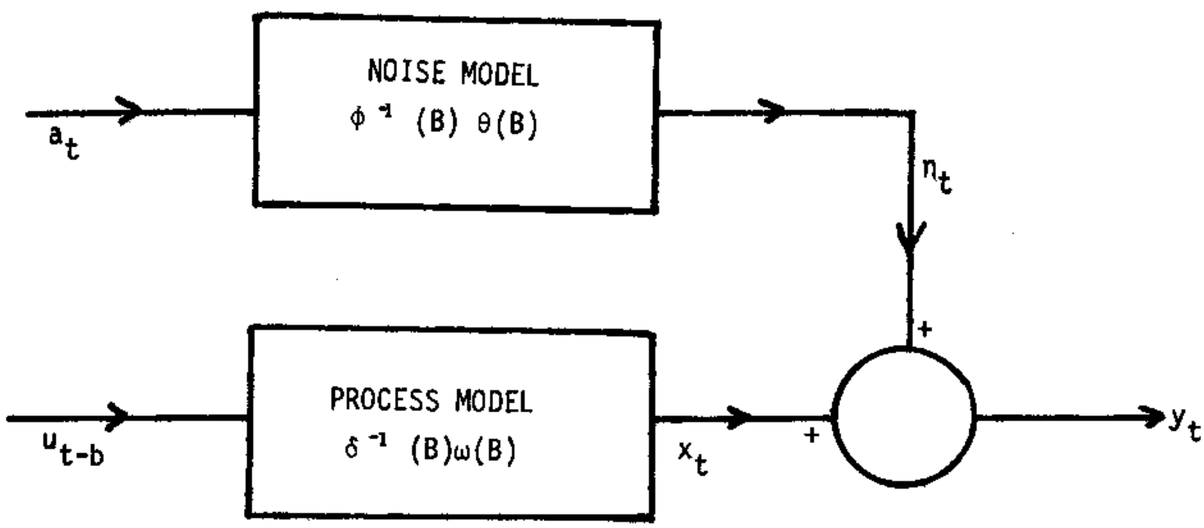

FIGURE 2.2 Representation of a general transfer function with superimposed noise model

input a_t, that accounts for random disturbances such as measurement noise. The output y_t is then given by the sum of the deterministic and stochastic components x_t and n_t respectively, ie:

$$y_t = x_t + n_t \tag{2.2}$$

x_t is obtained from the ARMA model

$$\delta(B)x_t = \omega(B)u_{t-b} \tag{2.3}$$

where $\delta(B) = 1 + \delta_1 B + \delta_2 B^2 + \ldots + \delta_r B^r$

and $\omega(B) = \omega_o + \omega_1 B + \omega_2 B^2 + \ldots + \omega_s B^s$

and b denotes the appropriate time delay between the observed input and the corresponding system output. Thus, b = 1 gives a time delay of one sample instant between the input and the systems response to it, via its output, y.

The noise component at time t, n_t, is generated by the ARMA model

$$\phi(B)n_t = \theta(B)a_t \qquad (2.4)$$

where the polynomials $\phi(B)$ and $\theta(B)$ are as defined above; a_t is again a white noise process. In practice, the noise component is not known, and has to be estimated from the data.

Substituting equations (2.3) and (2.4) into (2.2) gives the full transfer function as

$$y_t = \delta^{-1}(B)\ \omega(B)u_{t-b} + \phi^{-1}(B)\theta(B)a_t$$

2.1 General guide to model building

The problem of model building can be broken down into four main stages, ie:

(i) data validation and removal of non-stationarity

(ii) identification of model structure

(iii) estimation of model parameters

(iv) model verification

This approach follows that adopted by Box and Jenkins (1970).

2.1.1 Data validation and removal of non-stationarity

Before any analysis is attempted on the data, it should be examined carefully for trends, discontinuities, or spurious data points. This is most easily done if the series are displayed graphically. Non-stationarity may also be detected by examining the correlation functions of the sample. Any trends or seasonal variations observed in the data should be removed. This may be done through differencing, but these techniques often tend to amplify the noise component, and alternative means should be sought. However, in accordance with the Box and Jenkins method, differencing facilities are available in the package.

2.1.2 Identification of model structure

The numbers of autoregressive and moving average parameters of a model, together with the relevant time delay (if appropriate), may be termed the structure of the model.

The structure of a univariate model (ie a stochastic, or noise model) is identified by examining the correlation functions of the series. The number of parameters required can be inferred from the patterns observed in the functions.

The structure of a transfer function is identified from the impulse response function (or unit hydrograph) between the input series and the output series. Unfortunately, the impulse response function cannot be calculated directly from the data with sufficient accuracy, and the method of prewhitening must be used. (See Chapter 12, Moore (1977), section 2.2.1, and Box and Jenkins, section 11.2.1). In this method, the input series is taken to be the output of a stochastic model. An appropriate structure is identified, and the parameters estimated. If the input series is then passed through the inverse of the estimated model, it is said to have been "prewhitened". The cross-correlation function between the prewhitened input and the output series transformed in the same way, approximates to the impulse response function.

The interpretation of the different patterns and shapes of the various functions requires a certain expertise which can really only be gained from practical experience. However, the guidelines given by Box and Jenkins provide a good starting point, and these are summarised in Moore's Report (Chapter 2).

2.1.3 Parameter estimation

Once a possible structure has been identified, the model parameters may be estimated.

The process model parameters of a transfer function are estimated by an iterative Instrumental Variable (IV) technique (Young, Shellswell and Neethling, 1971; and Moore 1977 section 3.2), and is shown diagramatically in Figure 2.3. The method makes use of an auxiliary model, which is subjected to the same input sequence as the process to be identified. The auxiliary model output, $\hat{x}_t$, is then an estimate of the hypothetical noise-free output, x_t. The difference between $\hat{x}_t$ and the observed process output y_t, gives an estimate $\hat{n}_t$ of the noise component, ie:

$$\hat{n}_t = y_t - \hat{x}_t$$

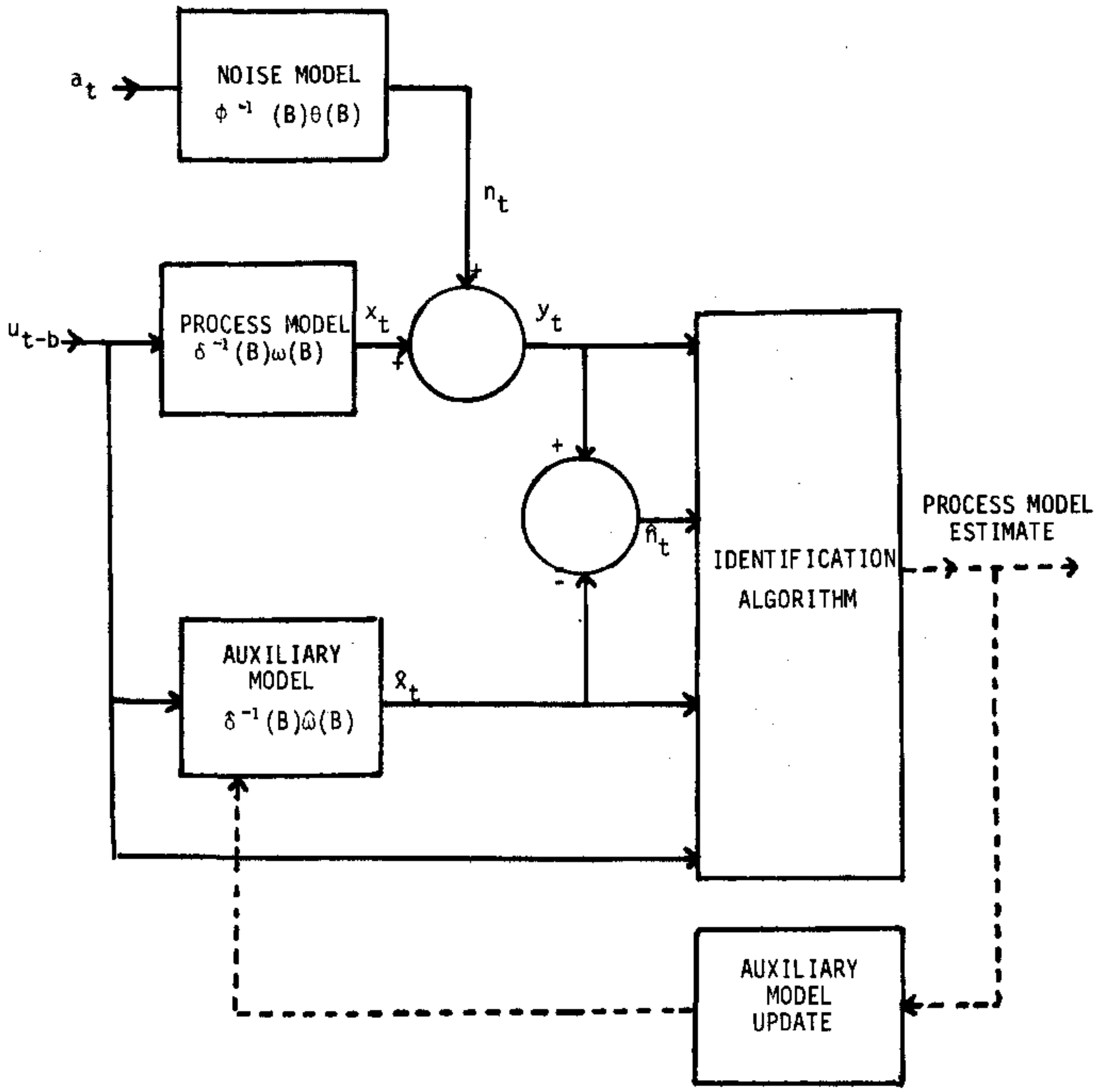

FIGURE 2.3 The Instrumental Variable approach to parameter estimation of a transfer function with superimposed noise

The observed input and output sequences, the auxiliary model output and the estimate of the noise, are then all input to the identification algorithm. This generates an estimate of the process model, which is used to update the auxiliary model. The parameter estimates are refined over subsequent iterations. On the final iteration, an estimate of the noise is made with the parameters in their final fitted form. This sequence may then be used in the identification and estimation of the noise model.

An Approximate Maximum Likelihood (AML) technique is used to estimate the parameters of a univariate model. (Young *et al* 1971; and Moore 1977 section 3.3). This employs a simple modification of the recursive least squares algorithm to furnish unbiased parameter estimates. The statistical efficiency of the parameter estimates is improved by repeated passes through the data. On the final iteration, an estimate is taken of the error, or residual series. This series may be used later to check the adequacy of the model.

2.1.4 Model verification

Model verification can be considered in two stages; firstly, applying diagnostic checks to the residual series, and secondly, by comparing forecasts given by the model with observed series.

When a univariate model has been estimated, the fitted parameters may be used to generate the residual series. Since this series should approximate white noise, there should be no significant values in its autocorrelation or partial autocorrelation functions. The series should also be independent of any deterministic input.

The ultimate test for a model is, of course, its ability to forecast accurately. It should give accurate predictions over the period of the data used in estimation, and more importantly, over a different period.

3. THE CONTROL LANGUAGE

The control language has been designed to follow English as far as possible. A certain degree of flexibility is given by allowing statements to be constructed in different ways. This is done mainly through the use of "noise" words, which have no meaning to the system, but may be included to make a statement easier to read, or remember.

3.1 CAPTAIN statements

There are two types of statement in the CAPTAIN control language, namely control statements and instructions.

3.1.1 Control statements

Control statements mark important points in a CAPTAIN program, and cause certain initialisations to take place. There are six control statements in the language, and these begin with the words

CUSTOMER

TITLE

FINISH

DEMAND

TEKTRONIX

ABORT

The first three of these are job identification statements, and are explained in chapter 4. The DEMAND and TEKTRONIX statements are "mode" controls, and their uses are given in 5.2 and 5.3 respectively. The ABORT statement is only useful in demand and Tektronix modes, and is fully explained in 5.4.

3.1.2 Instructions

An instruction is a statement that informs the system that a new analysis is to be started. It consists of two parts: a main sentence, and a set of qualifying phrases. The main sentence must be present as the first part of the statement, but the qualifying phrases need only be used if required, or if the situation demands it. If any of the phrases are omitted, either a standard value or a previous setting is assumed for the information they would normally introduce. The phrases may appear in any order.

An example of a main sentence is

READ SERIES VARIABLE NAMES RAINFALL AND RUNOFF

This sentence may be given without any qualifying phrases, in which case the system will assume that the data will be following the instruction on cards, and in free format. (see Chapter 7). If this is not the case, then the system must be informed through the use of the qualifying phrases relevant to the instruction. Examples of possible qualifying phrases are

STREAM 20

FORMAT (F8.5, F8.3)

In the above example of a main sentence, the words VARIABLE and AND are noise words, and may be omitted at the discretion of the user.

3.2 Use of the full stop

Normally a statement is actioned when the start of the next statement is encountered. However, it will often be the case, particularly if the package is being used interactively, that the form of the next analysis will depend on the results of the current one. So to action a particular analysis, the sequence

sp . sp

may be entered, where sp represents one or more spaces. Thus, a full stop indicates the end of a particular instruction.

3.3 Card preparation

CAPTAIN statements and data are presented to the system on cards (or equivalent) in free field format. In other words, items may be punched starting in any column, and are separated by one or more blanks or commas. Commas are treated as equivalent to blanks, and may be used anywhere, either instead of blanks, or freely mixed with them. The only exception to this rule occurs in the Transformation section of the package, where commas are not allowed except as separators between function arguments (see Chapter 10).

Although a new card is not necessary for a new statement, for the sake of clarity it is recommended that a fresh card is taken. A statement may be spread over more than one card if necessary; no continuation character is needed.

3.4 Syntax notation

This section describes the notational symbols and rules that are used to guide the construction of CAPTAIN commands. The conventions apply to all syntax skeletons specified in this manual. A complete syntax skeleton for all CAPTAIN commands is given in Appendix 2.

(i) All words that are part of the CAPTAIN control langauge are specified as uppercase words. (A list of these is given in Appendix 1).

(ii) All words that must be present in a command are underlined. Uppercase words that are not underlined are optional, and may be included at the discretion of the user.

(iii) All lower case words represent data that must be supplied by the user.

(iv) Elements of a command that involve a choice are surrounded by parentheses, ie: $\left\{\begin{matrix} x \\ y \end{matrix}\right\}$. Only one of x and y may be chosen.

(v) Optional functions that may be included or omitted as required, are surrounded by brackets, ie: [...].

Example (taken from Chapter 8):

SAVE SERIES list ON STREAM stream-number

[{ FORMAT format-specifications | UNFORMATTED }]

The words SAVE, SERIES and STREAM must be specified, while the word ON is optional; "list" and "stream-number" indicate that data must be supplied in these positions. No format need be given, but if one is present, then either

FORMAT format-specifications

or UNFORMATTED

may be specified, but not both.

Legal constructions of the command would be

(i) SAVE SERIES X ON STREAM 20

(ii) SAVE SERIES A1 A2 A3 STREAM 15 UNFORMATTED

(iii) SAVE SERIES RAINFALL ON STREAM 6
FORMAT (F10.7)

(iv) SAVE SERIES RAIN FLOW STREAM 12

4. JOB IDENTIFICATION COMMANDS

As stated in Chapter 3, there are three job identification commands, namely

CUSTOMER

TITLE

FINISH

Normally, a CAPTAIN program will start with a CUSTOMER card followed by a TITLE card, and end with a FINISH card.

4.1 CUSTOMER command

The command syntax is

CUSTOMER customer-text

"customer-text" consists of a maximum of 71 characters of alphanumeric text (ie: up to the end of the card), and should contain information to identify the user. The information presented cannot be extended onto a second card. More than one CUSTOMER card may be present in a program, but each subsequent one will cause the system to be completely reinitialised, and all information and data previously set up will be lost. The system will also revert to batch mode (see Chapter 5).

4.2 TITLE command

The command syntax is

TITLE title-text

"title-text" consists of a maximum of 74 characters of alphanumeric text (ie: up to the end of the card), and should give information to identify a particular run. The text cannot be extended onto a second card. The information presented is stored by the system, and output at certain positions in the print stream. More than one TITLE card may be present, but the title information on the first will be overwritten by the information on subsequent ones. This facility is useful if different headings are required for different parts of the run.

4.3 FINISH command

The syntax of the FINISH command is

FINISH

This command indicates the end of the sequence of instructions. On encountering it, the system is closed, and so any cards placed after it will not be processed by CAPTAIN. Any information on the FINISH card will be ignored by the system.

5. MODES OF OPERATION

The CAPTAIN package has been designed to operate in three different modes, ie: batch, demand and Tektronix. The second and third of these are interactive modes. As mentioned in Chapter 1, graphical output may be produced on either the Calcomp plotter or the Tektronix visual display unit. The graphs are directed to the Calcomp when running in either batch or demand mode, whereas in Tektronix mode, plots are displayed at the terminal. Graph plotting on the Calcomp is performed offline, and certain job control cards are required to achieve this. These are fully explained in Appendix 3.

5.1 Batch mode

Batch mode is entered automatically when the package is executed and is not initiated by the user. In this mode, the job is presented as a card deck (or equivalent). Print output is sent to the site lineprinter and graphical output is directed to the Calcomp plotter. If an error in the CAPTAIN control language is encountered (e.g.: a wrongly spelled keyword), the run is terminated.

5.2 Demand mode

If the package is being run from an online terminal, an interactive mode may be used. One advantage of this is that the output from one analysis may be examined before the next is initiated. Demand mode may be entered by typing the command

<u>DEMAND</u> <u>MODE</u> [<u>PRINT</u> <u>SWITCH</u> {<u>ON</u> / <u>OFF</u>}]

If the PRINT SWITCH phrase is omitted, PRINT SWITCH ON is assumed. In demand mode, abbreviated print output is displayed at the terminal and any graphs are directed to the Calcomp plotter. In addition, if PRINT SWITCH ON is used, the full print output is written to a mass-storage file and sent to the site lineprinter when the FINISH command is encountered. If an error in the control language is found, an error message is printed and the command may be retyped from the point at which the error was detected.

Once in demand mode, it is only possible to revert to batch mode by entering a new CUSTOMER card.

5.3 Tektronix mode

Tektronix mode is the second interactive mode but may only be used when running the package from a Tektronix 4010 VDU. Note that a Tektronix unit may also be operated in demand mode.

Tektronix mode may be entered via the command

<u>TEKTRONIX</u> <u>MODE</u> [<u>PRINT</u> <u>SWITCH</u> {<u>ON</u> / OFF}]

If the PRINT SWITCH phrase is omitted, PRINT SWITCH ON is assumed. All graphical output is displayed on the Tektronix unit, while print output sent to the terminal is kept to a minimum. If PRINT SWITCH ON is used (either explicitly or by default), the full print output is written to a mass storage file and sent to the site lineprinter when the FINISH card is encountered. If an error in the control language is found, an error message is printed and the command may be retyped from the point at which the error was detected.

After each graph has been plotted on the unit, a list of possible methods of continuation is written in the top right hand corner of the plot. A pair of crosshairs are displayed which may be positioned over the word giving the required action by means of the thumbwheels on the keyboard. Typing any character (with the exception of "RETURN") will then cause the package to proceed with the appropriate action. Since processing will only continue when a character is typed in, this gives the user time to examine a displayed plot before initiating any following ones.

At present, the possible actions are to take a hard copy of the plot before continuing, to continue normally, and to inhibit the plotting of any subsequent graphs. These actions are represented by the words

HARD COPY

CONTINUE

ABORT

respectively. Note that the ABORT option only has an effect on graphs produced in the current analysis. Graphs produced by a later analysis will not be affected.

The Tektronix screen is wiped clean before each graph is plotted and after the last in a group of graphs has been displayed.

Tektronix mode may be terminated either by the command

<u>TEKTRONIX</u> <u>END</u>

or by a DEMAND MODE command (5.2). In both cases the system will revert to demand mode. Once Tektronix mode has been entered, it is only possible to revert to batch mode by entering a new CUSTOMER card.

5.4 The ABORT command

If the package is being run in an interactive mode (ie: demand or Tektronix mode), it is possible to use the ABORT facility. Suppose that while a command is being typed in, the user realises that he has made an error (other than a syntax error). For example, the number

6. DATA HANDLING

The basic organisational unit of data is the series. This has its usual statistical definition, ie an ordered sequence of observations such as temperature readings taken every hour. Associated with each series is a series name (or variable name). For series presented to the system, this name must be supplied by the user. However, during execution, other series may be generated by the system for which the user may optionally supply names. Note that if a system generated series is to be used in a subsequent analysis, it must be named by the user, otherwise it will be deleted.

Also associated with each series known to the system is a summary containing items such as variable name, number of data values in the series, mean and standard deviation. The summary is generated automatically, and is used extensively by the system in the different analyses. The values in the summary for a particular series are output when the series is first presented to the system, or when the series is generated. If the package is being run in one of the interactive modes, the summaries will not appear on the terminal. However, they will be present on the full printout if PRINT SWITCH ON is used. The summaries may also be output through the use of the SUMMARY command (see Chapter 18), in which case they will appear on all print streams.

6.1 Missing values

At present, the system has no method of dealing with missing values. If the data to be analysed contain missing values, they must be estimated or replaced by the user before entry to the system.

6.2 Variable names

The user is required to supply names for all series presented to the system, and may also supply names for system generated series by using the NAME clause where appropriate. The NAME clause will be discussed in later chapters.

A variable name may be formed from the letters A-Z, the digits 0-9, and the hyphen (-), and must begin with a letter. Embedded blanks or spaces are not permitted. A keyword may not be used as a series name. A list of keywords is given in Appendix 1.

A name is stored by the system as an eight character string. If less than eight characters are presented as a name, they are stored left justified, and space filled to the right to eight characters. If more than eight characters are presented, then the name is truncated from the right. Thus, the names XYZ, RAINFALL, FLOW-1 and OVEREIGHT would be stored as

XYZ

RAINFALL

FLOW-1

OVEREIGH

The following names would not be valid:

123TEMP

OUT FLOW

STOCK/5

The first begins with a character other than a letter, the second contains an embedded blank, and the third contains the illegal character / .

6.3 System named series

If a system generated series is not named by the user, the system will supply a name for purposes of summaries, graphical output, etc. System supplied names are easily distinguished, since they make use of the otherwise illegal character $.

Examples of system supplied names are

WHITE$-X

NOISE$

ERR$I-1

System named series are deleted at the end of the analysis that generated them.

6.4 Restrictions on series

A maximum of 30 series may be known by the system at any one time. An attempt to name more than 30 series will result in an error.

All series known to the system are held in a disc file, and are brought into main memory in sections, as required. Hence, no restrictions are made on the possible lengths of series, other than the limits of disc file storage, ie. series may be any length, provided that the capacity of the file is not exceeded. In practice, this allows series lengths of several thousand points.

7. READING THE SERIES

All series to be input to the system must be presented via the READ command.

7.1 Command syntax

READ SERIES   VARIABLE {NAME / NAMES} list

[STREAM   stream-number]

[POINTS   no-of-points]

[{FORMAT / UNFORMATTED}   format-specifications]

7.2 The main sentence

READ SERIES   VARIABLE {NAME / NAMES} list

"list" is a set of user supplied names which are to be associated with the series being read in. A name must be given for each series to be input. A maximum of eight series may be read in with one READ command, so at least one name, and not more than eight names may be given. If more than one name is supplied, they may be separated either by blanks, commas, or the word AND.

7.3 Qualifying phrases

The one restriction in the use of qualifying phrases with the READ command occurs in their order, in that if a FORMAT phrase is specified, it must be the last phrase to appear.

7.3.1 STREAM stream-number

Normally, the data will follow the READ statement on cards (or equivalent). However, this may be inconvenient as the data may already be on tape or disc storage. If this is the case, then the STREAM phrase may be appended to the READ statement to inform the system where the data is to be found.

"stream-number" is an integer, and corresponds to a disc or a tape file (see Appendix 3).

7.3.2 POINTS no-of-points

If the number of data points in the series to be read is known, then it may be specified by the POINTS qualifying phrase, where "no-of-points" is an integer. Note that there is no limit imposed by the system on the length of the series, other than the limits of disc file storage.

7.3.3 {FORMAT / UNFORMATTED} format-specifications}

The FORMAT clause may be used when the series to be read are in a fixed format. It describes the arrangement of the items that constitute one data point.

The "format-specifications" have the same form as the specifications of a normal FORTRAN format statement, and must start with an open bracket and finish with a close bracket. There are however, three restrictions. Firstly, there must be at least one space between the word FORMAT and the first open bracket. Secondly, only the editing codes D,E,F,G,X and the control character / (slash) are allowed. Thirdly, the specifications cannot be split over two or more cards, i.e. they must be presented complete on one card.

The UNFORMATTED option must be specified if the data to be read in is in binary. Obviously, no format specifications would be needed in this case.

If no format clause is given, the data is assumed to be in free format, with items being separated by blanks and/or commas.

7.4 Format of the data

The data to be read is arranged to correspond with the format clause if one is specified, otherwise in free format decimal values. The one is specified, otherwise in free format decimal values. The data items must correspond to the "list" on the read statement. Thus, the first item read will be the first value for the first variable in the list, the second item will be the first value for the second variable, and so on. When a value has been assigned to each variable in the list, the next item read will be the second value for the first variable, etc. For example, if three series X, Y and Z are to be read in, the order of the data items read will be

$$X_1, Y_1, Z_1, X_2, Y_2, Z_2, X_3, Y_3, Z_3, \ldots\ldots\ldots X_n, Y_n, Z_n$$

In this case, the values X_1, Y_1 and Z_1 constitute the first data point, X_2, Y_2 and Z_2 the second data point, and so on.

All the series read in via one READ statement must be the same length.

If a format clause is used, one card image cannot contain more than one data point, although a point may be spread over two or more card images. Reading of the data is terminated by the word

<u>END</u>

or by the end of file if a STREAM phrase is present.

Note that if END is used, it must appear in a field covered by any of the editing codes in the FORMAT statement other than X. For example, if

```
FORMAT (F7.5, F6.1)
```

was specified, then END could appear anywhere between columns 1 to 13. inclusive. However, if the FORMAT statement was

```
FORMAT (15X, F10.3) ,
```

then END would have to appear between columns 16 and 25 inclusive.

If a POINTS phrase is used, reading is terminated after the given number of points. END may be present after the data, but in this case it is not necessary.

If no FORMAT clause is specified, the items may be arranged so that a card image contains many data points, or a data point spans several card images. Reading is terminated when the specified number of points (if any) has been reached, or when any non-numeric input is encountered, ie: a new statement, END, or the sequence space-full stop-space.

7.5 Examples

(i) READ SERIES VARIABLE NAMES RAINFALL RUNOFF

```
data
 :
 :
 :
 :
```

(ii) *READ SERIES NAME SUNSPOTS*
STREAM 15 UNFORMATTED

(iii) READ SERIES VARIABLE NAMES A B AND C POINTS 325 FORMAT

```
(2F7.5, 10X, E9.5)
data
 :
 :
 :
```

8. SAVING SERIES

In normal operation, all series generated by the system will be lost at the end of the run. However, it is possible to save any of the generated series using the SAVE command

8.1 Command syntax

SAVE SERIES list ON STREAM stream-number

$$\left[\left\{\begin{matrix}\underline{\text{FORMAT}} & \text{format-specifications} \\ \underline{\text{UNFORMATTED}} & \end{matrix}\right\}\right]$$

8.2 The main sentence

SAVE SERIES list ON STREAM stream-number

"list" is a set of names associated with series known to the system, separated by spaces, commas or the word AND. A maximum of eight names may be given. If more than one series is specified, the system checks the lengths of the series. Those that have the same number of points are written out in parallel, point by point. Series of different lengths are written out one after the other. For example, if series X, Y and Z are to be saved, having lengths n,n and m respectively, the output stream would look like

X_1 Y_1

X_2 Y_2

: :

: :

X_n Y_n

Z_1

:

:

:

Z_m

"stream-number" is an integer and corresponds to an output file (see Appendix 3). Note that it is possible to obtain listings of series by setting "stream-number" to correspond to one of the print streams. On the Univac 1108, stream 6 is the standard FORTRAN print stream, and is used for the normal output from CAPTAIN. Stream 7 is the auxiliary print stream of the package, and is used when running with the print switch on in one of the interactive modes (see 5.2 and 5.3). If the standard print stream is selected and the print switch is on, then the series will also be listed on the auxiliary print stream. The auxiliary stream may not be selected unless the print switch is on.

8.3 Qualifying phrase

8.3.1 { FORMAT format-specifications / UNFORMATTED }

The series are written in the required format through the use of a FORMAT phase. "format-specifications" has the same form as the specifications of a normal FORTRAN format statement and must start with an open bracket and finish with a close bracket. The three restrictions are that firstly, there must be at least one space between the word FORMAT and the first open bracket, secondly, only the editing codes D, E, F, G, X and the control character / (slash) are allowed, and thirdly, the specifications cannot be split over two or more cards.

If the UNFORMATTED option is used the series will be written in binary.

If no format phrase is present, UNFORMATTED is assumed unless one of the print streams has been specified, in which case a standard G format is used.

8.4 Example

```
SAVE  SERIES  ERROR-1 AND ERROR-2   STREAM 20
FORMAT (F8.5, 5X, F7.3)
```

9. DIFFERENCING SERIES

Trends and seasonal variations in time series may be removed by suitable differencing. However, in practice differencing tends to amplify the noise on the data, and it is usually preferable to employ some alternative means of reducing the series to stationary form. In accordance with the method of Box and Jenkins though, facilities are available in the package to difference and seasonally difference series.

9.1 Command syntax

DIFFERENCE [SEASONALLY] SERIES list1 [key1] [PERIOD key2]

[NAME DIFFERENCED SERIES list2]

9.2 The main sentence

DIFFERENCE [SEASONALLY] SERIES list1 [key1] [PERIOD key2]

The options SEASONALLY and PERIOD are only used for seasonal differencing. "list1" is a set of names of series to be differenced, the names being separated by spaces, commas, or the word AND. "key1" gives the degree of differencing required and may be an integer greater than zero or one of the words ONCE or TWICE. All the series in "list1" will be differenced "key1" times. If "key1" is omitted, 1 is assumed. "key2" is an integer greater than zero and is the period required for seasonal differencing. If SEASONALLY is specified, the PERIOD option must be used, and vice versa.

9.3 Qualifying phrase

9.3.1 NAME DIFFERENCED SERIES list2

The series produced by differencing may be named by using the NAME clause. "list2" is a set of supplied names to be attached to the differenced series, separated by spaces, commas, or the word AND.

If no NAME clause is used, the differenced series will overwrite the original series. If one name is supplied for each series to be differenced, then intermediate differences will be lost and the supplied names will be attached to the "key1" degree differences. If the series are to be differenced twice or more, and intermediate differences are required, then all differences must be named. (First differences are named first, followed by the second differences, and so on. Example (iii) below will clarify this.)

Note that the names on the NAME clause should be in the same order as the series whose differences they will be attached to (see examples (ii) and (iii)).

9.4 Examples

(i) DIFFERENCE SERIES X ONCE

(ii) DIFFERENCE SERIES AA BB NAME AA-1 BB-1

(iii) DIFFERENCE SERIES RAIN AND FLOW 2
NAME RAIN-1 FLOW-1 RAIN-2 FLOW-2

(iv) DIFFERENCE SEASONALLY SERIES SALES TWICE PERIOD 12
NAME DIFFERENCED SERIES SALES-2

10. TRANSFORMATION OF SERIES

The transformation of series facility allows various arithmetic operations to be performed on series. Series used in the CAPTAIN package may also be transformed by any of 19 available transformation functions.

10.1 Command syntax

TRANSFORM SERIES

. . . . .

Transformations

. . . . .

END

10.2 Transformations

The format of the transformations closely follows that of FORTRAN arithmetic statements with the general format:

v = e

where v is a series name and e is an arithmetic expression containing series names (and constants), operators and/or functions.
Note that if the series name includes a hyphen '-', then to use this in the transformations the series name must be placed in quotes to signify that the hyphen is not a minus sign.

i.e. To use a series called RECORD-1 in the transformations it must be enclosed thus 'RECORD-1', otherwise it will be interpreted as a series name RECORD minus 1.

The operations involving series names imply a loop over all the points in the series i.e. A = B + C, where B and C must be series of the same length, implies A(1) = B(1) + C(1), A(2) = B(2) + C(2), ... A(n) = B(n) + C(n).

Similarly A = B + 100 generates a new series A from B by adding the constant 100 to every point in B.

The following set of operators is allowed:

Operator	Meaning
+	Addition or unary plus
-	Subtraction or unary minus
*	Multiplication
/	Division
**	Exponentiation

Parentheses may be used to specify the order of operation in an expression. If parentheses are omitted, the following order of evaluation is observed; (the > denotes "is evaluated before")

Unary +, - following ** > ** > Unary +, - > *, / > +, -

Constants are floating point numbers, which include all the usual representations; for example, the values 1.34, 14, 52., 0.374, 1.2E-4, -7.6D+2 are valid. (Note spaces are not allowed within numbers.)

There are also 19 allowable mathematical functions available in the package (see section 10.3), and these may be used in a similar manner to functions in FORTRAN arithmetic statements.

e.g. RSRLOG = LOG(RSR) will generate a new series RSRLOG from the series RSR by taking the common log of each value.

Any number of transformation expressions, one per line but not continued over lines, may appear between the TRANSFORM SERIES and the END limiters. The transformations are read by the system and checked for syntax errors. Each expression is then executed in turn, the current expression being completed before the next is started. This allows a new series to be defined and used in a later expression in one entry to the transformation analysis

eg.
```
TRANSFORM SERIES
S1 = S2 + S3
DIFF = A - S1
END
```

Note: Although in theory each expression may be arbitrarily complex, there is a practical limit due to the amount of work space available. For this reason a maximum of eight series may be specified per expression.

10.3 Functions

The functions available for use in arithmetic operations may be arranged into three classes. The first of these contains the usual functions such as LOG,SIN, etc and may be used with both series names and constants as arguments.

The second class of functions are applicable only to series, and return a single value pertaining to the series as a whole, which may be used in generating new series.

The third class allows new series to be generated either by removing a number of points from the beginning or the end of known series.

10.3.1 Class 1 Functions

These are standard functions whose arguments may be series names or constants. For example, A = LOG(B) will generate a series A of equal length to B whose values are the common logs of the values of B. On the other hand A = C + LOG(15.5) will add the log of 15.5 to each point in C to form A.

Name	No. of Arguments	Example	Function Value	Remarks
LN	1	LN(A)	$\log_e(A_i)$ for all i	$A_i>0$ for all i
SIN	1	SIN(A)	$\sin(A_i)$	A_i in radians
COS	1	COS(A)	$\cos(A_i)$	"
TAN	1	TAN(A)	$\tan(A_i)$	"
LOG	1	LOG(A)	$\log_{10}(A_i)$	$A_i > 0$
EXP	1	EXP(A)	e^{A_i}	
ABS	1	ABS(A)	$\|A_i\|$	
MOD	2	MOD(A,B)	$A_i - Z*B_i$ where Z is the integer part of A_i/B_i	$B_i \neq 0$
MAX	2	MAX(A,B)	Returns maximum of A_i and B_i	
MIN	2	MIN(A,B)	Returns minimum of A_i and B_i	
SQRT	1	SQRT(A)	$\sqrt{A_i}$	$A_i \geq 0$
ASIN	1	ASIN(A)	Arcsin (A_i)	
ACOS	1	ACOS(A)	Arccos (A_i)	
ATAN	1	ATAN(A)	Arctan (A_i)	

In all the above the series name may be replaced by a constant.

10.3.2 Class 2 Functions

These functions are applicable to series only. They return a single value pertaining to a series as a whole, which may be used in generating new series.

Name	No. of Arguments	Example	Function
MEAN	1	MEAN(A)	This returns the mean value of the series A
VARNCE	1	VARNCE(A)	This returns the variance of the series A
LENG	1	LENG(A)	This returns the number of points in series A.

<u>Note:</u> These functions return their value by reference to the summary table kept for all series known to the system. Hence the arguments of these functions must be single series names and not an expression.

ie MEAN(B) is correct if B is known to the system, but MEAN(SERIES1 + SERIES2) is not allowed, as expressions may not be used as arguments to class 2 functions.

Also since these functions return a single value it is <u>illegal</u> to code X = MEAN(A) , for example, as X is a <u>series</u> name. Thus these functions may only be used where it is sensible to use a single value.

10.3.3 Class 3 Functions

These are series manipulation functions.

Name	No. of Arguments	Example	
TRUNC1	2	TRUNC1(A,n)	This will truncate n points from the end of A
TRUNC2	2	TRUNC2(A,n)	This will truncate n points from the beginning of A

10.4 Examples

(i)
```
TRANSFORM SERIES
A = RN1 + RU1 - SUM
B = T*Q + A + 3.4E-2
END
```

(ii)
```
TRANSFORM SERIES
z = 2.0*(TWO + THREE)
END
```

(iii)
```
TRANSFORM SERIES
SLOG = A + LOG(B)
END
```

(iv)
```
TRANSFORM SERIES
XSTD = (X - MEAN(X))/SQRT(VARNCE(X))
END
```

(v)
```
TRANSFORM SERIES
C = MEAN(A) + D
END
```

(vi)
```
TRANSFORM SERIES
'A-B'  =  A-B
Z      =  A+B
END
```

(vii)
```
TRANSFORM SERIES
X = TRUNC1(X,3)
END
```

11. SERIES DESCRIPTION

The series description facility of the package has applications at all stages of model building. The facility is used to calculate the correlation functions (autocorrelation, partial autocorrelation, and cross-correlation) of given series and to plot the series themselves.

Using the correlation functions:

(i) any periodicity or non-stationarity in the data can be determined,

(ii) the structure of stochastic ARMA models may be identified (in particular, the noise model and the model used to prewhiten the input series),

(iii) residual and error series may be checked that they approximate white noise, and that they are not significantly cross-correlated with the observed series.

The series and their correlation functions may be plotted, if required, on the graphical device in use. Examples of graphs produced on the Calcomp plotter are displayed in Figures 11.1 to 11.5. (The data used is the gas furnace data, analysed by Box and Jenkins (1970), and Young, Shelswell and Neethling (1971)).

11.1 Command syntax

DESCRIBE SERIES list

[{LAG / LAGS} k]

[{PLOT / PLOTTING} plot-list]

11.2 The main sentence

DESCRIBE SERIES list

"list" consists of one or two names of series known to the system. If two are given they may be separated by blanks, a comma, or the word AND. The first is taken to be the input series and the second is taken to be the output series. If the series are of different lengths, the longer series will be truncated from the beginning. If more than two names are given, the remainder are ignored.

The autocorrelation and partial autocorrelation functions are estimated for each series but no cross-correlation function is calculated if only one series is given. The correlation functions are printed, together with their corresponding 95% confidence limits.

11.3 Qualifying phrases

11.3.1 {LAG | LAGS} k

This phrase may be used to specify the number of lags required in the calculation of the correlation functions. "k" is an integer, and must be in the range

$$0 < k \leqslant 100$$

It must also satisfy the condition

$$k < N$$

where N is the number of points in the shortest series specified for the analysis.

If the phrase is omitted, the number of lags is taken to be the value given on the most recently declared LAGS phrase. If there have been no such declarations, it is taken to be the smaller of 50 and N/4, unless the package is being run in Tektronix mode, in which case the default is the smaller of 40 and N/4.

11.3.2 PLOTTING plot-list

If graph plots of any of the correlation functions or the series are required they must be requested by use of the PLOTTING phrase. "plot-list" consists of any combination (in any order) of the following specifications:

(i) { AUTOCORRELATION {FUNCTION | FUNCTIONS} | ACF }

(ii) { PARTIAL AUTOCORRELATION {FUNCTION | FUNCTIONS} | PACF }

(iii) { CROSS CORRELATION FUNCTION | XCF }

(iv) { SERIES | name-list }

Specifications are separated by blanks, commas, or the word AND. "name-list" consists of a subset of the series names given in "list" in the main sentence. The names can be in any order. If the SERIES option is used, all the series in "list" are plotted. The cross correlation function should not be requested if only one series is given. The correlation functions are plotted with their 95% confidence limits, as shown in figures 11.3, 11.4 and 11.5

11.4 Examples

(i) DESCRIBE SERIES SUNSPOTS

(ii) DESCRIBE SERIES RAINFALL AND RUNOFF PLOTTING SERIES AND AUTOCORRELATION FUNCTIONS

(iii) DESCRIBE SERIES GASIN GASOUT
PLOT GASIN GASOUT ACF PACF AND XCF LAGS 50

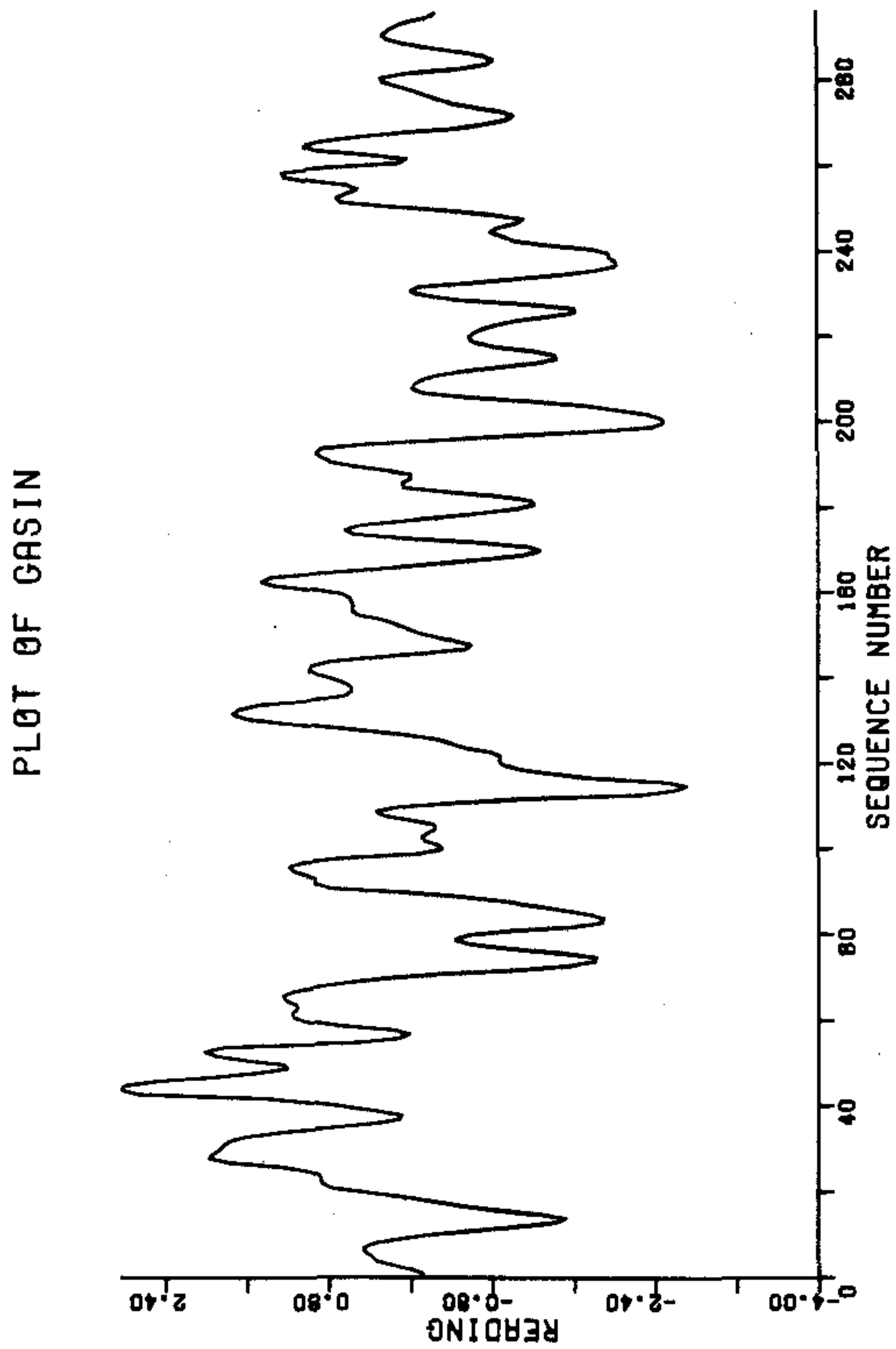

FIGURE 11.1 Graph of the observed input series

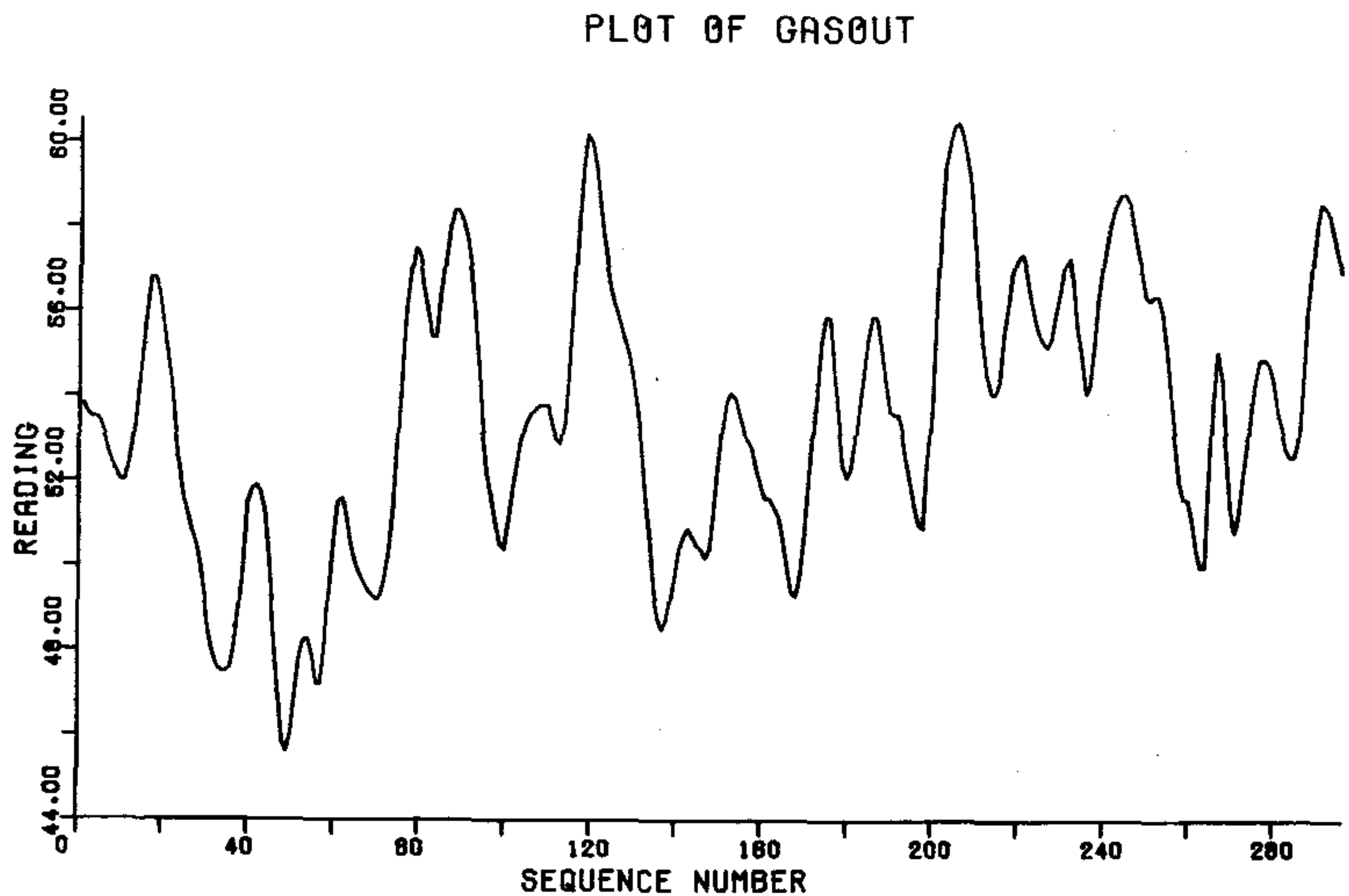

FIGURE 11.2 Graph of the observed output series

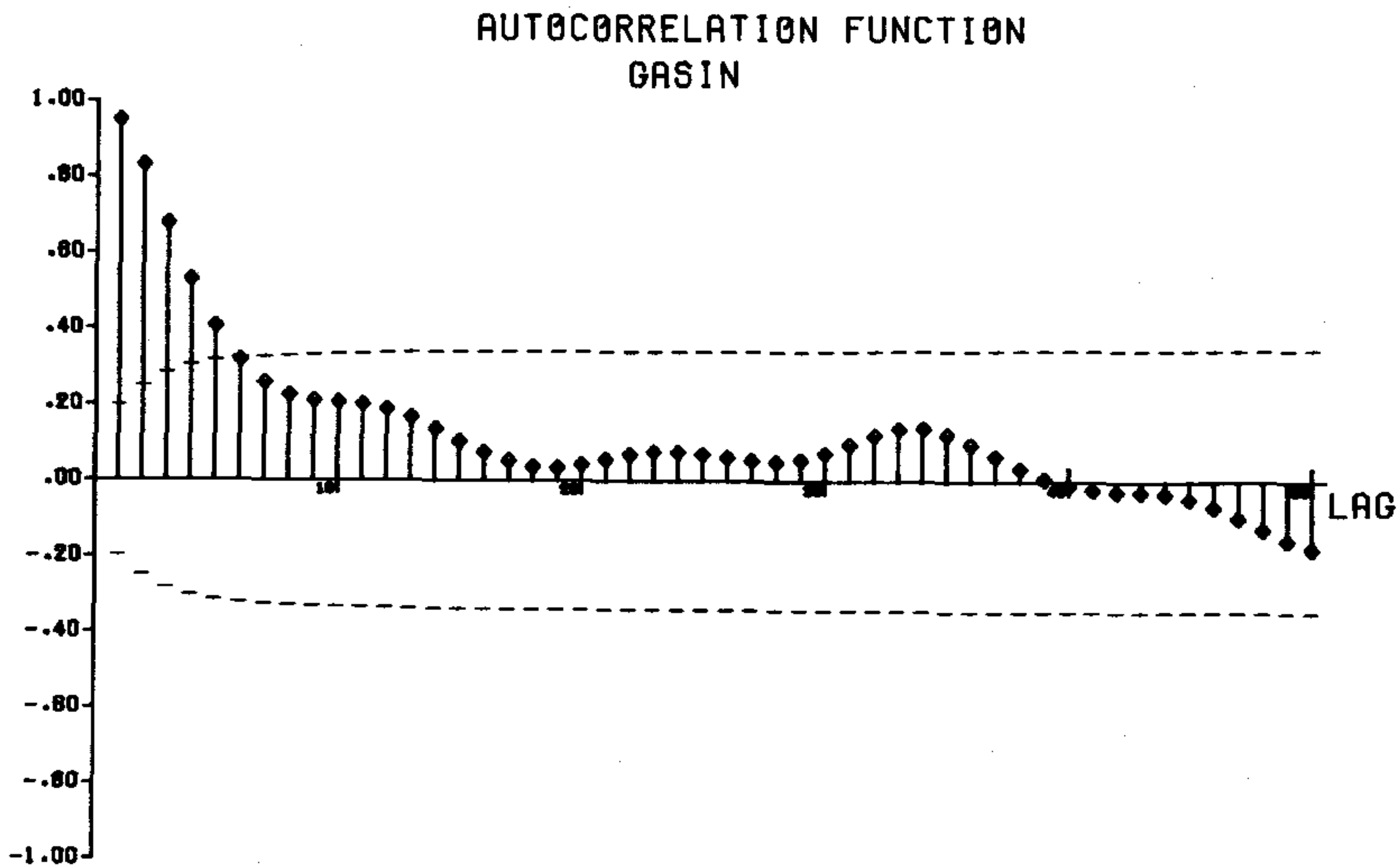

FIGURE 11.3 Estimated autocorrelation function of the observed input series, with 95% confidence limits

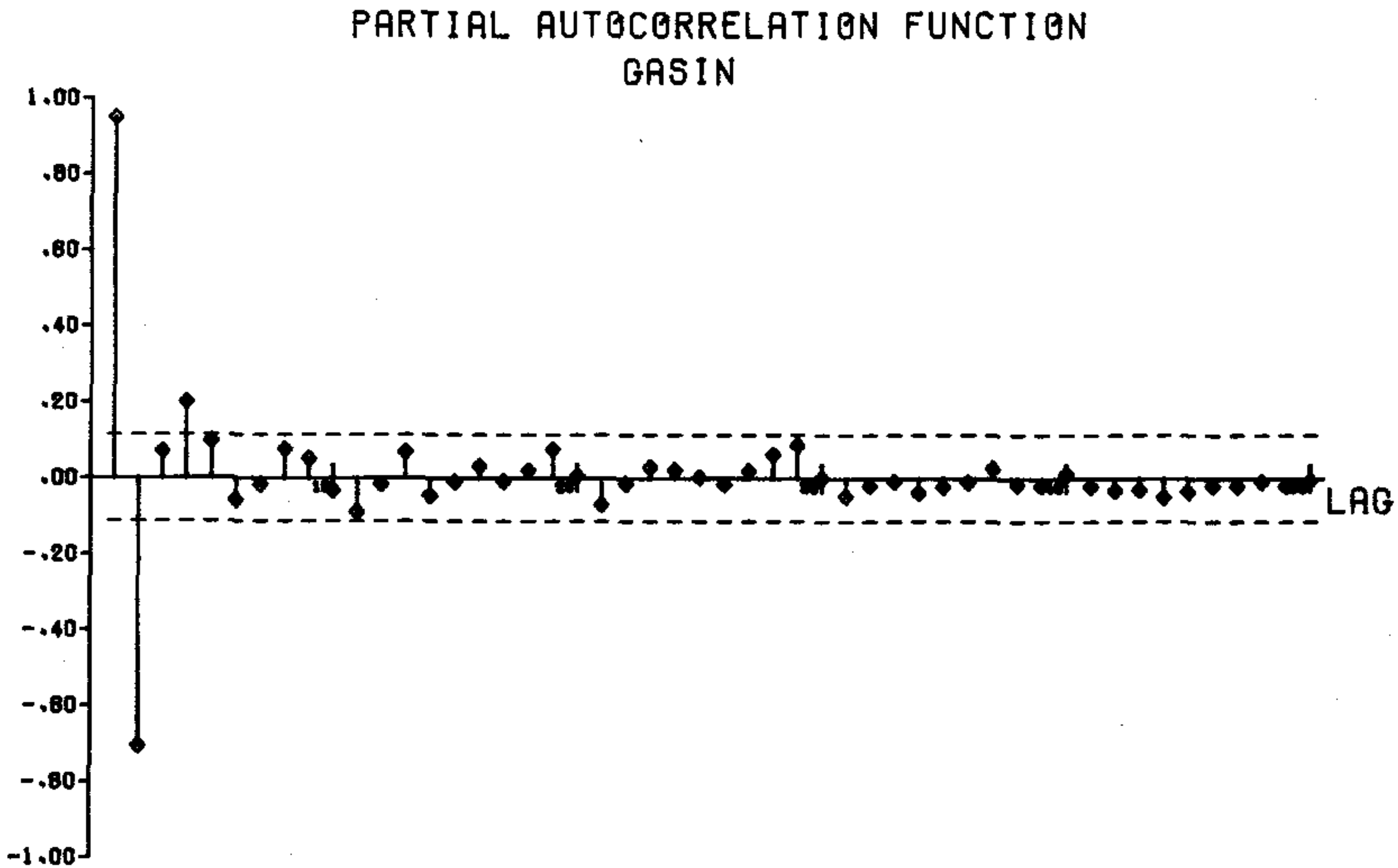

FIGURE 11.4 Estimated partial autocorrelation function of the observed input series, with 95% confidence limits

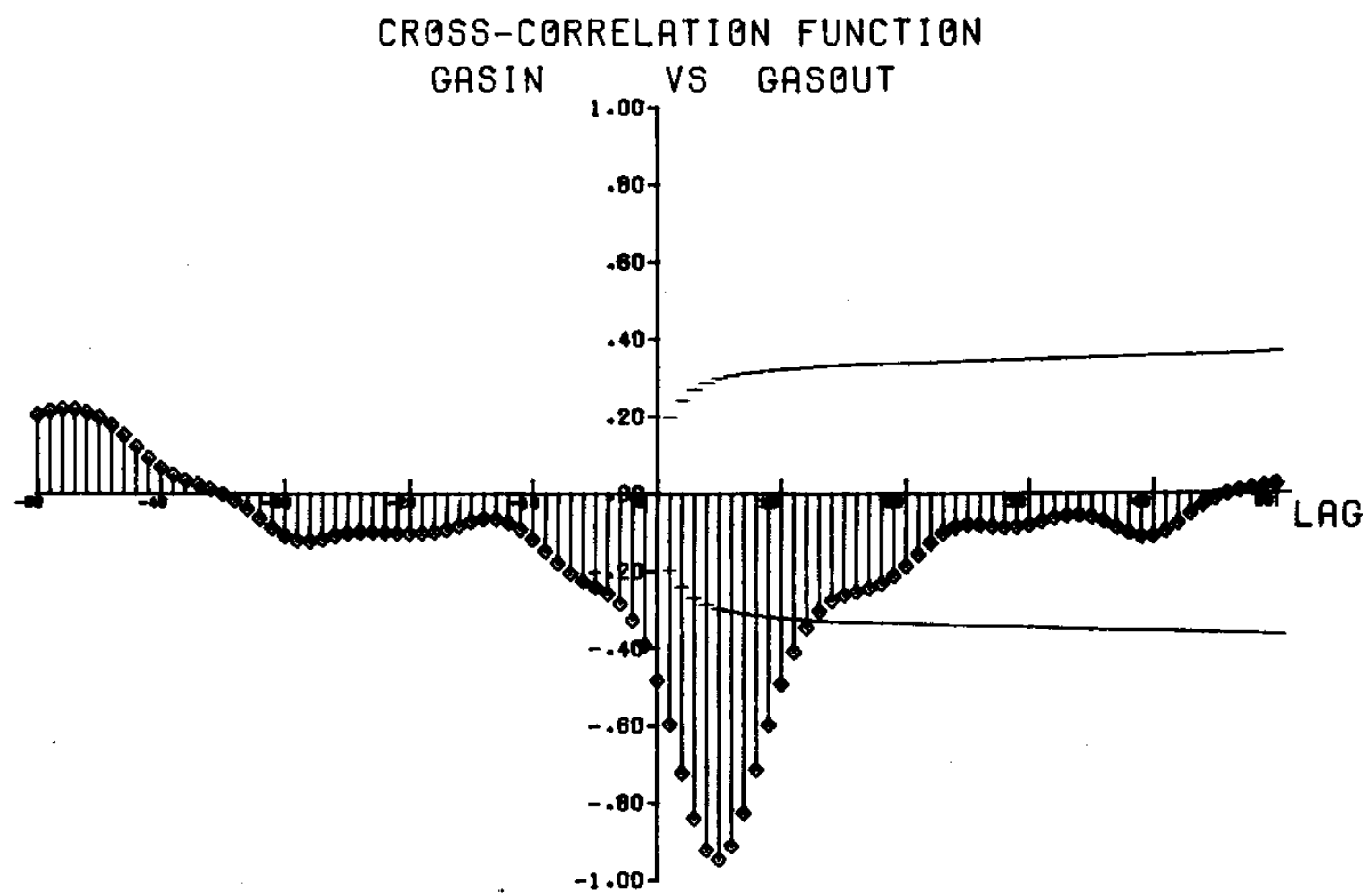

FIGURE 11.5 Estimated cross correlation function between the observed input and output series, with 95% confidence limits

12. PREWHITENING

The prewhitening facility in CAPTAIN is used to estimate the impulse response function of a transfer function model. Firstly, the observed input series is considered to be the output of a stochastic model and a normal identification procedure is carried out. The identified stochastic model structure is then used as input to the prewhitening procedure. The autoregressive parameters of the model are calculated directly from the autocorrelation function of the observed input and a simple Newton-Raphson iterative procedure is employed in estimating the moving average parameters. If the input series is then passed through the inverse of the model, it should resemble white noise and is said to have been "prewhitened". The cross correlation function between the prewhitened input series and the identically transformed output series is directly proportional to the impulse response function.

Another use for the prewhitening facility is to obtain rapid estimates of univariate model parameters. Used in this way, it provides a quick check on the adequacy of an identified model structure, before the parameters are estimated efficiently.

The autocorrelation and partial autocorrelation functions of the prewhitened input series are calculated and these may be used to check that it has been transformed into white noise. If input series and output series are specified, then the cross correlation function between the two transformed series and the impulse response function are also estimated. In addition, an estimate is made of the autocorrelation and partial autocorrelation functions of the ε-noise component. This ε-noise series is related to the actual noise series and its correlation functions may be used to identify the noise model, and to obtain rough parameter estimates.

Although the process is statistically inefficient, in practice the estimates of the impulse response function are good enough to allow reasonable identification of the transfer function.

Note that the ε-noise series autocorrelations and partial autocorrelations are estimated from the cross correlation function between the transformed series and are very sensitive to any inadequacies in the specified model. In particular, if the model selected does not transform the input series to white noise, estimates greater than one in absolute value can occur in the ε-noise series correlation functions. This is entirely due to their method of calculation but serves as a further indication of model inadequacy.

12.1 Command syntax

{ WHITEN / PREWHITEN } SERIES list1 STRUCTURE of MODEL structure

[PRINTING <u>CONVERGENCE</u> OF {<u>PARAMETER</u> / <u>PARAMETERS</u>}]

[<u>NAME</u> WHITENED SERIES list2]

[{<u>PLOT</u> / <u>PLOTTING</u>} plot-list]

[{<u>LAG</u> / <u>LAGS</u>} k]

12.2 The main sentence

{<u>WHITEN</u> / <u>PREWHITEN</u>} SERIES list1 <u>STRUCTURE</u> OF MODEL structure

"list1" consists of one or two names of series known to the system. If two series are given, they must be of equal length. The first series specified is taken as the input series and the second is taken as the output series. The names may be separated by blanks, a comma, or the word AND. If more than two are given, the remainder are ignored."structure" has the form

[{<u>AUTOREGRESSIVE</u> {<u>PARAMETER</u> / <u>PARAMETERS</u>} / <u>AR</u>}] p [{<u>MOVING AVERAGE</u> {<u>PARAMETER</u> / <u>PARAMETERS</u>} / <u>MA</u>}] q

The values p and q are integers in the range

$$0 \leq p, q \leq 10$$

and indicate the number of autoregressive and moving average parameters of the model to be used. Both p and q must be specified, with or without introductory words. If no introductory words are used, then the value p must precede the value q. However, if the introductory words are used the ordering is unimportant.

12.3 Qualifying phrases

12.3.1 PRINTING <u>CONVERGENCE</u> OF {<u>PARAMETER</u> / <u>PARAMETERS</u>}

The convergence of the moving average parameters may be output through the inclusion of this phrase. At each iteration, the contents of the relevant vectors and matrices are printed. The process is assumed to have converged when the elements of the error vector, F, are all less than 10^{-5} in absolute value. A maximum of 100 iterations is performed.

The phrase is meaningless if the value q (see 12.2) is set to zero, and will be ignored.

12.3.2 NAME WHITENED SERIES list2

If the transformed series are required for a subsequent analysis, they must be named by the NAME clause. "list2 " contains the names to be attached to the new series. If two series have been specified in "list1" in the main sentence, then "list2 " may contain one or two names. In the first case, only the prewhitened input series is named but in the second case both prewhitened series are named. If one series has been specified in "list1", then only one name may appear in "list2". If the prewhitened series are not named, they will not be retained by the system.

12.3.3 { PLOT | PLOTTING } plot-list

Graph plots of the whitened series, and the relevant correlation functions can be obtained by including a PLOTTING phrase. "plot-list" consists of any combination (in any order) of the following specifications

(i) { AUTOCORRELATION { FUNCTION | FUNCTIONS } | ACF }

(ii) { PARTIAL AUTOCORRELATION { FUNCTION | FUNCTIONS } | PACF }

(iii) { CROSS CORRELATION FUNCTION | XCF }

(iv) { WHITENED SERIES | name-list }

Specifications are separated by blanks, commas, or the word AND. "name-list" consists of a subset of the series names given in "list2" in the NAME clause (12.3.2). The names can be in any order. If the SERIES option is used, only the prewhitened input series will be plotted; the prewhitened output (if any) will not be drawn. If two series are presented for prewhitening, the autocorrelation and partial autocorrelation functions of the transformed output series are not plotted, since they are of no significance to the analysis. (The plural form of FUNCTION is retained for compatability with the PLOTTING clause of the DESCRIBE instruction - see 11.3). The cross correlation function is estimated from the transformed input and output series and yields the impulse response function on multiplying by the ratio of their standard deviations.

Examples of the graphs are displayed in Figures 12.1 to 12.4.

12.3.4. { LAG | LAGS } k

This phrase may be used to specify the number of lags required in the calculation of the various correlation functions. "k" is an integer, and must be in the range

$$0 < k \leq 100$$

It must also satisfy the condition

$$k < N-p$$

where N is the number of points in the original series and p is the number of autoregressive parameters, as defined in 12.2. If the phrase is omitted, the number of lags is taken to be the value given on the most recently declared LAGS phrase. If there have been no such declarations, it is taken to be the smaller of 50 and N/4, unless the package is being run in Tektronix mode in which case the default is the smaller of 40 and N/4.

12.4 Examples

(i) WHITEN SERIES X Y STRUCTURE AR 2 MA 1

(ii) PREWHITEN SERIES GASFLOW STRUCTURE OF MODEL 2 0 NAME WHITENED SERIES GAS-1 PLOT ACF AND PACF

(iii) WHITEN SERIES A AND B STRUCTURE OF MODEL AUTOREGRESSIVE PARAMETER 1 MOVING AVERAGE PARAMETERS 3 PRINTING CONVERGENCE OF PARAMETERS PLOTTING SERIES AND CROSS CORRELATION FUNCTION LAGS 45

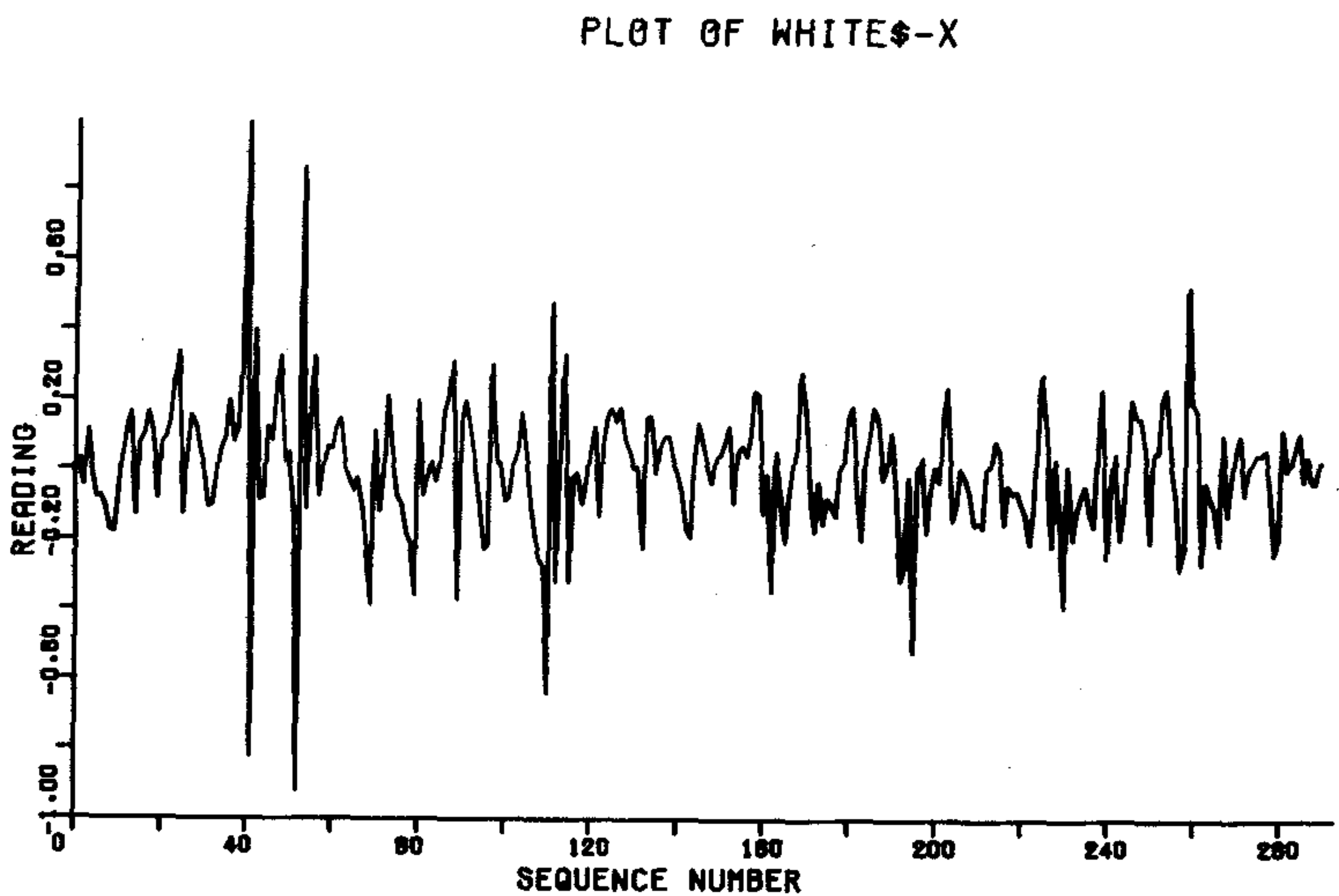

FIGURE 12.1 Graph of the prewhitened input series

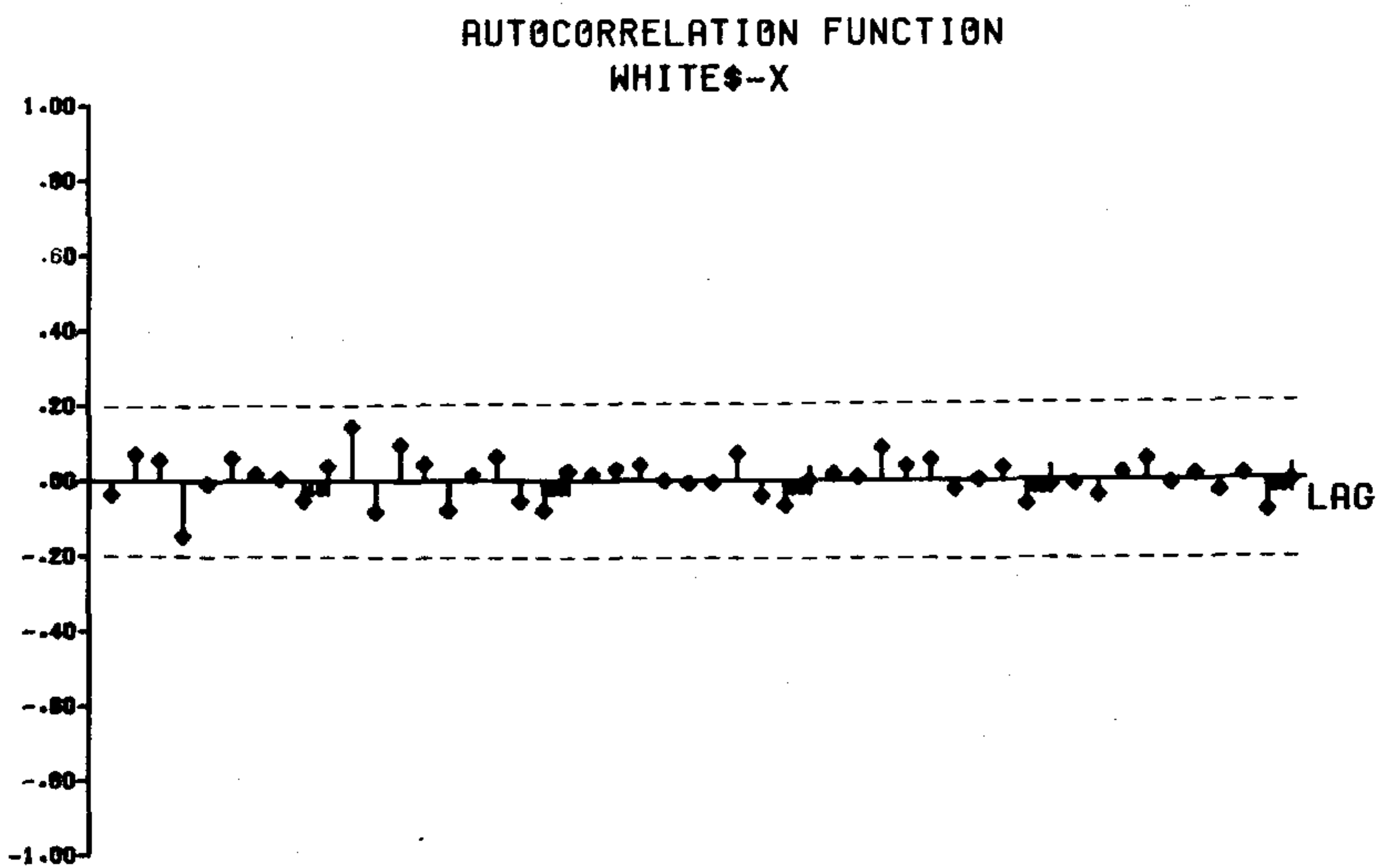

FIGURE 12.2 Estimated autocorrelation function of the prewhitened input series

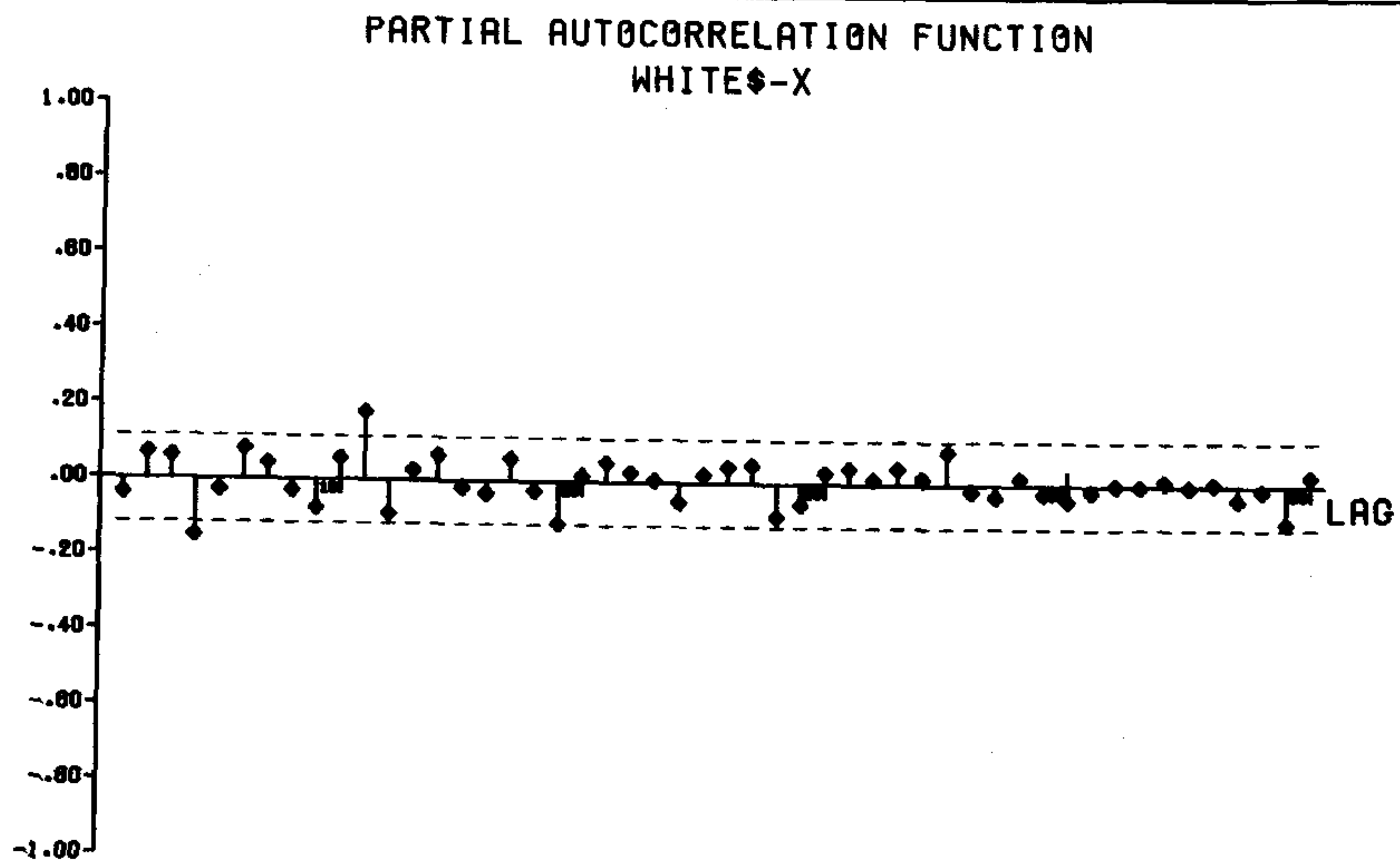

FIGURE 12.3 *Estimated partial autocorrelation function of the prewhitened input series*

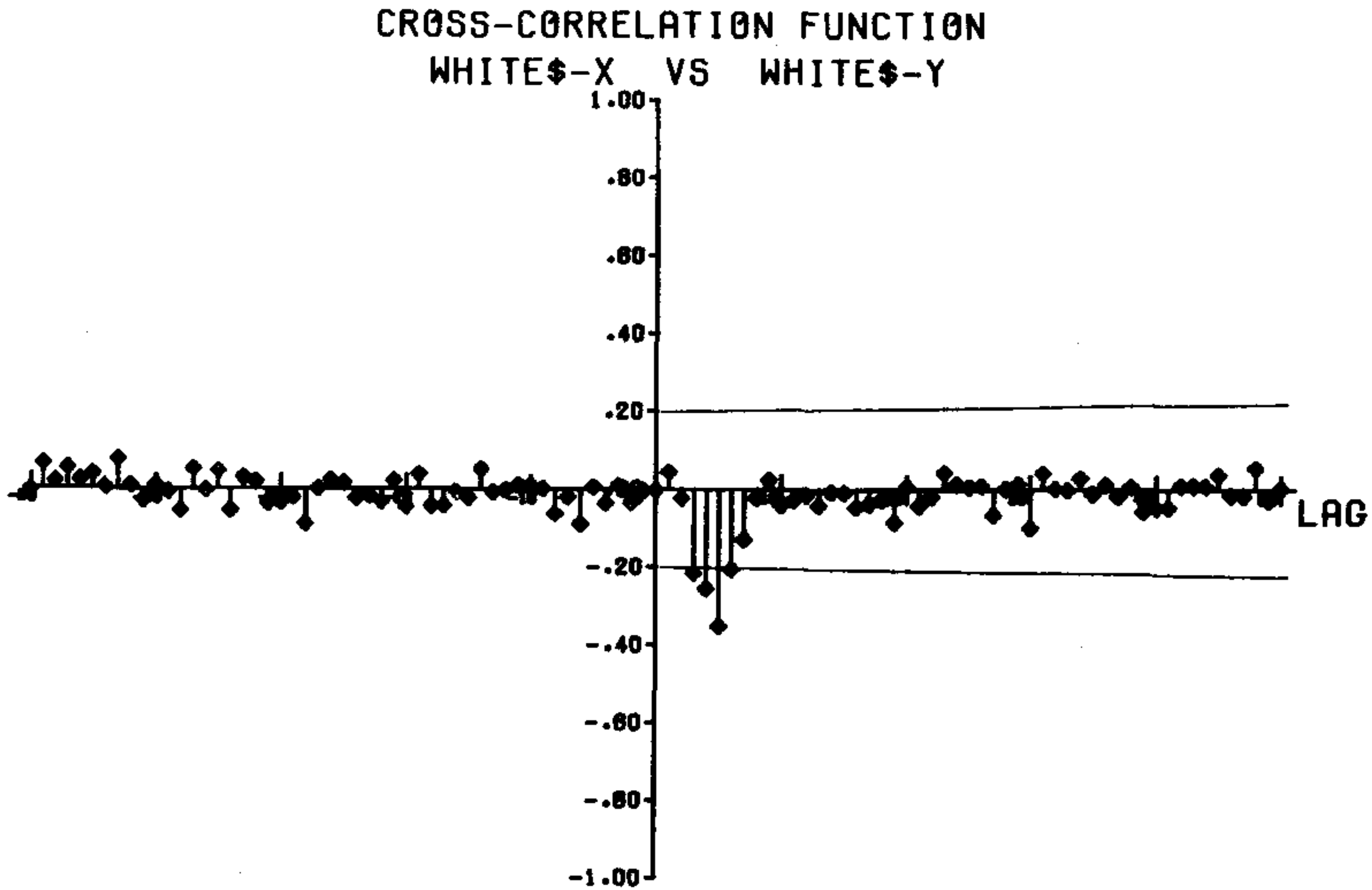

FIGURE 12.4 Estimated cross correlation function between the prewhitened input and similarly transformed output series. It is directly proportional to the impulse response function

13. PROCESS MODEL ESTIMATION

Once the structure of the process model has been identified using the facilities described in Chapters 11 and 12, the model parameters may be estimated. The estimation facility of the package utilises a recursive instrumental variable (IV) algorithm (Young, Shellswell and Neethling, 1971; and Moore, 1977, section 3.2).

The procedure uses an auxiliary model which is subjected to the same input series as the process to be estimated. Initial estimates of the auxiliary model parameters may be specified. If they are not specified, they are set to zero, and on the first pass through the data a least squares estimate is taken. Parameter estimates are refined over subsequent iterations.

The degree of confidence associated with the parameter estimates is reflected in the diagonal of a matrix, here termed the $\hat{P}$ matrix. Initially, this is set large, and as information contained in the data is incorporated in the parameter estimates, it decreases in magnitude.

After the specified number of iterations has been completed, a further pass through the data is made, with the $\hat{P}$ matrix set artificially high. This has the effect of amplifying the variation in the parameter estimates at each sample instant. These variations may be plotted to give an indication of model stability. Also over this final cycle, an estimate is made of the noise series using the parameter values calculated on the previous iteration. The noise series may be retained for subsequent analysis.

The output from the estimation includes a summary of the input data, values of the process and auxiliary model parameters at the end of each iteration, and the diagonal of the final $\hat{P}$ matrix, before it is set artificially high for the additional pass through the data. Also given is a summary of the estimated noise series, and the final parameter estimates, together with their standard errors.

13.1 Command Syntax

<u>ESTIMATE</u> <u>PROCESS</u> MODEL PARAMETERS OF SERIES list

<u>STRUCTURE</u> OF MODEL structure

[<u>SUBSCRIPTS</u> subscript-list]

[<u>SUBTRACT</u> <u>MEANS</u> FROM SERIES]

[<u>INITIAL</u> VALUES values-list]

[<u>ITERATIONS</u> iters]

[<u>NAME</u> NOISE SERIES name]

[{ <u>PLOT</u> / <u>PLOTTING</u> } { PARAMETER / PARAMETERS } [plot-list]]

13.2 The main sentence

ESTIMATE PROCESS MODEL PARAMETERS OF SERIES list

STRUCTURE OF MODEL structure

"list" consists of two series names, separated by blanks, a comma or the word AND. The first series specified is taken as the input series, the second is taken as the output series. The series must be the same length. The format of "structure" is

[{ AUTOREGRESSIVE / AR } { PARAMETER / PARAMETERS }] p [{ MOVING AVERAGE / MA } { PARAMETER / PARAMETERS }] q
[{ TIME DELAY / TD }] b

The values p and q are integers in the range

$$0 \leqslant p, q \leqslant 10$$

and indicate the number of autoregressive and moving average parameters of the model to be used.

The value b is an integer in the range

$$0 \leqslant b \leqslant 100$$

and indicates the pure time delay of the model.

All of p, q and b must be specified, with or without introductory words. If no introductory words are used, then the values must be given in the order

p q b

If introductory words are used, then the values may appear in any order (see example (iii), 13.4).

13.3 Qualifying phrases

13.3.1 SUBSCRIPTS subscript-list

This phrase may be used to indicate the subscripts of the parameters of the model. The format of "subscript-list" is

[{ AUTOREGRESSIVE / AR }] listll [{ MOVING AVERAGE / MA }] list2

"list1" contains integers, m_i, in the range $0 < m_i \leqslant 10$
"list2" contains integers, n_j, in the range $0 \leqslant n_j < 10$

Integers may be separated by spaces, commas, or the word AND.

The lengths of "list1" and "list2" are p and q respectively, where p and q are defined in 13.2. If the introductory words are omitted, "list1' must precede "list2", otherwise their order is unimportant. If either p or q is zero, then the corresponding subscript entry should be omitted (see example (iv), 13.4). The entire clause may be omitted, in which case the system will assume values for "list1" and "list2". The default subscript values are

1, 2, ... p for autroregressive parameters,
0, 1, ... q-1 for moving average parameters.

For example, if the values of p and q were 2 and 2, then the default subscripts would be

1 and 2 for the autoregressive parameters, and
0 and 1 for the moving average parameters.

The corresponding model would be, (assuming the time delay was zero)

$$y_t + \delta_1 y_{t-1} + \delta_2 y_{t-2} = \omega_o u_t + \omega_1 u_{t-1} + n_t$$

where u_t, y_t and n_t are the input, output and noise components respectively, at sample instant t.

13.3.2 <u>SUBTRACT</u> <u>MEANS</u> FROM SERIES

If this phrase is specified, the series to be used in the analysis will have their means removed, i.e. estimation will be performed on the deviations about the series means, rather than on the observed series themselves. In general, the means should be subtracted if the series have not been differenced.

13.3.3 <u>INITIAL</u> VALUES values-list

This phrase may be used to supply initial values for the auxiliary model parameters. The form of "values-list" is

[{ <u>AUTOREGRESSIVE</u> / <u>AR</u> }] list3 [{ <u>MOVING</u> <u>AVERAGE</u> / <u>MA</u> }] list4

"list3" and "list4" are lists of numbers having lengths p and q respectively (p and q are defined in 13.2). Commas, spaces or the word AND may be used to separate individual numbers. The order of the values within the lists is taken to be the same as the order of the subscripts in "list1" and "list2" (13.3.1). For example, if the second order moving average model

$$y_t = \omega_o u_t + \omega_1 u_{t-1} + n_t$$

is being used, and the subscripts clause is given as

SUBSCRIPTS MA 1 0

then on the INITIAL VALUES clause, the value ω_1 would have to precede the value of ω_o, i.e.

INITIAL VALUES MA ω_1 ω_o

The introductory words to "list3" and "list4" may be omitted but in this case "list3" must appear before "list4"; ordering is otherwise unimportant. If either p or q is zero, then the corresponding INITIAL VALUES entry should be omitted (see example (iv), 13.4).

The entire phrase may be omitted, in which case the initial values are set to zero and a least squares estimate is computed on the first pass through the data. No least squares estimate is taken if the phrase is included, even if the values presented are all zero.

13.3.4 <u>ITERATIONS</u> iters

This phrase may be used to specify the number of iterations required. "iters" is an integer greater than zero.

If the phrase is omitted, "iters" takes its value from the most recently given ITERATIONS clause. If no such clause has been given, it is assumed equal to 10.

13.3.5 <u>NAME</u> NOISE SERIES name

If the estimated noise series is required for subsequent analysis, it must be named using this clause. "name" is the variable name to be associated with the noise series (see 6.2 for construction of variable names). If the phrase is omitted, the noise series will be discarded at the end of the analysis.

13.3.6 { <u>PLOT</u> / <u>PLOTTING</u> } { PARAMETER / PARAMETERS } [plot-list]

By including this phrase the artificially enhanced parameter variations may be plotted to give an indication of the stability of the model. The form of "plot-list" is

{ <u>AUTOREGRESSIVE</u> / <u>AR</u> } list5 { <u>MOVING AVERAGE</u> / <u>MA</u> } list6

"list5" and "list6" consist of subsets of the integers in "list1" and "list2" (13.3.1) respectively (or their default values if no SUBSCRIPTS clause is given). Spaces, commas and the word AND may be used as separators. The variation of each parameter selected is plotted on the graphical device in use. A restriction in the use of this phrase is that the introductory words to "list5" and "list6" must be present. However, if no variations of autoregressive parameters are required, the entire entry should be omitted. The same holds for moving average parameters. The order of the entries (if both are present) is unimportant. If the "plot-list" option is not used, variations of all the estimated parameters will be plotted.

Examples of the plots are shown in Figures 13.1 and 13.2.

13.4 Examples

(i) ESTIMATE PROCESS MODEL PARAMETERS OF SERIES X AND Y STRUCTURE
OF MODEL AUTOREGRESSIVE PARAMETERS 2 MOVING AVERAGE
PARAMETER 1 TIME DELAY 10.

(ii) ESTIMATE PROCESS MODEL SERIES RAIN FLOW STRUCTURE 2 2 5
SUBSCRIPTS 1 2 0 1 INITIAL VALUES -1.3 0.4 0.1 0.1
NAME NOISE SERIES NOISE-1 PLOTTING

(iii) ESTIMATE PROCESS GASIN GASOUT
STRUCTURE TD 3 MA 3 AR 1
SUBTRACT MEANS FROM SERIES ITERATIONS 10
PLOT PARAMETERS AR 1 MA 0 AND 1

(iv) ESTIMATE PROCESS MODEL PARAMETERS SERIES AA AND BB
STRUCTURE OF MODEL AR 0 MA 2 TD 1
SUBSCRIPTS MOVING AVERAGE 0 AND 1 SUBTRACT MEANS
INITIAL VALUES MA 0.6 0.35
NAME NOISE-AB PLOTTING PARAMETER MA 1

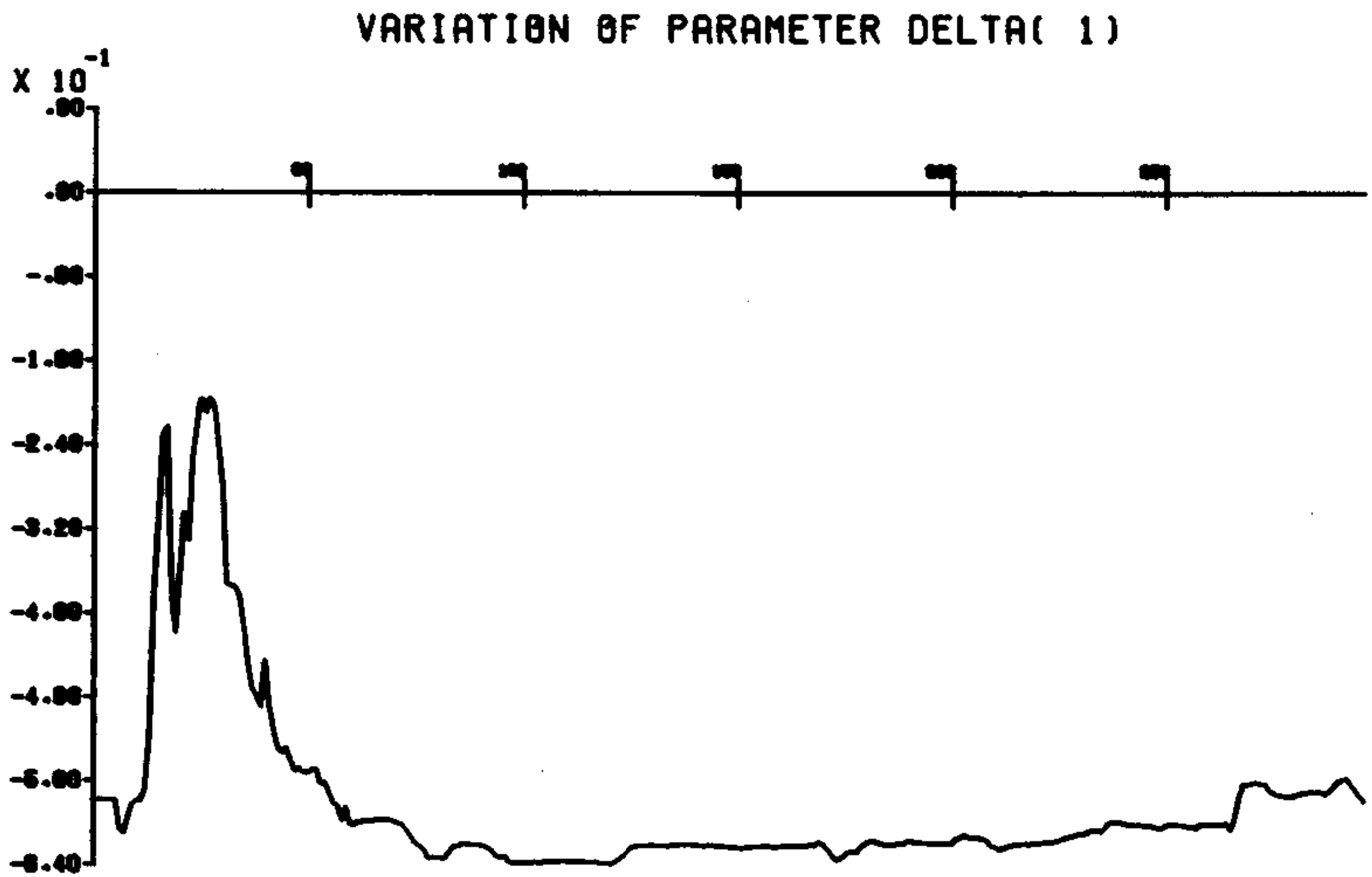

FIGURE 13.1 Variation of parameter δ_1 plotted over one pass through the data

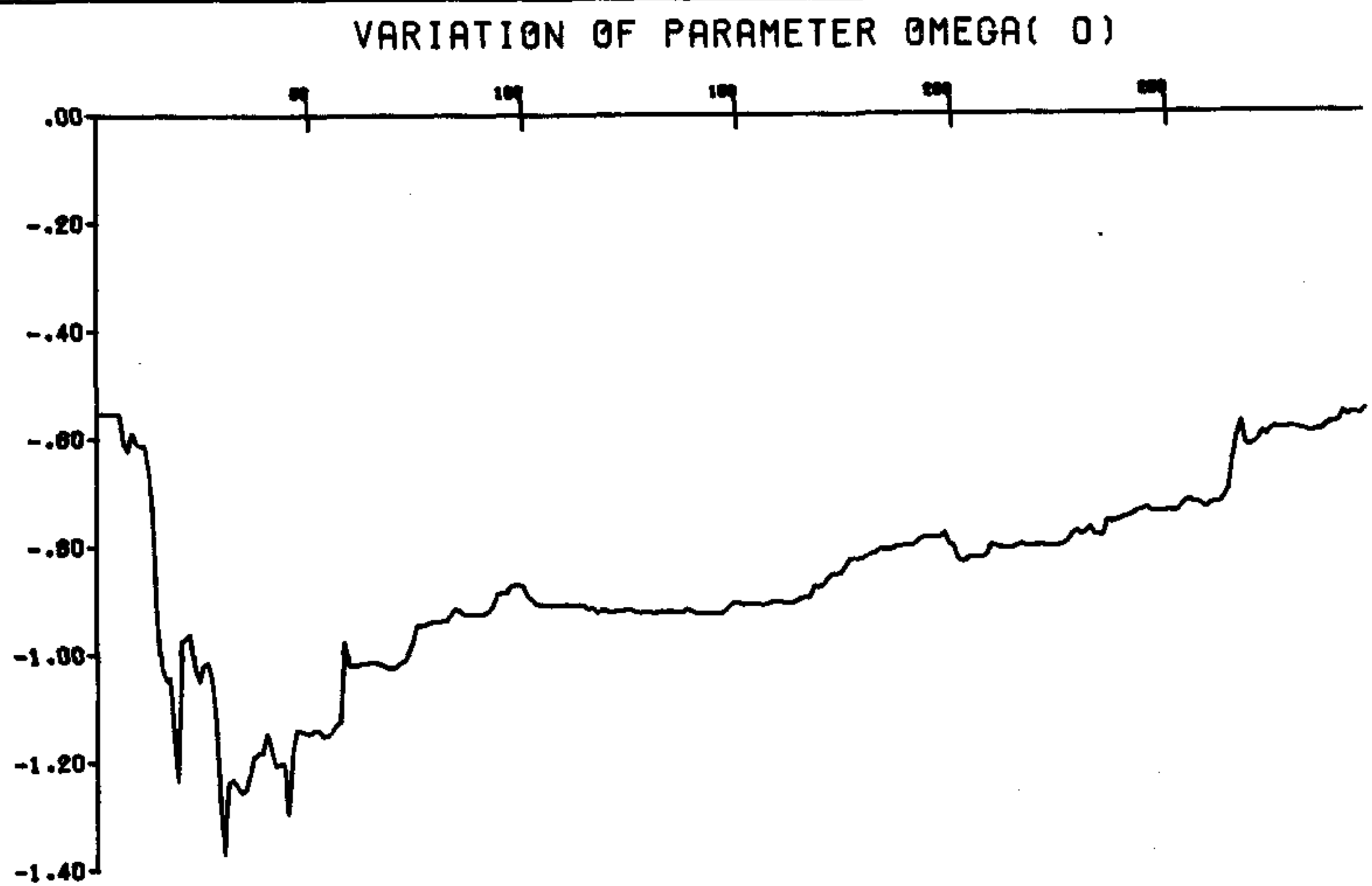

FIGURE 13.2   Variation of parameter ω_0 over one pass through the data

14. UNIVARIATE MODEL ESTIMATION

The univariate model estimation facility may be used to estimate the parameters of either a stochastic model, or the noise model component of a transfer function model. Both models relate a given series to white noise. In the former case, the series is an observed sequence whereas in the latter it is the noise series generated previously during process model estimation (as detailed in Chapter 13).

The univariate model parameters are estimated by an iterative procedure which incorporates a recursive approximate maximum likelihood (AML) algorithm (Young *et al*, 1971; and Moore, 1977 section 3.3). An option is available by which the model parameters may be set to some initial values.

The degree of confidence associated with the parameter estimates is reflected in the diagonal of a matrix (the $\hat{P}$ matrix). Initially, it is set large and, as information contained in the data is used in the estimation of the parameters, it diminishes.

The estimation proceeds for a specified number of iterations, after which an additional pass is made through the data with the P matrix set artificially high. This tends to amplify the variation of the parameter estimates. These variations may be plotted and used as an indication of the stability of the model. Also on this final pass, an estimate is made of the error, or residual series. If the identified model structure is adequate, this series should resemble white noise. It may be retained for subsequent diagnostic checking.

Output from the analysis includes a summary of the input data, the estimates of the model parameters at the end of each iteration, and the final diagonal of the $\hat{P}$ matrix before the additional pass is made. Also printed is a summary of the estimated error series and the final parameter values, together with their associated standard errors.

14.1 Command syntax

```
ESTIMATE { NOISE      } MODEL PARAMETERS OF SERIES name 1
         { STOCHASTIC }

         STRUCTURE OF MODEL structure

[SUBSCRIPTS subscript-list]

[SUBTRACT MEAN FROM SERIES]

[INITIAL VALUES  values-list]

[ITERATIONS   iters]
```

[NAME ERROR SERIES name2]

[{ PLOT / PLOTTING } { PARAMETER / PARAMETERS } plot-list]

14.2 The main sentence

ESTIMATE { NOISE / STOCHASTIC } MODEL PARAMETERS OF SERIES $name_1$

STRUCTURE OF MODEL structure

"name1" is the name of the series to be used in the estimation. Only one name may be given. The form of "structure" is

[{ AUTOREGRESSIVE / AR } { PARAMETER / PARAMETERS }] r [{ MOVING AVERAGE / MA } { PARAMETER / PARAMETERS }] s

The values r and s are integers in the range

$$0 \leqslant r, s \leqslant 10$$

and indicate the number of autoregressive and moving average parameters of the identified model. Both r and s must be specified, with or without introductory words. If no introductory words are used, then the values must be given in the order

r s

The ordering is immaterial if introductory words are used.

14.3 Qualifying phrases

14.3.1 SUBSCRIPTS subscript-list

This phrase may be used to indicate the parameter subscripts of the model to be estimated. The format of "subscript-list" is

[{ AUTOREGRESSIVE / AR }] list1 [{ MOVING AVERAGE / MA }] list2

"list1" and "list2" contain integers m_i and n_j in the range

$$0 < m_i, n_j \leqslant 10$$

and have lengths r and s respectively (see above). Spaces, commas and the word AND may be used to separate the integers. The introductory words may be omitted, in which case "list1" must precede "list2", otherwise their order is unimportant. If either r or s is zero, then the corresponding subscript entry should be omitted (see example (ii), 14.4). If the entire clause is omitted, the system will assume values for "list1" and "list2". The default settings are

1, 2, ... r   for autoregressive parameters

1, 2, ... s   for moving average parameters

14.3.2 SUBTRACT MEAN FROM SERIES

By including this phrase the mean will be removed from the series to be used in the analysis, ie:estimation takes place using the deviations about the series mean rather than on the series itself. In general, when a stochastic model is fitted, a series should have its mean removed if it has not been differenced; however, the mean should not be subtracted from noise series.

14.3.3 INITIAL VALUES values-list

The model parameters may be set to some initial values by including this clause. "values-list" has the form

[{ AUTOREGRESSIVE / AR }] list3   [{ MOVING AVERAGE / MA }] list4

"list3" and "list4" contain lists of numbers and have lengths r and s respectively (where r and s are as defined in 14.2). Individual numbers may be separated by blanks, commas or the word AND. The order of the values within the lists is taken to be the same as the order of the subscripts in "list1" and "list2" (14.3.1). For example, if the second order autoregressive model

$$z_t + \phi_1 z_{t-1} + \phi_2 z_{t-2} = a_t$$

is to be estimated, and

SUBSCRIPTS AR 2 1

has been specified, then the value of ϕ_2 will have to precede the value of ϕ_1, ie:

INITIAL VALUES ϕ_2 ϕ_1

The introductory words to "list3" and "list4" may be omitted, in which case "list3" must appear before "list4". Otherwise, the ordering is unimportant. The corresponding INITIAL VALUES entry should be omitted if either r or s is zero (see example (ii), 14.4).

Initial values of zero will be assumed if the phrase is omitted entirely.

14.3.4 ITERATIONS iters

The number of iterations required may be set by this phrase. "iters" is an integer, and must be greater than zero. The phrase may be omitted, in which case "iters" will assume its most recent setting. If no such declaration has been made previously, it is set to 10.

14.3.5 NAME ERROR SERIES name2

The estimated error or residual series may be retained for subsequent examination by including this phrase. "name2" gives the variable name to be associated with the error series (see 6.2 for construction of variable names). If the phrase is not included the series will be lost at the conclusion of the analysis.

14.3.6 {PLOT | PLOTTING} {PARAMETER | PARAMETERS} [plot-list]

Including this phrase will cause the artificially enhanced parameter variations to be plotted on the graphical device in use. They may then be used as an indication of model stability. The format of "plot-list" is

{AUTOREGRESSIVE | AR} list5 {MOVING AVERAGE | MA} list6

"list5" and "list6" contain subsets of the integers in "list1" and "list2" (14.3.1) respectively, or their default values if no SUBSCRIPTS clause is given. Spaces, commas and the word AND may be used as separators. Note that there is a restriction in the use of this clause, in that the introductory words to "list5" and "list6" must be present. However, if no variations of autoregressive parameters are required, the entire entry should be omitted; the same is true for moving average parameters. If both entries are present they may appear in any order.

Variations of all the estimated parameters will be plotted if the "plot-list" option is omitted.

Figures 14.1 and 14.2 are examples of graphs produced of the parameter variations.

14.4 Examples

(i) ESTIMATE NOISE MODEL PARAMETERS OF SERIES NOISE-1
STRUCTURE OF MODEL AUTOREGRESSIVE PARAMETER 1
MOVING AVERAGE PARAMETERS 0

(ii) ESTIMATE STOCHASTIC MODEL SERIES SUNSPOTS
STRUCTURE AR 3 MA 0
SUBSCRIPTS AR 1 2 3 SUBTRACT MEAN FROM SERIES
INITIAL VALUES - 1.5 1.0 - 0.2
NAME ERROR SERIES SUNERROR PLOTTING

(iii) ESTIMATE STOCHASTIC MODEL PARAMETERS SERIES STOCK
STRUCTURE OF MODEL 1 1 SUBTRACT MEAN
ITERATIONS 10 PLOT PARAMETER AR 1

(iv) ESTIMATE NOISE MODEL SERIES NOISE-AB
STRUCTURE AR 2 MA 1

```
INITIAL VALUES AUTOREGRESSIVE - 1.3 AND 0.5
   MOVING AVERAGE 0.35      ITERATIONS  8
NAME  ERROR-AB
PLOTTING PARAMETERS AR  1  AND  2   MA 1
```

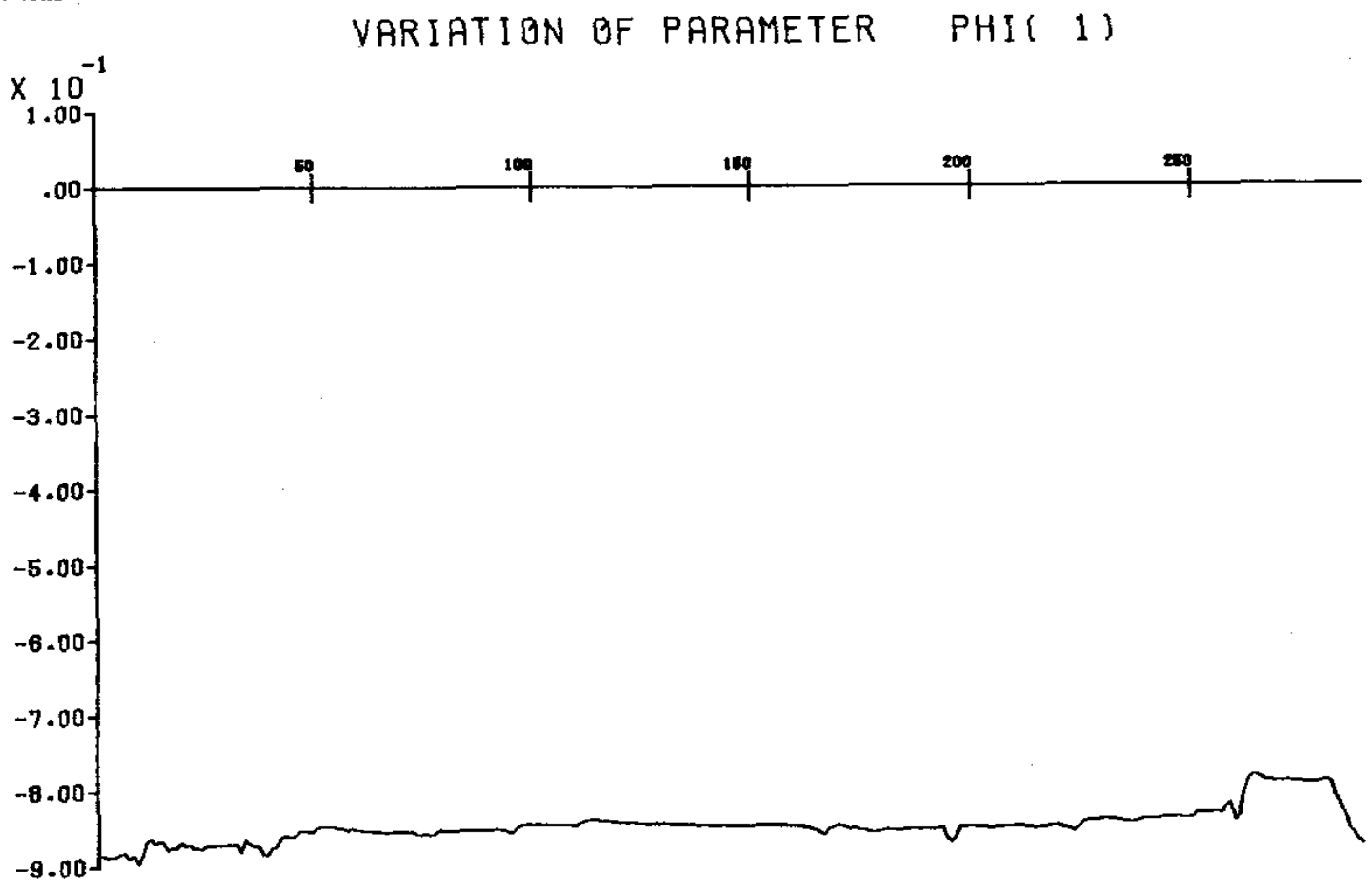

FIGURE 14.1 Variation of parameter ϕ_1 during noise model estimation, plotted over one pass through the data

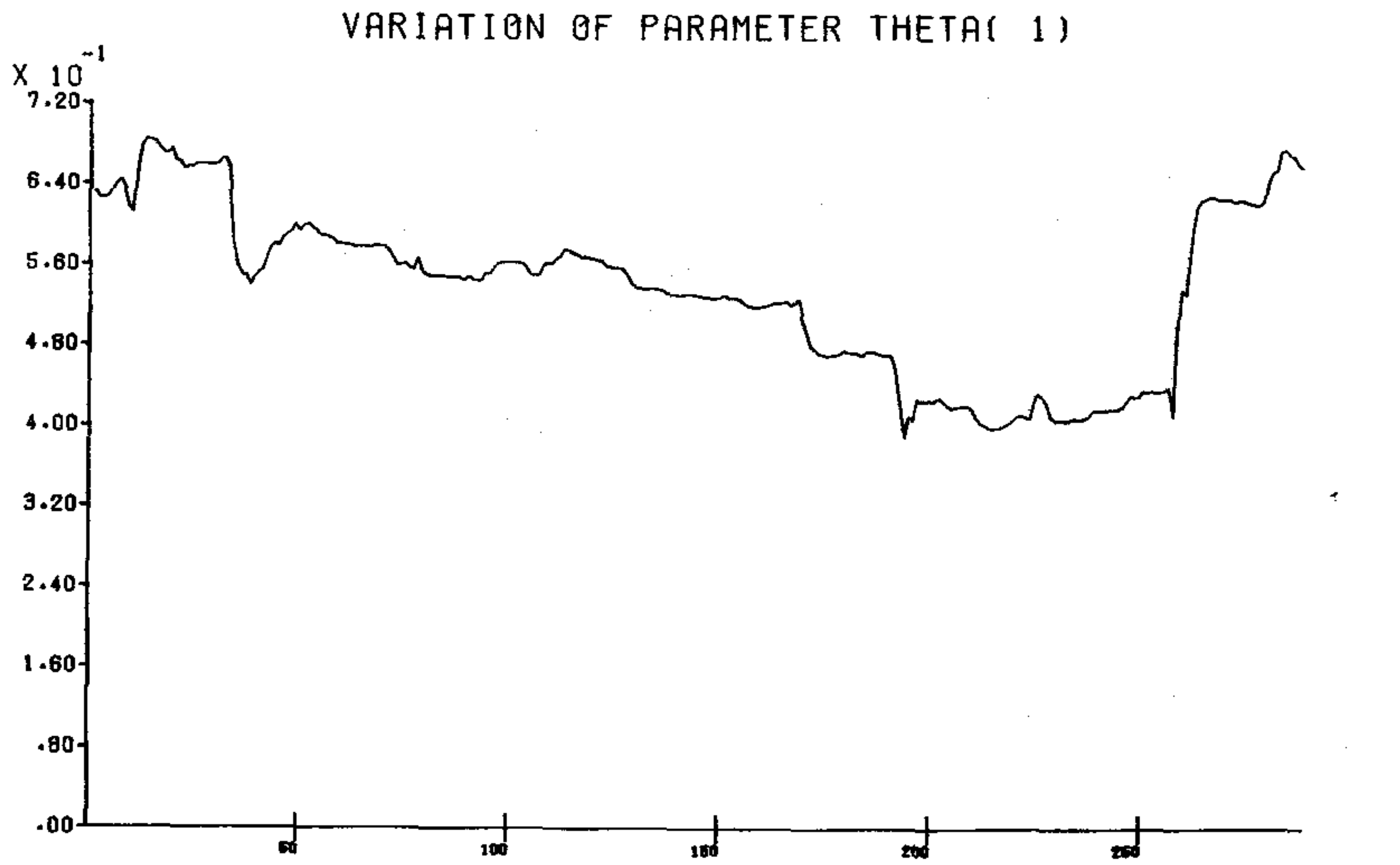

FIGURE 14.2 Variation of parameter θ_1 during noise model estimation, plotted over one pass through the data

15. INVESTIGATING PARAMETER VARIATIONS

If a trend or a seasonal variation is observed in the data, efforts should be made to remove it. Facilities exist in the package whereby series may be differenced or seasonally differenced to induce stationarity but this often has the adverse effect of amplifying the noise component, and ideally, alternative means should be sought. The INVESTIGATE facility can be used in situations such as this to examine the way in which the parameters of a model vary over time. It may thus be employed to investigate the efficacy of different transformations of the series to reduce the resulting model to time invariant parameter form.

The facility utilises extensions of the IV-AML algorithms which allow the parameters to vary according to a random walk model (Young *et al*, 1971). This involves a covariance matrix Q of the random parameter variations between samples. The expected rate of variation of the parameters between samples should be reflected in the initial choice of the diagonal elements of Q; for example, a parameter known to be time invariant should have its corresponding diagonal element set to zero. In general, the Q-matrix diagonal may be set to approximately one tenth of the corresponding diagonal elements of the final $\hat{P}$-matrix obtained from model estimation. However, this is only a rough guide, and experimental adjustment may be needed.

An illustration of the use of the INVESTIGATE facility may be found in Moore (1977), section 3.4.

The following are required as input to the analysis:

(i) the appropriate series

(ii) the model structure, and parameter estimates

(iii) the P-matrix and Q matrix diagonals

Only one pass is made through the data. Print output consists of the following:

(i) summary of the input data

(ii) the final parameter estimates

(iii) summary of the estimated noise/error series

(iv) diagonal of the final $\hat{P}$-matrix

(v) (optionally) the estimates at each sample instant of all or a subset of the parameters.

The parameter variations may also be plotted on the graphical device in use. On the pass through the data, an estimate of the noise/error series is made. This series may be retained for later analysis.

Both univariate models and transfer functions may be examined by the INVESTIGATE facility.

15.1 Command syntax

```
     INVESTIGATE  PARAMETER VARIATIONS USING SERIES list

                  STRUCTURE  OF MODEL  structure

    [SUBSCRIPTS   subscript-list]

     INITIAL   VALUES  values-list1

    [            (MEAN )                ]
    [SUBTRACT    (MEANS)    FROM SERIES ]

     P-MATRIX  DIAGONAL values-list2

     Q-MATRIX  DIAGONAL values-list3

    [         (ERROR)                 ]
    [NAME     (NOISE)   SERIES   name ]

    [(PRINT   )   (PARAMETER )             ]
    [(PRINTING)   (PARAMETERS)   print-list]

    [(PLOT    )   (PARAMETER )             ]
    [(PLOTTING)   (PARAMETERS)   plot-list ]
```

Note that the qualifying phrases beginning with the words

```
    INITIAL
    P-MATRIX
and Q-MATRIX
```

are not optional, ie: they must be present in the command.

15.2 The main sentence

```
INVESTIGATE PARAMHTER VARIATIONS USING SERIES list
     STRUCTURE OF MODEL structure
```

"list" consists of one or two series names. If two are given, they may be separated by blanks, a comma, or the word AND. If the model under investigation is a univariate model only one name must be given. Two names are required if the model is a transfer function, in which case the first is taken as the input series and the second is taken to be the output series. They must be the same length. The format of "structure" is

```
[ { AUTOREGRESSIVE   { PARAMETER  } } ]  p   [ { MOVING AVERAGE { PARAMETER  } } ] q
[ { AR               { PARAMETERS } } ]      [ { MA             { PARAMETERS } } ]

                    [ { TIME DELAY } ]   b
                    [ { TD         } ]
```

The values p and q are integers in the range

$0 \leqslant p, q \leqslant 10$

and indicate the number of autoregressive and moving average parameters of the model to be used. Both values must be given. The value b indicates the time delay associated with the model and is an integer in the range

$0 \leqslant b \leqslant 100$

The TIME DELAY entry must only be used if the model under investigation is a transfer function (ie: two series are specified).

The values may be given with or without their introductory words. If no introductory words are used the values must be given in the order

p q for univariate models

p q b for transfer functions

The order does not matter if introductory words are present.

15.3 Qualifying phrases

Note that the qualifying phrases beginning with the words

INITIAL
P-MATRIX
Q-MATRIX

must be *present in the instruction*.

15.3.1 SUBSCRIPTS subscript-list

This phrase may be used to indicate the subscripts of the parameters of the model under investigation. "subscript-list" has the form

```
[ { AUTOREGRESSIVE } ]  list1   [ { MOVING AVERAGE } ]  list2
[ { AR             } ]          [ { MA             } ]
```

For univariate models,

"list1" contains integers m_i in the range $0 < m_i \leq 10$
"list2" contains integers n_j in the range $0 < n_j \leq 10$

For transfer functions,

"list1" contains integers m_i in the range $0 < m_i \leq 10$

"list2" contains integers n_j in the range $0 \leq n_j < 10$

Spaces, commas and the word AND may be used to separate the integers. The lengths of "list1" and "list2" must be p and q respectively, where p and q are as defined in 15.2. If the introductory words are omitted, "list1" must precede "list2", otherwise ordering is unimportant. If either p or q is zero, the corresponding subscript entry should be omitted. If the entire clause is omitted, the system will assume values for "list1" and "list2". The default settings are

for univariate models	1, 2, ... p	for autoregressive parameters
	1, 2, ... q	for moving average parameters
for transfer functions	1, 2, ... p	for autoregressive parameters
	0, 1, ... q-1	for moving average parameters

15.3.2 INITIAL VALUES values-list1

This phrase must be present in the instruction and is used to supply estimates for the model parameters. The form of "values-list1" is

[{ AUTOREGRESSIVE | AR }] list3 [{ MOVING AVERAGE | MA }] list4

"list3" and "list4" are the lists of parameter values, and consist of numbers separated by blanks, commas or the word AND. They must be of lengths p and q respectively (p and q as defined in 15.2). The order of the values within the lists is taken to be the same as the order of the subscripts given in "list1" and "list2" (15.3.1). This is the case regardless of whether the SUBSCRIPTS clause is specified or default subscripts are assumed. Examples may be found in 13.3.3 (transfer functions), and 14.3.3 (univariate models).

The introductory words to the lists may be omitted but in this case "list3" must precede "list4" otherwise, ordering is unimportant. The corresponding INITIAL VALUES entry should be omitted if either p or q is zero (see example (i), 15.4)

15.3.3 SUBTACT { MEAN | MEANS } FROM SERIES

If this phrase is included, the series to be used in the analysis will have their means removed, ie: the analysis will be performed on the deviations about the series means, rather than on the series themselves. In general, the means should be subtracted if the series have not been differenced.

15.3.4 P-MATRIX DIAGONAL values-list2

Q-MATRIX DIAGONAL values-list3

These two phrases will be treated together, since their structures are essentially the same. Both phrases must be present in the INVESTIGATE instruction.

The form of "values-list2 " is identical to the form of "values-list3 ". Both consist of p + q numbers separated by blanks, commas or the word AND (p and q as defined in 15.2). The numbers in the lists will be taken as the starting values for the diagonal of the $\hat{P}$ matrix, and the diagonal values of the Q matrix. The first p values in each list correspond to the autoregressive parameters, the last q correspond to the moving average parameters. The order of the values in the lists is taken to be that given on the SUBSCRIPTS clause (15.3.1), whether specified or assumed. Thus, if a transfer function was under investigation, and

SUBSCRIPTS AR 2 1 MA 0 1

and P-MATRIX DIAGONAL p_1 p_2 p_3 p_4

were given, then the model parameters and their corresponding diagonal elements of the $\hat{P}$-matrix would be as follows:

	model parameter	diagonal element
autoregressive terms	δ_1	p_2
	δ_2	p_1
moving average terms	ω_0	p_3
	ω_1	p_4

15.3.5 NAME {ERROR / NOISE} SERIES name

On the pass through the data, an estimate is made of the noise series (or error series if the model under investigation is a univariate model). This series may be retained by including the NAME clause. "name" gives the variable name to be associated with the series (see 6.2 for construction of variable names). If the phrase is not included, the series will be lost at the conclusion of the analysis.

15.3.6 {PRINT / PRINTING} {PARAMETER / PARAMETERS} [print-list]

{ PLOT / PLOTTING } { PARAMETER / PARAMETERS } [plot-list]

Since the structures of these two phrases are identical, they will be treated together. The phrases control the output of the parameter variations. The PRINTING phrase directs listings of the variations to the appropriate streams, whereas the PLOTTING phrase causes them to be displayed graphically. A maximum of one phrase of each type may be included in any instruction.

If the PRINTING phrase is used without the "print-list" option, the variations of all the model parameters will be printed. Similarly, if the PLOTTING phrase is used without the "plot-list" option, the variations of all the parameters will be plotted.
"print-list" and "plot-list" have the same form, ie:

{ AUTOREGRESSIVE / AR } [$list_5$] { MOVING AVERAGE / MA } [list6]

"list5 " and "list6 " consist of subsets of the subscripts given in "list1 " and "list2 " (15.3.1) respectively (or their default values if no subscripts clause is given). Spaces, commas and the word AND may be used as separators. The variations of the parameters whose subscripts are specified will be output on the appropriate devices. If the variations are not required for any autoregressive parameters, then the list of subscripts ("list5 ") and the introductory word should be omitted. However, if AUTOREGRESSIVE (or AR) is specified, but no subscripts are given, the variations of all the autoregressive parameters will be output. The same rules apply for the specification of moving average parameters.

An example will serve to clarify the rules. Consider a transfer function with two autoregressive parameters (subscripts 1 and 2), and two moving average parameters (subscripts 0 and 1). To print the variations of all four parameters, any of the following could be used

```
PRINTING
PRINTING AR MA
PRINTING AR 1  2  MA  0  1
```

To plot the variations of just the moving average parameters, either

```
    PLOTTING MA
or  PLOTTING MA  0  1
```

could be specified.

15.4 Examples

(i) INVESTIGATE PARAMETER VARIATIONS SERIES RAIN STRUCTURE OF
MODEL AR 2 MA 0
INITIAL VALUES AUTOREGRESSIVE -1.31 0.57
SUBTRACT MEAN FROM SERIES

```
      P-MATRIX  DIAGONAL  15.0  12.0
      Q-MATRIX  DIAGONAL   1.2   1.0
      PRINTING PARAMETERS  AR  1  2

(ii)  INVESTIGATE VARIATIONS USING SERIES RAINFALL AND RUNOFF
      STRUCTURE 2  2  5   SUBSCRIPTS  AR  1  2  MA  0  1
      INITIAL VALUES  AR   -1.3  0.5    MA  0.02  0.01
      P-MATRIX   35.0  30.0   0.1   0.1
      Q-MATRIX    0.0   0.0   0.01  0.01
      NAME NOISE SERIES RNOISE
      PRINTING
      PLOTTING  AR  1  MA

(iii) INVESTIGATE PARAMETER VARIATIONS USING SERIES  X  Y
      STRUCTURE OF MODEL  AR  1  MA  3  TD  3
      INITIAL VALUES   -0.59   -0.59   -0.35  -0.35
      SUBTRACT MEANS
      P-MATRIX    23.0  15.0  12.5  12.5
      Q-MATRIX     2.0   1.1   1.0   1.1
      PLOT AUTOREGRESSIVE  1  MOVING AVERAGE  0  1  AND  2
```

16. FORECASTING (1) - SETTING UP THE MODEL

Before any forecasting can be done, the model to be used must be set up. This involves giving the system details of the model structure together with parameter values. A model introduced to the system in this way must be given a name by the user. Once set up the model can then be referenced by its name in other analyses.

16.1 Command syntax

```
   SET UP MODEL NAME          name
┌                                              ┐
│  PROCESS  MODEL  STRUCTURE  structure1       │
│ [SUBSCRIPTS  subscript-list1]                │
│ {VALUE  }  values-list1                      │
│ {VALUES }                                    │
│ [MEANS     value1  value2]                   │
│  NOISE MODEL STRUCTURE       structure2      │
│ [SUBSCRIPTS subscript-list2]                 │
│ {VALUE  }                                    │
│ {VALUES }  values-list2                      │
└                                              ┘
┌                                              ┐
│   STOCHASTIC MODEL STRUCTURE structure3      │
│  [SUBSCRIPTS subscript-list3]                │
│  {VALUE  }values-list3                       │
│  {VALUES }                                   │
│  [MEAN    value3]                            │
└                                              ┘
[OUTPUT ON STREAM number ]
```

If the model has been set up previously and saved to disc using the OUTPUT phrase, it may be recreated in this new run of the package by:

```
SET UP MODEL NAME  name
INPUT  ON STREAM  number
```

16.2 The main sentence

```
SET UP MODEL NAME name
```

This tells the package that a model is to be set up with name "name". The model can then be referenced in subsequent analyses by referring to it using its name. The construction of "name" follows the same rules as the construction of series names. These rules are given in 6.2.

16.3 Qualifying phrases

16.3.1

```
{PROCESS   }
{NOISE     }   MODEL STRUCTURE structure
{STOCHASTIC}
```

PROCESS and NOISE model phrases are required if a transfer function model is being used to forecast an output series; and if the input series is to be forecast the STOCHASTIC model phrase is required. The form of the list depends on whether PROCESS, NOISE or STOCHASTIC is specified.

For a PROCESS model the form of "structure" is

$$\left\{\begin{matrix}\text{AUTOREGRESSIVE}\\ \text{AR}\end{matrix}\right\} \; p \quad \left\{\begin{matrix}\text{MOVING AVERAGE}\\ \text{MA}\end{matrix}\right\} \; q \quad \left\{\begin{matrix}\text{TIME DELAY}\\ \text{TD}\end{matrix}\right\} \; b$$

whereas for NOISE and STOCHASTIC models the "structure" is

$$\left\{\begin{matrix}\text{AUTOREGRESSIVE}\\ \text{AR}\end{matrix}\right\} \; p \quad \left\{\begin{matrix}\text{MOVING AVERAGE}\\ \text{MA}\end{matrix}\right\} \; q$$

ie. a time delay is not applicable.

In the lists above p and q are integers in the range $0 \leqslant p, q \leqslant 10$ specifying the number of autoregressive and moving average parameters respectively. b is the time delay, which must be $\leqslant 100$. The integers p, q and b may be present with or without the introductory words (bracketed above), but if the introductory words are omitted then the list must be given in the order p, q, (b). If the introductory words are present the ordering is irrelevant.

16.3.1.1 <u>SUBSCRIPTS</u> subscript-list

"subscript list" in this phrase has the form

$$\left\{\begin{matrix}\text{AUTOREGRESSIVE}\\ \text{AR}\end{matrix}\right\} \; \text{list1} \quad \left\{\begin{matrix}\text{MOVING AVERAGE}\\ \text{MA}\end{matrix}\right\} \; \text{list2}$$

where 'list1" and "list2" are lists of integers defining the subscripts of the appropriate parameters. The SUBSCRIPTS qualifying phrase may be omitted, in which case the default values of the subscripts are set. These default values again depend on whether a PROCESS, NOISE or STOCHASTIC model is being considered. For a PROCESS model "list1" takes the values 1,2,3 ... p for the autoregressive subscripts, and 0,1,2, ... q-1 for the moving average subscripts; however for NOISE and STOCHASTIC models the lists take the form 1,2,3, ... $\left\{\begin{matrix}p\\ q\end{matrix}\right\}$

Again the introductory words may be omitted from the SUBSCRIPTS list but in this case the autoregressive subscripts must precede the moving average subscripts.

16.3.1.2 $\left\{\begin{matrix}\underline{\text{VALUE}}\\ \underline{\text{VALUES}}\end{matrix}\right\}$ value-list

"value-list" in this phrase has the form

$$\left\{\begin{matrix}\text{AUTOREGRESSIVE}\\ \text{AR}\end{matrix}\right\} \; \text{list3} \quad \left\{\begin{matrix}\text{MOVING AVERAGE}\\ \text{MA}\end{matrix}\right\} \; \text{list4}$$

where "list3" and "list4" are lists of real numbers which are the values of the particular model parameters. Again the introductory words in the list may be omitted, with the normal restrictions as noted above.

16.3.1.3 {MEAN / MEANS} list

This phrase allows mean values to be specified for the models. The form of "list" depends on the model under consideration. For a PROCESS model "list" has the form:

value1 value2

where "value1" refers to the input series and "value2" to the output series. For a STOCHASTIC model the form of "list" is:

value1

which is the mean for the stochastic model.

The phrase is not needed for the NOISE model specifications - in fact its use will result in an error.

The rules governing the necessity for a MEANS phrase (if applicable) are as follows:

(i) if the forecasting is to be carried out over a different period to that over which the parameters were estimated, then mean values should be specified.

(ii) if the parameters were estimated without means removed then the values should be set to zero

(iii) if the estimation was performed with means subtracted, and the period of the data for estimation and forecasting is the same, then the MEAN phrase need not be specified.

16.3.2 OUTPUT ON STREAM number

If the user wishes to keep the model just set up for a future run of the package, then this phrase will cause the model to be written out to stream "number". (See Appendix 3 for restrictions on the values "number" can take). The user should have previously assigned a file for this purpose and linked it to stream "number" by the @USE job control statement on the UNIVAC. In one run of the package many models (up to 10) may be written to the same stream, but between runs it is the responsibility of the user to remember which models are on which files. The first use of OUTPUT in a run of the package assumes a logically blank file. Hence the first use of this phrase in a run specifying a model file set up in a previous run will overwrite what is presently on the file.
Note also that this file should not be used to hold other data.

16.3.3 INPUT ON STREAM number

The use of this phrase presupposes that a model has been set up previously and saved using the OUTPUT phrase. This qualifying phrase replaces all those specified in sections 16.3.1 and 16.3.2. This phrase causes the package to search the file 'number' for a model whose name is given in the main sentence. On finding the model it is copied for use by this run of the package. A no-find situation is notified as an error to the user.

16.4 Examples

(i)
```
SET UP MODEL NAME MODEL-A
PROCESS STRUCTURE 2 2 5
SUBSCRIPTS MA 1 O AR 1 2
VALUES    1.35  -0.42  0.01  0.014
NOISE STRUCTURE  AR  2  MA 1
SUBSCRIPTS        AR 1 2  MA 1
VALUES           AR 1.75 -0.76 MA 0.05
OUTPUT STREAM  14
```

(ii)
```
SET UP MODEL NAME XYZ
STOCHASTIC MODEL STRUCTURE 1 1
SUBSCRIPTS    AR 1 MA 1
VALUES      0.75  0.01
MEAN        11.4
```

(iii)
```
SET UP NAME MOD1
INPUT STREAM 12
```

(iv)
```
SET UP MODEL NAME GASMODEL
PROCESS MODEL STRUCTURE AR 1 MA 3 TIME DELAY 3
VALUES MA -0.53 -0.37 -0.51 AR -0.57
MEANS  -5.68 53.5
NOISE STRUCTURE  AR  2  MA O
VALUES     -1.53   0.63
STOCHASTIC STRUCTURE AR 3 MA O
VALUES     -1.97   1.37  -0.34
MEAN       -5.68
OUTPUT ON STREAM 12
```

17. FORECASTING (2) - TESTING THE MODEL

The ultimate test of an estimated model is its ability to forecast accurately over the period of the data used for estimation and, more importantly, over a different period. The forecasting section of the package allows both of these tests to be made.

The forecasts are generated directly from the difference equation and are minimum mean square error forecasts. Thus, the forecasts are unbiased and the residuals (or random component) can be shown to be the one step ahead forecast errors (Box and Jenkins, Chapter 5). Three sorts of model may be used, namely, a stochastic model, a transfer function in which just the output series is forecast, and a combination of the two, in which both input and output series are forecast. The method whereby models may be set up to be used for forecasting is explained in Chapter 16.

Two types of forecast may be made.

Type I forecasts : these are predictions made at a fixed lead time over the whole length of a given series.

Type II forecasts : these are predictions made from a given origin for lead times 1, 2, 3, ..., up to a specified maximum.

Both types may be plotted on the graphical device in use and the error series associated with the forecasts may be retained for subsequent analysis. Examples of the two types are displayed in Figures 17.1 and 17.2.

17.1 Command syntax

```
FORECAST  USING SERIES list1  AND MODEL model-name
[type I specifications]
[type II specifications]
```

where "type I specifications" has the form

```
{ {PRINT   } }
{ {PRINTING} }  PREDICTIONS FOR LEAD {TIME } intlist1
{ {PLOT    } }                       {TIMES}
{ {PLOTTING} }
[ NO CONFIDENCE LIMITS ]
[ NAME  ERROR SERIES  list2 ]
```

and "type II specifications" has the form

```
  {PRINT    }
{ {PRINTING } }
{ {PLOT     } }  PREDICTIONS UP TO  intlist2  {STEP } AHEAD
  {PLOTTING }                                 {STEPS}

                {ORIGIN }  AT {POINT }  intlist3
                {ORIGINS}     {POINTS}

 [NO  CONFIDENCE  LIMITS]

 [NAME  ERROR SERIES list3 ]
```

17.2 The main sentence

```
 FORECAST  USING  SERIES  list1  AND  MODEL  model-name

 [type I specifications]

 [type II specifications]
```

"list1" consists of one or two names of series known to the system. If two are given, they may be separated by spaces, a comma, or the word AND. "model-name" is the name of a model which has been set up previously by the SET UP command (see Chapter 16). The appropriate number of series must be given for the type of model selected. For example, if the model is a transfer function, then two series must be specified, in which case the first is taken to be the input series and the second as the output series.

The "type I specifications" and "type II specifications" are explained in 17.3 and 17.4 respectively. Both of these specifications are optional. However, if neither of them is used the system will print type I forecasts for a lead time of 1, ie: the default is

PRINTING PREDICTIONS FOR LEAD TIME 1

Inclusion of one or both of these options will override the default.

If the model used is a transfer function in which both the input and the output series are forecast, then although the predictions of the input series will be printed, only the output series predictions will be plotted (if plotting is required).

17.3 Type I Specifications

```
  {PRINT    }
{ {PRINTING } }
{ {PLOT     } }  PREDICTIONS FOR LEAD {TIME }  intlist1
  {PLOTTING }                         {TIMES}
```

"intlist1" consists of a set of integers t_i in the range

$$1 \leq t_i \leq 110$$

separated by blanks, commas or the word AND. A maximum of five may be given. The integers give the lead times at which the type I forecasts are to be calculated.

If the PRINT (or PRINTING) option is used, the observed series, the forecast values for each lead time, and their associated 50% and 95% confidence limits are directed to the appropriate print streams. The PLOT (or PLOTTING) option includes the PRINT option, but in addition a graph is produced of the observed and forecast series for each specified lead time. The 50% and 95% confidence limits associated with the forecasts are also plotted.

17.4 Type II Specifications

$$\left\{ \begin{matrix} \left\{ \begin{matrix} \underline{\text{PRINT}} \\ \underline{\text{PRINTING}} \end{matrix} \right\} \\ \left\{ \begin{matrix} \underline{\text{PLOT}} \\ \underline{\text{PLOTTING}} \end{matrix} \right\} \end{matrix} \right\} \text{ PREDICTIONS } \underline{\text{UP}} \ \underline{\text{TO}} \text{ intlist2 } \left\{ \begin{matrix} \text{STEP} \\ \text{STEPS} \end{matrix} \right\} \text{ AHEAD}$$

$$\left\{ \begin{matrix} \underline{\text{ORIGIN}} \\ \underline{\text{ORIGINS}} \end{matrix} \right\} \text{ AT } \left\{ \begin{matrix} \text{POINT} \\ \text{POINTS} \end{matrix} \right\} \text{ intlist3}$$

"intlist3" consists of a maximum of 5 integers p_i, giving the starting points in the series from which type II forecasts are to be made. "intlist2" is a set of integers s_i which specify the corresponding lead times to be used. "intlist2" and "intlist3" must be the same length. The only restriction on the integers p_i and s_i is that they must satisfy

$$p_i + s_i \leq n$$

where n is the number of points in the series to be forecast. Blanks, commas and the word AND may be used to separate the integers in the lists.

Selection of the PRINTING option will cause the print output to be directed to the appropriate streams. For each specified origin, print output consists of the observed value, forecast value, forecast error, and the 50% and 95% confidence limits for each point up to the maximum lead time. The PLOTTING option includes the PRINTING option but in addition the results are displayed on the graphical device in use. A separate graph is produced for each origin specified, and the observed series, forecast values, and 50% and 95% confidence limits are plotted.

17.5 Qualifying phrases

There are two qualifying phrases, the NO LIMITS phrase, and the NAME phrase. They may be included in both "type I specifications" and "type II specifications" (refer to command syntax, 17.1)

17.5.1 NO CONFIDENCE LIMITS

Graphs sometimes become obscured if confidence limits are plotted. This is the case particularly when dealing with very long series or lead times, or if the confidence region is very narrow. In such situations it can be advantageous to omit the confidence limits from the graph. This may be done by including the above phrase. To suppress confidence limits from type I forecasts, the phrase must be included in "type I specifications" (17.1). Similarly, to suppress confidence limits from type II forecasts, the phrase must be included in "type II specifications" (17.1). Confidence limits are only suppressed from graphs, the print listings being unaffected by this phrase.

17.5.2 NAME ERROR SERIES list

To retain the error series for later analysis, they must be named by this clause. Type I forecast errors and type II forecast errors must be named separately. "list" is a set of names to be attached to the error series. Blanks, commas or the word AND may be used as separators between names. The number of names in "list" must be the same as the number of error series of the particular type that will be generated. For example, if

    PRINTING PREDICTIONS for LEAD TIMES 1 2 AND 4

is specified, then 3 names must be given for the Type I forecast errors.

17.6 Examples

(i) FORECAST USING SERIES SUNSPOTS AND MODEL SUNMODEL

(ii) FORECAST USING RAINFALL AND RUNOFF MODEL MOD-1A PLOTTING
PREDICTIONS FOR LEAD TIME 1
NAME ERROR SERIES E-1
PLOTTING PREDICTIONS UP TO 50 STEPS AHEAD ORIGIN AT
POINT 70 NAME ERROR SERIES E-50

(iii) FORECAST USING SERIES GASIN GASOUT MODEL GASMODEL
PRINTING LEAD TIMES 1 2 3 AND 4
NAME G1 G2 G3 G4
PLOTTING UP TO 30 50 50 ORIGINS 150 206 245
NO CONFIDENCE LIMITS NAME G5 G6 G7

(iv) FORECAST RAINFALL RUNOFF MODEL BESTMOD PLOT LEAD TIMES
12 24 48 NO LIMITS PRINT UP TO 24 48 STEPS AHEAD
ORIGINS AT POINTS 20 36 NAME ERROR SERIES E24 and E48.

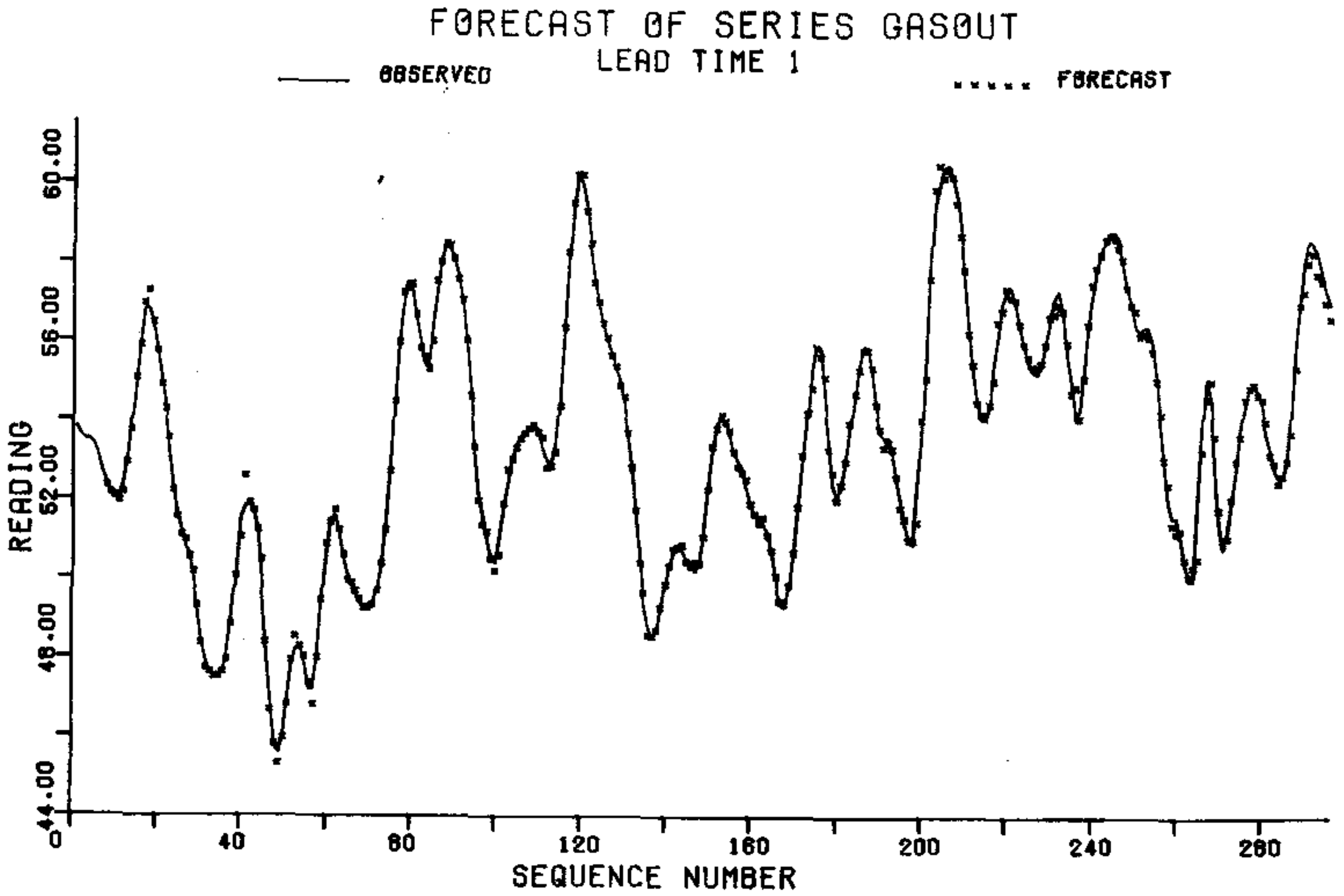

FIGURE 17.1 Type I forecasts, plotted without confidence limits

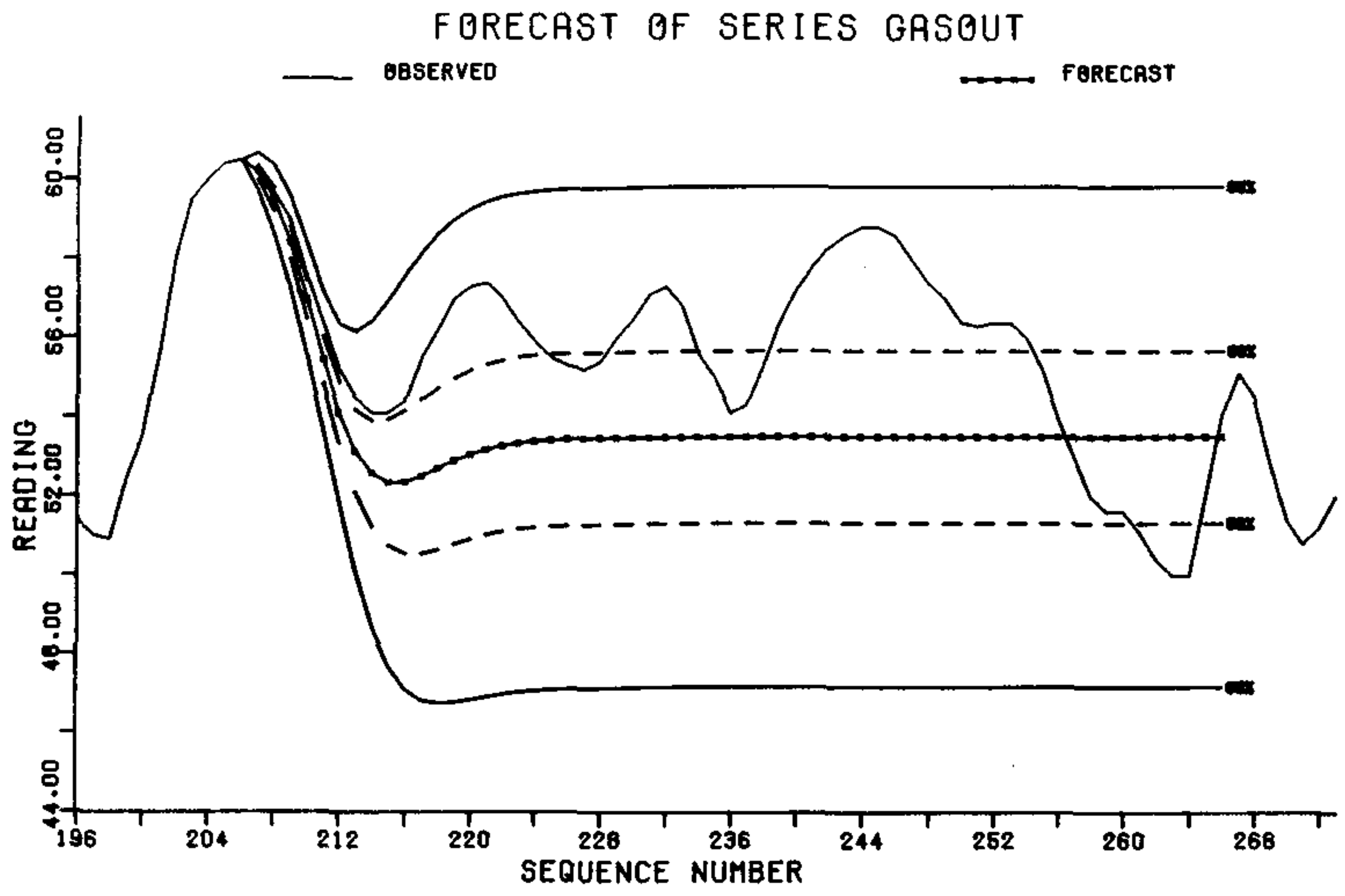

FIGURE 17.2 Type II forecasts

18. THE SUMMARY INSTRUCTION

The SUMMARY instruction has two uses. These are firstly to output the summaries of all, or a subset, of the series known to the system (See Chapter 6) and secondly, to generate summaries of models which have been set up via the SET UP command (Chapter 16).

The instruction has two forms to reflect these applications.

18.1 Command syntax

18.1.1 First Form

SUMMARY OF SERIES [list]

18.1.2 Second Form

SUMMARY OF MODEL model-name

18.2 Series Summaries

SUMMARY OF SERIES [list]

"list" is a set of names of series whose summaries are to be printed. Any number of names may be specified, up to the maximum of 30. Commas, spaces or the word AND may be used as separators. If the "list" option is not used (ie: no series names are given), then summaries of all series known to the system will be output. The items included in the summaries are

series name
number of points in the series
mean
variance
standard deviation
minimum value
maximum value

18.3 *Model Summaries*

SUMMARY OF MODEL model-name

"model-name" is the name of a model which has been created previously by a SET UP command. Only one model may be named on any one SUMMARY instruction. A complete listing of the selected model is output, together with its name.

18.4 Examples

(i) SUMMARY OF SERIES A B C AND D

(ii) SUMMARY

(iii) SUMMARY OF MODEL GASMODEL

References

Box, G. E. P. and Jenkins, G. M. (1970. Time Series Analysis forecasting and control. Holden-Day

Moore, R. J. (1977). Recursive Time Series Analysis Approach to Hydrological Modelling. *Inst. Hydrol. Rep. (in press)*

Shellswell, S. H. (1972). A Computer Aided Procedure for Time Series Analysis and Identification of Noisy Processes (CAPTAIN). *Tech. report.* CUED/B-Control/TR25, Department of Eng., University of Cambridge

Young, P.C., Shellswell, S. H. and Neethling, C. G. (1971). A recursive approach to time series analysis. *Tech. report.* CUED/B-Control/TR16, Dept. of Eng., University of Cambridge.

APPENDIX 1

LIST OF KEYWORDS

The following words are used as keywords within the package, and should be treated as reserved words:

(a) Alphabetical List

ABORT
ABS
ACF
ACOS
AHEAD
AND
AR
ASIN
AT
ATAN
AUTOCORRELATION
AUTOREGRESSIVE
AVERAGE
CONFIDENCE
CONVERGENCE
CORRELATION
COS
CROSS
CUSTOMER
DELAY
DEMAND
DESCRIBE
DIAGONAL
DIFFERENCE
END
ERROR
ESTIMATE
EXP
FINISH
FOR
FORECAST
FORMAT
FROM
FUNCTION
FUNCTIONS
INITIAL
INPUT
INVESTIGATE
ITERATIONS
LAG
LAGS
LEAD
LENG
LIMITS
LN
LOG
MA
MAX
MEAN
MEANS
MIN
MOD
MODE
MODEL
MOVING
NAME
NAMES
NO
NOISE
OF
OFF
ON
ONCE
ORIGIN
ORIGINS
OUTPUT
PACF
PARAMETER
PARAMETERS
PARTIAL
PERIOD
PLOT
PLOTTING
POINT
POINTS
PREWHITEN
PREDICTIONS
PRINT
PRINTING
PROCESS
P-MATRIX
Q-MATRIX
READ
RESIDUALS
SAVE
SEASONALLY
SERIES
SET
SIN
SQRT
STEP
STEPS
STREAM
STOCHASTIC
STRUCTURE
SUBSCRIPTS
SUBTRACT
SUMMARY
SWITCH
TAN
TD
TEKTRONIX
TESTING
TIME
TIMES
TITLE
TO
TRANSFORM
TRUNC1
TRUNC2
TWICE
UNFORMATTED
UP
USING
VALUE
VALUES
VARIABLE
VARIATIONS
VARNCE
WHITEN
WHITENED
XCF

(b) Group List

(i) Control Words

ABORT
CUSTOMER
DEMAND
FINISH
TEKTRONIX
TITLE

(ii) Primary Analysis Keywords

These are the initial words of main sentences which introduce the analyses.

DESCRIBE
DIFFERENCE
ESTIMATE
FORECAST
INVESTIGATE
PREWHITEN
READ
SAVE
SET
SUMMARY
TRANSFORM
WHITEN

(iii) Functions available in the Transformation Chapter

ABS	MEAN
ACOS	MIN
ASIN	MOD
ATAN	SIN
COS	SQRT
EXP	TAN
LENG	TRUNC1
LN	TRUNC2
LOG	VARNCE
MAX	

(iv) Secondary Analysis Keywords

These are keywords used within instructions for the analyses.

ACF	AUTOCORRELATION	CORRELATION
AHEAD	AUTOREGRESSIVE	CROSS
AND	AVERAGE	DELAY
AR	CONFIDENCE	DIAGONAL
AT	CONVERGENCE	DIFFERENCED

END
ERROR
FOR
FORMAT
FROM
FUNCTION
FUNCTIONS
INITIAL
INPUT
ITERATIONS
LAG
LAGS
LEAD
LIMITS
MA
MEAN
MEANS
MODE
MODEL
MOVING
NAME
NAMES
NO
NOISE
OF
OFF
ON
ONCE
ORIGIN
ORIGINS
OUTPUT
PACF
PARAMETER
PARAMETERS
PARTIAL
PERIOD
PLOT
PLOTTING
POINT
POINTS
PREDICTIONS
PRINT
PRINTING
P-MATRIX
Q-MATRIX
RESIDUALS
SEASONALLY
STEP
STEPS
STREAM
STOCHASTIC
STRUCTURE
SUBSCRIPTS
SUBTRACT
SWITCH
TD
TESTING
TIME
TIMES
TO
TWICE
UNFORMATTED
UP
USING
VALUE
VALUES
VARIABLE
VARIATIONS
WHITENED
XCF

APPENDIX 2

COMPLETE SYNTAX SKELETON

Conventions

(i) All words that are part of the CAPTAIN control language are specified as upper case words.

(ii) All words that must be present in a command are underlined. Upper case words that are not underlined are optional, and may be included at the discretion of the user.

(iii) All lower case words represent data that must be supplied by the user.

(iv) Elements of a command that involve a choice are surrounded by parentheses, i.e.: {x / y}
Only one of x and y may be chosen.

(v) Optional functions that may be included or omitted as required, are surrounded by brackets, i.e. [...]

The various commands listed below are in alphabetical order. Details of the commands may be obtained by following up the cross references given by the figures in round brackets, e.g. (5.2)

ABORT (5.4)

CUSTOMER customer-text (4.1)

DEMAND MODE [PRINT SWITCH {ON / OFF}] (5.2)

DESCRIBE SERIES list (11.1)

[{LAG / LAGS} k]

[{PLOT / PLOTTING} plot-list]

DIFFERENCE [SEASONALLY] SERIES list1 [key1] [PERIOD key2] (9.1)

[NAME DIFFERENCED SERIES list2]

ESTIMATE PROCESS MODEL PARAMETERS OF SERIES list (13.1)

STRUCTURE OF MODEL structure

[SUBSCRIPTS subscript-list]

[SUBTRACT MEANS FROM SERIES]

```
        [INITIAL  VALUES     values-list]
        [ITERATIONS    iters]
        [NAME  NOISE  SERIES    name]
         [{PLOT    } {PARAMETER } [plot-list]]
         [{PLOTTING} {PARAMETERS}            ]
ESTIMATE {NOISE     } MODEL  PARAMETERS OF SERIES  name1          (14.1)
         {STOCHASTIC}
                       STRUCTURE OF MODEL  structure
        [SUBSCRIPTS    subscript-list]
        [SUBTRACT  MEAN   FROM SERIES]
        [INITIAL   VALUES    values-list]
        [ITERATIONS    iters]
        [NAME   ERROR  SERIES    name2]
        [{PLOT    } {PARAMETER } plot-list]
        [{PLOTTING} {PARAMETERS}          ]
FINISH                                                             (4.3)
FORECAST  USING  SERIES  list1   AND  MODEL  modelname             (17.1)
       [ {{PRINT   }}                                               ]
       [ {{PRINTING}} PREDICTIONS FOR LEAD {TIME }intlist1          ]
       [ {{PLOT    }}                      {TIMES}                  ]
       [ {{PLOTTING}}                                               ]
       [               [NO  CONFIDENCE  LIMITS]                     ]
       [               [NAME ERROR SERIES  list2]                   ]
       [ {{PRINT   }}                                                      ]
       [ {{PRINTING}} PREDICTIONS  UP TO  intlist2   {STEP } AHEAD         ]
       [ {{PLOT    }}                                {STEPS}               ]
       [ {{PLOTTING}}                                                      ]
       [        {ORIGIN } AT  {POINT } intlist3                            ]
       [        {ORIGINS}     {POINTS}                                     ]
       [             [NO  CONFIDENCE  LIMITS]                              ]
       [             [NAME  ERROR SERIES  list3]                           ]
INVESTIGATE  PARAMETER VARIATIONS USING SERIES list                (15.1)
                      STRUCTURE OF MODEL  structure
        [SUBSCRIPTS    subscript-list]
         INITIAL     VALUES   values-list1
```

Underlined keywords: INITIAL, ITERATIONS, NAME, PLOT, PLOTTING, ESTIMATE, NOISE, STOCHASTIC, STRUCTURE, SUBSCRIPTS, SUBTRACT, MEAN, FINISH, FORECAST, MODEL, PRINT, PRINTING, LEAD, NO, LIMITS, UP, TO, ORIGIN, ORIGINS, INVESTIGATE.

```
[SUBTRACT  {MEAN  } FROM  SERIES]
           {MEANS }
P-MATRIX  DIAGONAL   values-list2
Q-MATRIX  DIAGONAL   values-list3
[NAME  {ERROR}  SERIES   name]
       {NOISE}
[{PRINT   }  {PARAMETER }  print-list]
[{PRINTING}  {PARAMETERS}            ]
[{PLOT    }  {PARAMETER }  plot-list]
[{PLOTTING}  {PARAMETERS}           ]
```

```
{PREWHITEN}  SERIES  list1  STRUCTURE OF MODEL  structure        (12.1)
{WHITEN   }
    [PRINTING  CONVERGENCE  OF {PARAMETER }]
                               {PARAMETERS}
    [NAME  WHITENED  SERIES  list2          ]
    [{PLOT    }     plot-list]
    [{PLOTTING}              ]
    [{LAG }          k]
    [{LAGS}           ]
```

```
READ  SERIES  VARIABLE  {NAME }   list                           (7.1)
                        {NAMES}
    [STREAM      stream-number]
    [POINTS      n]
    [{FORMAT    format-specifications}]
    [{UNFORMATTED                    }]
```

```
SAVE  SERIES  list  ON  STREAM   stream-number                   (8.1)
    [{FORMAT        format-specifications}]
    [{UNFORMATTED                        }]
```

```
SET UP  MODEL  NAME  name                                        (16.1)
    {model - specifications              }
    {INPUT ON STREAM   stream-number1    }
```

where "model-specifications" has the form

```
[PROCESS  MODEL  STRUCTURE   structure
    [SUBSCRIPTS    subscript-list1 ]
    {VALUE }
    {VALUES}       values-list1
    [MEANS         mean1  mean2 ]
```

```
|NOISE MODEL STRUCTURE structure 2                  |
|  [SUBSCRIPTS   subscript-list2 ]                  |
|  {VALUE  }                                        |
|  {VALUES }   values-list2                         |
[STOCHASTIC MODEL STRUCTURE structure3              ]
|  [SUBSCRIPTS   subscript-list3 ]                  |
|  {VALUE  }                                        |
|  {VALUES }     values-list3                       |
|  [MEAN         mean3 ]                            |
[OUTPUT ON STREAM   stream-number2 ]

SUMMARY OF MODEL modelname                              (18.3)

SUMMARY OF SERIES  [list]                               (18.2)

TEKTRONIX END                                           (5.3)

TEKTRONIX MODE PRINT SWITCH {ON }                       (5.3)
                            {OFF}
TITLE      title-text                                   (4.2)

TRANSFORM SERIES                                        (10.1)
      transformations
      END

WHITEN     (SEE PREWHITEN)                              (12.1)
```

APPENDIX 3

CAPTAIN ON THE UNIVAC 1108

This section describes how to use the Captain package as implemented on the Institute of Hydrology's UNIVAC 1108 computer. A knowledge of basic UNIVAC job control language is assumed. Users are referred to the internal publications "Using the UNIVAC is Easy" and "Using Files on the UNIVAC"; together with the "1100 Series Programmer Reference Manual" (UP4144) for further details.

The absolute program is held in the element IHLIBS*STATS.CAPTAIN and may be run using the following runstream:

```
@RUN     runid,acct,project,-----
@ASG,A   IHLIBS*STATS
@XQT     IHLIBS*STATS.CAPTAIN
      ---
      ---
      Captain Commands.
      ---
      ---
@FIN
```

The sample runstream, above, is suitable provided no plotting on the Calcomp plotter is required, and no Captain produced series or models are to be read from (or saved to) user files.

If the user intends either to save series and/or models for a future run of the package; or to read series and/or models from a file, then he should make certain, before executing the program, that he has attached the files to the input/output units he is to use. The package uses units 1-9 internally as temporary work streams, plotter streams and input/output streams. The user must therefore restrict his own files to units 10-29. The required file may either have numeric names in the range 10-29 for this purpose or the unit numbers may be attached to a file using the @USE JCL command.

```
eg.  @USE     10,SAVEFILE.
```

If the user attempts to write to a unit for which no assignment has been made then the system will assign a small temporary file which will be lost at run termination. However if an attempt is made to read from a unit for which no assignment has been made the program will terminate in error.

Consider a run which is to save some generated series to a file called SERIESFILE, and to read a model from a previously produced file called 11. Then a typical runstream might be:

```
@RUN      -------
@ASG,A    IHLIBS*STATS.
@ASG,CP   SERIESFILE.
```

```
@ASG,A      11.
@USE        10,SERIESFILE
@XQT        IHLIBS*STATS.CAPTAIN
                .
                .
                .
                .
SAVE  SERIES  series-name ON STREAM 10
                .
                .
                .
                .
SET UP MODEL NAME   model-name
INPUT ON STREAM 11
                .
                .
                .
@FIN
```

When working in TEKTRONIX MODE no special action is required for plotting. However if the user is to produce plots on the Calcomp plotter, the standard graph plotter spooling system control commands must be set up prior to entering Captain. Calcomp plotting with the Captain package requires 15 inch paper and 2 pens (usually black and one other colour). If ink pens are to be used a size of 0.3 mm is recommended.

Thus a typical runstream might be:

```
@RUN        PLxxxx, ----------
@ASG,A            IHLIBS*STATS
@PLDIR                                ) Plotter spooling system
 15,1,0.3,BLACK,0.3,RED               ) commands.  See the
                                      ) plotting manual and
@XQT              IHLIBS*STATS.CAPTAIN relevant user notes for
  .                                    current details.
  .
  .
  .
@FIN
```

This run would plot 1 copy of each graph.