Parametric Identification of Time Series

Gustavsson, Ivar

1968

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Gustavsson, I. (1968). *Parametric Identification of Time Series*. (Research Reports TFRT-3003). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

PARAMETRIC IDENTIFICATION OF TIME SERIES

I. GUSTAVSSON

REPORT 6803 APRIL 16 1968
LUND INSTITUTE OF TECHNOLOGY
DIVISION OF AUTOMATIC CONTROL

PARAMETRIC IDENTIFICATION OF TIME SERIES

I. Gustavsson

Abstract

A technique is described for parametric estimation of rational power spectra from sampled data. The estimation of the coefficients of the transfer function is obtained by using the maximum likelihood criterion. A FORTRAN algorithm for the identification procedure is given. Some computational results are presented. Furthermore the problem of tests of the model order is discussed. This work has been carried out with the support of the Swedish Technical Research Council.

Report 6803 April 16 1968
Lund Institute of Technology
Division of Automatic Control

INTRODUCTION

In this report we consider the identification of disturbances which can be described as stationary random processes having rational spectral densities. Such disturbances can always be represented as filtered white noise. The estimation procedure described gives the coefficients of the pulse transfer function of the filter directly and also the coefficient accuracy. The identification procedure is based on the method of maximum likelihood which can be shown to give estimates of the coefficients which are asymptotically consistent, normal and efficient for increasing sampling length. A FORTRAN algorithm for the identification procedure is given in the report. The technique has been applied both to simulated time series and to time series received from real processes. In the report some of the results of the computational experiments are presented.

The method may be compared with the conventional method of analysing time series by calculating autocorrelation functions and spectral densities with different numerical methods using spectral windows and so on. One of the advantages of the method given here is that the result may be used directly for prediction. It is also easier to determine the order of the system. The question how to choose the order of the model is discussed. The problem of several local maxima of the likelihood function is shortly mentioned. The autocorrelation functions and spectral densities may be easily computed from the obtained model without the numerical difficulties that may arise when calculating spectral densities by conventional methods.

In section 1 the problem is stated and in section 2 the FORTRAN program for the technique used to solve the problem is described. In section 3 the problem of tests of order is discussed and section 4 contains some examples.

1. STATEMENT OF THE PROBLEM

We now formulate the identification problem:
Given the observations $\{y(t), t = 1, 2, \ldots, N\}$, a time series, find an estimate of the parameters of the system model

$$A(z^{-1})\ y(t) = \lambda\ C(z^{-1})\ e(t) \qquad (1)$$

where $\{e(t)\}$ is a sequence of independent normal (0,1) random variables. Furthermore z denotes the shift operator

$$z\ x(t) = x(t + 1) \qquad (2)$$

and A(z) and C(z) are polynomials

$$A(z) = 1 + a_1 z + \ldots + a_n z^n$$
$$C(z) = 1 + c_1 z + \ldots + c_n z^n \qquad (3)$$

The following assumptions are made:

(i) the functions $A(z^{-1})$ and $C(z^{-1})$ have all their zeros inside the unit circle

(ii) there are no factors common to polynomials A(z) and C(z).

Then this problem is that of parametric estimation of rational power spectra.

The technique used is described in references {1} and {2}, but they consider a somewhat different problem. They describe a technique for identification of a discrete time system from input/output samples, that is the model used is

$$A(z^{-1})\ y(t) = B(z^{-1})\ u(t) + \lambda\ C(z^{-1})\ e(t) \qquad (4)$$

where $\{u(t)\}$ is the input, $\{y(t)\}$ the output and B(z) a polynomial

$$B(z) = b_o + b_1 z + \ldots + b_n z^n \qquad (5)$$

and the other notations as before.

A short summary of the theory is given here. The problem stated above is a statistical parameter estimation problem. It will be solved by the method of maximum likelihood.

It follows from (1) that the numbers $\{\varepsilon(t)\}$, defined by

$$C(z^{-1})\ \varepsilon(t) = A(z^{-1})\ y(t) \tag{6}$$

are independent and normal $(0,\lambda)$. $\{\varepsilon(t)\}$ are called the residuals. The logarithm of the likelihood function now becomes

$$L = -\frac{1}{2\lambda^2} \sum_{t=1}^{N} \varepsilon^2(t) - N \log \lambda + \text{const.} \tag{7}$$

Maximizing this function is equivalent to minimizing the loss function

$$V(\theta) = \frac{1}{2} \sum_{t=1}^{N} \varepsilon^2(t) \tag{8}$$

where θ is the column vector $(a_1, \ldots, a_n, c_1, \ldots, c_n)$. When $\hat{\theta}$, such that $V(\hat{\theta})$ is minimal, has been found, the maximum likelihood estimate of λ will be

$$\hat{\lambda}^2 = \frac{2}{N} V(\hat{\theta}) \tag{9}$$

To minimize the loss function $V(\theta)$ the following Newton-Raphson algorithm is used

$$\theta^{k+1} = \theta^k - \{V_{\theta\theta}(\theta^k)\}^{-1}\ V_\theta(\theta^k) \tag{10}$$

where

V_θ = the gradient vector of $V(\theta)$

$V_{\theta\theta}$=the matrix of second partial derivatives of $V(\theta)$

Differentiating (8) gives

$$\frac{\partial V}{\partial \theta_i} = \sum_{t=1}^{N} \varepsilon(t) \cdot \frac{\partial \varepsilon(t)}{\partial \theta_i} \tag{11}$$

$$\frac{\partial^2 V}{\partial \theta_i \, \partial \theta_j} = \sum_{t=1}^{N} \frac{\partial \varepsilon(t)}{\partial \theta_i} \frac{\partial \varepsilon(t)}{\partial \theta_j} + \sum_{t=1}^{N} \varepsilon(t) \cdot \frac{\partial^2 \varepsilon(t)}{\partial \theta_i \, \partial \theta_j} \tag{12}$$

Due to the well-known fact that a Newton-Raphson procedure will not converge from a poor approximation, the first iterations will only use the approximate second partial derivatives (that is the first term of (12)). Thus we use a Gauss-Newton method for the first iterations. This ensures most often convergence to a minimum as the matrix of the approximate second derivatives usually is positive definite {3}.However, the convergence may be slow, and near the minimum point the exact second derivatives should be used to speed up the convergence rate.

Differentiating of (6) gives

$$\begin{aligned} C(z^{-1}) \frac{\partial \varepsilon(t)}{\partial a_j} &= z^{-j} \, y(t) \\ C(z^{-1}) \frac{\partial \varepsilon(t)}{\partial c_j} &= -z^{-j} \, \varepsilon(t) \end{aligned} \tag{13}$$

The last equation of (13) can be differentiated once more:

$$\begin{aligned} C(z^{-1}) \frac{\partial^2 \varepsilon(t)}{\partial a_i \, \partial c_j} &= -z^{-i-j+1} \frac{\partial \varepsilon(t)}{\partial a_1} \\ C(z^{-1}) \frac{\partial^2 \varepsilon(t)}{\partial c_i \, \partial c_j} &= -\, 2z^{-i-j+1} \frac{\partial \varepsilon(t)}{\partial c_1} \end{aligned} \tag{14}$$

where the relation

$$\frac{\partial \varepsilon(t)}{\partial a_i} = z^{-i+1} \frac{\partial \varepsilon(t)}{\partial a_1} = \frac{\partial \varepsilon(t-i+1)}{\partial a_1} \qquad i \leq t+1 \tag{15}$$

has been used.

Similar formulas also hold for the other derivatives.

Now the algorithm for minimizing the loss function becomes:

1. Put $\theta^k = \theta^o$ (starting value of θ)
2. Evaluate $V_\theta(\theta^k)$ and $V_{\theta\theta}(\theta^k)$ using (11), (12), (13) and (14).
3. Calculate θ^{k+1} from (10) and repeat from 2.

The recursive formula (10) requires an initial value. Putting $c_i = 0$, $i = 1, 2, \ldots, n$ we obtain in one step the least squares estimate a^o of a. Then the initial value for the iteration (10) is taken as $\theta^o = (a_1^o, \ldots, a_n^o, 0, \ldots, 0)^T$.

The standard deviations of the estimates will be computed from the estimate of the covariances given by the matrix $\hat{\lambda}^2 \{V_{\theta\theta}(\hat{\theta})\}^{-1}$. These standard deviations are automatically calculated by this identification procedure, because $\{V_{\theta\theta}(\hat{\theta})\}^{-1}$ is used in the Newton-Raphson algorithm (10). We may also compute the standard deviations of the loss function and of the parameter λ. The standard deviation of the loss function is $\hat{\lambda}^2\sqrt{\frac{N}{2}}$ and that of the parameter λ is approximately $\frac{\hat{\lambda}}{\sqrt{2N}}$ {4}.

2. PROGRAM

The program used for the identification procedure is a modification of a program developed in reference {2}. This program and the original modification are in ALGOL. The program was then translated to FORTRAN and the program with the last improvements is now available only in FORTRAN but with a small effort it is possible to complete the ALGOL version. In Appendix the computer program is given.

The program consists of the main program and three subroutines, PERPS, LFGSD and GJRV. The first of these subroutines makes the identification procedure. LFGSD calculates the loss function, $V(\theta)$, the gradient, $V_\theta(\theta)$ and the second partial derivatives, $V_{\theta\theta}(\theta)$. The last one is a matrix-inversion routine. In the following the main program and the subroutines are commented. Further comments may be found in the subroutines themselves.

The program has been tested on the CDC 3600, and in Table 1 we give the time necessary for one iteration for a record length of 1000 data. The dependence of order is almost linear. The dependence of record length is also approximately linear.

Order of system	Time in seconds
1	2.5
2	3.5
3	4.5
4	5.5

Table 1 - Time for one iteration for a record of 1000 data.

MAIN PROGRAM

This program must contain:

1. Program identifier
2. The same common statement as in subroutine PERPS
3. Input data in the array DAT.(Notice that the chosen model requires that the mean of the time series is zero.)Eventual starting values for some of the coefficients (in the array C).
4. A call sequence

```
   CALL PERPS (..,..,..,..,..,..)
       :
```

5. Format specifications for input data and eventual extra output.
6. END

SUBROUTINE PERPS (NO, NP, L1, L2, IT, IPRINT)

This subroutine performs the identification. When called it executes a number of iterations according to (10).

The parameters of the call are defined as follows:

NO Order of system model (1)

NP Number of data in the analysed time series

L1 Parameter determining which starting values of the coefficients $a_1, \ldots, a_n, c_1, \ldots, c_n$ will be used for the iteration.

L1=1 A least squares estimation of $a_1, a_2, \ldots, a_n$ is made starting from $a_1 = \ldots = a_n = c_1 = \ldots = c_n = 0$.

L1=2 A least squares estimation of $a_1, a_2, \ldots, a_n$ is made starting from $a_1 = \ldots = a_n = 0$ but where $c_1, \ldots, c_n$ have been given special values in the main program.

L1=0 A common estimation of $a_1, \ldots, a_n, c_1, \ldots, c_n$ is made starting from the values of these coefficients obtained in the preceding iteration.

L1=-1 A common estimation of $a_1, \ldots, a_n, c_1, \ldots, c_n$ is made starting from values of these coefficients given in the main program.

L2 Parameter determining the number of iterations.

L2=0 The iteration procedure continues until maximum correction of the coefficients $a_1, \ldots, a_n, c_1, \ldots, c_n$ is smaller than 0.0001 or if approximate second derivatives are used until this maximum correction is smaller than 0.01.

L2=N N iterations are made.

IT Parameter determining whether exact or approximate second derivative matrix will be used (12).

IT=0 Approximate second derivatives used.

IT=1 Exact second derivatives used.

IPRINT Parameter determining whether the matrix of second derivatives and its inverse are printed out or not.
IPRINT=0 The matrices are not printed out.

A normal call sequence is

```
   CALL PERPS (NO, NP, 1, 1, 0, 0)
   CALL PERPS (NO, NP, 0, N, 0, 0)
   CALL PERPS (NO, NP, 0, 0, 1, 1)
```

where NO, NP and N have given values. This call sequence first gives a least squares estimation of $a_1, \ldots, a_n$ and then N iterations starting the first of them from $a_1, \ldots, a_n$ obtained from the least squares estimation and $c_1 = \ldots = c_n = 0$. Then the procedure will continue until maximum correction of coefficients is smaller than 0.0001 and these last iterations use the exact second derivatives. N is chosen as to secure convergence to the minimum point. As a rule N may be chosen two or three times as large as the order of the system NO, if data are not too bad. If N is chosen too small, the exact second derivatives may be used before we have reached a point from which convergence to the minimum point will occur. In this case convergence to a maximum point may occur if there exists such a point, and then you have to execute the program once more with the obtained coefficients as starting values and use the approximate second derivatives for a few more iterations.

The subroutine PERPS initially puts $\alpha = 1$ in the Newton-Raphson algorithm

$$\theta^{k+1} = \theta^k - \alpha \{V_{\theta\theta}(\theta^k)\}^{-1} V_\theta(\theta^k) \qquad (16)$$

If the loss function calculated in the new point $V(\theta^{k+1})$ is larger than $V(\theta^k)$ then α is halved until $V(\theta^{k+1})$ is smaller than $V(\theta^k)$ or maximum correction of coefficients is smaller than 0.0001.

PERPS needs the subroutines LFGSD and GJRV. They are automatically called from PERPS.

Notations used in subroutine PERPS:

C(i)	i = 1,2,...,2n	$\theta = (a_1,..,a_n,c_1,...,c_n)$
V		$V(\theta^k)$
V1(i)	i = 1,2,...,2n	$V_\theta(\theta^k)$
V2(i,j)	i = 1,2,...,2n, j = 1,2,...,2n	$V_{\theta\theta}(\theta^k)$
SPR		λ
CC(i)	i = 1,2,...,2n	$-\{V_{\theta\theta}(\theta^k)\}^{-1} V_\theta(\theta^k)$
ALFA		α
EE		$\varepsilon(t)$
V2COND		Condition number of second partial derivative matrix

The condition number used is defined as follows {5}. The condition number, M, of a matrix A is calculated from the formula

$$M = 2 \cdot n \cdot \max_{i,k} |a_{ik}| \cdot \max_{i,k} |b_{ik}| \qquad (17)$$

where $A = (a_{ik})$ and $A^{-1} = (b_{ik})$ and n = the order of matrix A. Large condition number is most often an indication that numerical difficulties may occur when the matrix is used in the calculations. It is a measure of the singularity of the matrix.

SUBROUTINE LFGSD (NO, NP, IT)

This subroutine calculates the loss function, V_θ, the gradient, $V_\theta(\theta)$ and the second partial derivative matrix, $V_{\theta\theta}(\theta)$.

The parameters of the call are defined as:

NO Order of system model (1)

NP Number of data in time series

IT Parameter determining whether exact or approximate second derivatives are computed.

These parameters are automatically transferred from the call of PERPS to the call of LFGSD.

In order to save computing time and memory capacity state variables $E_i(t)$, $i = 1,n$ are chosen as follows:

$$\begin{bmatrix} E_1(t) \\ E_2(t) \\ \vdots \\ E_{n-1}(t) \\ E_n(t) \end{bmatrix} = \begin{bmatrix} -c_1 & 1 & 0 & \dots & 0 \\ -c_2 & 0 & 1 & & 0 \\ \vdots & & & & \\ -c_{n-1} & 0 & 0 & & 1 \\ -c_n & 0 & 0 & & 0 \end{bmatrix} \begin{bmatrix} E_1(t-1) \\ E_2(t-1) \\ \vdots \\ E_{n-1}(t-1) \\ E_n(t-1) \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix} y(t-1) + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} y(t) \tag{18}$$

$$e(t) = E_1(t)$$

The procedure makes it possible to sum (8), (11) and (12) successively. Only n state variables, 2n first derivatives and 4n second derivatives have to be saved.

The computation of the approximative second derivative matrix follows a special procedure exploiting the fact that the elements of the sums are essentially the same. Only the last terms of the sums are different. Then it is only necessary to compute one column vector of for instance the derivatives with respect to c_n, that is the only elements necessary to compute for each t are the following

$$\frac{\partial^2 V}{\partial a_i \, \partial a_n} \qquad i = 1, 2, \dots, n$$

$$\frac{\partial^2 V}{\partial a_i \, \partial c_n} \qquad i = 1, 2, \dots, n$$

$$\frac{\partial^2 V}{\partial a_n \, \partial c_i} \qquad i = 1, 2, \dots, n-1$$

$$\frac{\partial^2 V}{\partial c_i \, \partial c_n} \qquad i = 1, 2, \dots, n$$

For instance consider the following part of the matrix:

$$\begin{bmatrix} \frac{\partial^2 V}{\partial a_1 \partial a_1} & \cdots & & \frac{\partial^2 V}{\partial a_1 \partial a_n} \\ & \ddots & & \vdots \\ & & \frac{\partial^2 V}{\partial a_{n-2} \partial a_{n-1}} & \frac{\partial^2 V}{\partial a_{n-2} \partial a_n} \\ & & \frac{\partial^2 V}{\partial a_{n-1} \partial a_{n-1}} & \frac{\partial^2 V}{\partial a_{n-1} \partial a_n} \\ & & & \frac{\partial^2 V}{\partial a_n \partial a_n} \end{bmatrix} =$$

$$= \begin{bmatrix} \sum_{t=1}^{N} \frac{\partial \varepsilon(t)}{\partial a_1} \cdot \frac{\partial \varepsilon(t)}{\partial a_1} & \cdots & & \sum_{t=1}^{N} \frac{\partial \varepsilon(t)}{\partial a_1} \cdot \frac{\partial \varepsilon(t-n+1)}{\partial a_1} \\ & \ddots & & \vdots \\ & & \sum_{t=1}^{N-n+3} \frac{\partial \varepsilon(t)}{\partial a_1} \cdot \frac{\partial \varepsilon(t-1)}{\partial a_1} & \sum_{t=1}^{N-n+3} \frac{\partial \varepsilon(t)}{\partial a_1} \cdot \frac{\partial \varepsilon(t-2)}{\partial a_1} \\ & & \sum_{t=1}^{N-n+2} \frac{\partial \varepsilon(t)}{\partial a_1} \cdot \frac{\partial \varepsilon(t)}{\partial a_1} & \sum_{t=1}^{N-n+2} \frac{\partial \varepsilon(t)}{\partial a_1} \cdot \frac{\partial \varepsilon(t-1)}{\partial a_1} \\ & & & \sum_{t=1}^{N-n+1} \frac{\partial \varepsilon(t)}{\partial a_1} \cdot \frac{\partial \varepsilon(t)}{\partial a_1} \end{bmatrix}$$

To get the element $\partial^2 V/\partial a_{n-1}\partial a_{n-1}$ take the element $\partial^2 V/\partial a_n \partial a_n$ and add one more term etc. (Notice that N is much greater than n.)

Also for the computation of the correcting term necessary to obtain the exact second derivatives, we can save a great deal of computing time

$$\left(\frac{\partial^2 V}{\partial a_i \partial c_j} \right)_{corr} = \sum_{t=1}^{N} \frac{\partial^2 \varepsilon(t)}{\partial a_i \partial c_j} = \sum_{1}^{N} \frac{\partial^2 \varepsilon(t-i-j-2)}{\partial a_1 \partial c_1}$$

and we find that the correction term is identical for derivatives for which the sum of the indices are the same.

SUBROUTINE GJRV (A, N, EPS, IERR, IA)

This subroutine inverts asymmetric matrices. The routine uses the Gauss-Jordan method. It is used for the inversion of the matrix of the second partial derivatives. Notice that in this subroutine the original matrix is destroyed. More comments are given in the head of the subroutine.

Additional programs

Various programs have been developed for the analysis of the results obtained e.g. a program for testing the normality of the residuals, a program for computing the autocorrelation function and spectral densities from the transfer function.

3. TESTS OF THE ORDER OF MODEL

If the order of the system is not known before the identification, another problem arises: What is the minimal order of a model that represents the system? To answer this problem we need a test. There are many methods to test the order of the model, but no one of them seems to be entirely satisfactory, and they must thus be used with caution.

Below the following methods are mentioned:

a) The method of the second derivative matrix
b) The statistical F-test
c) The examination of the residuals $\{\varepsilon(t)\}$
d) The examination of the roots of the polynomials $A(z^{-1})$ and $C(z^{-1})$
e) The examination of the standard deviations of the parameters
f) The examination of the physical background

To illustrate the various methods, results from the examples of identification in the next section are taken.

The second derivative matrix $V_{\theta\theta}(\hat{\theta})$ can be used as an indication that there are redundant parameters in the model {1}. Therefore the matrix will be singular if a model of too high order is chosen. If we have a measure of the singularity of the matrix and repeat the identification with increasing order, we would have a test of the order of the model. This test is, however, not easy to use, because it can be difficult to find a suitable measure of the singularity (some sort of condition number may be used) and because the singularity may be caused by other reasons. This method is not used here, but as a comparison we give in Table 2 the condition numbers calculated from (18) for example 2. The system is here of second order.

Order of model	Condition number
1	6
2	221
3	194740

Table 2 - Condition numbers for different orders of the model in example 2.

Next method is a statistical test. Also for this method we repeat the identification with increasing order. Now let V_n denotes the minimal value of the loss function for the n-th order model. It follows from {1} that the parameter estimates for a large number of data are asymptotically normal $(\theta_o, \lambda^2 V_{\theta\theta}^{-1})$, where θ_o stands for the correct value of θ. Assuming that asymptotic theory may be applied, we test the hypothesis that the system is of order n, that is the null hypothesis is

$$H_o: \quad a_{n+1}^o = \ldots = a_{n+k}^o = c_{n+1}^o = \ldots = c_{n+k}^o = 0$$

(θ_i^o stands for the correct value of θ_i)

Then

$$F_{n+k,n} = \frac{V_n - V_{n+k}}{V_{n+k}} \cdot \frac{N - 2(n + k)}{2 \cdot k} \qquad (19)$$

has an $F(2k, N - 2(n + k))$ distribution under null hypothesis.

Most often the test is used with $k = 1$, that is we test the model of order $(n + 1)$ against the model of order n. It is used at a risk level of 5%, that is if the test quantity is greater than 3,0 (N is supposed to be greater than 100) the loss function has been reduced significantly. Tests of this kind are frequently used in the examples. The test is, however, not always reliable. We give here an example. The results are taken from example 5, series I:A. In Table 3 the obtained test quantities are given.

n \ n+k	2	3	4	5	6	8
1	1380	701	482	362	296	216
2	-	8.4	12.2	8.7	9.1	8.2
3	-	-	15.8	8.8	9.3	8.1
4	-	-	-	1.8	5.9	6.1
5	-	-	-	-	10.0	7.5
6	-	-	-	-	-	6.2

Table 3 - The test quantities obtained when testing (n+k)-order model against n-order model for series I:A in example 5 using F-test.

From this table it is clear that this test and the use of a risk level of 5% give that the system is of at least eighth order. If we study the roots of the polynomials $A(z^{-1})$ and $C(z^{-1})$ we find that some of them may be considered equal. (Table 4).

Order of model	Roots of $A(z^{-1})$	Roots of $C(z^{-1})$
6	$-0.634 \pm 0.730\, i$	$-0.637 \pm 0.742\, i$
8	$-0.923 \pm 0.341\, i$	$-0.929 \pm 0.340\, i$
	$-0.055 \pm 0.995\, i$	$-0.044 \pm 0.987\, i$

Table 4 - Some of the roots of the polynomials $A(z^{-1})$ and $C(z^{-1})$ for the sixth- and the eighth-order models for series I:A in example 5.

The fact that the other roots of the denominator are almost unchanged when increasing the order of the system, is an indication that they represent some essential characteristics of the system. In this example the F-test gives a very high order, but the system is very likely only of fourth order. This can be justified by use of the test of the residuals, described below, and by considering the physical background, which must never be forgotten. A known fact in this example is that the so called α- and β-rhythms should be there, and they are obtained from the fourth order model. Probably there are no more marked rhythms in these records.

The method to examine if the polynomials $A(z^{-1})$ and $C(z^{-1})$ have any roots in common, has already been used above and are not commented any more.

Another method of testing the order of the model is to study the standard deviations of the estimated parameters. As seen from the examples a small reduction of loss function is most often combined with a remarkable increase of the standard deviations and that a great reduction of loss function is combined with decreasing standard deviations. (For instance example 5, series I:B, $n = 2,3,4,6$) This method has, however, the same disadvantages as the statistical test of the loss function and they are in fact asymptotically equivalent. Another difficulty combined with these two tests is that there may be no significant reduction of the loss function when increasing the order, but suddenly a remarkable reduction occurs.(For instance example 4, series C). If this phenomenon can occur for a very high order system, is not known. However, there may be a risk that a system is supposed to be of first order, when it may be of a much higher order. Unless something about the magnitude of the order is known before, this disadvantage may make it impossible to determine the real order of the system.

There is one method left, that is the examination of the residuals. Firstly if the model order is not less than the order of the system, then the residuals form a series of independent normal variables. This fact is thus a test of the order of the system. Secondly the residuals can be interpreted as being one step ahead prediction errors, a fact that can be used in other applications.

To test the order of the system the residuals must be examined for increasing order of the model. The residuals are obtained from (6). The examination is also motivated by the fact that the identification procedure was based on the assumption that the residuals were normal and uncorrelated. If these assumptions do not hold, the results from the procedure may not be good.

Tests of normality and independence of the residuals have been performed for most of the examples. The test of normality may be done in a number of ways. One of the methods is to compute a probability density function and to compare it to the theoretical normal distribution. Another is the so called chi-square goodness-of-fit test {6}, {7}. To test the independence the autocorrelation function for the residuals is computed. If they are independent the function should equal zero when $\tau \neq 0$. We give an example of a test of independence of the residuals for increasing order of the model. The example is taken from example 4, time series C and in figure 1 the autocorrelation functions of the residuals from the 1st, 2nd, 3rd and 4th-order models are shown. (Notice that the curves in the figures have been obtained by connecting the values of the sampling points with lines.) The system is supposed to be of fourth order.

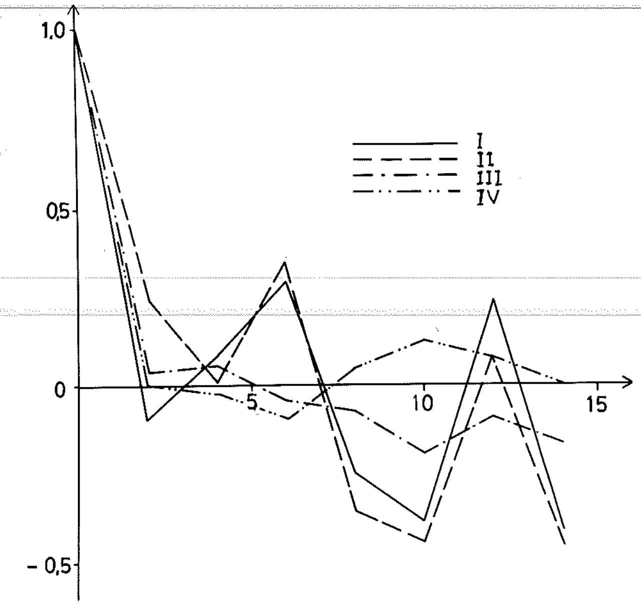


Fig. 1 - Autocorrelation functions for the residuals of the first (I), second (II), third (III), and fourth (IV) order models. (Time series C, example 4)

We end this section by recommending not to forget to study the physical background of the problem, because the test may be contradictory and lead to wrong conclusions.

4. PRACTICAL EXPERIMENTS TO DETERMINE SYSTEM MODELS FOR TIME SERIES

The identification procedure described in the previous section has been tested for many different time series, both simulated data and data from practice. If nothing else is said the FORTRAN version of the program has been used at the identification.

Example 1

Identify the following system

$$y(t) \quad = \quad 0.9 \frac{1 + 0.5z^{-1}}{1 - 0.9z^{-1}} e(t) \qquad (20)$$

In the experiment 600 variables y were generated using equation (20). The random numbers $\{e(t)\}$ were obtained from a subroutine RANSS, generating random floating point numbers distributed according to normal distribution with a mean of 0 and a variance of 1. The routine has been tested and the numbers obtained are very nearly normally distributed (0,1) and independent.

First choose n = 1 and the identification procedure proceeds as shown in Table 5 (notations: ex stands for exact second derivatives used; V_n loss function; COND stands for condition number, defined in section 2).

Step	a_1	c_1	V_1	COND
0	0	0	2547	10200
1	-0.94	0	289	20
2	-0.894	0.42	236.1	11
3	-0.895	0.503	234.4	8
4	-0.896	0.498	234.4	8
5ex	-0.896	0.498		
σ_i	0.019	0.037		
λ	0.884			

Table 5 - Successive estimates of the parameters (1st order model)

For n = 2 we get the results given in Table 6.

Step	a_1	a_2	c_1	c_2	V_2	COND
0	0	0	0	0	2547	20400
1	-1.32	0.40	0	0	243	1300
2	-1.44	0.49	-0.05	-0.24	234.2	37100
3	-1.73	0.75	-0.33	-0.39	234.0	27700
4	-1.78	0.80	-0.38	-0.42	234.0	21500
5	-1.84	0.85	-0.44	-0.46	233.8	13800
6	-1.879	0.884	-0.482	-0.479	233.3	6900
7	-1.890	0.893	-0.494	-0.483	233.3	5100
8	-1.884	0.889	-0.488	-0.479	233.2	4755
9ex	-1.884	0.888	-0.487	-0.479	233.2	4793
10ex	-1.884	0.888	-0.487	-0.479		
σ	0.029	0.026	0.045	0.041		
λ	0.882					

Table 6 - Successive estimates of the parameters (2nd order model)

We now test the null hypothesis that the system is of first order and find that $F_{2,1} = 1.5$ and that the hypothesis thus has to be accepted.

From this we conclude that the system identified is of first order and the parameters are given in Table 7.

Parameter	Estimated	True
a_1	-0.90 ± 0.02	-0.9
c_1	0.50 ± 0.04	0.5
λ	0.88 ± 0.02	0.9
V	234.4 ± 13.5	

Table 7 - Estimated and true parameter values

Notice that the first step of the identification procedure gives the least squares estimate of the parameters and the significant difference between this estimate and the obtained likelihood estimate.

Diagrams show the series of random numbers used, the generated time series {y(t)} and the residuals obtained from the first order model (figure 2). Figure 3 shows the autocorrelation function for the original system (20) and for the obtained first order model and the autocorrelation function calculated direct from the time series {y(t)} according to

$$R(\tau) = \frac{1}{N - \tau} \sum_{t=1}^{N-\tau} y(t)\, y(t + \tau) \qquad (21)$$

The series {y(t)} is supposed to have zero mean when (21) is used.
Figure 4 shows the autocorrelation function for the residuals obtained from the first order model. Tests of normality show that the hypothesis about normality of the residuals may be accepted. ($\chi^2 = 15 < \chi^2_{0.05} = 35$). Note the remarkable increase of the condition number when increasing the order of the model. This is also indicating that the system is of first order.

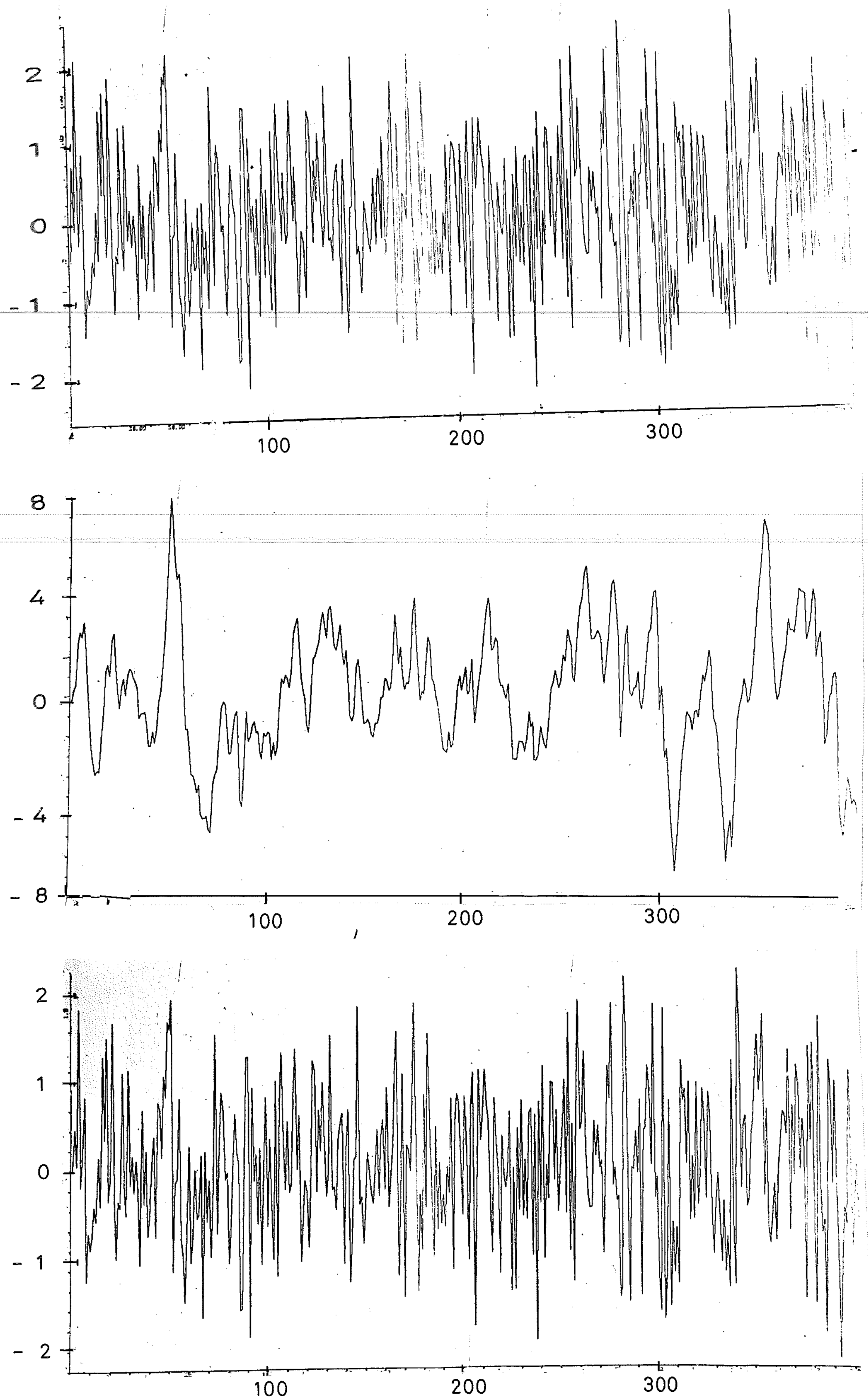


Fig. 2 - Sequence of random numbers, time series and residuals from the first order model (only the first parts shown)

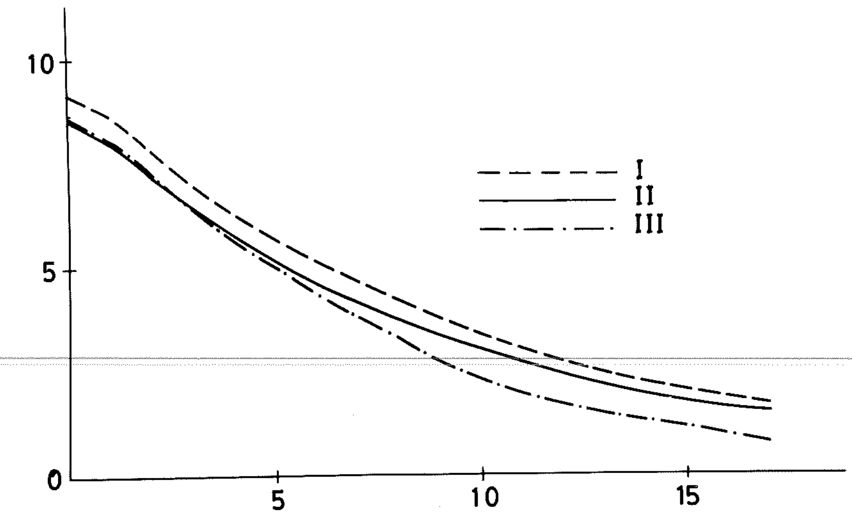


Fig. 3 - Autocorrelation functions for the original system (I), for the identified system (II) and for the generated time series (III)

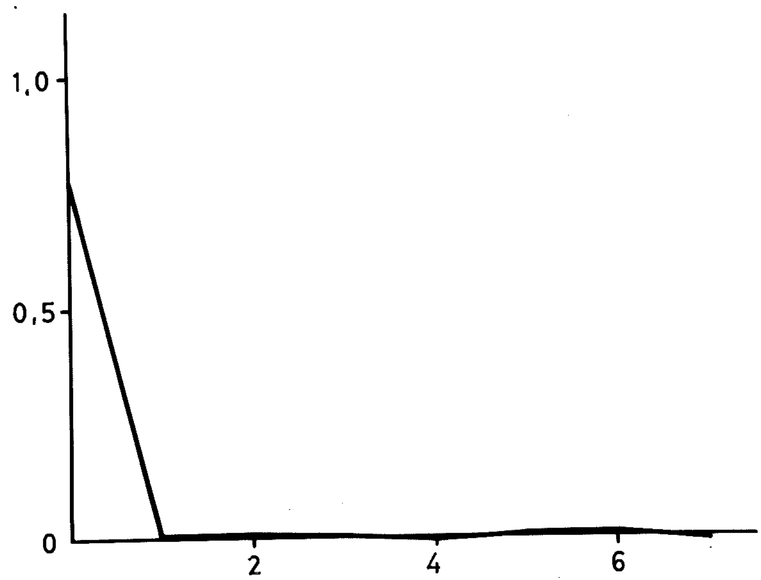


Fig. 4 - Autocorrelation function for the residuals from the first order model (example 1)

Example 2

Identify the following system

$$y(t) = 0.75 \frac{1 - 1.0z^{-1} + 0.4z^{-2}}{1 - 1.5z^{-1} + 0.7z^{-2}} e(t) \qquad (22)$$

600 variables {y(t)} are generated using (22) and the same set of {e(t)} as in example 1. For this example and the following ones the proceeding of the identification procedure is not tabled. Only the obtained likelihood estimate is given (Table 8).

n	a_1	a_2	a_3	c_1	c_2	c_3	λ	V_1
1	-0.71±0.04	-	-	-0.16±0.05	-	-	0.75	170.9
2	-1.56±0.08	0.73±0.06	-	-1.06±0.09	0.39±0.05	-	0.74	162.1
3	-1.58±1.14	0.74±1.77	-0.01±0.82	-1.08±1.14	0.41±1.20	-0.05±0.45	0.73	162.0

Table 8 - Estimates of parameters

Test first the null hypothesis that the system is of first order. This gives $F_{2,1} = 16.1$ and the hypothesis is rejected. Test then the null hypothesis that the system is of second order. This gives $F_{3,2} = 0.2$ and the hypothesis is accepted.

We then conclude that the analysed system is of second order and the parameters are given in Table 9.

Parameter	Estimated	True
a_1	-1.56 ± 0.08	-1.5
a_2	0.73 ± 0.06	0.7
c_1	-1.06 ± 0.09	-1.0
c_2	0.39 ± 0.05	0.4
λ	0.74 ± 0.02	0.75
V	162.1 ± 9.2	

Table 9 - Estimated and true parameters

Diagrams (figure 5 and 6) show the series of random numbers used, the generated time series $\{y(t)\}$, the residuals obtained from the second order model, the autocorrelation functions for the original system (22), for the obtained second order model and for the time series $\{y(t)\}$. Tests show that the residuals are normally distributed ($\chi^2 = 4 < \chi^2_{0.05} = 35$) and the autocorrelation function for them is shown in figure 7.

Now note that for these two examples no numerical difficulties at all appeared at the computations and that the obtained estimates are rather good compared with the parameters of the simulated systems. The tests of order work very well too. The reasons for this are the relative long series used and the fact that the generated set $\{e(t)\}$ is very nearly normally distributed.

The difficulties arise when we start identifying time series from practice, that is where the assumptions made not always hold, and where we often have to try to identify "short" time series. "Short" is here used relative to the existing time constants of the system. As a rule it is much more difficult to determine the order of the system than in the examples above. Numerical difficulties arise which may prolong the time necessary for the identification. When the time series is "short" an additional difficulty sometimes arises. The procedure used converges to a local minimum of the loss function, but in this case this may not be unique and it may be very difficult to find the global minimum. The only way is to identify the system with different starting values of the coefficients $c_1, \ldots, c_n$. This is exemplified later in this section.

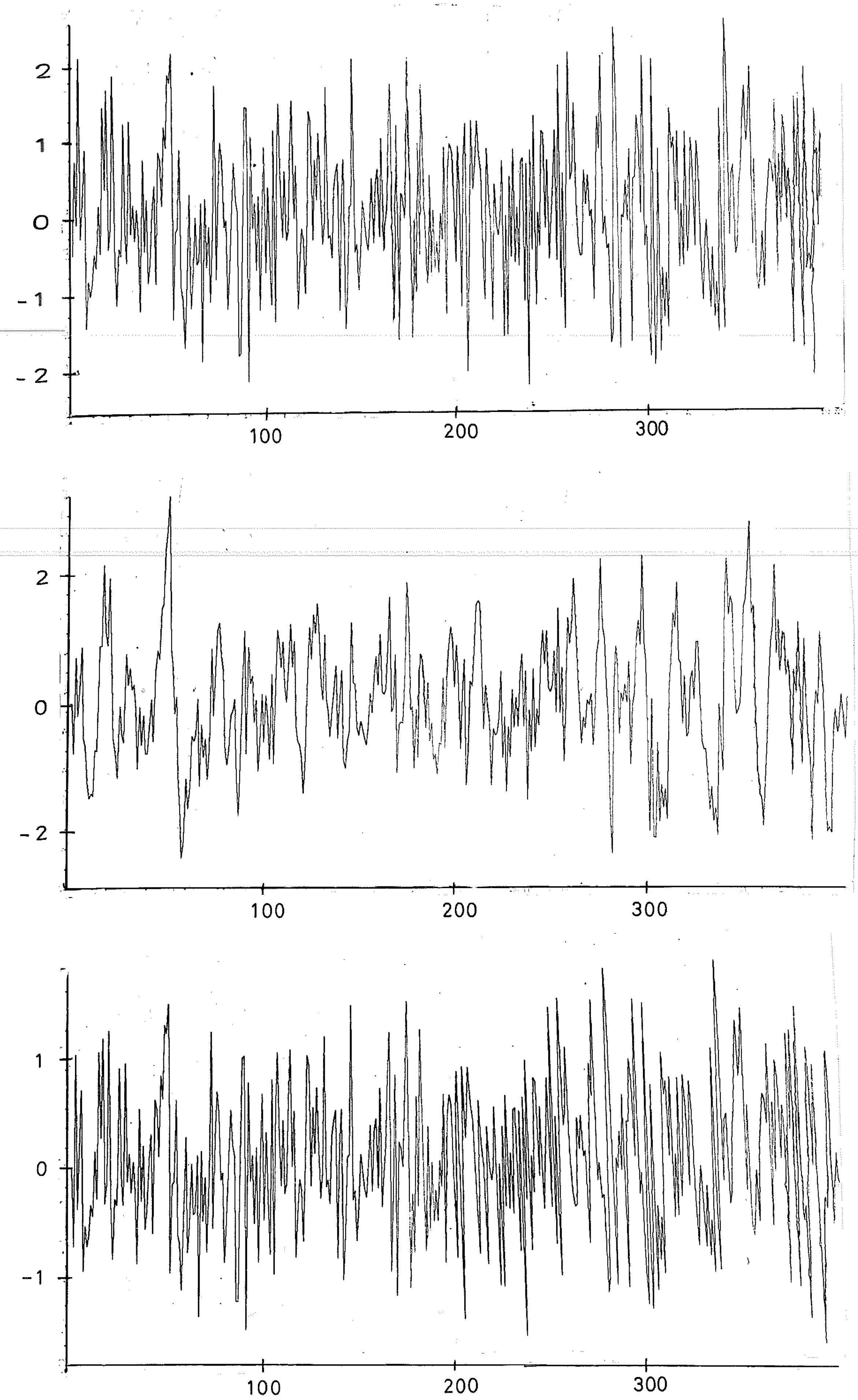


Fig. 5 - Sequence of random numbers, time series and residuals from the second order model (only the first part shown)

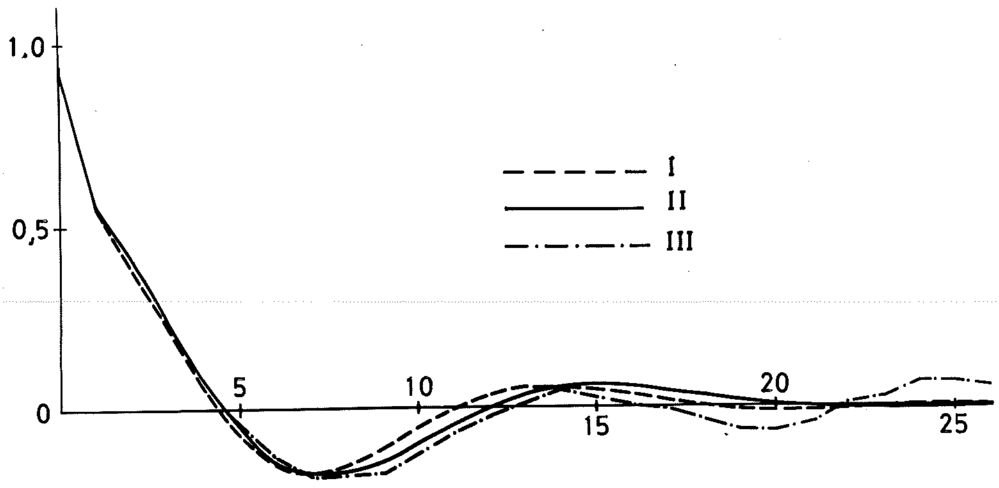


Fig. 6 - Autocorrelation functions for the original system (I), for the identified system (II) and for the generated time series (III)

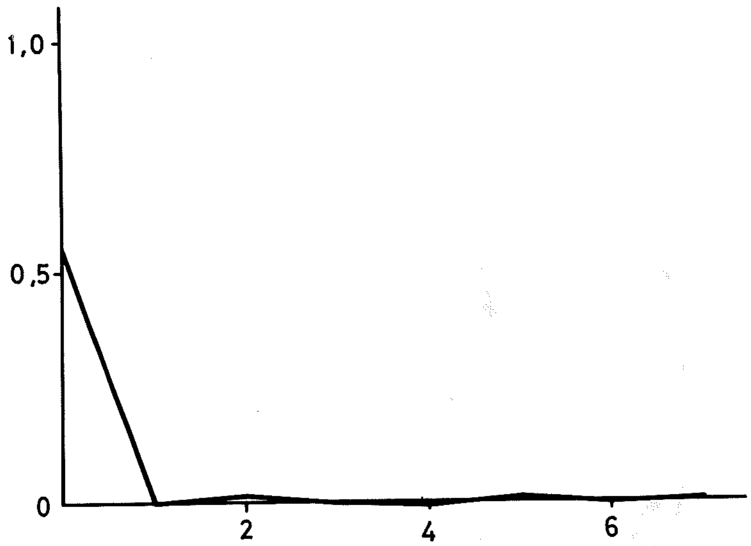


Fig. 7 - Autocorrelation function for the residuals from the second order model (example 2)

Example 3

In this example time series obtained from gyro measurements are identified. A special method was used for the measuring. It is shortly described here.

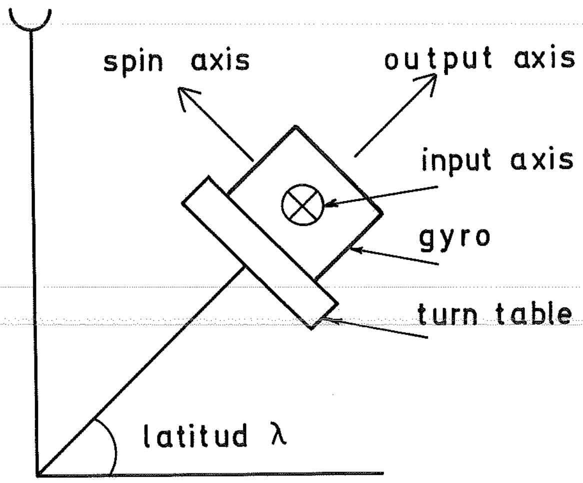


Fig. 8 - Experimental arrangement

We have a gyro on a turntable. Suppose the table is fixed but that the gyro can rotate around the output axis. If a disturbance torque is applied to the output axis and it turns the input axis the angle φ from the east-west position, the input axis is precessed. This causes a torque around the output axis trying to turn the input axis back to east-west position ($\varphi = 0$). When this torque is equal the disturbance torque, a position of equilibrium ($\varphi \neq 0$) is achieved. φ is now a measure of the drift. Now instead fix the gyro on a rotatory table and let the output from the gyro turn the table. The angle φ is now easier to measure, it is the angle the table has been turned from its initial position. It is constant if the drift is constant. Otherwise a curve $\varphi(t)$ as in figure 9 is obtained. The curve is a part from one of the original measurements. The curve is then smoothed by hand and the value of the sample points are read. The two identified series have been obtained from FOA, Stockholm, where the measurements have been performed.

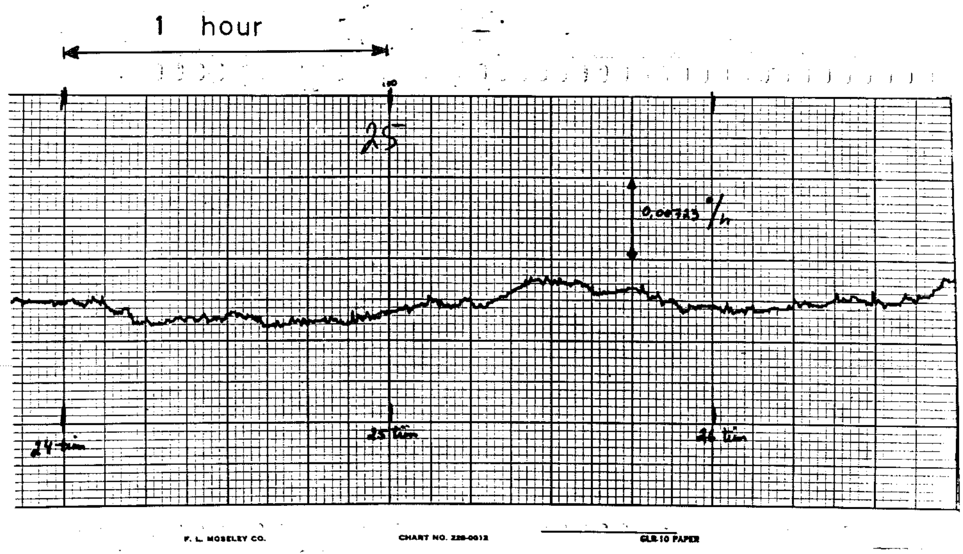


Fig. 9 - Part of the original measurements (series I)

The first series (I) is about 75 hours long and the sample interval is 15 minutes. The identified time series is plotted in figure 10. The second series (II) is about 125 hours long with a sample interval of 6 minutes. This series is also plotted (figure 11). We will already here warn of the obtained results of this example. There are strange parts of the series that may indicate missing of data at the measurements or an eventual fault in the writing device. If we only identify the first half of series I we obtain, however, a result that is near what we expected, that is a time constant of 2-3 hours. This is only reliable if we suppose that the behaviour of the gyro during the second half is not typical for the gyro drift.

However, the results of the identification is given in Table 10.

Notations:	I	Series I
	I:A	Series I (the first 160 data)
	I:B	Series I (the last 157 data)
	II	Series II
	II:A	Series II (Data 433-1159)

Series	Order of system n	Coefficients $a_1 \dots a_n$	Coefficients $c_1 \dots c_n$	λ	V
I	1	-0.948 ± 0.019	0.329 ± 0.055	4.19	2691.6
	2	-1.462 ± 0.117 0.468 ± 0.115	-0.301 ± 0.108 -0.342 ± 0.049	4.07	2537.2
	3	-1.400 ± 0.302 0.658 ± 0.438 -0.248 ± 0.176	-0.216 ± 0.301 -0.087 ± 0.150 -0.098 ± 0.119	4.02	2486.6
	4	-0.964 ± 0.141 0.702 ± 0.102 -1.070 ± 0.113 0.347 ± 0.135	0.223 ± 0.133 0.512 ± 0.077 -0.438 ± 0.124 -0.311 ± 0.058	3.96	2411.6
I:A	1	-0.861 ± 0.045	0.299 ± 0.084	4.23	1433.6
	2	0.077 ± 0.259 -0.809 ± 0.220	1.236 ± 0.272 0.276 ± 0.124	4.23	1433.3
I:B	1	-0.934 ± 0.031	0.308 ± 0.082	4.61	1560.8
	2	-1.509 ± 0.193 0.519 ± 0.187	-0.351 ± 0.181 -0.301 ± 0.072	4.53	1505.4
II	1	-0.990 ± 0.004	0.090 ± 0.028	0.339	71.88
	2	-1.814 ± 0.129 0.816 ± 0.128	-0.738 ± 0.130 -0.095 ± 0.028	0.339	71.78
II:A	1	-0.993 ± 0.005	-0.008 ± 0.040	2.91	3079.3
	2	-1.758 ± 0.111 0.758 ± 0.110	-0.782 ± 0.115 -0.056 ± 0.042	2.90	3043.4

Table 10 - Estimates

Diagrams show the autocorrelation functions for some of the cases (figures 12 - 14).

If we test the residuals for normality we find that it is most unlikely that they are normally distributed. (Examples: Series I: $\chi^2 = 300 > \chi^2_{0.001} = 44$ and Series II: $\chi^2 = 850 > \chi^2_{0.001} = 68$). The residuals are plotted for some of the series (figures 10 and 11).

Now we have an example where the assumption of normality does not hold and therefore the obtained results may not be quite correct. This may also be an explanation of the difficulties that appeared.

First the presence of a root near the rand of the unit circle prolongs the time for the identification, because much more iterations must be performed. This is a consequence of that the system is near its stability limit. Sometimes it also happens that a root outside the unit circle is obtained. The presence of such a root may be caused by sudden large variations in the series. In such cases it may not be possible to obtain a stable system using the model we have chosen (1). Furthermore we can give another example of the same kind. The time series in figure 15 has been identified. No significant reduction of loss function occurs when the order of the system is increased. The first order system is not stable.

Results: $a_1 = -1.015 \pm 0.012$
$c_1 = -0.113 \pm 0.067$

This must be caused by the rapidly increasing part at the end of the series.

To demonstrate this two time series were generated and identified. 300 variables $\{y(t)\}$ were generated from the system

$$y(t) = 0.5 \frac{1 + 0.3z^{-1}}{1 - 0.9z^{-1}} e(t)$$

where $\{e(t)\}$ were random floating point numbers, normally distributed with a mean of 0 and a variance of 1. Then two series were generated:

$$\text{I:}\quad y'(t) = \begin{cases} y(t) & t = 1, \ldots, 270 \\ y(t) + k \cdot (t - 270) & t = 271, \ldots, 300 \end{cases}$$

$$\text{II:}\quad y''(t) = \begin{cases} y(t) & t = 1, \ldots, 95 \\ y(t) + k \cdot (t - 95) & t = 96, \ldots, 105 \\ y(t) + 10k & t = 106, \ldots, 195 \\ y(t) + 10k + k(t - 195) & t = 196, \ldots, 205 \\ y(t) + 20k & t = 206, \ldots, 300 \end{cases}$$

These series ($\{y'(t)\}$ and $\{y''(t)\}$) were then identified for different values of k. The results are given in Table 11.

Series	k	a_1	c_1	λ	V
I,II	0	-0.864	0.295	0.467	32.70
I	0.03	-0.887	0.289	0.471	33.25
	0.1	-0.916	0.282	0.476	33.96
	0.2	-0.965	0.270	0.485	35.24
	1.0	-1.042	0.274	0.527	41.70
II	0.05	-0.885	0.285	0.471	33.25
	0.1	-0.931	0.262	0.482	34.91
	0.3	-0.981	0.217	0.526	41.55
	0.8	-0.996	0.158	0.716	76.93

Table 11 - Estimates for different values of k

From these results we conclude that if the time series are not stationary, results similar those obtained for gyro data may occur. The increase of the parameter a_1 is equivalent with an increase of the time constant of the system.

Then in some cases several minima appeared. Series I has been more carefully examined by choosing many different values of $c_1, \ldots, c_n$.

Results in Table 12.

	a_1	a_2	c_1	c_2	Loss function
Minimum 1	0.037	-0.933	1.317	0.331	2689.5
Minimum 2	-1.462	0.468	-0.301	-0.342	2537.2
Maximum	-0.238	-0.673	1.039	0.234	2691.4

Table 12 - Example of several minima

The maximum can only be obtained if we do not use approximate second derivatives for a sufficient number of iterations. The value of the loss function at different points of the c-parameter space is shown in figure 16. We also give another example of a case when two local minima of the loss function were found. The series analysed is series B, n = 2 in example 4.

	a_1	a_2	c_1	c_2	Loss function
Minimum 1	-1.632	0.632	-0.439	-0.391	80537.4
Minimum 2	-0.216	-0.673	1.060	0.198	84203.2

Table 13 - Example of several minima

The value of the loss function at different points is given in figure 17.

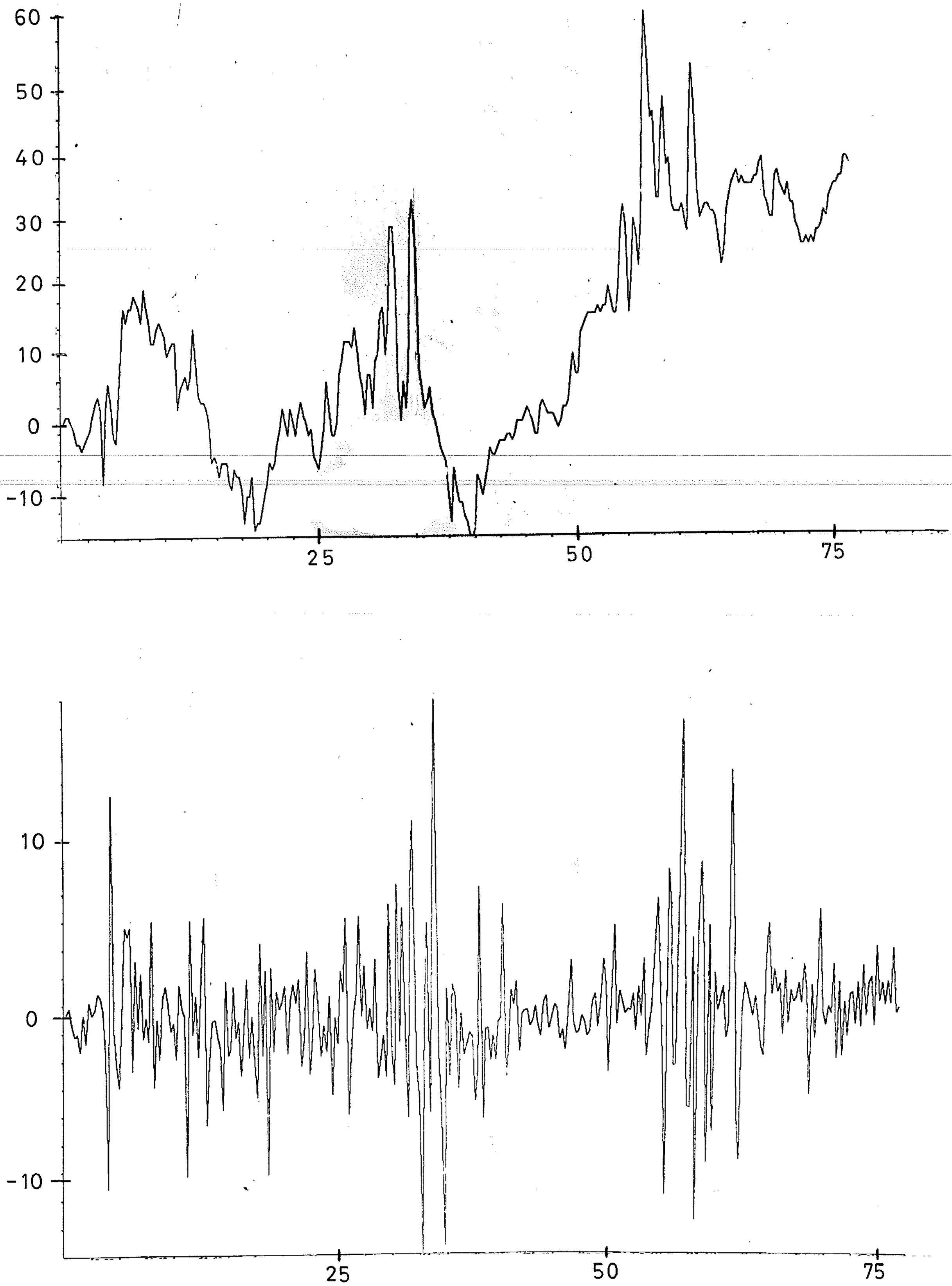


Fig. 10 - The gyro time series I and the residuals for the first order model

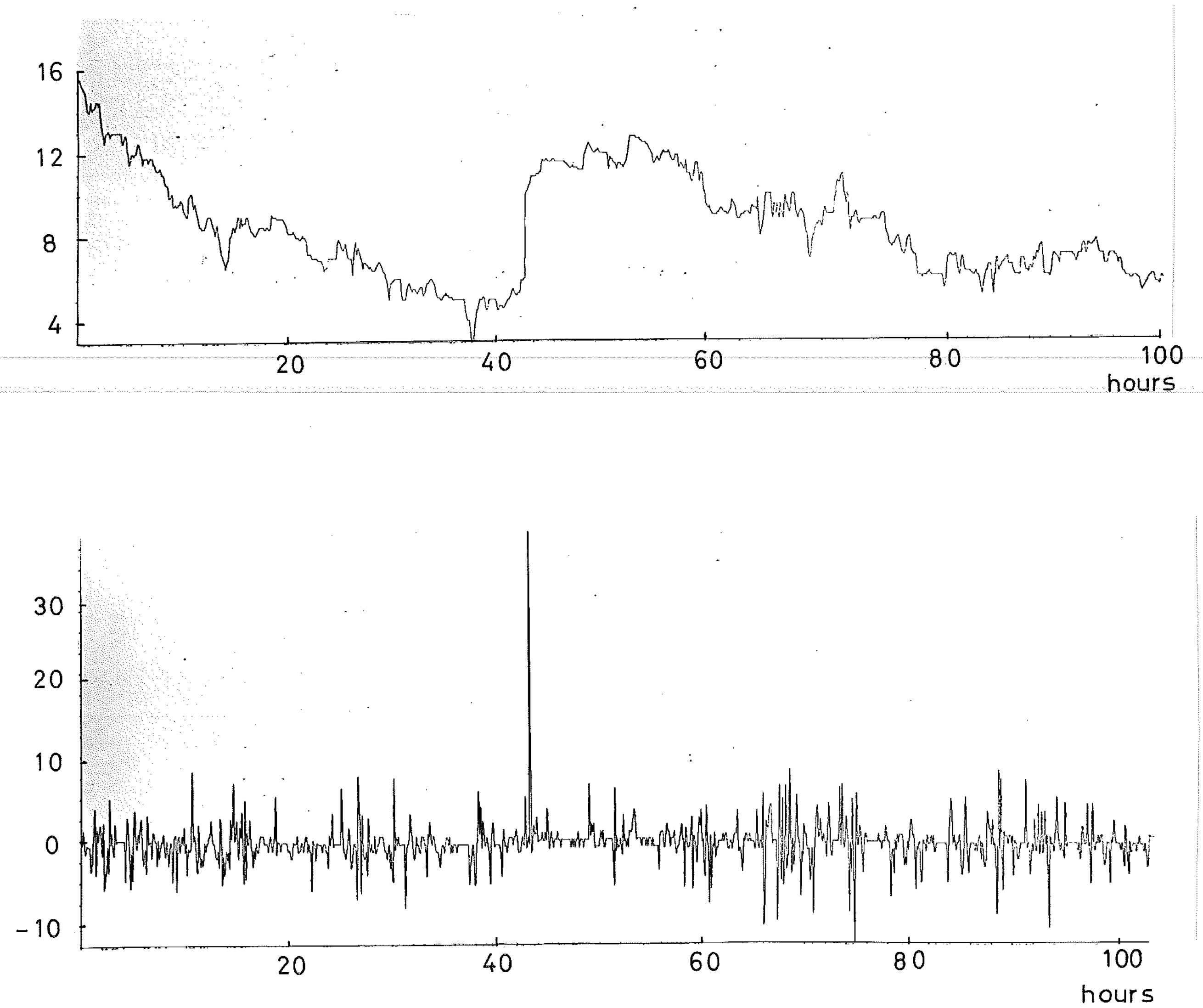


Fig. 11 - Part of the gyro series II and the residuals for the first order model

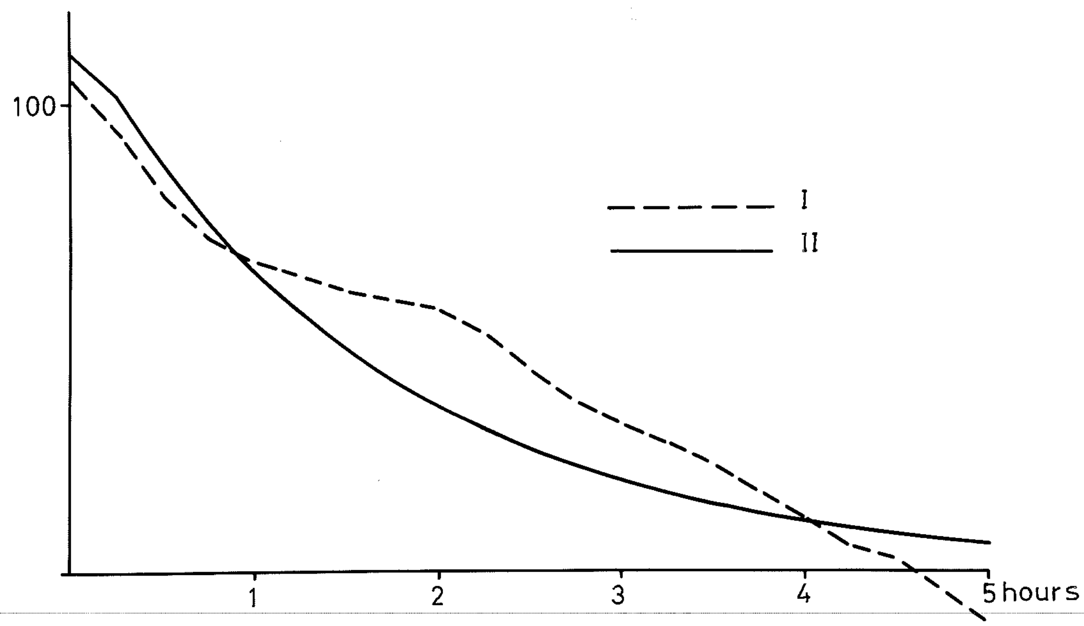


Fig. 12 - Autocorrelation functions for the time series I:A (I) and for the identified first order model (II)

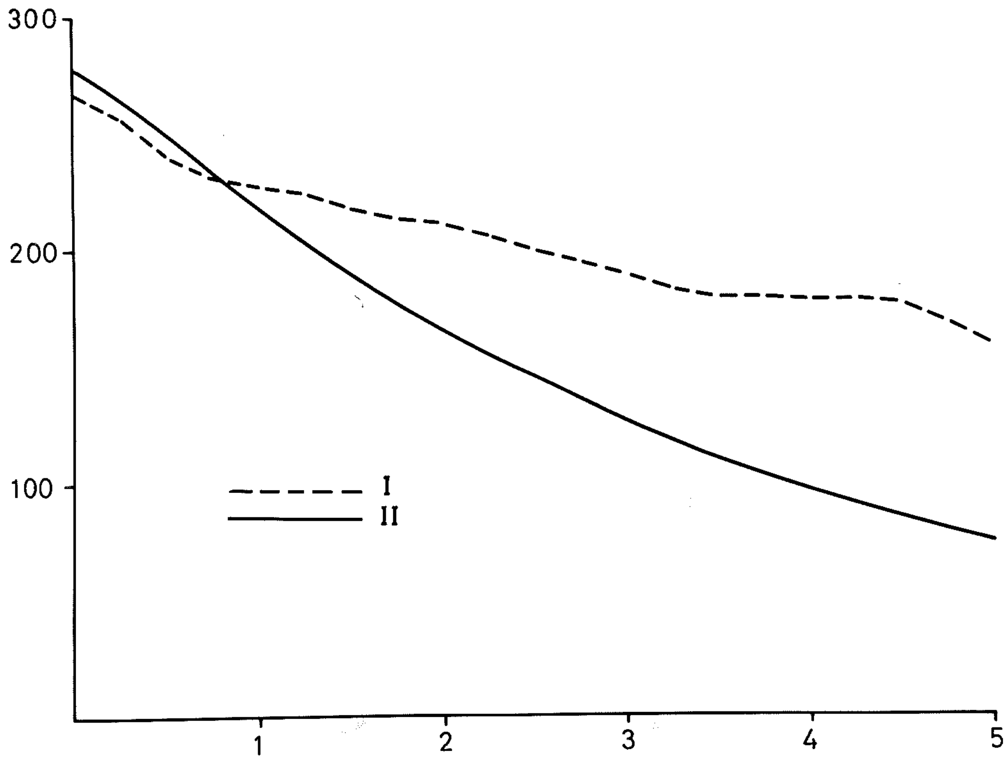


Fig. 13 - Autocorrelation functions for the time series I:B (I) and for the identified first order model (II)

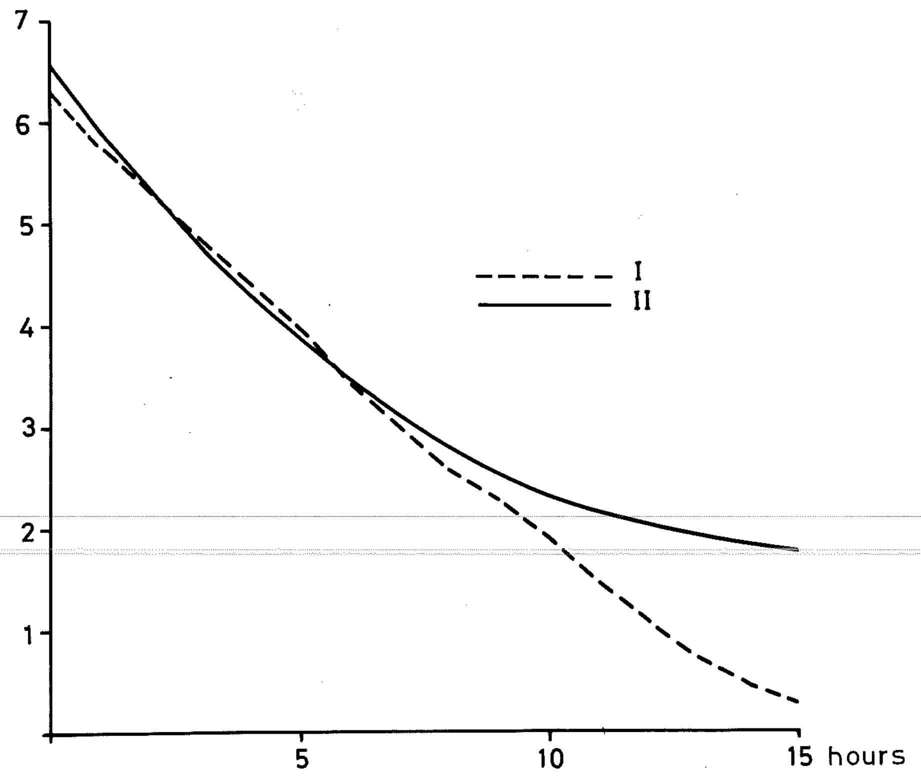


Fig. 14 - Autocorrelation functions for the time series II (I) and for the identified first order model (II)

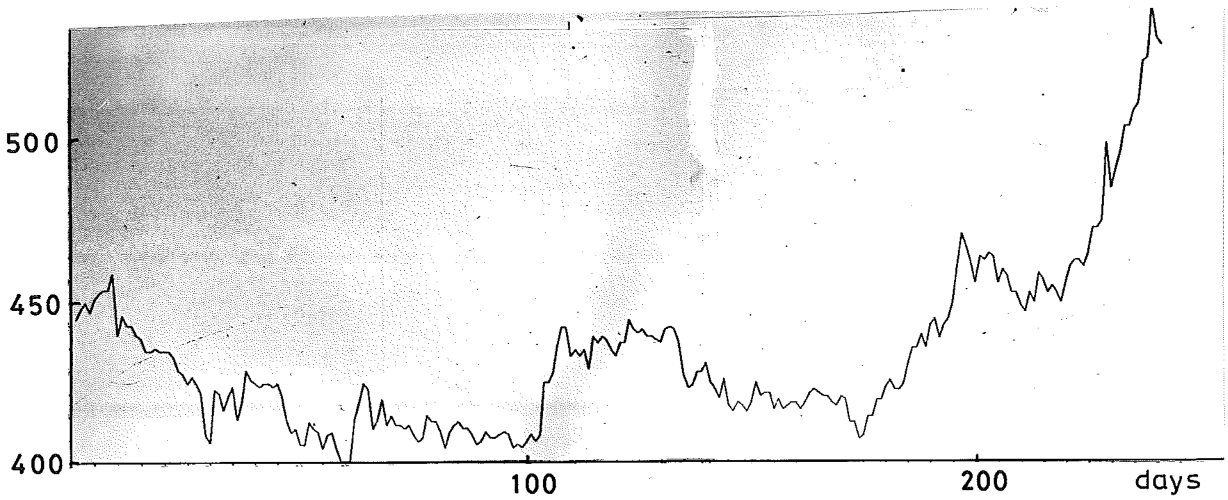


Fig. 15 - Time series from stock market

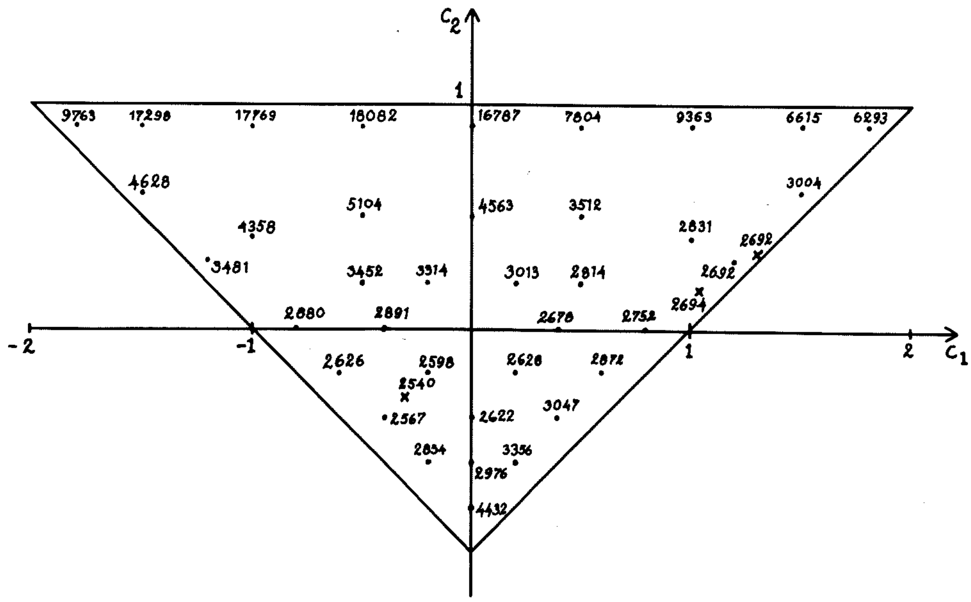


Fig. 16 - The value of the loss function for different values of the c-coefficients for time series I, example 3, second order model. Local minima and maxima denoted by x

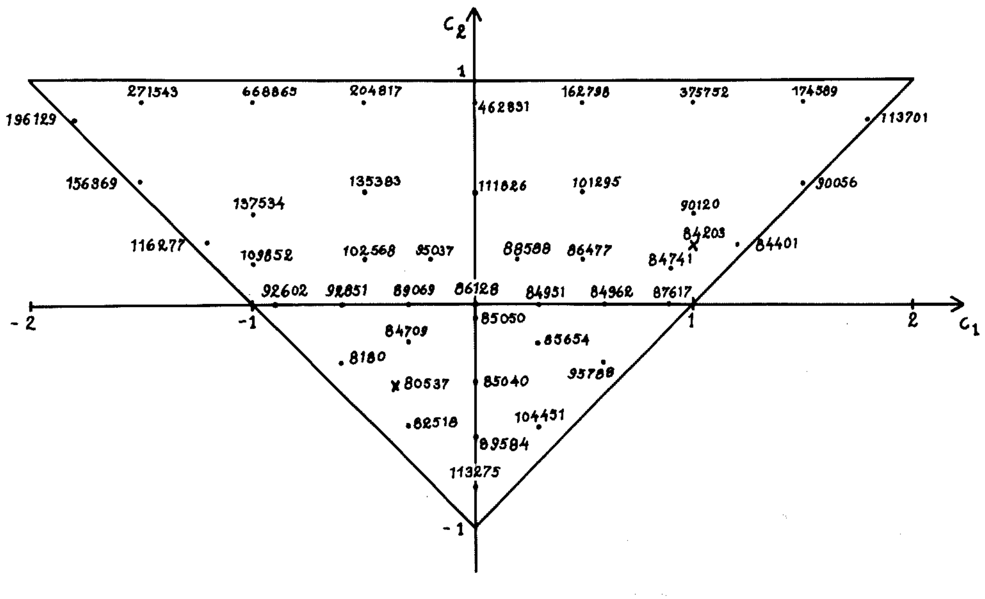


Fig. 17 - The value of the loss function for different values of the c-coefficients for time series B, example 4, second order model. Local minima denoted by x

Example 4

The procedure has also been used for identification of a number of economical time series. The series are taken from {8}. Three series are examined below. The series are

A the monthly sales of Warmdot filters (1953-1962)

B the number of passengers on international airlines (1949-1960)

C the monthly imports by company B (1950-1959)

The time series are plotted in figures 18-20. They are rather short and this may explain some of the difficulties that appear when they are identified.

These series are interesting because of the obvious cycles of series A, B and C and because of the apparent trend of the series B. Therefore these series are tests of how well the procedure will find such characteristics. The autocorrelation functions computed from the series are shown in figures 21-23 together with them obtained from the system models given by the identification procedure.

The procedure gives the following results:

Time series	Order of system	Coefficients $a_1 \ldots a_n$	Coefficients $c_1 \ldots c_n$	λ	V_n
A	n = 1	-0.475 ± 0.131	0.114 ± 0.140	33.56	67579
	n = 2	-1.718 ± 0.021 0.986 ± 0.021	-1.589 ± 0.077 0.799 ± 0.078	27.30	44730
	n = 3	-0.682 ± 0.069 -0.793 ± 0.114 1.022 ± 0.069	-0.553 ± 0.101 -0.846 ± 0.112 0.828 ± 0.099	27.26	44578
B	n = 1	-0.932 ± 0.033	0.344 ± 0.086	34.34	84910
	n = 2	-1.632 ± 0.108 0.632 ± 0.107	-0.439 ± 0.115 -0.391 ± 0.082	33.45	80537
	n = 3	-2.687 ± 0.042 2.637 ± 0.085 0.948 ± 0.046	-1.616 ± 0.110 0.659 ± 0.211 0.147 ± 0.111	30.73	68003
C	n = 1	0.352 ± 0.114	0.824 ± 0.062	412.53	10210980
	n = 2	-0.648 ± 0.117 -0.351 ± 0.119	-0.185 ± 0.065 -0.833 ± 0.067	410.42	10106904
	n = 3	0.471 ± 0.106 0.470 ± 0.107 -0.527 ± 0.107	1.179 ± 0.110 1.103 ± 0.116 0.267 ± 0.114	293.58	5171315
	n = 4	-0.517 ± 0.106 0.236 ± 0.048 -0.752 ± 0.049 0.764 ± 0.106	0.098 ± 0.175 0.176 ± 0.101 -0.546 ± 0.089 0.112 ± 0.171	277.37	4615986

Table 14 - Estimates

From these results it is clear that the procedure actually find both trends and cycles. Let us make some remarks.

Series A

The series A represents obviously a second order system and increasing the order of the model to n = 4 gives nothing. Note that for this series we obtain the same root of $A(z^{-1})$ and $C(z^{-1})$ for n = 3. We find that the resulting system

$$y(t) = 27.3 \frac{1 - 1.589z^{-1} + 0.799z^{-2}}{1 - 1.718z^{-1} + 0.986z^{-2}} e(t)$$

is an oscillating system with a period of 12 sampling intervals, that is one year. This is what was expected from a direct study of the series.

The residuals for n = 2 are plotted (figure 18) and tested for normality and independence. They are normal and independent.

Series B

The series B is partly described by a third order system

$$y(t) = 30.7 \frac{1 - 1.616z^{-1} + 0.659z^{-2} + 0.147z^{-3}}{1 - 2.687z^{-1} + 2.637z^{-2} - 0.948z^{-3}} e(t)$$

We see that $z \simeq 1$ is a root of the denominator and this fact is a proof of an existing trend in the examined series. Dividing the denominator by $(1 - z^{-1})$ we obtain $1 - 1.687z^{-1} + 0.950z^{-2}$ and this indicates that the system is oscillating with a period of one year.

If we increase the order of the model to n = 5 we, however, obtain significant reduction of the loss function both when increasing the order from n = 3 to n = 4 and from n = 4 to n = 5.

For n = 5 we obtain an indication that there is also a period of 6 months, something that may be seen from the plot of the series. In this case the increasing of order is combined with computational difficulties because of the root on or very near the rand of the unit circle. For n = 5 the approximate second derivatives must be used for more than 30 iterations. The obtained model is unstable.

A way out of this is to estimate and remove the trend before the identification procedure is applied to the time series. At least we can say that if $A(z^{-1})$ has a root on or very near the rand of the unit circle, numerical difficulties may arise.

The residuals for n = 3 is plotted (figure 19). The tests of normality and independence show that we cannot reject the hypothesis that the residuals are normal and that they are independent.

To test the proposition above the time series obtained as the differences of the original time series, was also identified. This is a method to remove the trend. We have

$$y'(t) = (1 - z^{-1})\ y(t) = \frac{C(z^{-1})}{A(z^{-1})}\ e(t) \qquad (23)$$

The model found here was of fourth order and the polynomial $A(z^{-1})$ of (23) has approximatively the same roots as the fifth order model for the original time series except the root on the rand of the unit circle, that is the periods of 6 and 12 months respectively are found. No such severe numerical difficulties appeared as when identifying the original series, but again two minima were found.

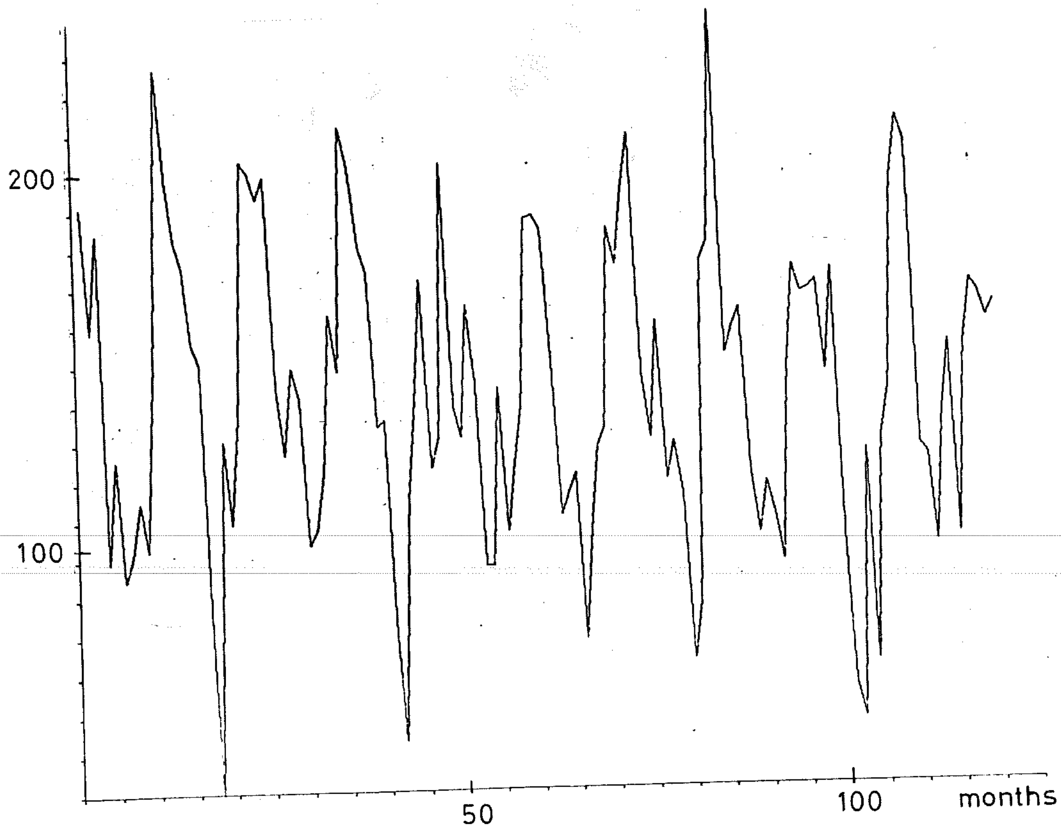
As an example we also give the one-step ahead predictor $\hat{y}(t|t-1)$ for this series, n = 3. As said before the obtained residuals may also be interpreted as the one-step ahead prediction errors (figure 24).

Series C

The series C seems to represent a fourth order system

$$y(t) = 277.4 \frac{1 - 0.098z^{-1} + 0.176z^{-2} - 0.546z^{-3} + 0.112z^{-4}}{1 - 0.517z^{-1} + 0.236z^{-2} - 0.752z^{-3} + 0.764z^{-4}} e(t)$$

The system is oscillating with periods of 3 respectively 12 months and this fact may be observed direct from the time series. The residuals are normal and independent. They are plotted in figure 20.



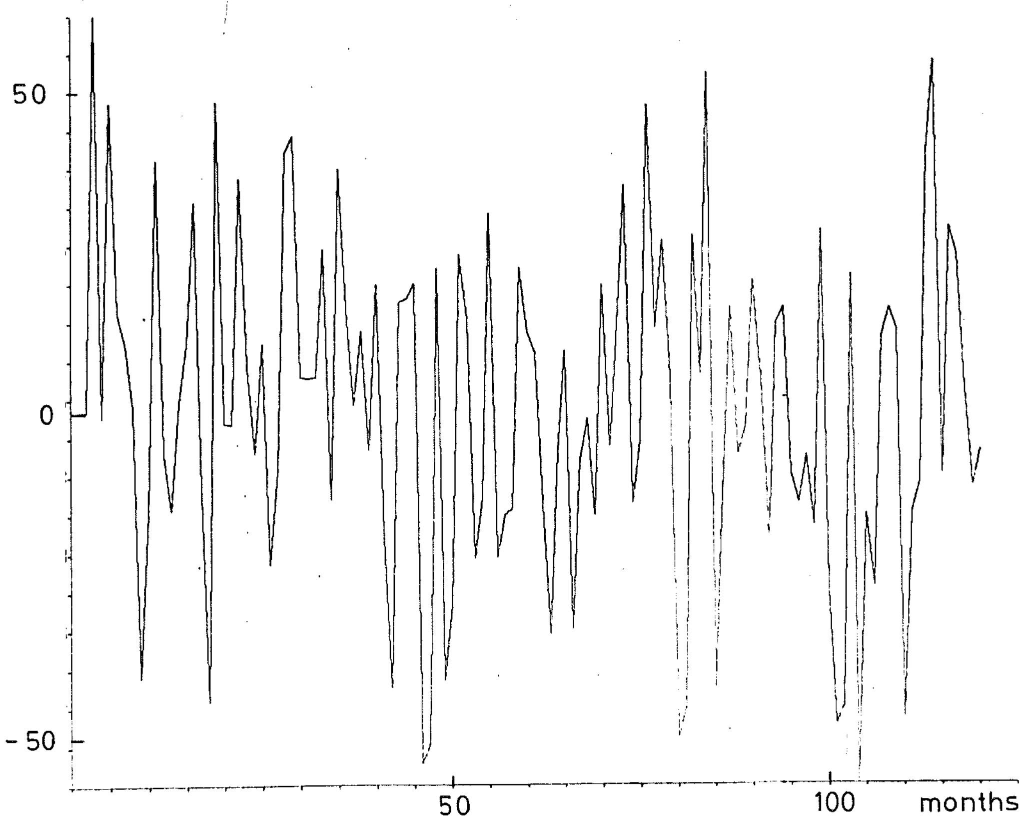


Fig. 18 - Time series A and the residuals for the second order model

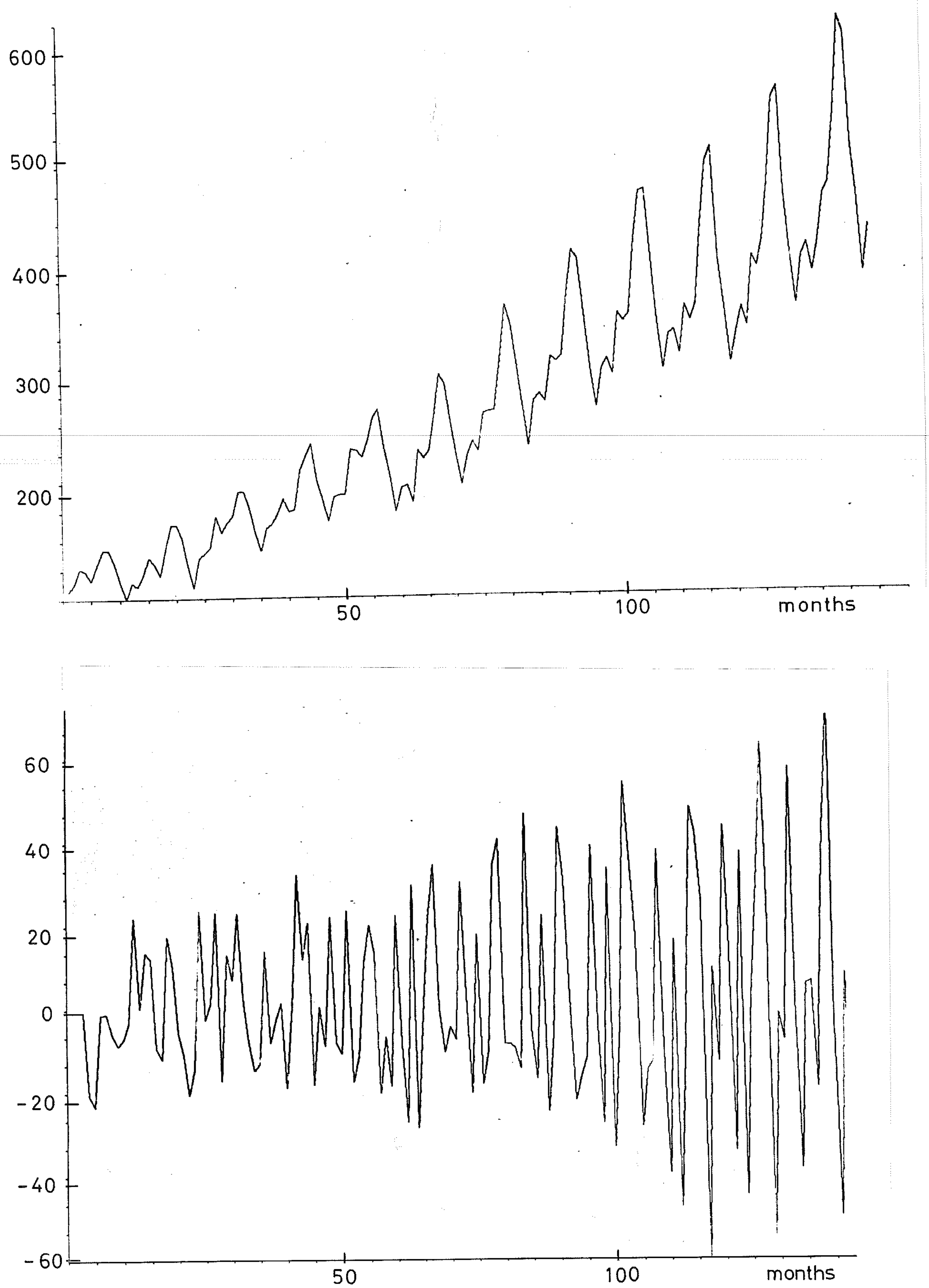
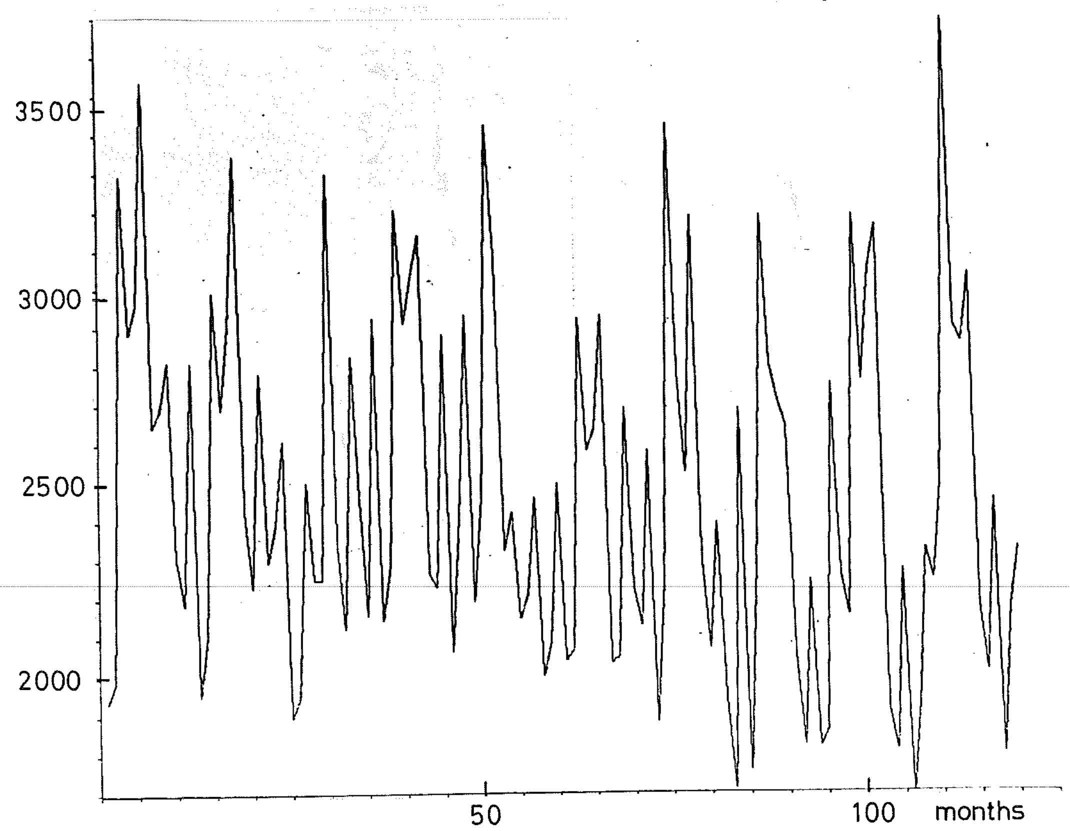


Fig. 19 - Time series B and the residuals for the third order model



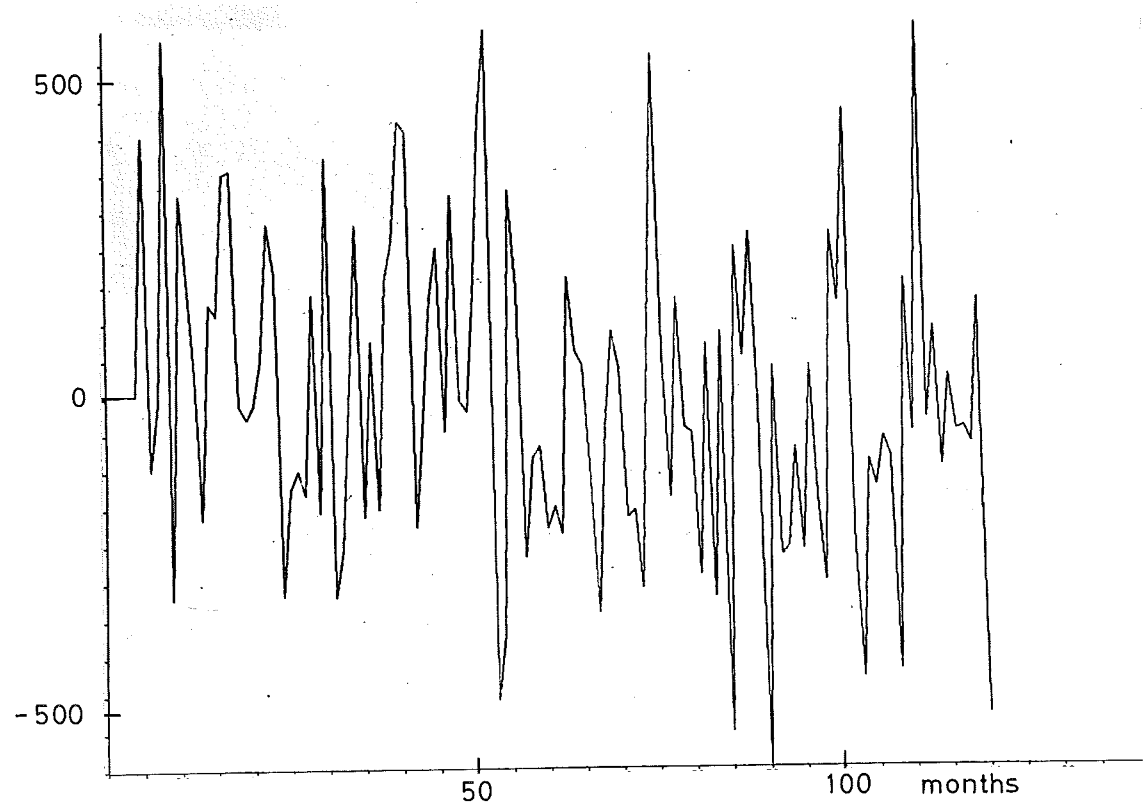


Fig. 20 - Time series C and the residuals for the fourth order model

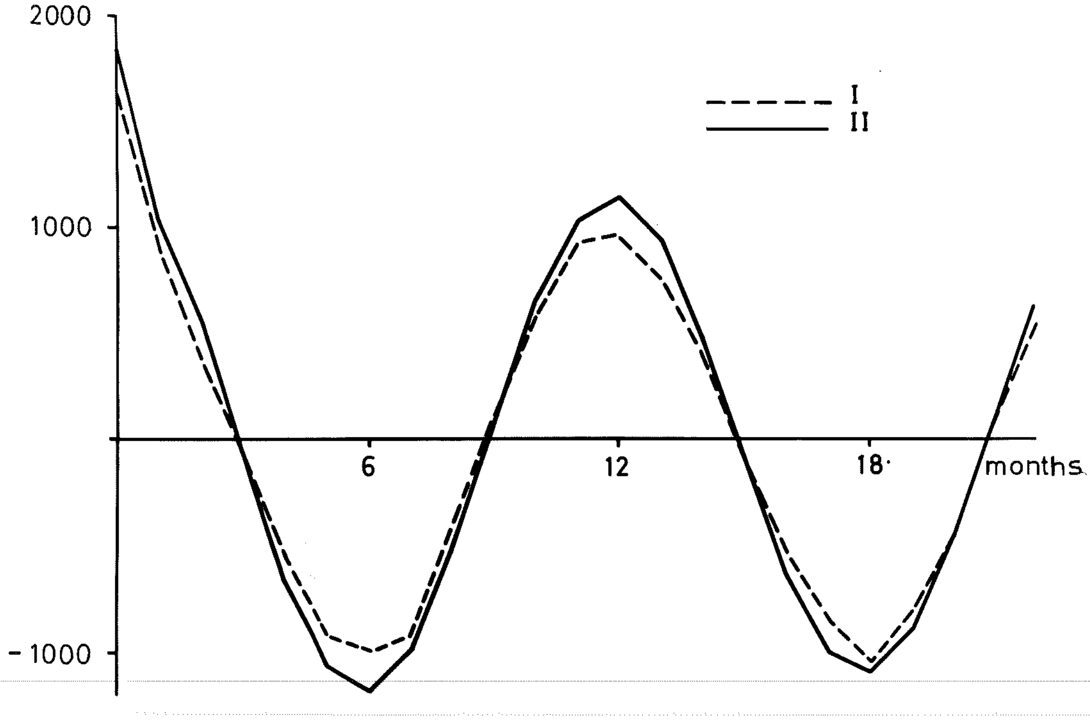


Fig. 21 - Autocorrelation functions for time series A (I) and for the identified model (II)

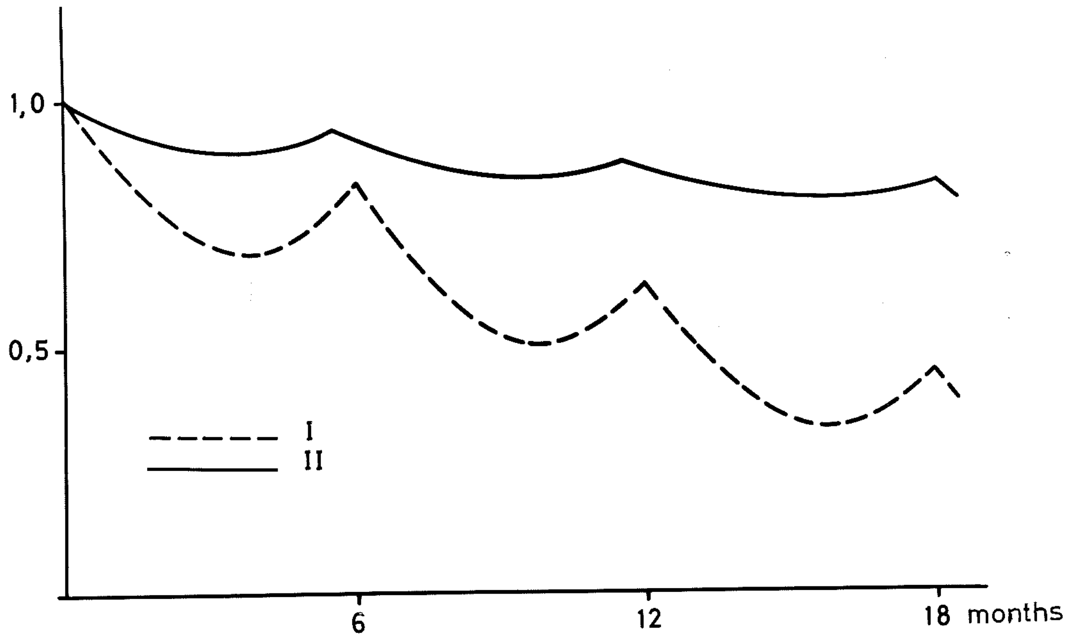
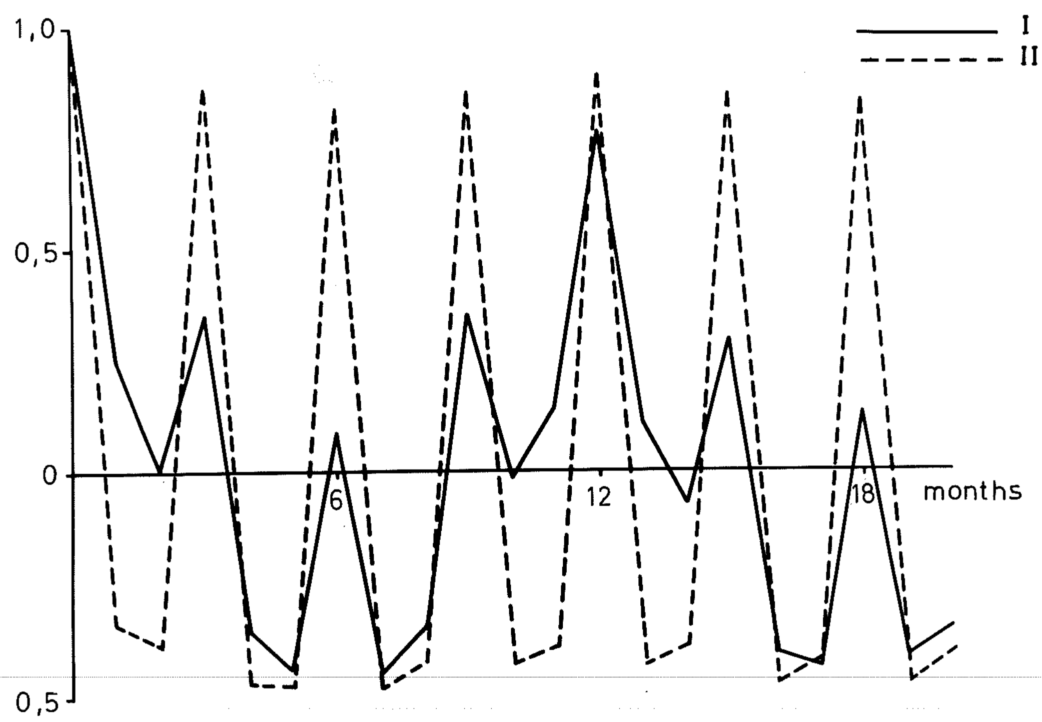
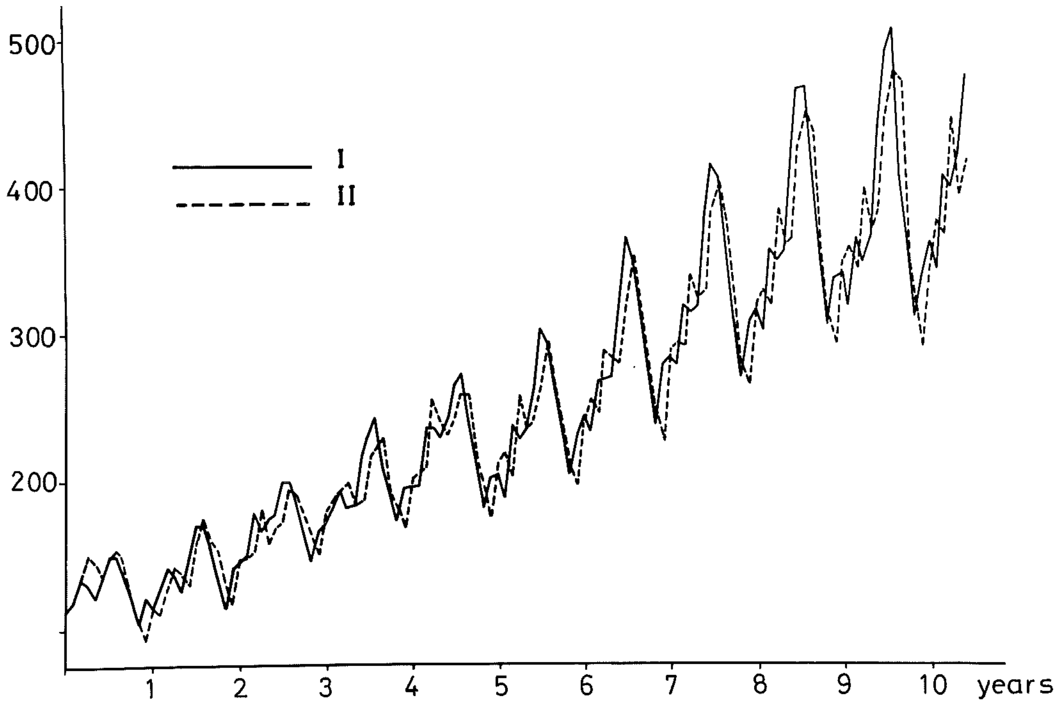


Fig. 22 - Autocorrelation functions for time series B (I) and for the identified model (II)



<u>Fig. 23</u> - Autocorrelation functions for time series C (I) and for the identified model (II)



<u>Fig. 24</u> - Time series B (I) and the one-step-ahead predictor (II) using the third order model

Example 5

This example is a medical application. By placing two electrodes on the scalp and leading via suitable amplifiers to a recorder, a record is obtained of the electrical activity of the brain. It is called an electroencephalogram (eeg). Characteristic for this is the presence of different rhythms, the alfa-rhythm (8-13 Hz), the beta-rhythm (14-30 Hz), the delta-rhythm (1-3.5 Hz) and the theta-rhythm (4-7 Hz). The alfa-rhythm is the most striking one and is normally present when eeg-record is taken of a person who refrains of mental activity and who keeps his eyes closed (average amplitude 50 μV). The theta-rhythm has a low amplitude (about 10 μV). The largest waves are the delta group which usually appear during sleep, attaining an amplitude of 100 μV. The presence of a tumour may also cause delta wave activity. Then eeg-tracings which show evidence of a focus of delta wave production, may give a possibility of localizing tumours. Of course it is then valuable to have a method determining the present rhythms, and that is why this identification procedure is applied to eeg-records {9}.

The data used are two eeg-records obtained simultaneously from the same person at different points on the scalp. The sampling interval was 0.0053 sec and the total amount of data of each series was 5838 , that is the length of records was about 31 sec.

These records are also investigated at the Royal Institute of Technology, Stockholm (Institutionen för teletransmissionsteori), wherefrom we received the records. They use a different method for the identification. Figure 25 shows a part of the plotted time series. There are also diagrams showing the autocorrelation functions for series I:A and I:D, computed direct from given data (figures 26-27).

The two time series, here called I and II, have now been examined in the following way. The series have been divided into four almost equal parts (A, B, C and D, where A, B and C have 1470 data each and D 1428 data), which have been identified separately. The whole series I has also been identified. Below some of the results are given (Table 15-16).

Series	Order of system n	Coefficients $a_1 \ldots a_n$	Coefficients $c_1 \ldots c_n$	λ	V_n
I	2	-1.7773±0.0071	0.2600±0.0151	9.213	247758
		0.8886±0.0070	0.1109±0.0144		
	4	-1.6553±0.0076	0.3885±0.0146	9.004	236651
		1.6718±0.0081	1.1944±0.0153		
		-1.6538±0.0094	0.3041±0.0155		
		0.8748±0.0080	0.1926±0.0155		
I:A	1	-0.9316±0.0095	0.7926±0.0116	15.608	179045
	2	-1.7727±0.0150	0.2496±0.0307	9.196	62162
		0.8815±0.0146	0.0765±0.0296		
	3	-2.3570±0.1006	-0.3487±0.1020	9.144	61460
		1.8960±0.1826	-0.1166±0.0312		
		-0.4957±0.0925	-0.1269±0.0281		
	4	-3.3986±0.0406	-1.4094±0.0482	9.047	60157
		4.6300±0.1148	0.5488±0.0716		
		-2.9808±0.1245	0.0541±0.0473		
		0.7753±0.0497	0.0499±0.0369		
	5	-2.9054±0.4768	-0.9150±0.4756	9.036	60012
		2.9639±1.6225	-0.1358±0.6735		
		-0.6849±2.2061	0.3406±0.2677		
		-0.7423±1.4138	0.1345±0.0375		
		0.4109±0.3654	-0.0290±0.0527		
	6	-2.2534±0.0585	-0.2755±0.0633	8.974	59198
		1.2911±0.1521	-0.5327±0.0654		
		-0.2079±0.1833	-0.7875±0.0621		
		1.4031±0.1666	0.6732±0.0432		
		-1.8902±0.1144	0.2153±0.0336		
		0.6950±0.0421	0.1074±0.0327		
	8	-1.5469±0.0440	0.4169±0.0482	8.899	58207
		0.0848±0.0774	-0.3146±0.0463		
		0.7075±0.0841	-0.3650±0.0332		
		-0.6190±0.0825	-0.7566±0.0354		
		0.8509±0.0868	-0.2120±0.0472		
		0.3762±0.0933	0.7884±0.0398		
		-1.4566±0.0782	0.3026±0.0328		
		0.7000±0.0420	0.1902±0.0292		

I:B	1	-0.9333±0.0094	0.7887±0.0118	14.472	1453941
	2	-1.7908±0.0146 0.8494±0.0141	0.2047±0.0322 0.0105±0.0308	8.596	54306
	3	-1.7822±0.2163 0.8508±0.3866 0.0329±0.1949	0.2119±0.2147 -0.0443±0.0527 -0.1333±0.0291	8.540	53598
	4	-1.2801±0.3955 -0.2205±0.5616 0.7793±0.2396 -0.1444±0.1970	0.7144±0.3956 -0.1120±0.2720 0.1824±0.0417 -0.0609±0.0604	8.537	53561
	6	-1.3789±0.0578 -1.2463±0.0631 2.7405±0.1083 -0.2778±0.1232 -1.4133±0.0528 0.6471±0.0667	0.6028±0.0624 -1.3904±0.0627 -0.5504±0.1109 0.7060±0.0927 0.3324±0.0421 0.1064±0.0353	8.329	50993
I:C	1	-0.9305±0.0096	0.7917±0.0117	19.114	268520
	2	-1.7944±0.0124 0.9085±0.0122	0.1957±0.0290 0.1921±0.0263	9.708	69272
	3	-1.6753±0.5124 0.6718±0.9131 0.1276±0.4594	0.3064±0.5151 0.1802±0.1320 -0.0880±0.1305	9.671	68738
	4	-2.6720±0.0612 3.3864±0.1476 -2.4147±0.1335 0.8168±0.0452	-0.6860±0.0703 0.9455±0.0423 0.3836±0.0320 0.1584±0.0335	9.638	68278
	6	-3.7868±0.1125 6.1251±0.4636 -5.4206±0.8534 2.7663±0.8659 -0.7443±0.4728 0.0713±0.1092	-1.8820±0.1093 1.5651±0.2559 -0.8917±0.2697 0.4741±0.1411 0.3712±0.0721 0.2583±0.0416	9.234	62665

I:D	1	-0.9297±0.0098	0.8272±0.0112	16.048	183872
	2	-1.7491±0.0159 0.8655±0.0156	0.3909±0.0308 0.1410±0.0292	9.148	59748
	3	-1.3425±0.5501 0.1430±0.9590 0.3623±0.4734	0.7940±0.5506 0.2786±0.2246 0.0142±0.0953	9.139	59633
	4	-3.4228±0.0209 4.7747±0.0523 -3.1851±0.0531 0.8664±0.0208	-1.3448±0.0341 0.5224±0.0509 0.1499±0.0454 0.0129±0.0342	8.875	56233
II:1	1	-0.9331±0.0094	0.7701±0.0129	16.008	188345
	2	-1.7624±0.0160 0.8673±0.0156	0.2292±0.0311 0.1293±0.0285	9.609	67869
	3	-2.3117±0.0995 1.8056±0.1811 0.4490±0.0914	-0.3389±0.1010 -0.0617±0.0287 -0.1728±0.0272	9.531	66774
	4	-2.2099±0.2039 1.1065±0.5528 0.5806±0.5171 -0.4743±0.1679	-0.2328±0.2063 -0.5435±0.1472 -0.2424±0.0329 0.0176±0.0529	9.522	66635
	6	-1.7814±0.0521 -0.9633±0.0458 3.1828±0.0905 -0.4999±0.0794 -1.7726±0.0504 0.8357±0.0835	0.2316±0.0586 -1.7780±0.0655 -0.5817±0.0665 0.7688±0.0814 0.2641±0.0369 0.0952±0.0354	9.372	64555

Table 15 - Estimates

As a supplement the roots of the polynomials are also given:

Series	Order of system n	Roots of $A(z^{-1})$	Roots of $C(z^{-1})$
I	2	0.889 ± 0.314 i	0.130 ± 0.307 i
	4	0.885 ± 0.315 i -0.058 ± 0.994 i	-0.147 ± 0.421 i -0.047 ± 0.983 i
I:A	1	-0.932	0.793
	2	0.886 ± 0.310 i	-0.125 ± 0.247 i
	3	0.905 ± 0.294 i 0.548	-0.195 ± 0.366 i 0.739
	4	0.926 ± 0.300 i 0.773 ± 0.470 i	0.097 ± 0.230 i 0.802 ± 0.393 i
	5	0.951 ± 0.297 i 0.810 ± 0.306 i -0.634 ± 0.730 i	-0.156 ± 0.307 i 0.931 ± 0.286 i -0.637 ± 0.742 i
	8	0.951 ± 0.297 i 0.800 ± 0.306 i -0.923 ± 0.341 i -0.055 ± 0.995 i	-0.165 ± 0.428 i 0.930 ± 0.285 i -0.929 ± 0.340 i -0.044 ± 0.987 i
I:B	1	-0.933	0.789
	2	0.895 ± 0.305 i	-0.102 ± 0.005 i
	3	0.909 ± 0.300 i -0.036	-0.343 ± 0.404 i 0.474
	4	0.909 ± 0.298 i 0.211 -0.749	-0.285 ± 0.295 i -0.678 0.534
	6	0.935 ± 0.319 i 0.748 ± 0.323 i -0.993 ± 0.120 i	-0.185 ± 0.291 i 0.878 ± 0.348 i -0.993 ± 0.117 i

Table 16 - Roots of polynomials $A(z^{-1})$ and $C(z^{-1})$ for some of the series

Notice that the coefficients of the polynomials are only given with an accuracy of four decimals and that a small change of them may cause a very large change of the roots. Furthermore the standard deviations of the parameters are rather great. Note, however, that especially for the polynomial $A(z^{-1})$ some of the roots are almost unchanged when increasing the order of the system. For instance consider series I:A. The roots

$0.886 \pm 0.310\, i$

appear for $n = 2$.

Then we have for higher order systems almost the same roots:

$n = 3$	$0.905 \pm 0.294\, i$
$n = 4$	$0.926 \pm 0.300\, i$
$n = 6$	$0.951 \pm 0.297\, i$
$n = 8$	$0.951 \pm 0.297\, i$

As mentioned in section 3 the F-test gives that the system is of very high order, at least sixth. Even if we use a model of eighth order we obtain a significant reduction of the loss function (for instance series I:A). However if we examine the roots of the polynomials $A(z^{-1})$ and $C(z^{-1})$ we find that the order of the system is perhaps only four. Already for the fourth order system the α- and β-rhythms are most often obtained.

Notice also for this example the mentioned fact that a small reduction of loss function is combined with significantly increasing standard deviations of the parameters (series I:A $n = 4$ and $n = 5$). Notice also the example of the fact that the loss function sometimes only reduces slowly for increasing order of model and then suddenly have a considerable reduction (series I:B for instance). Why this is the case is not fully explained.

Figures 26-27 show the autocorrelation functions of the identified system model for series I:A and I:D ($n = 4$).

Figure 28 shows the residuals for series I:A ($n = 4$). They are normal and independent. Series I:A has a lower test quantity when tested for normality than series I:C and this is an indication of better normality. This is a result also obtained at the investigation at the Royal Institute of Technology with a different method.

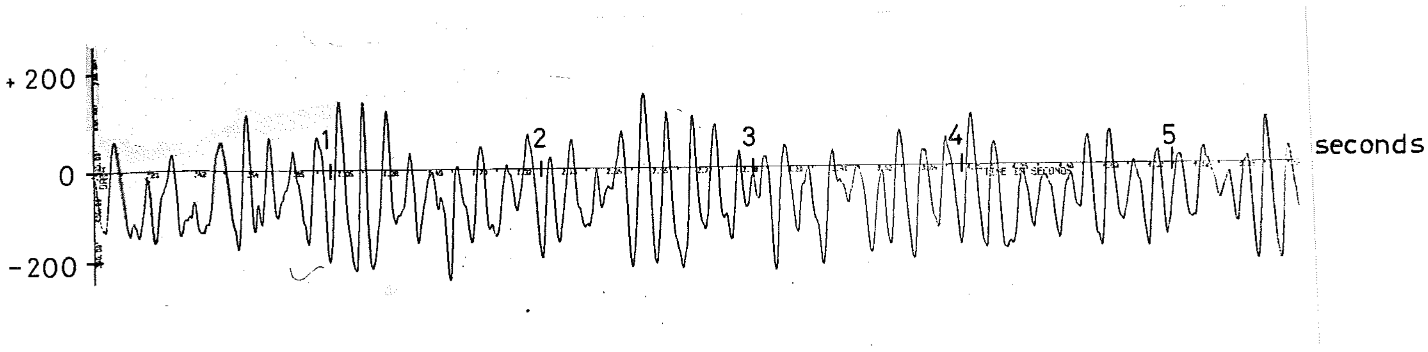


Fig. 25 - Part of the EEG-records

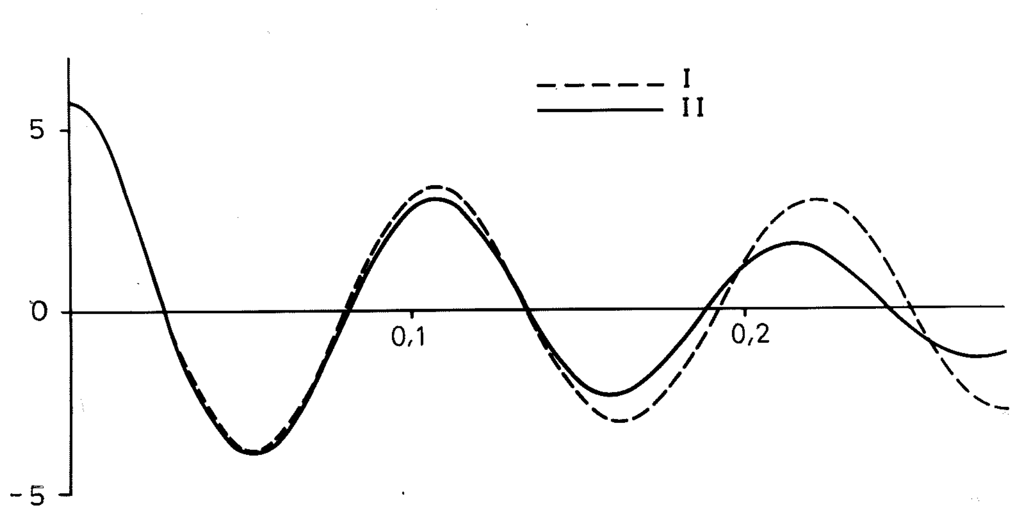


Fig. 26 - Autocorrelation functions for series I:A (I) and for the identified model (II)

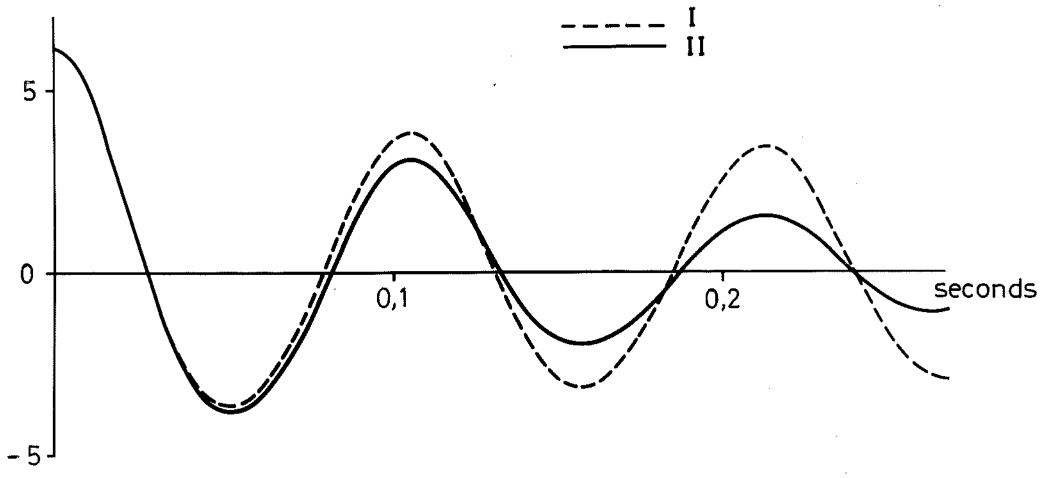


Fig. 27 - Autocorrelation functions for series I:D (I) and for the identified model (II)

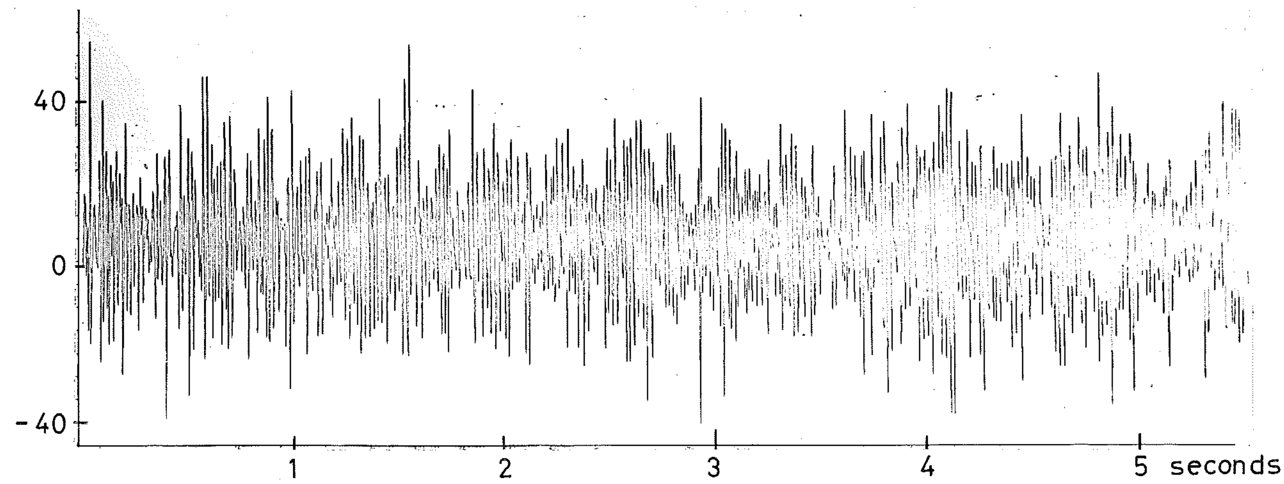


Fig. 28 - Residuals from the fourth order model (time series I:A)

Example 6

To test the effectiveness of the identification procedure when very short time series are identified, the following numerical experiment was carried out.

An artificial time series $\{y(t)\}$ was generated using the recursive equations

$$x_1(t + 1) = x_2(t)$$
$$x_2(t + 1) = -0.7x_1(t) + 1.5x_2(t) + 0.0548v(t)$$
$$y(t) = x_1(t) + 0.158w(t) \tag{24}$$

$\{v(t)\}$ and $\{w(t)\}$ are sequences of normal (0,1) and independent variables. They were generated by the subroutine RANSS.

It can be shown that the random process $\{y(t)\}$ generated by (24) is equivalent to the process generated by

$$y(t) = 0.19738 \frac{1 - 1.12808z^{-1} + 0.44858z^{-2}}{1 - 1.5z^{-1} + 0.7z^{-2}} e(t)$$

where $\{e(t)\}$ is a sequence of independent normal (0,1) variables.

Two series of 600 values of y(t) were generated. The first 100 variables of each of them were not used in order to achieve stationarity. Then the rest of the series was divided into series of 50 values each. In all 20 series were obtained and identified with a second order model. The sequences y(t) are plotted in figure 29.

The following results were obtained by the identification procedure:

Time series	λ	a_1	a_2	c_1	c_2
1	0.1647	-1.4949	0.6722	-1.0779	0.4038
2	0.1950	-1.5678	0.8030	-1.3946	0.6573
3	0.1658	-0.6122	0.3363	-0.4436	0.5526
4	0.1880	-0.1648	-0.3036	0.2422	0.1122
5	0.1715	-1.3275	0.6297	-1.0559	0.8608
6	0.1813	-0.3845	-0.1567	-0.0935	0.3928
7	0.1500	1.0108	0.2739	1.4590	0.8976
8	0.1740	-0.3335	-0.4933	-0.1188	-0.5626
9	0.2117	0.2216	-0.7810	0.7076	-0.3094
10	0.1505	-0.3524	-0.1495	0.0846	0.2033
11	0.1323	0.0099	-0.2275	0.5441	0.3301
12	0.1868	0.2308	-0.2963	0.6089	-0.2282
13	0.2430	0.2257	-0.4036	0.7196	-0.0726
14	0.1799	0.6091	-0.1802	0.9056	-0.1348
15	0.1826	-0.0087	-0.6942	0.3432	-0.1662
16	0.1874	-1.7636	1.0077	-1.4472	0.7061
17	0.1889	-0.8946	-0.2403	-0.7383	-0.3918
18	0.1540	-1.7655	0.8124	-2.0982	1.2694
19	0.1980	-1.6809	0.8625	-1.1208	0.3798
20	0.1265	-1.7308	0.9721	-1.4184	0.6838
True values	0.1974	-1.5000	0.7000	-1.1281	0.4486

Tables 17 - Estimates

We see that the results often differ very much from the correct values. This is also indicated by the obtained great standard deviations, for instance time series 3:

$a_1 = -0.6122 \pm 0.4894$

$a_2 = 0.3363 \pm 0.3363$

$c_1 = -0.4436 \pm 0.4218$

$c_2 = 0.5526 \pm 0.3003$

This depends on the shortness of the series. Numerical difficulties arose when identifying. More iterations than normally had to be done. Series 12 could not be identified when starting at the ordinary starting value. Even some of the models (16, 18) have roots outside the unit circle. If we, however, compare the autocorrelation functions obtained direct from the time series according to (21) and the autocorrelation functions obtained from the system models above, we find that the last ones most often are the best ones. Oscillations and other phenomena are often damped out at the identification. Strikingly is also that the functions are almost identical for the first time lags. Examples are given below. They show the two autocorrelation functions mentioned above and the exact correlation function for system (23) (figures 30-34). Some of the mentioned difficulties may be explained by the fact that some of the time series perhaps actually represent a first order or a third order system, but that they are only identified with a second order model.

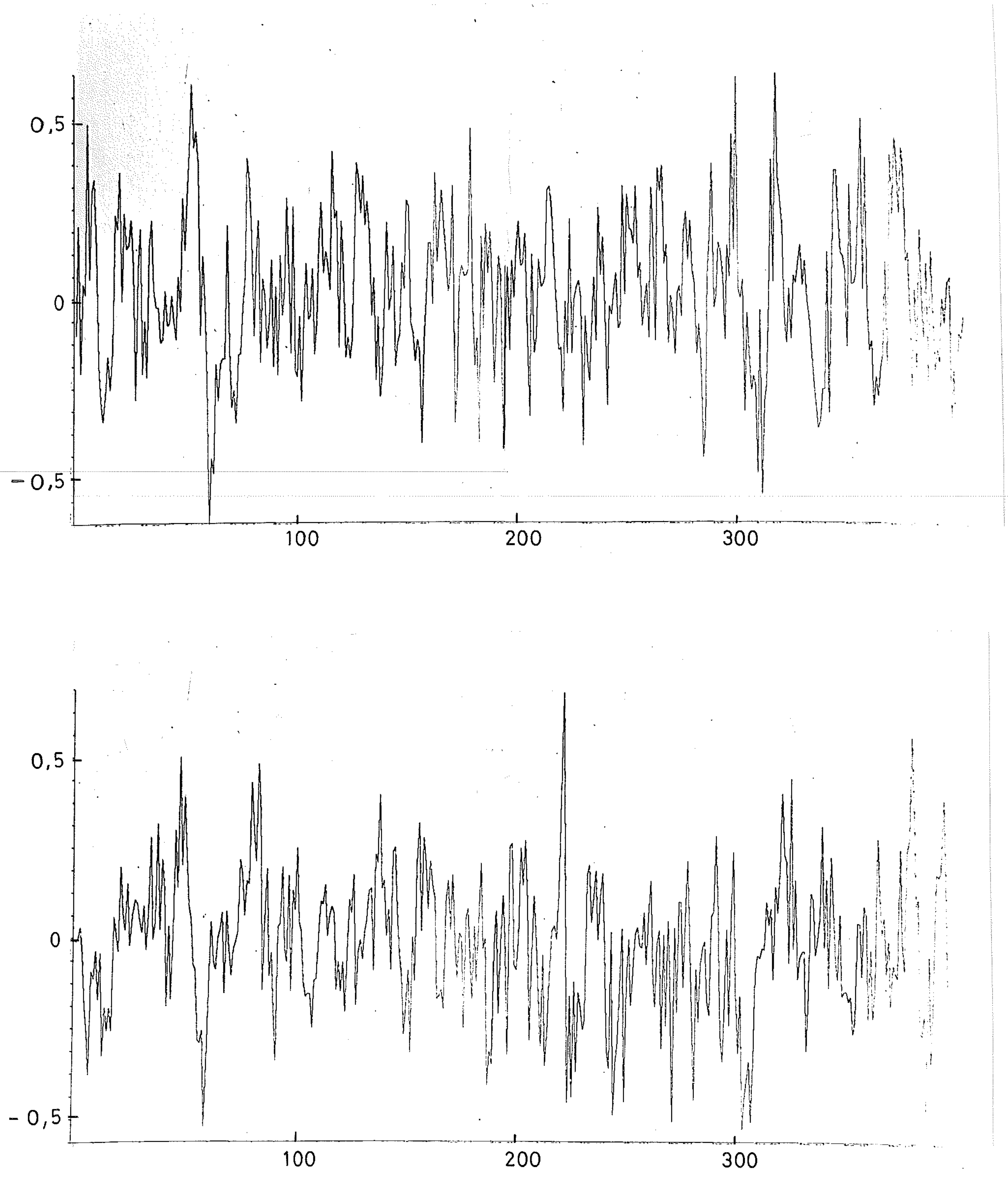


Fig. 29 - The first parts of the time series used in example 6

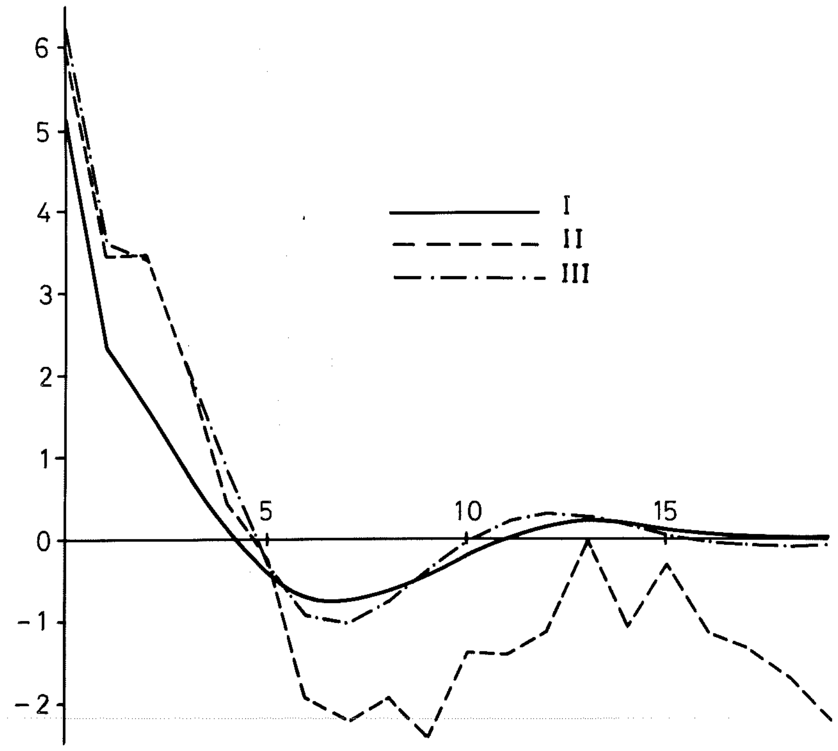


Fig. 30 - Autocorrelation functions for the generating system (I), the time series (II) and for the identified model (III). Series 5.

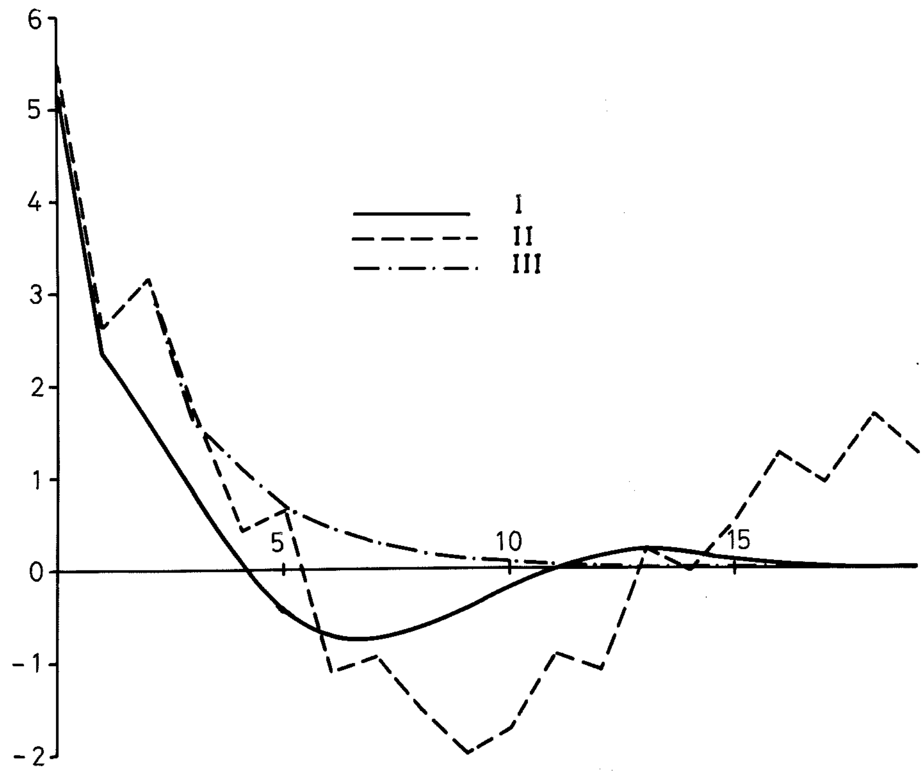


Fig. 31 - Autocorrelation functions for the generating system (I), the time series (II) and for the identified model (III). Series 6.

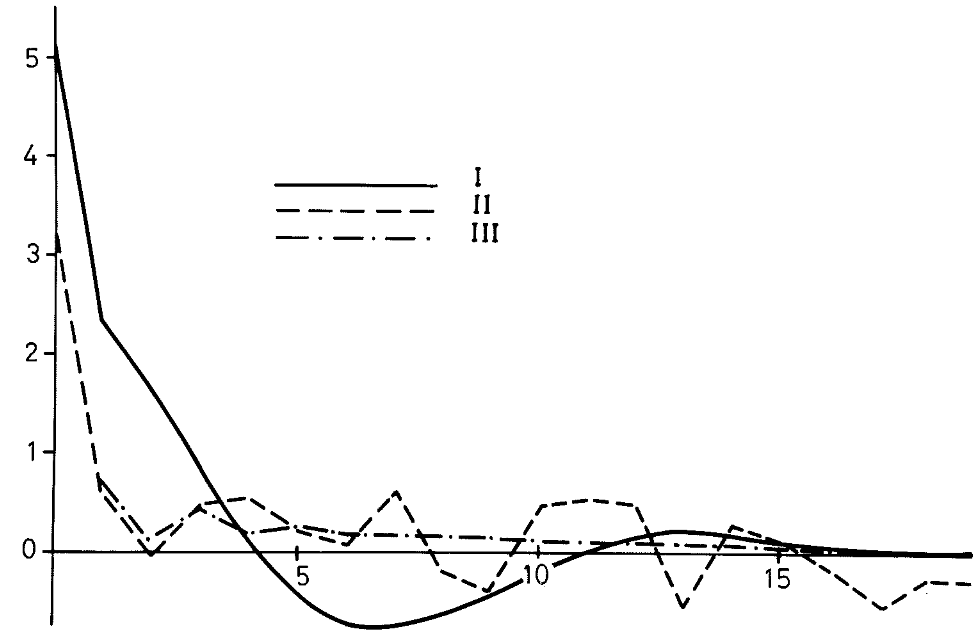


Fig. 32 - Autocorrelation functions for the generating system (I), the time series (II) and for the identified model (III). Series 8.

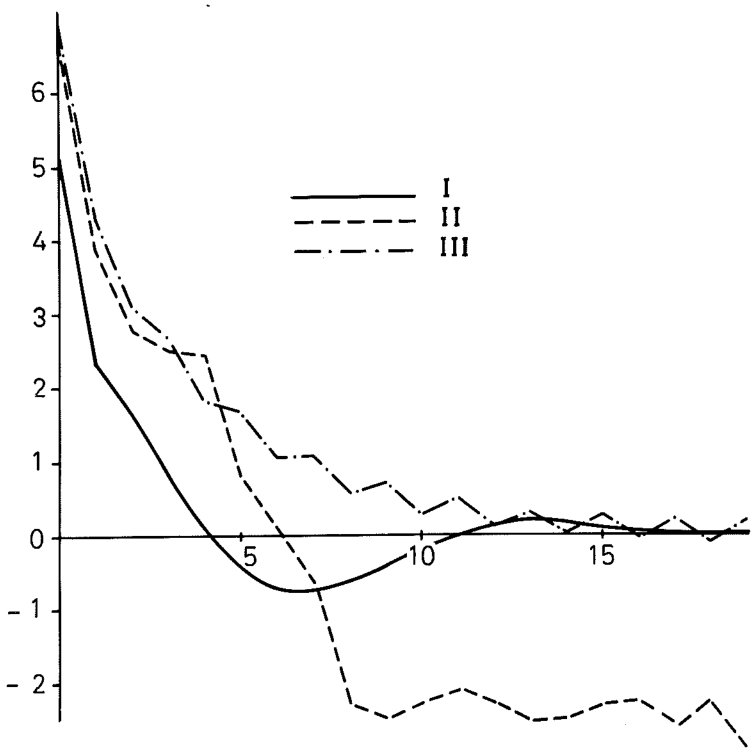


Fig. 33 - Autocorrelation functions for the generating system (I), the time series (II) and for the identified model (III). Series 9.

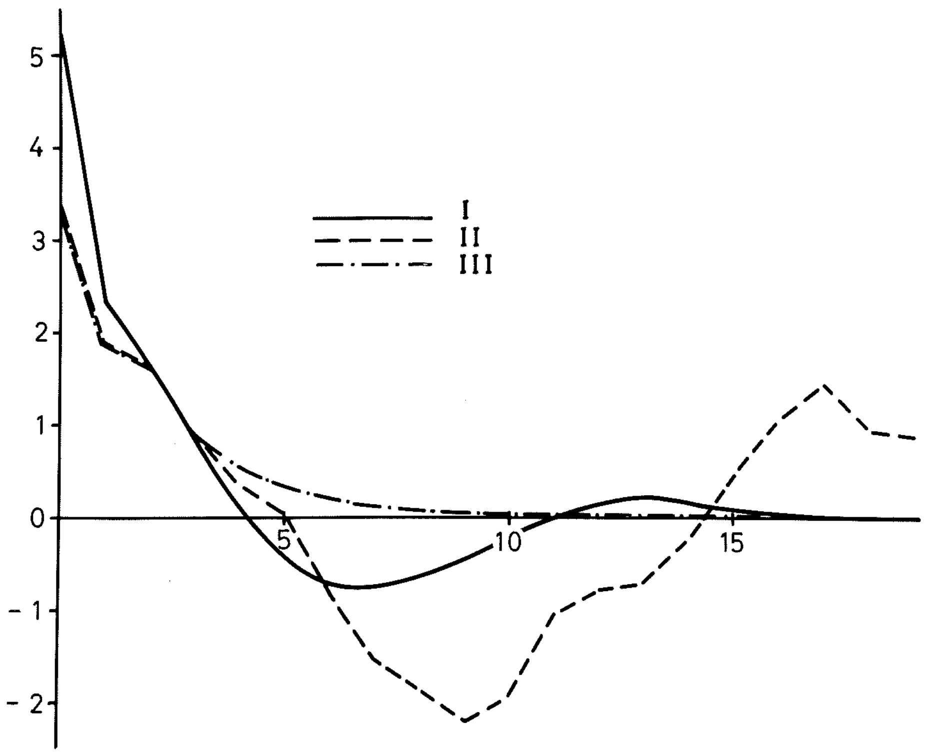


Fig. 34 - Autocorrelation functions for the generating system (I), the time series (II) and for the identified model (III). Series 10.

REFERENCES

{1} K-J Åström and T Bohlin "Numerical Identification of Linear Dynamic Systems from Normal Operating Records", Proceedings of the IFAC Conference on Self-Adaptive Control Systems, Teddington (1965).

{2} E Andersson "Processidentifiering med datamaskin" (1966).

{3} G A Bekey and R B McGhee "Gradient Methods for the Optimization of Dynamic System Parameters by Hybrid Computation", in Computing Methods in Optimization Problems, Balakrishnan and Neustadt, Academic Press, New York (1964).

{4} M G Kendall and A Stuart "The advanced theory of statistics", Griffin & Co, London (1963).

{5} C-E Fröberg "Lärobok i numerisk analys", Studentlitteratur, Lund (1964).

{6} H Cramér "Sannolikhetskalkylen och några av dess användningar", Almqvist & Wiksell, Uppsala (1961).

{7} J S Bendat and A G Piersol "Measurements and Analysis of Random Data", John Wiley, New York (1966).

{8} R G Brown "Smoothing, Forecasting and Prediction of Discrete Time Series", Prentice Hall (1963).

{9} C A Keek and E Neil "Samson Wrights Applied Physiology", Oxford University Press (1964).

```
      PROGRAM IDENT
C     NB    RUN THROUGH THE PROGRAM NB TIMES
C     I=0   INTEGER DATA
C     I=1   REAL DATA
C     NP    NUMBER OF DATA
C     JA=0  ALL DATA IN TIME SERIES USED
C     JA=1  ONLY DATA MB-MC USED
C     JO=0  NO STARTING VALUES OF C(I)
C     JO=1  STARTING VALUES FOR C(I),I=JP,JJ
C     JT=0  IDENTIFICATION PROCEDURE USE DATA WITH MEAN=0
C     JT=1  IDENTIFICATION PROCEDURE USE INPUT DATA
C     DATACARD 0  NB
C                 FORMAT(I5)
C     DATACARD 1  I,NP,JA,MB,MC,JJ,JP,JO,JT
C                 FORMAT(9I5)
C     DATACARD 2  FORMAT SPECIFICATIONS FOR INPUT DATA
C                 FORMAT(A5,A5,A5)
C     DATACARD 3  IA,IB,IC,ID,IE,IF,IG,IH,IJ,IK,IL,IM,IN,IO
C                 FORMAT(14I2)
C                 THE IDENTIFICATION MADE FOR ORDER OF SYSTEM IA-IB
C                 AND THE CALL SEQUENCE USED IS
C                 CALL PERPS(  ,  ,IC,ID,IE,IM)
C                 CALL PERPS(  ,  ,IF,IG,IH,IN)
C                 CALL PERPS(  ,  ,IJ,IK,IL,IO)
C     DATACARD 4- INPUT DATA. FORMAT ACCORDING TO FORMAT SPECIFICATIONS
C                 ABOVE
C      LAST DATACARD  EVENTUAL STARTING VALUES FOR C(I)
C                 FORMAT(10F5.4)
      DIMENSION IDATA(7000),IV(3)
      COMMON EE,V,Y,E(20),C(40),EC(40),V1(40),VCC(60),ECC(60),
     1 V2(40,40),DAT(7000)
      READ 104,NB
  104 FORMAT(I5)
      DO 10 NA=1,NB
      READ 100,I,NP,JA,MB,MC,JJ,JP,JO,JT
  100 FORMAT(9I5)
      READ 102,(IV(K),K=1,3)
  102 FORMAT(A5,A5,A5)
      READ 101,IA,IB,IC,ID,IE,IF,IG,IH,IJ,IK,IL,IM,IN,IO
  101 FORMAT(14I2)
      IF(I)  2,1,2
    1 READ IV,(IDATA(K),K=1,NP)
      DO 17 IO=1,NP
   17 DAT(IO)=IDATA(IO)
      GO TO 6
    2 READ IV,(DAT(K),K=1,NP)
    6 IF(JA)  7,9,7
    7 ND=MC-MB+1
      ME=MB-1
      DO 8 K=1,ND
      MF=ME+K
    8 DAT(K)=DAT(MF)
      NP=ND
    9 IF(JO)  12,11,12
   12 READ 103,(C(K),K=JP,JJ)
  103 FORMAT(10F5.4)
```

```
11 IF(JT) 19,20,19
20 S=0.0
   DO 13 K=1,NP
13 S=S+DAT(K)
   S=S/NP
   DO 14 K=1,NP
14 DAT(K)=DAT(K)-S
19 DO 10 KF=IA,IB
   CALL PERPS(KF,NP,IC,ID,IE,IM)
   CALL PERPS(KF,NP,IF,IG,IH,IN)
10 CALL PERPS(KF,NP,IJ,IK,IL,IO)
   CALL EXIT
   END
```

```
      SUBROUTINE PERPS(NO,NP,L1,L2,IT,IPRINT)
C
C     ROUTINE FOR PARAMETRIC ESTIMATION OF RATIONAL POWER SPECTRA
C
C     NO=  ORDER OF SYSTEM
C     NP=  LENGTH OF RECORD
C     L1=-1 GIVES COMMON ESTIMATION FOR STARTING VALUES
C     L1=0 GIVES COMMON ESTIMATION
C     L1=1 GIVES LEAST SQUARE ESTIMATE FOR STARTING VALUES
C     L1=2 GIVES LEAST SQUARE ESTIMATE FROM SPEC. ERROR-COEFF.
C     L2=  NUMBER OF ESTIMATIONS
C     L2=0  GIVES ESTIMATIONS UNTIL MAX.COEFF.CORR.LE.0.0001
C     IT=0  GIVES APPROXIMATIVE SECOND DERIVATIVES
C     IT=1  GIVES EXACT SECOND DERIVATIVES
C     IPRINT=0  MATRIX AND INVERSE MATRIX OF SECOND DERIVATIVES NOT
C     PRINTED OUT
C
C     C =  COEFF. VECTOR
C         (OUTPUT COEFF.,ERROR COEFF.)
C     CC=COEFF. CORR. VECTOR
C     V =  LOSS FUNCTION
C     V1=  GRADIENT OF V
C     V2=  SECOND DERIVATIVES OF V
C
C     ALFA= REDUCTION FACTOR FOR COEFF.-CORR.
C             USED WHEN THE LOSS FUNCTION IS GREATER THAN
C             THE PREVIOUS LOSS FUNCTION
C     REQUIRE DATA IN THE ARRAY DAT
C     OUTPUT(1) IN DAT(1). OUTPUT(2) IN DAT(2) AND SO ON
C
C     SUBROUTINE REQUIRED
C               LFOSD
C               SORV
C
      DIMENSION CC(40)
      COMMON EE,V,Y,E(20),C(40),EC(40),V1(40),VCC(80),ECC(80),
     1 V2(40,40),DAT(7000)
C
      NO=2*NO
      IF(L1-1) 5,3,1
    1 DO 2  I=1,NO
    2 C(I)=0.0
      GO TO 5
    3 DO 4  I=1,NO
    4 C(I)=0.0
C     COEFF. ZERO
    5 CONTINUE
C
C     START LOOP L
      IF(L2.EQ.0)  GO TO 1001
      DO 1000  L=1,L2
 1001 ALFA=1.0
      IF(L1.GE.1.OR.L1.LT.0)  V=1.0E15
      VO=V
   40 CALL LFOSD(NO,NP,IT)
      IF(V.LE.VO)  GO TO 42
```

```
      TK1=0.0
      DO 41 I=1,N0
      CC(I)=0.5*CC(I)
      IF(ABS(CC(I))-TK1)  41,41,44
   44 TK1=ABS(CC(I))
   41 C(I)=C(I)-CC(I)
      ALFA=0.5*ALFA
      PRINT 111,V
      PRINT 106,ALFA
      PRINT 109,(C(I),I=1,N0)
      IF(TK1-0.0001)    1002,1002,45
   45 GO TO 40
   42 CONTINUE
      IF(L1-1)  9,6,6
    6 M=N0+1
      DO 8 I=M,M0
      V1(I)=0.0
      DO 7  J=1,M0
    7 V2(J,I)=V2(I,J)=0.0
    8 V2(I,I)=1.0
C     DERIVATIVES ZERO
    9 CONTINUE
C     PRINT V,ES(EE),V1,V2
      SPR=SQRT(2.0*V/NP)
      PRINT 100,  V,SPR
      PRINT 101, (V1(I), I=1,M0)
      IF(IT.EQ.0) PRINT 102
      IF(IT.EQ.1) PRINT 103
      IF(IPRINT)  300,301,300
  300 DO 10 I=1,M0
   10 PRINT 104,(V2(I,J),J=1,M0)
  301 V2M=0.0
      DO 11  I=1,M0
      DO 11  J=1,M0
   11 V2M=MAX1F(ABSF(V2(I,J)),V2M)
C
      CALL GJRV(V2,M0,1.0E-08,IERR,40)
      IF(IERR+1) 20,19,20
   19 PRINT 120
      RETURN
C     PRINT V2-INVERS
   20 IF(IPRINT)  302,303,302
  302 PRINT 105
      DO 21  I=1,M0
   21 PRINT 106, (V2(I,J), J=1,M0)
  303 V2IM=0.0
      DO 12 I=1,M0
      DO 12 J=1,M0
   12 V2IM=MAX1F(ABSF(V2(I,J)),V2IM)
      V2COND=M0*V2M*V2IM
      PRINT 107, V2COND
C     COMPUTE COEFF.-CORR. FROM NEWTON-R.
      TK=0.0
      DO 24  I=1,M0
      CC(I)=0.0
      DO 22 J=1,M0
```

APPENDIX:5

```
   22 CC(I)=CC(I)-V2(I,J)*V1(J)
      IF(ABS(CC(I))-TK) 24,24,23
   23 TK=ABS(CC(I))
   24 CONTINUE
   33 DO 25 I=1,M0
   25 C(I)=C(I)+CC(I)
C     PRINT COEFF. AND LS(COEFF.)
      PRINT 109, (C(I), I=1,M0)
      DO 26  I=1,M0
   26 V2(I,I)=SQRT(ABS(SMR*SMR*V2(I,I)))
      PRINT 110, (V2(I,I), I=1,M0)
      IF(L2)  1000,30,1000
   30 IF(TK.LE.0.01.AND.IT.EQ.0)    GO TO 1003
   31 IF(TK-0.0001)  1002,1002,1001
 1000 CONTINUE
      GO TO 1003
 1002 PRINT 112
 1003 CONTINUE
C
  100 FORMAT(1H1/////5X,15HLOSS FUNCTION =,E16.8/5X,
     1 29HSTANDARD DEVIATION OF ERRORS=,E16.8)
  101 FORMAT(/5X,6HGRAD V/(8E15.7))
  102 FORMAT(/5X,32HAPPROXIMATIVE SECOND DERIVATIVES)
  103 FORMAT(/5X,24HEXACT SECOND DERIVATIVES)
  104 FORMAT(8E15.7)
  105 FORMAT(/5X,6HINVERS)
  106 FORMAT(8E15.7)
  107 FORMAT(5X,8HV2COND.=,E16.8)
  108 FORMAT(/5X,40HTHE PREVIOUS STEP HAS BEEN REDUCED WITH ALFA= ,
     1 E16.8)
  109 FORMAT(/5X,10HNEW COEFF./(8E15.7))
  110 FORMAT(5X,28HSTANDARD DEVIATION OF COEFF./(8E15.7))
  111 FORMAT(/5X,15HLOSS FUNCTION =,E16.8)
  112 FORMAT(/5X,35HMAX.COEFF.CORR. IS LESS THAN 0.0001)
  120 FORMAT(/5X,42HA PIVOT ELEMENT HAS BEEN LESS THEN 1.0E-08)
      RETURN
      END
```

APPENDIX:6

```
      SUBROUTINE LFGSD(NO,NP,IT)
C
C     LFGSD COMPUTES LOSS FUNKTION V, GRADIENT OF V AND
C     SECOND DERIVATIVES OF V, FOR SUBROUTINE PERPS
C
C     NO=  ORDER OF SYSTEM
C     NP=  LENGTH OF RECORD
C     IT=0  GIVES APPROXIMATIVE SECOND DERIVATIVES
C     IT=1  GIVES EXACT SECOND DERIVATIVES
C
C     C =  COEFF. VECTOR
C          (OUTPUT COEFF.,ERROR COEFF.)
C     Y =  OUTPUT DATA
C     EE=  ERROR
C     E =  STATE VECTOR OF ERROR
C     EC=  FIRST DERIVATIVES OF ERROR VECTOR
C     ECC= SECOND DERIVATIVES OF ERROR VECTOR
C     V =  LOSS FUNCTION
C     V1=  GRADIENT OF V
C     V2=  SECOND DERIVATIVES OF V
C     VCC= VECTOR WITH TERMS FOR EXACT V2
C     REQUIRE DATA IN THE ARRAY DAT
C     OUTPUT(1) IN DAT(1), OUTPUT(2) IN DAT(2) AND SO ON
C
C     SUBROUTINE REQUIRED
C             NONE
C
      COMMON EE,V,Y,E(20),C(40),EC(40),V1(40),VCC(80),ECC(80),
     1 V2(40,40),DAT(7010)
C
      MO=2*NO
      KK=NP-NO+1
      Y=EE=V=0.0
      DO 1  I=1,NO
    1 E(1)=0.0
      E(2)=0.0
      MM=2*MO
      DO 2  I=1,MM
    2 ECC(I)=VCC(I)=0.0
      DO 3  I=1,MO
      EC(I)=V1(I)=0.0
      DO 3  J=1,MO
    3 V2(I,J)=0.0
C
C     START LOOP K
      DO 1000  K=1,NP
      A1=C1=A2=C2=0.0
      DO 5  I=1,NO
      IP=NO+I
      A1=A1-C(IP)*EC(I)
      C1=C1-C(IP)*EC(IP)
      IF(IT-1)  5,4,5
    4 A2=A2-C(IP)*ECC(I)
      C2=C2-C(IP)*ECC(NO+IP)
    5 CONTINUE
      IO=NO+1
```

```
      IF(IT-1)   9,8,9
    8 MM=2*MO-1
      DO 7  I=1,MM
    7 ECC(MM+2-I)=ECC(MM+1-I)
      ECC(1)=A2-EC(1)
      ECC(NO+IQ)=C2-D*EC(IQ)
    9 CONTINUE
      MM=MO-1
      DO 6  I=1,MM
    6 EC(MO+1-I)=EC(MO-I)
      EC(1)=A1+Y
      EC(IQ)=C1-EE
C     DERIVATIVES OF ERROR COMPUTED
C     COMPUTE STATE VECTOR E
      EE=-C(IQ)*E(1)+E(2)+C(1)*Y
      MM=NO-1
      DO 10  I=2,MM
   10 E(I)=-C(NO+I)*E(1)+E(I+1)+C(I)*Y
      E(NO)=-C(NO)*E(1)+C(NO)*Y
C     STEP TO NEXT POINT
      Y=DAT(K)
      EE=EE+Y
      E(1)=EE
C     ERROR COMPUTED
      V=V+EE*EE
      IF(V.GT.1.0E15) GO TO 35
C     LOSS FUNKTION COMPUTED
      DO 11  I=1,NO
      IQ=NO+I
      V1(I)=V1(I)+EE*EC(I)
   11 V1(IQ)=V1(IQ)+EE*EC(IQ)
C     GRAD V COMPUTED
C
      IA=NP-K+1
      N1=NO-1
C
C     START COMPUTATION OF V2
      IF(K.GT.KK)  GO TO 100
C     HERE ONLY VECTORS IN APP. V2 ARE COMPUTED
      DO 12 J=1,3
      J1=J*NO
      DO 12 JJ=1,J
      JJ1=(JJ-1)*NO
      DO 12 I=1,NO
   12 V2(JJ1+I,J1)=V2(JJ1+I,J1)+EC(JJ1+1)*EC(J1+1-I)
      DO 13 J=2,3
      J1=J-1
      J11=J1*NO
      DO 13 JJ=1,J1
      JJ1=JJ*NO
      DO 13 I=1,N1
   13 V2(JJ1,J11+I)=V2(JJ1,J11+I)+EC(J11+1)*EC(JJ1+1-I)
      GO TO 200
C     NOW ALL ELEMENTS IN APP.V2 ARE SUCCESSIVELY COMPUTED
  100 DO 23 J=1,3
      J1=J*NO
```

```
      J2=(J-1)*NO
      DO 23 JJ=1,J
      JJ1=(JJ-1)*NO
      DO 22 I=1,IA
   22 V2(JJ1+I,J1)=V2(JJ1+I,J1)+EC(JJ1+1)*EC(J1+1-I)
      DO 23 I=IA,N1
   23 V2(JJ1+IA,J2+I)=V2(JJ1+IA+1,J2+I+1)+EC(JJ1+1)*EC(J2+I+1-IA)
      DO 25 J=2,3
      J1=J-1
      J2=(J-1)*NO
      DO 25 JJ=1,J1
      JJ1=(JJ-1)*NO
      JJ2=JJ*NO
      DO 24 I=1,IA
   24 V2(JJ2,J2+I)=V2(JJ2,J2+I)+EC(J2+1)*EC(JJ2+1-I)
      IA1=IA+1
      DO 25 I=IA1,N1
   25 V2(JJ1+I,J2+IA)= V2(JJ1+I+1,J2+IA+1)+EC(J2+1)*EC(JJ1+I+1-IA)
C     APP. V2 COMPUTED
  200 IF(II-1) 28,26,28
   26 NM=2*NO
      DO 27  I=1,NM
   27 VCC(I)=VCC(I)+EE*ECC(I)
   28 CONTINUE
C     TERMS VCC FOR EX. V2 COMPUTED
C     END LOOP K
 1000 CONTINUE
      V=V/2.0
      IF(IT-1)  33,31,35
C     ADD TERMS VCC AND V2 WILL BE EXACT
   31 DO 33  J=1,NO
      IQ=NO+J
      DO 32  I=1,NO
      IR=I+J-1
   32 V2(I,IQ)=V2(I,IQ)+VCC(IR)
      DO 33  I=1,J
      IR=I+J-1
      V2(NO+I,IJ)=V2(NO+I,IJ)+VCC(NO+NO+IR)
C     EX. V2 COMPUTED
   33 CONTINUE
      DO 34  I=1,MO
      MM=I+1
      DO 34  J=MM,MO
   34 V2(J,I)=V2(I,J)
   35 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GJRV(A,N,EPS,IERR,IA)
C
C     INVERTS ASYMMETRIC MATRICES, HAS EMERGENCY EXIT,
C     REQUIRES N**2+4*N WORDS OF ARRAY STORAGE
C
C     A IS THE NAME OF THE MATRIX TO BE INVERTED
C     N IS THE ORDER OF A
C     EPS IS A VALUE TO BE USED AS A TOLERANCE FOR
C     ACCEPTANCE OF THE SINGULARITY OF A GIVEN MATRIX
C     IERR IS AN INTEGER VARIABLE WHICH WILL CONTAIN ZERO
C     UPON RETURN IF INVERSION IS COMPLETED OR -1 IF SOME
C     PIVOT ELEMENT HAS AN ABSOLUTE VALUE LESS THAN EPS
C     IA IS THE DIMENSION PARAMETER
C     MAXIMUM ORDER OF A=40
C     THE ORIGINAL MATRIX IS DESTROYED
C     IF IERR IS RETURNED =-1 THEN THE INVERSION HAS FAILED
C     OTHERWISE THE RESULTING INVERSE IS PLACED IN A
C
C     SUBROUTINE REQUIRED
C              NONE
C
      DIMENSION A(IA,IA),B(40),C(40),IP(40),IQ(40)
      IERR=0
      DO 140 K=1,N
      PIVOT=0.0
      DO 120 I=K,N
      DO 2 J=K,N
      IF(ABSF(A(I,J))-ABSF(PIVOT)) 2,2,1
    1 PIVOT=A(I,J)
      IP(K)=I
      IQ(K)=J
    2 CONTINUE
  120 CONTINUE
      IF(ABSF(PIVOT)-EPS) 100,100,3
    3 IF(IP(K)-K) 4,6,4
    4 DO 5 J=1,N
      IPX=IP(K)
      Z=A(IPX,J)
      A(IPX,J)=A(K,J)
    5 A(K,J)=Z
    6 IF(IQ(K)-K) 7,9,7
    7 DO 8 I=1,N
      IPX=IQ(K)
      Z=A(I,IPX)
      A(I,IPX)=A(I,K)
    8 A(I,K)=Z
    9 DO 13 J=1,N
      IF(J-K) 11,10,11
   10 B(J)=1.0/PIVOT
      C(J)=1.0
      GO TO 12
   11 B(J)=-A(K,J)/PIVOT
      C(J)=A(J,K)
   12 A(K,J)=0.0
      A(J,K)=0.0
   13 CONTINUE
```

```
      DO 14 I=1,N
      DO 14 J=1,N
   14 A(I,J)=A(I,J)+C(I)*B(J)
  140 CONTINUE
      DO 20 KP=1,N
      K=N+1-KP
      IF(IP(K)-K) 15,17,15
   15 DO 16 I=1,N
      IPX=IP(K)
      Z=A(I,IPX)
      A(I,IPX)=A(I,K)
   16 A(I,K)=Z
   17 IF(IQ(K)-K) 18,20,18
   18 DO 19 J=1,N
      IPX=IQ(K)
      Z=A(IPX,J)
      A(IPX,J)=A(K,J)
   19 A(K,J)=Z
   20 CONTINUE
      GO TO 21
  100 IERR=-1
   21 RETURN
      END
```