THE INSTRUMENTAL VARIABLE METHOD
AND RELATED IDENTIFICATION SCHEMES

by

A.J. Smets
THE INSTRUMENTAL VARIABLE METHOD AND RELATED IDENTIFICATION SCHEMES

by

A.J. Smets

TH-Report 70-E-15

November 1970

Submitted in partial fulfillment of the requirements
for the degree of Ir. (M.Sc.) at the Eindhoven University of Technology.
The work was carried out in the Measurement and Control Group
under directorship of Prof. dr. ir. P. Eykhoff.
Advisor ir. A.J.W. van den Boom.
THE INSTRUMENTAL VARIABLE METHOD AND RELATED IDENTIFICATION SCHEMES

A.J. Smets
Eindhoven University of Technology
Department of Electrical Engineering
Eindhoven, Netherlands

Summary

It is assumed that sampled input and output data of a linear process become available in time. These data will be used immediately to update the previous estimate \( \hat{\theta} \) of the parameter vector. Consequently recursive estimation techniques will be used.

The instrumental variable (I.V.) method provides an estimator with quick convergence, if the output noise level is not excessively high and if the instrumental variables are taken from the model output sequence. The model parameters are adjusted in an intermittent way (time-varying system) according to the estimates of the process parameters. If high noise powers occur, the initial model vector has a strong effect on the speed convergence. It is shown that the method indicated by Wong (18) to obtain the "optimal" I.V. matrix, can be described in a much more elegant way. This leads to the filtering of input, output and model output data with a (noise-whitening) filter having the same parameters as those of the -inverse- noise filter. In this way the close analogy between the optimal I.V. method and the generalized least-squares method becomes apparent. Moreover, some extended matrix schemes are discussed.
Contents

1. Introduction

2. Discussion of some estimation schemes
   2.1. Classical regression analysis
      2.1.1. Forward model
      2.1.2. Forward and backward model
   2.2. Generalized least-squares estimator
   2.3. Generalized least-squares estimator with extended matrix

3. The instrumental variable method
   3.1. Statement of the problem
   3.2. The optimal instrumental variable matrix \( Z^* \)
   3.3. Realisation of the asymptotic optimal I.V. matrix (Wong)
   3.4. Alternative realisation of the asymptotic optimal I.V. matrix
   3.5. Simple construction of the optimal I.V. algorithm

4. Experimental results and discussion
   4.1. Estimation schemes
   4.2. The iterative algorithm
   4.3. The open-loop process with only one forward parameter
   4.4. The closed-loop process with only one forward parameter
   4.5. The open-loop process with two forward parameters
   4.6. The process description introduced by Åström
   4.7. The generalized extended matrix methods
   4.8. The sub-optimal I.V. method

5. Additional remarks and suggestions

List of Symbols

Appendix I : Definitions, conventions
Appendix II : Recursive formulae
Appendix III: Exponential weighting of input and output data
Appendix IV : Closed-loop parameter estimation

Literature
1. Introduction

The identification (including parameter estimation) of an industrial process is a necessary act, if we want to have a better knowledge about this process. This knowledge can be essential to design a specific regulator, or to improve the performance of an already regulated process (e.g., chemical plants). In biological processes, identification can lead to the estimate of so-called "standard" parameters, from which a model of the process studied can be formed. A patient for instance can now be compared with such a model, and the differences between them can give an indication of a disturbance in the patient himself.

As most processes have a "closed-loop" configuration (e.g., the human being reacting on a stimulus or regulating a certain action; biological processes) it is essential to have an estimator which can estimate quite accurately the open-loop parameters of such a process as well as its closed-loop parameters. The instrumental variable estimator is one (of the few) estimators which is capable of such an action.

Now, even if we could make a mathematical model of a process, this model can be too complex to handle (e.g., destillation columns and, in general, any chemical plant). In this case we will have to reduce this mathematical model with our background knowledge and our physical intuition as a guide, into a more suitable, approximate model description. In this way we will only estimate those parameters of the process, which are quite relevant.

The purpose of identification is a.o.:

1) to design a regulator (to control an industrial process);
2) to estimate certain specific parameters (biological, economical);
3) depending on the aim, we will try to estimate not only the parameters of the process, but also the parameters of the "noise" (where "noise" stands for all disturbances that make the real process output differ from the measured process output).

It will be clear that the quality of the estimates will depend on:

1) the estimator chosen;
2) the (approximated) model structure;
3) the noise power;
4) the number of observations available;
5) our a priori knowledge of the process and our interpretation of it.
The instrumental variable (I.V.) algorithm will work well only if we have some a priori knowledge about the structure of the process (order of the difference equation describing the process). It is well known that a variety of methods have already been proposed to solve this "pure" identification of the process (cfr. Åström (9), Woodside (1)). So it will be assumed that the order of the process has been determined making use of one of these methods, before the estimation of the process parameters starts.

In the following chapters we will give:

1) a few estimation schemes (chapter 2);
2) the I.V. estimator, as well as its optimized form (chapter 3);
3) the results obtained using some estimators discussed in chapters 2 and 3 (chapter 4);
4) some ideas about future work especially on the I.V. estimator.

It will be assumed in chapter 4 that the I.V. algorithm only starts after some knowledge about the process parameters has been obtained. The reason for this is quite obvious, as will be seen in chapter 3 (when all model parameters are assumed to be zero, the model output will have no relation at all with the true process output).

This knowledge will be obtained by the application of the least-squares (L.S.) estimator on a few (10 x m where m is the total number of parameters to be estimated) input and output samples. Furthermore we will assume that new input-output samples are used immediately to update the estimates. This necessitates an iterative form of our algorithm used.

As the estimator will be applied on simulated processes, this simulations will have a transient (at the beginning of the simulation, the process output is generated without any knowledge about its past values; these past values are put equal to zero). To eliminate this transient we calculated the response of the process to be studied, to a unit impulse, and if the transient resulting from this input signal was smaller than 0.01 (e.g. \( x_T \leq 0.01 \); the index T indicates a delay of T times \( \tau \) where \( \tau \) is the sample period), we decided to reject the first T input and output observations generated.
It is the author's conviction that the use of instrumental variables in the field of parameter estimation, will be implied more and more (in a way, the Tally estimate described by Peterka (20) is an I.V. approach). Indeed this method has two basic fundamentals which make it quite appealing:

1) the background knowledge and the physical feeling can be used directly in the estimator;
2) its mathematical simplicity makes it easy to implement it.
2. Discussion of some estimation schemes

2.1 Classical regression analysis

2.1.1 Forward model

Consider the linear process $P$ of which the relation between input and output can be described by the following difference equation (D.E.):

$$x_k = \sum_{i=0}^{p} b_i u_{k-i}.$$  \hspace{1cm} (2.1)

Fig. 2.1: A linear process. The output is disturbed with additive noise.

Let the output $x_k$ be disturbed by an additive noise signal $n_k$, e.g.

$$y_k = x_k + n_k,$$

where $y_k$ is the observable disturbed output of the process. The sequences $\{u_k\}_{1}^{N+p}$ and $\{y_k\}_{1}^{N+p}$ are available, where $N$ is considerably larger than the total number of parameters (in this case $p+1$) to be estimated. In this case, we can estimate the coefficients $b_i$ $(i = 0, 1, \ldots, p)$ by considering the following set of equations:

$$
\begin{align*}
  y_{p+1} &= b_0 u_{p+1} + b_1 u_p + \ldots + b_p u_1 + n_{p+1} \\
  y_{p+2} &= b_0 u_{p+2} + b_1 u_{p+1} + \ldots + b_p u_2 + n_{p+2} \\
  &\vdots \\
  y_{p+N} &= b_0 u_{p+N} + b_1 u_{p+N-1} + \ldots + b_p u_N + n_{p+N} 
\end{align*}
$$

(2.2)

Using matrix notation, the set of equations (2.2) can be written in a more convenient way:

$$Y = Ub + n$$

(2.3)
According to classical regression analysis, the least-squares estimator of $\theta$ is given by (e.g. Deutsch (2), Goldberger (3)):

$$\hat{\theta} = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{y} \quad (2.4)$$

This is a linear estimator.

If $\{n_k\}_{k=1}^{N+p}$ is uncorrelated with $\{u_k\}_{k=1}^{N+p}$, this estimator gives an unbiased estimate of $\theta$ as:

$$\mathbb{E}( \mathbf{U}^T \mathbf{U}^{-1} \mathbf{U}^T \mathbf{n} ) = 0,$$

and thus:

$$\mathbb{E}( \hat{\theta} ) = \theta \quad (2.5)$$

If the characteristics of the noise sequence $\{n_k\}_{k=1}^{N+p}$ are known in the form of its covariance matrix, the best linear unbiased least-squares estimator of $\theta$ is given by (e.g. Goldberger (3)):

$$\hat{\theta} = (\mathbf{U}^T \mathbf{L}^{-1} \mathbf{U})^{-1} \mathbf{U}^T \mathbf{n} \quad (2.6)$$

where $\mathbb{E}( \mathbf{n} \mathbf{n}^T )$ is the covariance matrix of the noise $\{n_k\}_{k=1}^{N+p}$ and $\mathbb{E}( n_k ) = 0$.

In this case we can prove that:

$$\text{cov} \hat{\theta} = \mathbb{E}( (\hat{\theta} - \theta) (\hat{\theta} - \theta)^T ) = (\mathbf{U}^T \mathbf{L}^{-1} \mathbf{U})^{-1} \quad (2.7)$$

Eq.(2.6) is known as the generalized Gauss-Markoff estimator; any choice for $\mathbf{L}$ other than $\mathbb{E}( \mathbf{n} \mathbf{n}^T )$ will give a covariance matrix of $\hat{\theta}$ that is greater than the matrix given in eq.(2.7).

Already in 1960, Levin (4) used the estimator given in eq.(2.4) to estimate the points of the impulse response $h(t)$. Indeed we can write:
if we suppose all $h_p^i$ for $i > 0$ to be zero. In this way we will always have a truncation error (van den Boom & Melis (5)), but this truncation can be made arbitrarily small by choosing $p$ large enough.

![Graph](image)

**Fig. 2.2**: The impulse response of a first order system.

Levin supposed the sequence $\{n_k\}_{k=1}^{N+p}$ to be a white noise sequence. In this case we get:

$$\hat{\mathbf{h}} = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{y}$$

(2.9)

and this is an unbiased linear estimator of $\mathbf{h}$, where:

$$\mathbf{h} = (h_0, h_1, \ldots, h_p).$$

### 2.1.2 Forward and backward model (combined-, generalized model)

The description of a linear process by its impulse response (forward model) is in a lot of cases not a practical one. Indeed in general the points $h_{p+i}$ ($i > 0$) of the impulse response are not zero at all and thus there will be always a truncation error when estimating $\mathbf{h}$ (eq. (2.9)). Of course this error can be made arbitrarily small, but practical reasons limit this procedure (time-consuming manipulations with matrices having large dimensions; finite sequence length $N$). Furthermore for white additive noise $\mathbf{n}$ the standard deviation $\sigma$ in each parameter is given by:

$$\sigma = \sqrt{\frac{\psi_{nn}(0)}{\psi_{uu}(0)}}$$

(2.9a)

(van den Boom & Melis (5)), where $\psi_{nn}$ and $\psi_{uu}$ are the autocorrelation functions of $\mathbf{n}$ resp. $\mathbf{u}$, and $l$ is the total length of the observation sequence (e.g. $l = (N+p)x\tau$). It thus will have no sense to estimate these $h_{p+i}$ that have a lower value than $\sigma$ as their estimate would be "lost" in its standard deviation. Especially in the case of a closed-loop process.
(Prinsen (6)) this would give no further information about the impulse response of the open-loop process.

A simple example will make it clear that there is much sense in estimating forward as well as backward parameters of a D.E.

Suppose the first order process described by the relation (with a < 1):

\[ x_k = u_k + a u_{k-1} + a^2 u_{k-2} + \ldots + a^j u_{k-j} + \ldots = \sum_{i=0}^{\infty} a^i u_{k-i} \quad (2.10) \]

We can write then:

\[ a x_{k-1} = a u_{k-1} + a^2 u_{k-2} + \ldots + a^j u_{k-j} + \ldots \]

\[ = \sum_{i=1}^{\infty} a^i u_{k-i} = \sum_{i=0}^{\infty} a^i u_{k-i} - u_k \quad (2.11) \]

or:

\[ x_k - a x_{k-1} = u_k \quad (2.11a) \]

The D.E. (2.11a) describes the same physical linear process as eq.(2.10), but it is clear that it will be easier (when taking the necessary precautions) to estimate two parameters (1, a) than all points of the impulse response.

If however, we describe the process P by the following D.E.:

\[ x_k + \sum_{i=1}^{q} a_i x_{k-i} = \sum_{i=0}^{p} b_i u_{k-i} \quad (2.12) \]

it can be proved (Shaw (7)) that classical regression analysis in general gives biased estimates of both forward and backward parameters of the D.E.(2.11), if there is additive noise on the output.

Suppose the sequences \( \{ u_k \}_{k=1}^{N+q} \) and \( \{ y_k \}_{k=1}^{N+q} \) are given, where \( y_k = x_k + n_k \), then eq.(2.12) can be written:

\[ y_k = \sum_{i=0}^{p} b_i u_{k-i} - \sum_{i=1}^{q} a_i y_{k-i} + n_k + \sum_{i=0}^{q} a_i n_{k-i} \quad (2.13) \]

The noise sequence \( \{ n_k \}_{k=1}^{N+q} \) is considered to have elements taken from a stationary process, which has zero mean, and the elements \( n_k \) are uncorrelated with \( x_i \) and \( u_j \) (\( i, j = 1, 2, \ldots, N+q \)).

We define:

\[ b^T = (b_0, b_1, \ldots, b_p, -a_1, -a_2, \ldots, -a_q) \],

\[ Y^T = (y_{q+1}, y_{q+2}, \ldots, y_{q+N}) \],

\[ N^T = (n_{q+1}, n_{q+2}, \ldots, n_{q+N}) \],

(2.14)
and:
\[
\Omega(u,y) = \begin{bmatrix}
  u_{q+1} & \cdots & u_{q+p-1} & y_q & \cdots & y_1 \\
  \vdots & & \vdots & \vdots & & \vdots \\
  u_{q+N} & \cdots & u_{q+N-p} & y_{q+N-1} & \cdots & y_N \\
\end{bmatrix} = (U|Y)
\]

and analogously:
\[
\Omega(u,x) = (U|X) \quad \text{and} \quad \Omega(0,n) = (0|\emptyset)
\]  

(we will always consider the case with \( q \geq p \); this is no essential restriction).

Now, the least-squares estimator of \( \beta \) is given by:
\[
\hat{\beta} = \left\{ \Omega^T(u,y) \Omega(u,y) \right\}^{-1} \Omega^T(u,y) y
\]  

As:
\[
y = \Omega(u,y) \beta + n - \Omega(0,n) b
\]

we can combine eq.(2.16) with eq.(2.17) to give:
\[
\hat{\beta} = \beta + \left( \Omega^T(u,y) \Omega(u,y) \right)^{-1} \left( \Omega^T(u,y) n - \Omega^T(u,y) \Omega(0,n) b \right)
\]

and thus:
\[
\lim_{N \to \infty} \Delta \hat{\beta} = \lim_{N \to \infty} \left( \hat{\beta} - \beta \right)
\]

\[
= \lim_{N \to \infty} \left( \left( \Omega^T(u,y) \Omega(u,y) \right)^{-1} \left( \Omega^T(u,y) n - \Omega^T(u,y) \Omega(0,n) b \right) \right)
\]

We will show that the expression (2.18a) is in general different from the zero vector, and thus \( \hat{\beta} \) is an asymptotic biased estimator of \( \beta \) (Shaw (7), Evers (8)).

We suppose that \( u_k \) and \( n_k \) are taken from well-behaved stationary stochastic processes, so the time averages of their products are consistent estimators of their auto- and crosscorrelation functions.

Let \( \Omega^T(u,y) \Omega(u,y) \) and consider the relevant parts of eq.(2.18a) one by one:

a) \( \lim_{N \to \infty} \left[ \frac{1}{N} \right] p \) = \( \Gamma \) (a positive definite matrix);

b) \( \lim_{N \to \infty} \frac{\Omega^T(u,y) \Omega(0,n)}{N} = \begin{bmatrix}
  0 \\
  \psi_{nn}(0) & \cdots & \psi_{nn}(1-q) \\
  \vdots & & \vdots \\
  \psi_{nn}(q-1) & \cdots & \psi_{nn}(0) \\
\end{bmatrix} \)
with:  
\[
\begin{align*}
\psi_{nn}(k) &= \psi_{nn}(-k) \\
\psi_{yn}(k) &= \psi_{nn}(k) \\
\psi_{yn}(k) &= 0
\end{align*}
\]
for all \(k\),

where:  
\(
\psi_{ab}(k) = E[a(j)b(j+k)]
\)

\(c)\)

\[
\lim_{N \to \infty} \frac{\Omega_{n}^{T}(u,y)}{N} = \begin{bmatrix}
Q
\psi_{nn}(1) \\
\psi_{nn}(2) \\
\vdots \\
\psi_{nn}(q)
\end{bmatrix}
\]

and so:

\[
\lim_{N \to \infty} ( \hat{\beta} - b ) = \Gamma^{-1} \begin{bmatrix}
0 \\
\psi_{nn}(1) \\
\vdots \\
\psi_{nn}(q)
\end{bmatrix} - \begin{bmatrix}
0 \\
\psi_{nn}(0) \cdots \psi_{nn}(1-q) \\
\vdots \\
\psi_{nn}(q-1) \cdots \psi_{nn}(0)
\end{bmatrix} \begin{bmatrix}
b_0 \\
b_1 \\
\vdots \\
a_q
\end{bmatrix}
\]

or:

\[
\lim_{N \to \infty} \Delta\hat{\beta} = \Gamma^{-1} \begin{bmatrix}
0 \\
\psi_{nn}(1) + \sum_{i=1}^{q} a_i \psi_{nn}(1-i) \\
\vdots \\
\psi_{nn}(q) + \sum_{i=1}^{q} a_i \psi_{nn}(q-i)
\end{bmatrix}
\]

\(2.18b)\)

As \(\Gamma^{-1}\) is a non-singular matrix, \(\lim_{N \to \infty} \Delta\hat{\beta} \neq 0\) (and in general each component of \(\Delta\hat{\beta}\) can be different from zero). The \(\lim_{N \to \infty} \Delta\hat{\beta}\) is only equal to the zero vector if:

\[
\psi_{nn}(j) + \sum_{i=1}^{q} a_i \psi_{nn}(j-i) = 0 \quad (j = 1, 2, \ldots, q).
\]

This will only be the case when the noise sequence \(\{n_k\}_{k=1}^{N+q}\) is derived from a white noise sequence \(\{\xi_k\}_{k=1}^{N+q}\) in the following way:

\[
n_k + a_1 n_{k-1} + \ldots + a_q n_{k-q} = \xi_k.
\]

\(2.19)\)
Remark 1: This coincides with the special case, considered by Åström (10):

\[(1 + A(z^{-1}))y_k = (b_0 + B(z^{-1}))u_k + \xi_k.\]  
\[(2.20)\]

The L.S. estimator will yield asymptotically unbiased estimates of \(a_i\) and \(b_j\) (Åström (10), Evers (8)). Eq. (2.20) can then be written:

\[y_k = \frac{b_0 + B(z^{-1})}{1 + A(z^{-1})} u_k + \frac{1}{1 + A(z^{-1})} \xi_k\]

and so:

\[(1 + A(z^{-1}))n_k = \xi_k\]

which is equal to eq. (2.19).

Remark 2: We have proved in a rather elaborate way, using the a priori knowledge about the structure of the process (order of the D.E.), that the use of the L.S. estimator in general will yield biased results.

Rejecting this a priori knowledge of the order of the D.E. in some cases can lead to an asymptotically unbiased estimate of \(b_0\).

Indeed, consider the D.E. of the following type (Åström (9)):

\[(1 + A)\tilde{y}_k = (b_0 + B)\tilde{u}_k + (1 + C)\xi_k\]  
\[(2.21a)\]

where:

\[A = \sum_{i=1}^{q} a_i z^{-i}; B = \sum_{i=1}^{p} b_i z^{-i} \text{ and } C = \sum_{i=1}^{s} c_i z^{-i}\]

and suppose \(1 + D = 1/(1 + C)\) exists e.g. \(D\) can be written as:

\[D = \sum_{i=1}^{r} d_i z^{-i} \quad \text{(finite } r\text{).}\]

We can rewrite eq. (2.21a):

\[(1 + A)(1 + D)\tilde{y}_k = (b_0 + B)(1 + D)\tilde{u}_k + \xi_k.\]  
\[(2.21b)\]

If we now write:

\[1 + A^* = (1 + A)(1 + D)\]

\[b_0 + B^* = (b_0 + B)(1 + D)\]

where:

\[A^* = \sum_{i=1}^{q+r} a_i^* z^{-i}\]

and:

\[B^* = \sum_{i=1}^{p+r} b_i^* z^{-i},\]  
\[(2.22)\]
and combine eq.(2.22) with eq.(2.21b), it is seen that:

\[ (1 + A^*)Y_k = (b_0 + B^*)u_k + \xi_k \]  

(2.23)

Using the L.S. estimator on \( b^* \) (e.g. estimating the \( a_i^* \) and \( b_i^* \)), we will obtain consistent estimates of the parameters \( a_i^* \) and \( b_i^* \).

The common part in the polynomials \( 1 + A^* \) and \( b_0 + B^* \) can then be eliminated, and an unbiased estimate of \( b \) is thus obtained (Valis (11), Åström & Eykhoff (12)). Note the correlation between the ideas of this remark and those of the following paragraph.

2.2 Generalized least-squares estimator

\[ \text{Fig. 2.3: The symbolic representation of a linear process described by the D.E. (2.24). The generation of the samples } u_k^* \text{ and } y_k^* \text{ is shown.} \]

Suppose that the linear process \( P \) can be described by the following D.E. (eq.(2.20)):

\[ (1 + A)y_k = (b_0 + B)u_k + (1 + C)\xi_k \]  

(2.24)

where:

\[ (1 + C)\xi_k = e_k \]  

(2.25)

e.g.:

\[ (1 + A)y_k = (b_0 + B)u_k + e_k. \]

As in the remark 2 in paragraph 2.1, we suppose \( 1 + D = 1/(1 + C) \) exists, and:

\[ D = \sum_{i=1}^{r} d_i z^{-i}. \]

Eq.(2.25) can then be written as:

\[ \xi_k = (1 + D)e_k. \]  

(2.26)
It has been shown in 2.1.2 that the normal least-squares estimator \( \mathbf{b} \) of \( \mathbf{b} \), where \( \mathbf{b}^T = ( b_0, b_1, \ldots, b_p, -a_1, -a_2, \ldots, -a_q ) \), is biased. Only in the case where \( \mathbf{e}_k = \xi_k \) (cfr. eq. (2.20)) will the L.S. estimator be consistent. If we filter the sequences \( \{u_k\}_{N+q} \) and \( \{y_k\}_{N+q} \) and \( \{e_k\}_{N+q} \) in the following way:

\[
\begin{align*}
\mathbf{u}_k^* &= u_k + d_1 u_{k-1} + \cdots + d_r u_{k-r} = (1+D)u_k \\
\mathbf{y}_k^* &= y_k + d_1 y_{k-1} + \cdots + d_r y_{k-r} = (1+D)y_k \\
\mathbf{e}_k^* &= e_k + d_1 e_{k-1} + \cdots + d_r e_{k-r} = (1+D)e_k = \xi_k,
\end{align*}
\]

then we can rewrite eq. (2.24) as follows:

\[
(1 + A)\mathbf{y}_k^* = (bo + B)\mathbf{u}_k^* + \xi_k.
\] (2.28)

It can be seen from eq. (2.28) that, when using the sequences \( \{u_k^*\}_{N+q} \) and \( \{y_k^*\}_{N+q} \), a generalized L.S. estimator is given by:

\[
\mathbf{b}^* = (\Omega^T(u^*_0, y^*_0)\Omega(u^*_0, y^*_0))^{-1} \Omega^T(u^*_0, y^*_0) \mathbf{y}^*.
\] (2.29)

where \( \mathbf{b}^* \) is a consistent estimate of \( \mathbf{b} \).

Remark 1: The same idea has also been indicated by Eykhoff (14) when instrumenting the Markoff-estimate (eq. 2.6 applied on the matrix \( \Omega(u, y) \) instead of \( U \)). He assumes that \( \Sigma^{-1} \) can be written in the following way:

\[
\Sigma^{-1} = D^T D
\]

(\( D \) is in this remark a matrix and not the polynomial \( D(z^{-1}) \)). Using this notation \( \mathbf{b} \) can be written as:

\[
\mathbf{b} = (\Omega^T(u, y)) D \Omega(u, y)^{-1} (\Omega(u, y))^T (D\mathbf{y}),
\]

which means that the matrix \( D \) represents a "noise-whitening" filter. Applying this filter on the sequences \( \{u_k\}_{N+q} \) and \( \{y_k\}_{N+q} \) leads to an asymptotic unbiased estimate of \( \mathbf{b} \).

In practice, the parameters \( d_i \) of the (inverse) noise filter are not known; we will try to estimate \( \mathbf{d}_T = ( d_1, d_2, \ldots, d_r ) \) consistently, and then filter the sequences \( \{u_k\}_{N+q} \) and \( \{y_k\}_{N+q} \) according to eq. (2.27).

The following procedure has been proposed by Clarke (13):

1) making use of the sequences \( \{u_k\}_{N+q} \) and \( \{y_k\}_{N+q} \), a L.S. solution \( \mathbf{b}_0 \) is found;
2) with the available \( \bar{\beta} \) and the sequences \( \{u_k\}^{N+q}_1 \) and \( \{y_k\}^{N+q}_1 \), a sequence \( \{\bar{\varepsilon}_k\}^{N+q}_1 \) can be computed, where:

\[
\bar{\varepsilon}_k = y_k + \alpha_1 y_{k-1} + \ldots + \alpha_q y_{k-q} - \beta_0 u_k - \ldots - \beta_p u_{k-p};
\]

(2.30)

3) eq. (2.26) can be written in matrix notation, having \( \{\bar{\varepsilon}_k\}^{N+q}_1 \) available:

\[
\bar{\varepsilon} = - \tilde{\bar{d}} \hat{d}_i^T + \xi',
\]

(2.31)

and thus a consistent estimate of \( \bar{d}_i \) (which need not be equal to \( \hat{d} \), as we postulated the noise term in eq. (2.31) to be white noise; our hope is that the \( \bar{d}_i \) thus obtained will be a good first approximation of \( \hat{d} \)) will be given by:

\[
\hat{y}_i = - (\hat{\varepsilon}_i^T \hat{\varepsilon})^{-1} \hat{\varepsilon}_i^T \hat{\varepsilon};
\]

(2.32)

4) filtering the sequences \( \{u_k\}^{N+q}_1 \) and \( \{y_k\}^{N+q}_1 \) with the estimations \( \hat{y}_j \) ( \( j = 1, 2, \ldots, r \) ) will give the sequences \( \{u_k^*\}^{N+q}_1 \) and \( \{y_k^*\}^{N+q}_1 \);

5) a L.S. solution \( \hat{\beta}_i^* \) according to eq. (2.29) using the sequences \( \{u_k^*\}^{N+q}_1 \) and \( \{y_k^*\}^{N+q}_1 \) is computed, whereafter a new sequence \( \{\bar{\varepsilon}_k\}^{N+q}_1 \) will be computed.

If the so-called loss-function \( V = \sum_{k=1}^{N+q} \bar{\varepsilon}_k^2 \) is smaller for the last sequence than for the next to last, the procedure now starts from 3), else it will be stopped. The next to last estimated parameter vector is chosen to be the estimate of the process parameter vector.

---

**Fig. 2.4**: Schematical diagram of the procedure proposed by Clarke.

The procedure described above is an explicit estimation scheme and it yields, after using \( N+q \) input-output pairs and some runs through the procedure, a solution for \( \hat{\beta}_i^* \) and \( \hat{y} \). We see that the process parameters
as well as the (inverse) noise parameters will be estimated.

A recursive algorithm, which is almost the same as the method described by Clarke, has been given by Hastings-James & Sage (15). Hastings-James & Sage as well as Clarke are not able to prove the convergence of their algorithm, although their results do indicate, that it is indeed convergent in practice, even with noise/signal ratios of 7 (15).

Remark 2: By weighting the input and output values (e.g. exponentially, see Appendix III), it is possible to approximate the method of Clarke in a recursive way (Hastings-James & Sage (15)). This weighting is necessary. In an iterative scheme without weighting, past values of \( u^* \), \( y^* \) and \( \hat{\theta} \) are held in the memory of the algorithm. Hence they will influence the updating of the parameter vector \( \hat{\theta} \). As these past values have been obtained by very poor (in the beginning; small N) estimates of \( \hat{\theta} \) and \( \hat{\gamma} \), retaining them in the memory could produce large deviations in the final estimates of \( \hat{\theta} \) and \( \hat{\gamma} \) (large N). By weighting these past values, they will be gradually removed from the memory, and their influence on the updating of \( \hat{\theta} \) and \( \hat{\gamma} \) will be eliminated. However, a consequence of this weighting is, that the estimates of \( \hat{\theta} \) and \( \hat{\gamma} \) for large N will have a lower bound on their standard deviations (appendix III).

2.3 Generalized least-squares estimator with extended matrix

Again, consider the process \( P \) described by the D.E.:

\[
(1 + A)u_k = (b_0 + B)u_k + (1 + C)\xi_k.
\] (2.20)

The sequences \( \{u_k\}_1^{N+q} \) and \( \{y_k\}_1^{N+q} \) are available. The sequence \( \{\xi_k\}_1^{N+q} \) is a white noise sequence. According to 2.2 we suppose:

\[
e_k = (1 + C)\xi_k.
\] (2.25)

and:

\[
\xi_k = (1 + D)e_k.
\] (2.26)

and so:

\[
e_k = -d_1 e_{k-1} - \ldots - d_r e_{k-r} + \xi_k.
\] (2.27)

Combining eqs. (2.20) and (2.27) we get:

\[
y_k = (u_k, u_{k-1}, \ldots, u_{k-p}, y_{k-1}, \ldots, y_{k-q}, e_{k-1}, \ldots, e_{k-r}) b' + \xi_k
\] (2.33)

where:

\[
b'^T = (b_0, b, \ldots, b_p, -a_1, \ldots, -a_q, -d_1, \ldots, -d_r)
\] (2.34)
Writing eq. (2.33) in matrix notation we get:
\[
\mathbf{y} = ( \mathbf{U}^{\top} \mathbf{Y}^{\top} \mathbf{E} ) \mathbf{b}' + \xi = \mathbf{\Omega}(\mathbf{u}, \mathbf{y}, \mathbf{e}) \mathbf{b}' + \xi. \tag{2.35}
\]

The estimator:
\[
\mathbf{b}' = (\mathbf{\Omega}^{\top}(\mathbf{u}, \mathbf{y}, \mathbf{e}) \mathbf{\Omega}(\mathbf{u}, \mathbf{y}, \mathbf{e}))^{-1} \mathbf{\Omega}^{\top}(\mathbf{u}, \mathbf{y}, \mathbf{e}) \mathbf{y} \tag{2.36}
\]
gives a consistent estimate of \( \mathbf{b}' \). Indeed:
\[
\lim_{N \to \infty} \left[ (\mathbf{\Omega}^{\top}(\mathbf{u}, \mathbf{y}, \mathbf{e}) \mathbf{\Omega}(\mathbf{u}, \mathbf{y}, \mathbf{e}))^{-1} \mathbf{\Omega}^{\top}(\mathbf{u}, \mathbf{y}, \mathbf{e}) \xi \right] = \lim_{N \to \infty} \mathbf{\Omega}^{\top}(\mathbf{u}, \mathbf{y}, \mathbf{e}) \xi = 0 \tag{2.37}
\]

(note the analogy with Clarke).

The estimator given in eq. (2.36) can be put in a recursive form. The sequence \( \{ \mathbf{e}_k \}_{1}^{N+q} \) is not available and thus we will compute an estimate of it using eq. (2.30). It thus will be clear that we will apply weighting too on this recursive algorithm (analogy with Hastings–James & Sage).

Computing time for the algorithm of Hastings–James & Sage is about the same as the time needed to solve eq. (2.36), but now there is no need to filter the data explicitly (cfr. eq. (2.27)).

The results obtained using this algorithm are given in chapter 4, 4.7.1.

Remark 1: We will show the strong analogy between the method proposed by Clarke, and the method described in this paragraph. The sequences \( \{ \mathbf{u}_k \}_{1}^{N+q} \), \( \{ \mathbf{y}_k \}_{1}^{N+q} \) and \( \{ \mathbf{e}_k \}_{1}^{N+q} \) are once again available.

a) The method given by Clarke is given below:
\[
\begin{align*}
\mathbf{d} & = - (\mathbf{E}^{\top} \mathbf{E})^{-1} \mathbf{E}^{\top} \mathbf{e} + (\mathbf{E}^{\top} \mathbf{E})^{-1} \mathbf{E}^{\top} \xi \tag{2.38} \\
\mathbf{y}_k & = \mathbf{y}_{k-1} + \sum_{i=1}^{r} \mathbf{d}_i \mathbf{y}_{k-1-i} ; \quad \mathbf{u}_k = \mathbf{u}_{k-1} + \sum_{i=1}^{r} \mathbf{d}_i \mathbf{u}_{k-1-i} \tag{2.39} \\
\mathbf{b} & = (\mathbf{\Omega}^{\top}(\mathbf{u}_r, \mathbf{y}_r, \mathbf{e}) \mathbf{\Omega}(\mathbf{u}_r, \mathbf{y}_r, \mathbf{e}))^{-1} \mathbf{\Omega}^{\top}(\mathbf{u}_r, \mathbf{y}_r, \mathbf{e}) (\mathbf{y}_r - \xi) \tag{2.40}
\end{align*}
\]

b) The extended matrix method:

premultiplying eq. (2.35) with \( \mathbf{\Omega}^{\top}(\mathbf{u}, \mathbf{y}, \mathbf{e}) \) gives the following set:
\[
\begin{align*}
\mathbf{\Omega}^{\top}(\mathbf{u}, \mathbf{y}) \mathbf{y} & = \mathbf{\Omega}^{\top}(\mathbf{u}, \mathbf{y}) \mathbf{\Omega}(\mathbf{u}, \mathbf{y}) \mathbf{b} - \mathbf{\Omega}^{\top}(\mathbf{u}, \mathbf{y}) (\mathbf{Ed} - \xi) \tag{2.41} \\
\mathbf{E}^{\top} \mathbf{y} & = \mathbf{E}^{\top} \mathbf{\Omega}(\mathbf{u}, \mathbf{y}) \mathbf{b} - \mathbf{E}^{\top} \mathbf{Ed} + \mathbf{E}^{\top} \xi \tag{2.42}
\end{align*}
\]
Now we know that: \( \mathbf{e} = \mathbf{y} - \Omega(u,y) \mathbf{b} \)

so eq.(2.42) can be written as:

\[
\mathbf{E}^T \mathbf{e} = - (\mathbf{E}^T \mathbf{E}) \mathbf{d} + \mathbf{E}^T \mathbf{f}
\]

(2.44)

and it is seen that eq.(2.44) and eq.(2.38) are the same.

We know that:

\[
\mathbf{E} \mathbf{d} = \begin{bmatrix}
\mathbf{e}^q & \cdots & \mathbf{e}^{q+1-r} \\
\mathbf{e}^{q+1} & \cdots & \mathbf{e}^{q+2-r} \\
& \ddots & \ddots \\
\mathbf{e}^{q+N-1} & \cdots & \mathbf{e}^{q+N-r}
\end{bmatrix}
\begin{bmatrix}
d_1 \\
d_2 \\
\vdots \\
d_r
\end{bmatrix} = \mathbf{d} \mathbf{e}'.
\]

Defining the vectors \( \mathbf{y}' \) and \( \mathbf{u}' \) in an analog way as \( \mathbf{e}' \), we get:

\[
\mathbf{e}' = \mathbf{y}' - \Omega'(u,y) \mathbf{b}
\]

and thus:

\[
\mathbf{E} \mathbf{d} = \mathbf{D} \mathbf{y}' - \mathbf{D} \Omega'(u,y) \mathbf{b}.
\]

(2.45)

The matrix \( \Omega'(u,y) \) is defined as follows:

\[
\Omega'(u,y) = \begin{bmatrix}
u_{q+1-r} & \cdots & u_{q+1-p-r} & y_q & \cdots & y_{1-r} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
u_{q} & \cdots & u_{q-p} & y_{q-1} & \cdots & y_0 \\
u_{q+1} & \cdots & u_{q+1-p} & y_q & \cdots & y_1 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
u_{q+N-1} & \cdots & u_{q+N-1-p} & y_{q+N-2} & \cdots & y_{N-1}
\end{bmatrix}
\]
Substituting eq. (2.45) in eq. (2.41) we get:
\[
\Omega^T(u,y) (\chi + D\chi') = \Omega^T(u,y) (\Omega(u,y) + D\Omega'(u,y))\delta + \Omega^T(u,y)\xi
\]
\[(2.46)\]

Now:
\[
\begin{bmatrix}
    y_{q+1} \\
    y_{q+2} \\
    \vdots \\
    y_{q+N}
\end{bmatrix}
+ D
\begin{bmatrix}
    d_r \cdots \cdots d_1 \cdots \cdots 0 \cdots \cdots 0 \\
    0 \cdots \cdots d_r \cdots \cdots 0 \cdots \cdots 0 \\
    \vdots \\
    0 \cdots \cdots 0 \cdots \cdots d_r \\
\end{bmatrix}
\begin{bmatrix}
    y_{q+1-r} \\
    y_{q+2-r} \\
    \vdots \\
    y_{q+N-1}
\end{bmatrix}
\]
\[
\begin{bmatrix}
    y_{q+1} + d_1 y_{q+1} + \cdots + d_r y_{q+1-r} \\
    y_{q+2} + d_1 y_{q+2} + \cdots + d_r y_{q+2-r} \\
    \vdots \\
    y_{q+N} + d_1 y_{q+N} + \cdots + d_r y_{q+N-r}
\end{bmatrix} = \begin{bmatrix}
    y_{q+1}^* \\
    y_{q+2}^* \\
    \vdots \\
    y_{q+N}^*
\end{bmatrix}
\]
\[(2.47)\]

So it can be seen that: \( \chi + D\chi' = \chi^* \)
and in an analogous way:
\[\Omega(u,y) + D\Omega'(u,y) = \Omega(u^*,y^*) \]
\[(2.48)\]

Eq. (2.46) can be rewritten:
\[
b = (\Omega^T(u,y)\Omega(u^*,y^*))^{-1} \Omega^T(u,y) (\chi - \xi).
\]
\[(2.49)\]

As we can see eq. (2.49) and eq. (2.40) do correspond quite well. The only difference to be seen is that while eq. (2.40) gives a L.S. solution of
\[
\chi^* = \Omega(u^*,y^*) b + \xi
\]
\[(2.28b)\]
eqq (2.49) gives a sort of "instrumental variable" (see chapter 3) solution of this same equation, where the "instrumental variable matrix" \(Z\) is equal to \(\Omega(u,y)\).

The results given in chapter 4, 4.7.2, are obtained by taking \(\Omega^T(u,w)\) instead of \(\Omega^T(u,y)\) in eq. (2.49). This means that the algorithm:
\[
b' = (\Omega^T(u,w,e)\Omega(u,y,e))^{-1} \Omega^T(u,w,e) \chi
\]
has been instrumented. The elements $w_k$ are taken from the output sequence of the model of the process. The choice of the matrix $\Omega^T(u,w)$ will be discussed in chapter 3.

Remark 2: Another closely related method, proposed by Young(16) will be given below.

The extended matrix $\Omega(u,y,\xi)$ is used in this method. Indeed we can write eq.(2.20) in matrix form as follows:

$$y = \Omega(u,y,\xi) \mathbf{b}^n + \xi$$

(2.50)

where:

$$\mathbf{b}^n = (b^T, c^T)$$

and:

$$c^T = (c_1, c_2, ..., c_s)$$

A L.S. solution of eq.(2.50) gives estimates of the parameters $a_i$ and $b_j$ as well as the parameters $c_i$, which are directly related with the D.E. description (2.20).

The filling of the matrix $\Omega(u,y,\xi)$ by calculated values of $x_k$ can be done by using the relation:

$$x_k = y_k + a_1 y_{k-1} + ... + a_q y_{k-q} - b_0 u_k - ... - b_p u_{k-p}
- \epsilon_1 x_{k-1} - ... - \epsilon_s x_{k-s}$$

(2.51)

(where for $k = 0$ all $\tilde{\epsilon}_i = 0$; the $\tilde{\epsilon}_i$ are the estimates of the parameters $c_i$).

As the iterative solution proceeds $\tilde{\xi}$ will be filled with estimations of the $\mathbf{c}$ vector and if the algorithm is convergent, the samples $x_k$ will tend to the samples $\xi_k$. In this case $\tilde{\xi}$ will be a better and better estimator of $x$, as well as $\tilde{\xi}$ will be a better and better estimator of $c$.

The algorithm:

$$\mathbf{b}' = (\Omega^T(u,y,\xi)\Omega(u,y,\xi))^{-1} \Omega^T(u,y,\xi) y$$

(2.52)

as well as its modified form:

$$\mathbf{b}' = (\Omega^T(u,w,\xi)\Omega(u,w,\xi))^{-1} \Omega^T(u,w,\xi) y$$

(2.53)

have been instrumented.

The results obtained are given for both algorithms in chapter 4, 4.7.3 resp. 4.7.4.
3. The instrumental variable method

3.1 Statement of the problem

Consider the process \( P \) described by the following D.E.:

\[
(1 + A(z^{-1}))x_k = \{b_0 + B(z^{-1})\}u_k
\]

where:

\[
A(z^{-1}) = a_1 z^{-1} + a_2 z^{-2} + \ldots + a_q z^{-q},
\]

\[
B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \ldots + b_p z^{-p}.
\]

The roots of the polynomial:

\[
1 + A(z^{-1}) = 0
\]

all lay inside the unit circle in the z-plane (i.e., the system is assumed to be stable).

Now suppose this process to be disturbed by an additive noise signal \( n_k \) (not necessarily "white" noise), with zero mean

\[
E(n_k) = 0.
\]

In this case, eq. (3.1) can be rewritten as a relation between the observable (disturbed) output \( y_k \) and the input \( u_k \):

\[
(1 + A) y_k = (b_0 + B) u_k + e_k
\]

with:

\[
e_k = (1 + A) n_k.
\]

It will be assumed that the noise \( n_k \) is the random process output from a general filter driven by a white noise source \( \xi_k \), and the input-output relation is:

\[
(1 + F) n_k = (1 + G) \xi_k.
\]

Combining eq. (3.6) and eq. (3.5) we get (cfr. Hastings-James & Sage (15)):

\[
(1 + A) y_k = (b_0 + B) u_k + \frac{(1+A)(1+G)}{(1+F)} \xi_k
\]

and putting

\[
\frac{(1+A)(1+G)}{(1+F)} = 1 + C = 1 + c_1 z^{-1} + \ldots + c_s z^{-s},
\]

we get (cfr. Aström, Bohlin & Wensmark (9)):

\[
(1 + A) y_k = (b_0 + B) u_k + (1 + C) \xi_k.
\]

Let \( q > \max(p,s) \) - which is not essential. If the sequences \( \{u_k\}^{N+q}_{k=1} \) and \( \{y_k\}^{N+q}_{k=1} \) are given, eq. (3.9) gives rise to the following set of equations:
\[
\begin{align*}
\begin{cases}
y_{q+1} = b_0 u_{q+1} + \cdots + b_p u_{q+1-p} - a_1 y_{q} - \cdots - a_q y_1 + \xi_{q+1} + \cdots + c_s \xi_{q+1-s} \\
y_{q+2} = b_0 u_{q+2} + \cdots + b_p u_{q+2-p} - a_1 y_{q+1} - \cdots - a_q y_2 + \xi_{q+2} + \cdots + c_s \xi_{q+2-s} \\
\vdots \\
y_{q+N} = b_0 u_{q+N} + \cdots + b_p u_{q+N-p} - a_1 y_{q+N-1} - \cdots - a_q y_N + \xi_{q+N} + \cdots + c_s \xi_{q+N-s}
\end{cases}
\end{align*}
\]

Using matrix notation eq. (3.10) becomes:

\[
Y = \Sigma(u,y) B + \Xi C
\]

with:

\[
\begin{align*}
X^T &= \begin{pmatrix} y_{q+1}, y_{q+2}, \ldots, y_{q+N} \end{pmatrix}, \\
B^T &= \begin{pmatrix} b_0, b_1, \ldots, b_p, -a_1, \ldots, -a_q \end{pmatrix}, \\
C^T &= \begin{pmatrix} 1, c_1, c_2, \ldots, c_s \end{pmatrix},
\end{align*}
\]

and:

\[
\Sigma(u,y) = \begin{bmatrix}
  u_{q+1} & \cdots & u_{q+1-p} & y_q & \cdots & y_1 \\
  u_{q+2} & \cdots & u_{q+2-p} & y_{q+1} & \cdots & y_2 \\
  \vdots & & \vdots & \vdots & & \vdots \\
  u_{q+N} & \cdots & u_{q+N-p} & y_{q+N-1} & \cdots & y_N
\end{bmatrix} = (U' Y),
\]

\[
\Xi = \begin{bmatrix}
  \xi_{q+1} & \cdots & \xi_{q+1-s} \\
  \xi_{q+2} & \cdots & \xi_{q+2-s} \\
  \vdots & & \vdots \\
  \xi_{q+N} & \cdots & \xi_{q+N-s}
\end{bmatrix}
\]

Let:

\[
\Xi C = \underline{e}
\]

with:

\[
\underline{e}^T = \begin{pmatrix} e_{q+1}, e_{q+2}, \ldots, e_{q+N} \end{pmatrix},
\]

and define an \( N \times m \) matrix \( Z \), where \( m = p+q+1 \).

We can then write the following expression:

\[
\underline{e} = (Z^T \Sigma(u,y))^{-1} Z^T Y = B + (Z^T \Sigma(u,y))^{-1} Z^T \underline{e}.
\]

We now assume that the elements of \( Z \) are chosen so that:

1) \( \text{plim } \frac{1}{N} Z^T \Sigma(u,y) \) gives rise to a non-singular matrix;

2) \( \text{plim } \frac{1}{N} Z^T \underline{e} = 0 \) (zero vector).

( the definition of plim is given in appendix I ).
In this case it can be proved, using Slutsky's theorem (appendix 1), that
\[ \operatorname{plim} \hat{b} = \operatorname{plim} (Z^T \hat{\Omega}(u,y))^{-1} Z^T y = b. \] (3.14)
A matrix \( Z \), which satisfies the conditions mentioned above, is called an "instrumental variable" matrix (Reiersoll (17), Wong (18)).

The rate of convergence of the estimator \( \hat{b} \) (which is indeed a vector random variable) to the unknown parameter vector \( b \), will depend heavily on the actual choice of \( Z \). A matrix \( Z^* \) which minimizes the asymptotic variance of the estimate exists (Wong (18)).

3.2 The optimal instrumental variable matrix \( Z^* \)

The asymptotic covariance matrix of the estimator \( \hat{b} \) is given by:
\[ \operatorname{cov} \hat{b} = \lim_{N \to \infty} \left\{ (Z^T \hat{\Omega}(u,y))^{-1} Z^T e e^T Z (\hat{\Omega}^T(u,y)Z)^{-1} \right\} \]
\[ = \lim_{N \to \infty} (Z^T \hat{\Omega}(u,x))^{-1} Z^T \Phi Z (\hat{\Omega}^T(u,x)Z)^{-1} \] (3.15)

(Wong (18): assumptions on p. 42 and his appendix IV), where \( \Phi = E(e e^T) \) is a positive definite matrix, and \( \hat{\Omega}(u,x) \) has the same construction as \( \Omega(u,y) \) defined in 3.1, with the elements \( y_i \) replaced by the corresponding elements \( x_i \) (where \( x_k \) is the unmeasurable system output) i.e., \( \hat{\Omega}(u,x) = (U^T X) \).

It has been proved by Wong (18) that the rate of convergence in covariance is \( 1/N \).

Let:
\[ Z^* = \Phi^{-1} \Omega(u,x) \] (3.16)

then eq. (3.15) can be rewritten with \( Z \) replaced by \( Z^* \):
\[ \lim_{N \to \infty} (\hat{\Omega}^T(u,x)\Phi^{-1} \hat{\Omega}(u,x))^{-1} \hat{\Omega}^T(u,x)\Phi^{-1} \hat{\Omega}(u,x) (\hat{\Omega}^T(u,x)\Phi^{-1} \hat{\Omega}(u,x))^{-1} \]
\[ = \lim_{N \to \infty} (\hat{\Omega}^T(u,x)\Phi^{-1} \hat{\Omega}(u,x))^{-1} \]
(3.17)

And it has been proved by Wong (18) that
\[ (Z^T \hat{\Omega}(u,x))^{-1} Z^T \Phi Z (Z^T \hat{\Omega}(u,x)Z)^{-1} \geq (\hat{\Omega}^T(u,x)\Phi^{-1} \hat{\Omega}(u,x))^{-1}. \] (3.18)

Any other choice of \( Z \) than the one given in eq. (3.16) will result in an asymptotic covariance matrix that is larger than the one obtained with
Z = Z^*. So the matrix Z^* = \Phi^{-1}\Omega(u,x) will be called "optimal instrumental variable matrix".

Now in practice \( \phi \) an optimal I.V. matrix can not be constructed, because:

1) the sequence \( \{x_k\}_{k=1}^{N+q} \) is not available and so we cannot construct a matrix \( \Omega(u,x) = (u|x) \);
2) we do not know the parameters \( c_i \), and so we cannot compute the matrix \( \Phi \).

However, the idea is clear. Whenever we know something about the parameters \( c_i \) and we have a large number of observations available, we will try to generate an I.V. matrix as "close" to Z^* as possible.

### 3.3 Realization of the asymptotic optimal I.V. matrix (Wong (18))

This paragraph is based on the method given by Wong (18), and on the process description given in eq. (3.5). In the following paragraph, this same method but now based on the process description given in eq. (3.9) will be considered. It will be seen that this latter scheme to obtain Z^* without matrix inversion is much easier to follow (and to do in practice) than the former one. So, in reading this thesis, this paragraph can be skipped without any loss of continuity, for the essence of it will be described in section 3.4.

We will assume that we know the parameters of the process completely i.e.

\[ b, \Phi, \Omega(u,x) \]

are given, as well as the parameters of the general noise-filter i.e.

\[ f_i \text{ and } g_j \quad (i = 1, \ldots, v; j = 1, \ldots, u) \]

So the discussion given here will be concerned with the idealized mathematical problem of computing approximately the matrix Z^* without any matrix inversion.

Rewrite eq. (3.6):

\[ n_k + \sum_{i=1}^{v} f_i n_{k-i} = \xi_k + \sum_{i=1}^{u} g_i \xi_{k-1} \quad (k = \ldots, -1, 0, +1, \ldots) \]

where \( \xi_k \) is a sample taken from a "white" noise process with the properties
and
\[ E(\xi_k) = 0 \]
\[ E(\xi_k \xi_j) = \delta_{kj} \frac{e}{j} \]

(where \( \delta_{kj} \) is the Kronecker-delta).

Fig. 3.1: Noise-filter

If now in eq. (3.6) the initial conditions \( n_k = 0 \) \((k \leq 0)\) are used, this will give rise to an error in the sequence \( \{n_k\}^\infty_1 \) derived from the sequence \( \{\xi_k\}^-\infty^+ \) although this error will decrease as \( k \) grows (cfr. initial conditions in a D.E. are important only for transients). It has been assumed that the polynomial
\[ 1 + f_1 z^{-1} + \ldots + f_M z^{-M} = 0 \]
has its roots inside the unit circle in the \( z \)-plane (stable noise-filter).

Remark 1: Note that if \( n_k = \xi_k \) (e.q. all \( f_i \) and \( g_i \) equal to zero) for all \( k \), there will be no transient, and consequently no error.

From eq. (3.5), we see that:
\[ e = \mathbf{N} \mathbf{a} \]

where:
\[ a^T = (1, a_1, \ldots, a_q) \]

and
\[
\mathbf{N} = \begin{bmatrix}
 n_{q+1} & n_q & \cdots & n_1 \\
 n_{q+2} & n_{q+1} & \cdots & n_2 \\
 \vdots & \vdots & \ddots & \vdots \\
 n_{q+N} & n_{q+N-1} & \cdots & n_N
\end{bmatrix}
\]

We will define now the following matrices:
Then eq. (3.19) can be rewritten as:

\[ \mathbf{e} = \mathbf{M} \mathbf{a} = \mathbf{A} \mathbf{n} \]  

(3.22)

where:

\[ \mathbf{n}^T = (n_1, n_2, \ldots, n_{q+1}, \ldots, n_N, \ldots, n_{q+N}) \]

and eq. (3.6) can be rewritten as:

\[ \mathbf{F} \mathbf{n} = \mathbf{G} \mathbf{\xi} \]  

(3.23)

where

\[ \mathbf{\xi}^T = (\xi_1, \xi_2, \ldots, \xi_{q+1}, \ldots, \xi_N, \ldots, \xi_{q+N}) \]

and thus:

\[ \mathbf{n} = \mathbf{F}^{-1} \mathbf{G} \mathbf{\xi} \]  

(3.24)
Combining eq. (3.24) and eq. (3.22) we get:

\[ e = AF^{-1}G \xi. \]  

(3.25)

Remark 2: \( F^{-1} \) does always exist, because \( \det(F) = 1 \). At the same time \( F^{-1} \) is a lower triangular matrix (L.T.M.) as well as \( F \) is a L.T.M.

Substituting eq. (3.25) in \( \phi = E (e e^T) \) gives:

\[ \phi = (AF^{-1}G)(AF^{-1}G)^T \sigma^2. \]  

(3.26)

We normalize \( \sigma^2 \) to one (this introduces no restriction).

From eq. (3.26) and eq. (3.16) it can be seen that

\[ (AF^{-1}G)(AF^{-1}G)^TZ^* = \Omega(u,x). \]  

(3.26a)

We will define the matrices \( V^*, H^*, T^* \) and \( W^* \) using the following relations

\[
\begin{align*}
V^* &= A^T Z^* \\
H^* &= F^{-1} V^* \\
W^* &= G^T H^* \\
T^* &= F^{-1} G W^* \\
\Omega(u,x) &= A T^*
\end{align*}
\]  

(3.27)

Using the relations (3.27 a,b,c,d,e), the following scheme will be proposed for the evaluation of the \( N \times m \) matrix \( Z^* \) without applying any matrix inversion.

1. \( A \) and \( \Omega(u,x) \) are known: generation of an \((N+q)\times m\) matrix \( T^* \) according to eq. (3.27e). We write eq. (3.27e) explicitly:

\[
\begin{bmatrix}
  u_{q+1} & \cdots & u_{q+2-j} & \cdots & u_{q+1-p} & x_q & \cdots & x_{m+1-j} & \cdots & x_1 \\
  \vdots & \ddots & \vdots & & \vdots & & \vdots & & \vdots & \vdots \\
  u_{q+k} & \cdots & u_{q+k+1-j} & \cdots & u_{q+k-p} & x_{q+k-1} & \cdots & x_{m+k-j} & \cdots & x_k \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  u_{q+N} & \cdots & u_{q+N+1-j} & \cdots & u_{q+N-p} & x_{q+N-1} & \cdots & x_{m+N-j} & \cdots & x_N \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
  \phi \end{bmatrix}
\]

\[
\begin{bmatrix}
  (j=1,2,\ldots,p+1) \\
  (j=p+2,p+3,\ldots,p+q+1=m)
\end{bmatrix}
\]
From eq. (3.28) results:

\[ \sum_{i=0}^{q} a_i t_{k+q-i,j} = u_{q+i+k-j} \]  \hspace{1cm} (3.29a)

\[ \sum_{i=0}^{q} a_i t_{k+q-i,j} = x_{m+k-j} \]  \hspace{1cm} (3.29b)

(\( k = 1, 2, \ldots, N; \ j = 1, 2, \ldots, p+1 \)),
and

(\( k = 1, 2, \ldots, N; \ j = p+2, p+3, \ldots, m \)),

with \( a_0 = 1 \).

These equations give rise to the filterscheme given in Fig. 3.2. This filter has only a backward part, for which the parameters are equal to those of the backward part of the process \( P \).

**Remark 3**: as \( A \) has the dimension \( N \times (N+q) \), the equations (3.29) give a set of \( N \) equations with \( N+q \) unknown output values \( t_{k,j} \) (for each \( j \)). By choosing \( q \) of the unknown output values (initial conditions) in such a way that transients of the noise filter are eliminated, we will get the optimal matrix \( Z \). Otherwise a suboptimal matrix \( \tilde{Z} \) will result.

**Fig. 3.2**: the filters giving the same D.E. as given by eqs. (3.29 a, b).
2. \(T^*\) and \(G\) are known: generation of an \((N+q)\times m\) matrix \(W^*\) according to eq. (3.27d). Writing eq. (3.27d) explicitly, we obtain the following set of equations:
\[
\sum_{i=0}^{b} f_{i}^* k+1-i, j = \sum_{i=0}^{1} g_{i}^* w_{k+1-i}^* \tag{3.30}
\]
(k = 1, 2, \ldots, N+q; j = 1, 2, \ldots, m),
and \(f_0 = g_0 = 1\).
These equations describe a filter as given in Fig. 3.3 with the initial conditions:
\[
\begin{align*}
t^*_0, j &= t^*_{-1}, j = \ldots = t^*_{-1+2}, j = 0 \\
w^*_0, j &= w^*_{-1}, j = \ldots = w^*_{-b+2}, j = 0
\end{align*}
\]
(j = 1, 2, \ldots, m).

![Fig. 3.3: equivalent filter of eq. (3.30).](image)

3. \(G\) and \(W^*\) are known: generation of an \((N+q)\times m\) matrix \(H^*\) according to eq. (3.27c). Again, writing eq. (3.27c) explicitly gives the set of equations:
\[
\sum_{i=0}^{q+N-1} g_{i} h_{k+i,j} = W_{k,j} \tag{3.31}
\]
(k = N+q, N+q-1, \ldots, 1; j = 1, 2, \ldots, m)
with \(g_0 = 1\).
These equations describe a filter as given in Fig. 3.4, with the initial conditions:
\[
\begin{align*}
h_{q+N+1, j} &= h_{q+N+2, j} = \ldots = h_{q+N+1, j} = 0
\end{align*}
\]
(j = 1, 2, \ldots, m).
So, if the sequence \(\{W_{k,j}^*\}_{k=1}^{N+q}\) is used in reverse order, resulting from the transpose in eq. (3.27c), as the input of the filter given in Fig. 3.4, the output will deliver the sequence \(\{H_{k,j}^*\}_{k=1}^{N+q}\).
Fig. 3.4 : equivalent filter of eq. (3.31).

4. $F$ and $H^*$ are known : generation of an $(N+q) \times m$ matrix $V^*$ according to eq. (3.27b). Writing this equation explicitly, we obtain:

$$
\sum_{i=0}^{b} f_{i} h_{k+i,j}^* = v_{k,j}^* \quad (k = N+q, N+q-1, \ldots, 1; j = 1, 2, \ldots, m),
$$

with $f_0 = 1$.

This is once again a description of a filter like that given in Fig. 3.5, with as initial conditions:

$$
h_{q+N+1,j}^* = h_{q+N+2,j}^* = \cdots = h_{q+N+b,j}^* = 0 \quad (j = 1, 2, \ldots, m).
$$

Fig. 3.5 : equivalent filter of eq. (3.32).

5. $A$ and $V^*$ are known : generation of an $N \times m$ matrix $Z^*$ according to eq. (3.27a). First rewrite eq. (3.27a) in the following way:

$$
V^* = \begin{bmatrix} V^* \\ \vdots \\ V^* \end{bmatrix} = \begin{bmatrix} A \\ \vdots \\ A \end{bmatrix} Z^* \quad (3.33)
$$

or

$$
\bar{V}^* = \bar{A} Z^* \quad (3.33a)
$$

and

$$
\check{V}^* = \check{A} Z^* \quad (3.33b)
$$

where :

$\omega_{ij}$

$\psi_{ij}$

$\chi_{ij}$

$\phi_{ij}$

$\lambda_{ij}$

$\mu_{ij}$

$\nu_{ij}$

$\alpha_{ij}$

$\beta_{ij}$

$\gamma_{ij}$

$\delta_{ij}$

$\epsilon_{ij}$

$\zeta_{ij}$

$\eta_{ij}$

$\theta_{ij}$

$\iota_{ij}$

$\kappa_{ij}$

$\lambda_{ij}$

$\mu_{ij}$

$\nu_{ij}$

$\xi_{ij}$

$\rho_{ij}$

$\sigma_{ij}$

$\tau_{ij}$

$\upsilon_{ij}$

$\phi_{ij}$

$\chi_{ij}$

$\psi_{ij}$

$\omega_{ij}$
Writing eq. (3.33b) explicitly, gives the following set of equations:

\[
\sum_{i=0}^{q} a_i \dot{z}_{k+i,j} = \dot{v}_{q+k,j}
\]  

(3.34)

\( k = 1, 2, \ldots, N; \ j = 1, 2, \ldots, m \)

with \( a_0 = 1 \).

This is a description of the filter given in Fig. 3.6, with initial conditions:

\[
\hat{z}_{N+1,j} = \hat{z}_{N+2,j} = \cdots = \hat{z}_{N+q,j} = 0
\]

(\( j = 1, 2, \ldots, m \)).

In general, the sequence \( \{ \hat{z}_{k,j} \}^{N+q}_N \) found in this way will not satisfy eq. (3.33a). If, however, all initial conditions are the same as stated throughout this discussion, the sequence will indeed satisfy eq. (3.33a), (Wong (18)) and will contain the components of the optimal I.V. matrix.

Remark 4: the scheme given here has already been worked out by Wong (18), taking only a backward process. It has been given here using a general process description (forward and backward part). In this case, it is clearly seen that all filters will be working on the sequence \( \{ \hat{v}_{k,j} \}^{N+q}_1 \) as well as on the sequence \( \{ \dot{u}_{k,j} \}^{N+q}_1 \).
Remark 5: It has been indicated by Wong (18) that the matrix $Z$ computed with the filters discussed before, and using as initial conditions for the first filter

$$t_1,j = t_2,j = \ldots = t_q,j = 0$$

( $j = 1,2,\ldots,m$ ), is a "good" approximation of the asymptotic optimal I.V. matrix (Wong (18), p. 70).

$$\begin{bmatrix}
    z_{j+1}^{qN} & a \\
    z_{j+1}^{qN} & a
\end{bmatrix}
$$

( $j = 1,2,\ldots,m$ )

Fig. 3.6: equivalent filter of eq. (3.34).

Remark 6: The model output sequence $\{w_k\}_{k=1}^{N+q}$ (Fig. 3.7) is an instrumental variable sequence (Wong (18), p. 90). Indeed, a matrix $Z = (U,W)$, where the elements of $W$ are the $w_k$ corresponding with the $y_k$ in the matrix $Y$, satisfies the condition:

$$\lim_{N \to \infty} \frac{1}{N} Z^T e = 0$$

and, on an intuitive basis:

$$\lim_{N \to \infty} \frac{1}{N} (Z \Omega(u,y))^{-1}$$

exists. Consequently $Z = (U,W)$ is an I.V. matrix.

It is this matrix that has been used in the estimation programs for which the results given in chapter 4, sections 4.3, 4.4, 4.5 and 4.6 holds.

Fig. 3.7: generation of the model output sequence $\{w_k\}_{k=1}^{N+q}$. 
3.4 Alternative realization of the asymptotic optimal I.V. matrix

We will assume that we know the parameters of the process completely i.e.

\[ b, \xi, \Omega(u, x) \]

are given, and the process is described by eq. (3.9). The optimal I.V. matrix is

\[ Z^* = \phi^{-1} \Omega(u, x), \]

where:

\[ \phi = \mathbb{E}(\xi \xi^T) \]

and where \( \xi \) has been defined in eq. (3.12). Writing \( \xi \) explicitly gives:

\[
\begin{bmatrix}
\xi_{q+1} + \cdots + c_s \xi_{q+1-s} \\
\xi_{q+2} + \cdots + c_s \xi_{q+2-s} \\
\vdots \\
\xi_{q+N} + \cdots + c_s \xi_{q+N-s}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\xi_{q+1} + \cdots + c_s \xi_{q+1-s} \\
\xi_{q+2} + \cdots + c_s \xi_{q+2-s} \\
\vdots \\
\xi_{q+N} + \cdots + c_s \xi_{q+N-s}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\xi_{q+1} + \cdots + c_s \xi_{q+1-s} \\
\xi_{q+2} + \cdots + c_s \xi_{q+2-s} \\
\vdots \\
\xi_{q+N} + \cdots + c_s \xi_{q+N-s}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\xi_{q+1} + \cdots + c_s \xi_{q+1-s} \\
\xi_{q+2} + \cdots + c_s \xi_{q+2-s} \\
\vdots \\
\xi_{q+N} + \cdots + c_s \xi_{q+N-s}
\end{bmatrix}
\]

So:

\[
\xi \xi^T = C \xi \xi^T C^T
\]

and

\[
\phi = \mathbb{E}(\xi \xi^T) = C \mathbb{E}(\xi \xi^T) C^T = C C^T
\]

(as we normalized \( \sigma^2_n = 1 \))

Combining eq. (3.36) and eq. (3.16) we get:

\[
C C^T Z^* = \Omega(u, x).
\]
Z* can be solved without matrix inversion, by introducing the matrix H in the following way:

\[ \Omega(u,x) = C H \]  
(3.38a)

and

\[ H = C^T Z^* \]  
(3.38b)

(note the similarity between eq. (3.37) and eq. (3.26a)).

Eq. (3.38a) and (3.27e) have the same structure, and so eq. (3.28) can be applied, where now the parameters \(a_i\) are replaced by the parameters \(c_i\) and the elements \(t_{ij}\) are replaced by the elements \(h_{ij}\). So the resulting explicit equations from (3.38a) are given by:

\[ \sum_{i=0}^{s} c_i h_{k+s-i,j} = u_{q+1+k-j} \]  
(k = 1, 2, ..., N; j = 1, 2, ..., p+1),

and

\[ \sum_{i=0}^{s} c_1 h_{k+s-i,j} = x_{m+k-j} \]  
(k = 1, 2, ..., N; j = p+2, p+3, ..., m)

with \(c_0 = 1\)

These equations produce the filterscheme proposed in fig. 3.8.

At the same time, remark 3 made in section 3.3 holds also.

![Diagram](image)

Fig. 3.8: Equivalent filter according to eq. (3.39).

Writing eq. (3.38b) explicitly gives:

\[
\begin{bmatrix}
  h_{1j} \\
  h_{2j} \\
  \vdots \\
  h_{N+j}
\end{bmatrix}
\begin{bmatrix}
  c_0 \\
  c_1 \\
  \vdots \\
  c_1
\end{bmatrix}
= \begin{bmatrix}
  u_{q+1} \\
  u_{q+2} \\
  \vdots \\
  u_{q+N}
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
  h_{1j} \\
  h_{2j} \\
  \vdots \\
  h_{N+j}
\end{bmatrix}
\begin{bmatrix}
  z_{1j} \\
  z_{2j} \\
  \vdots \\
  z_{Nj}
\end{bmatrix}
= \begin{bmatrix}
  x_{m+1} \\
  x_{m+2} \\
  \vdots \\
  x_{m+N}
\end{bmatrix}
\]

(j = 1, 2, ..., m),

(3.40)
which is equivalent to:

\[
\sum_{i=0}^{s} c_i z_{k+i,j} = h_{k+s,j} (k = 1, 2, \ldots, N; j = 1, \ldots, m)
\]

with \( c_0 = 1 \).

The initial conditions are: \( z_{N+1,j} = \ldots = z_{N+s,j} = 0 \) \( j = 1, \ldots, m \).

This is equivalent to the filterscheme given in fig. 3.9. (Note the similarity with point 5 in section 3.4.)

![Fig. 3.9: Filter giving the D.E. of (3.41)](image)

It is seen that this method eliminates a lot of computation time, and reduces possible untrue initial conditions. At the same time, however, we reject the information that the factor \((1+A)\) is implied in the polynomial \((1+C)\).

### 3.5 Simple construction of the optimal I.V. algorithm

It has been proved by Wong (18) that the optimal I.V. matrix \( Z^* \) is given by:

\[
Z^* = \phi^{-1} \Omega(u,x)
\]

where, as defined in section 3.2:

\[
\Omega(u,x) = (U|X),
\]

and

\[
\phi = E(e e^T)
\]

Consider the equivalent noise term in the D.E. given in eq. (3.9):

\[
e_k = \xi_k + c_1 \xi_{k-1} + c_2 \xi_{k-2} + \ldots + c_s \xi_{k-s}
\]

and in matrix notation:

\[
e = C \xi
\]

where \( e, C \) and \( \xi \) are already defined in section 3.4.
Putting, \(q_{q+1-s} = \ldots = q_{q+2-s} = \ldots = q_{q} = 0\) (initial conditions), we can write the following matrix form:

\[ e = C^* \xi^* , \quad (3.42) \]

where

\[
C^* = \begin{bmatrix}
1 & 0 & & & & \\
& c_1 & 1 & & & \\
& & \vdots & \ddots & \ddots & \\
& & & c_s & 1 & \\
& & & 0 & \vdots & \\
& & & 0 & \cdots & 0 \cdot c_s & 1
\end{bmatrix}
\]

and

\[
\xi^* = (q_{q+1}, q_{q+2}, \ldots, q_{q+N})
\]

We will evaluate now the inverse of \(C^*\), denoted by \(D^* = C^{-1} \). As \(C^*\) is a L.T.M., \(D^*\) also is a L.T.M. \(D^*\) exists as \(\text{det.}(C^*) = 1\).

Consider the following matrix identity:

\[
\begin{bmatrix}
1 & 0 & & & & \\
& c_1 & 1 & & & \\
& & \vdots & \ddots & \ddots & \\
& & & c_s & 1 & \\
& & & 0 & \vdots & \\
& & & 0 & \cdots & 0 \cdot c_s & 1
\end{bmatrix} \cdot \begin{bmatrix}
1 & 0 & & & & \\
& d_1 & 1 & & & \\
& & \vdots & \ddots & \ddots & \\
& & & d_r & 1 & \\
& & & 0 & \vdots & \\
& & & 0 & \cdots & 0 \cdot d_r & 1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & & & & \\
& 0 & 1 & & & \\
& & \vdots & \ddots & \ddots & \\
& & & 0 & 1 & \\
& & & 0 & \vdots & \\
& & & 0 & \cdots & 0 \cdot 1 & 1
\end{bmatrix} = I,
\]

(3.43)

where \(c_0 = d_0 = 1\) and: \(c_j (j > s) = 0\);
\(d_j (j > r) = 0\).

Eq. (3.43) gives rise to the following set of equations relating all \(c_j\) and \(d_i\):

\[
\sum_{i=1}^{j+1} c_{j+1-i} \cdot d_{i-1} = 0 \quad (\text{all } j > 0)
\]

(3.43a)
Example: \[ j = 2 + c_2 d_0 + c_0 d_1 = 0 \quad \text{or: } \quad d_1 = -c_2 \]
\[ j = 3 + c_2 d_0 + c_1 d_1 + c_0 d_2 = 0 \quad \text{or: } \quad d_2 = -c_2 + c_1 \]

Remark 1: The \( d_i \) resulting from eq. (3.43a) are exactly the coefficients of the polynomial \((1 + c(z^{-1}))^{-1} = (1 + c_1 z^{-1} + \ldots + c_s z^{-s})^{-1}\). Indeed eq. (3.43a) could be obtained from inspection of the following relation between \( c_j \) and \( d_i \):
\[ (1 + c_1 z^{-1} + \ldots + c_s z^{-s})(1 + d_1 z^{-1} + \ldots + d_r z^{-r}) = 1 \]  
(3.43b)

Remark 2: In eq. (3.43) and (3.43a,b) it is assumed that all \( d_i \) (i > r) are small enough to be neglected.

Combining eq. (3.42) and eq. (3.16), we get:
\[ \Phi = (D^*)^{-1}(D^*)^{-T} \]
and thus: \[ Z^* = D^* \Omega(u,x). \]  
(3.44)

The optimal I.V. algorithm is then given by:
\[ \beta^* = (Z^T \Omega(u,y))^{-1} Z^T y \]
\[ = \{(\Omega^T(u,x)D^*)D^* \Omega(u,y)}^{-1}(\Omega^T(u,x)D^*)D^* y. \]

Now define:
\[ \Omega^*(u,x) = D^* \Omega(u,x); \quad \Omega^*(u,y) = D^* \Omega(u,y) \]
and
\[ Y^* = D^* Y, \]  
(3.45)

we get:
\[ \beta^* = (\Omega^*T(u,x)\Omega^*(u,y))^{-1} \Omega^*T(u,x)Y^*. \]  
(3.46)

This relation implies that, by filtering the sequences \( \{u_k\}_{1}^{N+q} \), \( \{y_k\}_{1}^{N+q} \) and \( \{x_k\}_{1}^{N+q} \) with a filter which only has a forward part, of which the parameter values are given by the inverse parameters of the equivalent noise filter (given in eq. (3.9)), we will arrive at an algorithm which is not only consistent, but has optimal properties with respect to the asymptotic covariance matrix.

Remark 3: As the sequence \( \{x_k\}_{1}^{N+q} \) as well as the parameters \( d_i \) are unknown, we will introduce in practice the following approximations:

1) instead of taking \( \Omega(u,x) \), we will use the matrix \( \Omega(u,w) \) where the
elements $w_i$ are taken from the model output sequence $\{w_k\}_{k=1}^{N+q}$ (see fig. 3.7). It has been shown that this matrix is a "good" approximation of the matrix $\Omega(u,x)$ (remark 6, section 3.3).

2) The parameters $d_i$ can be estimated using the classical L.S. estimator. Indeed, the equivalent noise was given by

$$e_k = \{1+C(z^{-1})\} \xi_k,$$

and thus:

$$\xi_k = \{1+D(z^{-1})\} e_k$$

or:

$$e_k = -d_1 e_{k-1} - d_2 e_{k-2} - \cdots - d_r e_{k-r} + \xi_k$$

Now suppose the sequence $\{e_k\}_{k=1}^{N+q}$ is available. Then eq. (3.48b) can be written in matrix notation:

$$\begin{bmatrix} e \\ \xi \end{bmatrix} = -E \begin{bmatrix} d \\ \xi \end{bmatrix}$$

with:

$$e_T = (e_{q+1}, e_{q+2}, \ldots, e_{q+N}),$$
$$\xi_T = (\xi_{q+1}, \xi_{q+2}, \ldots, \xi_{q+N}),$$
$$d_T = (d_1, d_2, \ldots, d_r)$$

and:

$$E = \begin{bmatrix} e_{q-1} & e_{q-2} & \cdots & e_{q+1-r} \\ \cdots & \cdots & \cdots & \cdots \\ e_{q+N-1} & \cdots & \cdots & e_{q+N-r} \end{bmatrix}$$

An unbiased (L.S.) estimator of $\hat{d}$ is given by:

$$\hat{d} = -E_T (E E)^{-1} E_T e$$

(3.50)

It is clear that the sequence $\{e_k\}_{k=1}^{N+q}$ is not available at all. If, however, an estimate $\hat{d}$ has been obtained (e.g. by applying a simple I.V. estimator on $\{u_k\}_{k=1}^{N+q}$ and $\{y_k\}_{k=1}^{N+q}$), we can use this estimate to generate the sequence $\{-e_k\}_{k=1}^{N+q}$:

$$-e_k = (u_k, u_{k-1}, \ldots, u_{k-q}, y_{k-1}, \ldots, y_{k-q}) \hat{d} \cdot y_k$$

(3.51)
(with initial conditions $y_0 = y_{-1} = \ldots = y_{-q} = u_0 = u_{-1} = \ldots = u_{-p} = 0$)

Taking now the estimator given in eq. (3.50) and applying it to the sequence $\{e_k\}_{k=1}^{N+q}$, we will get an estimate $\gamma_1$ of $d$.

With this $\gamma_1$ we can filter the sequences $\{u_k\}_{k=1}^{N+q}$, $\{y_k\}_{k=1}^{N+q}$ and $\{w_k\}_{k=1}^{N+q}$, yielding $\Omega_1^*(u,w)$, $\Omega_1^*(u,y)$ and $\gamma_1^*$, and now use the estimator (3.46), i.e.:

$$
\beta_1 = \left[\Omega_1^T(u,w)\Omega_1^*(u,y)\right]^{-1} \Omega_1^*(u,w) y_1^*
$$

(3.52)

Now again eq. (3.51) will give, with this $\beta_1^*$ instead of $\beta_0^*$, a new sequence $\{e_k\}_{k=1}^{N+q}$, and so on. Using a similar criterium as mentioned in section 2.2, the algorithm can be stopped.

The method proposed in this section is closely related to the method introduced by Clarke (13) which is given in chapter 2, section 2.2.

In practice we will apply a recursive scheme. After each iteration step the vectors $\gamma$ and $\beta$ are updated using the new information and an exponentially weighted part of the past information. (cfr. chapter 2, section 2.3).

The results obtained with the method proposed in this paragraph are given in chapter 4, section 4.8.
4. Experimental Results and Discussion

4.1 Estimation schemes

Consider a physical process $P$, which can be described by the following difference equation (D.E.):

$$\sum_{i=0}^{q} a_i x_{k-i} = \sum_{i=0}^{p} b_i u_{k-i}$$  \hspace{1cm} (4.1)

and let $q \geq p$ (this is no restriction). Suppose the process output $x_k$ is disturbed by an additive noise signal $n_k$, and thus $y_k = x_k + n_k$ is the disturbed observable output. The sequences $\{u_k\}_{0}^{N+q}$ and $\{y_k\}_{0}^{N+q}$ are available. Eq. (4.1) can now be rewritten as follows:

$$\sum_{i=0}^{q} a_i y_{k-i} = \sum_{i=0}^{p} b_i u_{k-i} + e_k$$  \hspace{1cm} (4.2a)

with:

$$e_k = \sum_{i=0}^{q} a_i n_{k-i}$$  \hspace{1cm} (4.2b)

4.1.1 Algorithm 1 (section 3.1.)

Rewrite Eq. (4.2a) in the following way:

$$u_k = \sum_{i=0}^{q} l_{a_i} y_{k-i} - \sum_{i=1}^{p} l_{b_i} u_{k-i} - \frac{1}{b_o} e_k$$  \hspace{1cm} (4.3)

with: $l_{a_i} = a_i / b_o$ (i = 0, 1, ..., q)

and: $l_{b_i} = b_i / b_o$ (i = 1, 2, ..., p)

Eq. (4.3) leads to the following L.V. estimator:

$$\hat{\beta} = (1Z^T 1\Omega(u, y))^{-1} 1Z^T u$$ \hspace{1cm} (4.4)

where: $\hat{\beta} = (-\beta_1, -\beta_2, \ldots, -\beta_p, \beta_0, \beta_1, \ldots, \beta_q)$

$$u = (u_{q+1}, u_{q+2}, \ldots, u_{q+N})$$
and:

\[ 1\Omega(u, y) = 1(U \mid Y) = \begin{bmatrix} u_q & \ldots & u_{q+p-1} & y_{q+1} & \ldots & y_1 \\ u_{q+1} & \ldots & u_{q+2+p-1} & y_{q+2} & \ldots & y_2 \\ \vdots & & \vdots & & \vdots & \vdots \\ u_{q+N-1} & \ldots & u_{q+N+p-1} & y_{q+N} & \ldots & y_N \end{bmatrix} \]

\[ 1Z = 1(U' \mid w) = 1\Omega(u, w) \quad (4.5) \]

The elements \( y_i \) \((i = 1, \ldots, N+q)\) in \( 1\Omega(u, y) \) are replaced by the corresponding elements \( w_i \) \((i = 1, \ldots, N+q)\) to form the matrix \( 1\Omega(u, w) \). These \( w_i \) are taken from the sequence \( \{w_k\}_{k=1}^{N+q} \) where \( w_k \) is the output of the "model" of the process \( P \) at time \( t = kT \).

Fig. 4.1

4.1.2. Algorithm 2 (section 3.1.)

Rewrite eq. (4.2a) in the following way:

\[ y_k = \sum_{i=0}^{p} 2b_i u_{k-i} - \sum_{i=1}^{q} 2a_i y_{k-i} + \frac{1}{a_o} \epsilon_k \quad (4.6) \]

with: \( 2a_i = a_i^* / a_o^* \) \((i = 1, 2, \ldots, q)\)

and: \( 2b_i = b_i^* / a_o^* \) \((i = 0, 1, \ldots, p)\).

Eq. (4.6) leads to the following I.V. estimator:

\[ 2\beta = (2Z^T 2\Omega(u, y))^{-1} 2Z^T Y \quad (4.7) \]

where: \( 2\beta = (2\beta_0, 2\beta_1, \ldots, 2\beta_p, -2\alpha_1, -2\alpha_2, \ldots, -2\alpha_q) \),

\( Y = (y_{q+1}, y_{q+2}, \ldots, y_{q+N}) \)

and:

\[ 2\Omega(u, y) = 2(U \mid Y) = \begin{bmatrix} u_{q+1} & \ldots & u_{q+p-1} & y_1 & \ldots & y_{q-1} \\ \vdots & & \vdots & & \vdots & \vdots \\ u_{q+N-1} & \ldots & u_{q+N+p-1} & y_{q+N-1} & \ldots & y_N \end{bmatrix} \]
with the same convention for the elements \( \omega_i \) (\( i = 1, 2, \ldots, N+q \)) for the matrix \( \Omega(u,w) \).

It will be shown in appendix IV that algorithm 2 gives excellent results in most cases. Algorithm 1 can be used with success in closed-loop situations, where we want to know the parameters of the open-loop process.

In sections 4.3, 4.4 and 4.5 algorithm 1 will be applied.
In sections 4.7 and 4.8 algorithm 2 will be applied.
In section 4.6 both algorithms have been used to show the difference between their resulting estimates.

4.2 The iterative algorithm

Consider the estimator \( \beta \) given in eq. (4.4):

\[
\beta_N = \left( \begin{array}{c}
\Omega_N(u,y) \\
\Omega_N(u,y)
\end{array} \right)_{-1} \Omega_N \left( \begin{array}{c}
u_N \\
u_N
\end{array} \right)
\]

and in an analogous way

\[
\beta_{N-1} = \left( \begin{array}{c}
\Omega_{N-1}(u,y) \\
\Omega_{N-1}(u,y)
\end{array} \right)_{-1} \Omega_{N-1} \left( \begin{array}{c}
u_{N-1} \\
u_{N-1}
\end{array} \right)
\]

Comparing eq. (4.4) and eq. (4.4a) and keeping in mind the definitions of the matrices and vectors given in (4.5), it can be seen that:

\[
\Omega_N(u,y) = \left[ \begin{array}{c}
\Omega_{N-1} \\
\Omega_{N-1}
\end{array} \right] \quad \Omega_N = \left[ \begin{array}{c}
\Omega_{N-1} \\
\Omega_{N-1}
\end{array} \right] \quad \text{and} \quad \nu_N = \left[ \begin{array}{c}
\nu_{N-1} \\
\nu_{N-1}
\end{array} \right]
\]

where

\[
\begin{array}{c}
\nu_N^T = (u_{q+N-1}, u_{q+N-2}, \ldots, u_{q+p}, y_{q+N}, \ldots, y_N), \\
\nu_N^T = (u_{q+N-1}, u_{q+N-2}, \ldots, u_{q+p}, w_{q+N}, \ldots, w_N).
\end{array}
\]

Defining:

\[
P_N^{-1} = \Omega_N(u,y)
\]

we see that:

\[
P_N^{-1} = \left( \begin{array}{c}
\Omega_N^T \\
\Omega_N^T
\end{array} \right)_{-1} \Omega_N \left( \begin{array}{c}
\nu_N \\
\nu_N
\end{array} \right) = P_{N-1} + \nu_N \nu_N^T
\]

and thus \( P_N \) can be written (cfr. Westenberg (19)):

\[
P_N = P_{N-1} - P_{N-1} \nu_N \left( 1 + \nu_N^T P_{N-1} \nu_N \right)^{-1} \nu_N^T P_{N-1}
\]

At the same time:

\[
\beta_N = P_N \Omega_N \nu_N = P_N \left( \begin{array}{c}
\Omega_N \\
\Omega_N
\end{array} \right)_{-1} \nu_N + \nu_N \nu_N^T \nu_{N+q}
\]
and this leads to:

$$\beta_N = \beta_{N-1} - P_{N-1} \mathbf{w}_N (1 + \mathbf{w}_N^T P_{N-1} \mathbf{w}_N)^{-1} (\mathbf{y}_N^T \mathbf{w}_N \beta_{N-1} - u_{N+q})$$  \hspace{1cm} (4.4c)

or:

$$\beta_N = \beta_{N-1} - P_{N-1} \mathbf{w}_N (\mathbf{w}_N^T \beta_{N-1} - u_{N+q})$$  \hspace{1cm} (4.4d)

The iterative scheme has been started with a L.S. estimator, where:

$$P_0 = a \mathbf{I} \quad (a = 10^6) \quad \text{and} \quad \beta_0 = 0$$

After m iterations the iterative scheme given by eq. (4.4b,c) is switched on. In an analogous way as Westenberg (19) it can be proved that:

$$\lim_{a \to \infty} P_N \rightarrow (1_2 \mathbf{z}_N^T \Omega_{N}(u,y))^{-1}$$

and:

$$\lim_{a \to \infty} \beta_N \rightarrow P_N 1_2 \mathbf{z}_N \mathbf{a}_N$$

The algorithm given by eq. (4.7) can be solved in an iterative manner, making use of eq. (4.4b,c) too. The matrices and vectors involved are defined in eq. (4.8).

4.3 The open-loop process with only one forward parameter.

Consider the following D.E.:

$$u_k = \sum_{i=0}^{q} a_i x_{k-i}$$ \hspace{1cm} (4.9)

![Diagram](image)

**Fig. 4.2:** The process described in section 4.3

The process output $x_k$ is disturbed by an additive white noise signal $\xi_k$, e.g.:

$$y_k = x_k + \xi_k$$

and eq. (4.9) can be written:

$$u_k = \sum_{i=0}^{q} a_i y_{k-i} - \sum_{i=0}^{q} a_i \xi_{k-i}$$ \hspace{1cm} (4.9a)
Remark I: In all simulated processes to be discussed in the following paragraphs, the input signal $u_k$ has been derived from a white noise sequence in the following way:

$$u_k = 0.6 u_{k-1} + \zeta_k$$

(4.10)

unless otherwise specified.

The white noise sequences $\{\zeta_k\}$ (input) and $\{\xi_k\}$ (output noise) have probability functions uniform on $[-1, +1]$ and $[-B, +B]$ respectively (where $B$ is variable).

The model parameter vector $\mathbf{\theta}^M$ will be updated in the following way:

$$\mathbf{\theta}_k^M = \mathbf{\theta}_k (\lambda = 0, 5, 10, \ldots)$$

The simulated process described by the D.E. (4.9) has the following parameters: $a_0 = 1; a_1 = 1.3; a_2 = -.22; a_3 = -.832; a_4 = -.269$ (e.g. $q = 4$, cfr. Wong (18)). The input signal was a white noise sequence (no filtering).

The results obtained with algorithm I are given in Table 4.1.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.883</td>
<td>1.173</td>
<td>-1.05</td>
<td>-0.675</td>
<td>-0.245</td>
</tr>
<tr>
<td>200</td>
<td>0.926</td>
<td>1.229</td>
<td>-1.40</td>
<td>-0.754</td>
<td>-0.245</td>
</tr>
<tr>
<td>300</td>
<td>0.964</td>
<td>1.277</td>
<td>-1.67</td>
<td>-0.805</td>
<td>-0.287</td>
</tr>
<tr>
<td>400</td>
<td>0.980</td>
<td>1.299</td>
<td>-1.75</td>
<td>-0.825</td>
<td>-0.292</td>
</tr>
<tr>
<td>500</td>
<td>0.978</td>
<td>1.297</td>
<td>-1.80</td>
<td>-0.825</td>
<td>-0.286</td>
</tr>
<tr>
<td>600</td>
<td>0.987</td>
<td>1.296</td>
<td>-1.93</td>
<td>-0.820</td>
<td>-0.276</td>
</tr>
<tr>
<td>700</td>
<td>0.996</td>
<td>1.309</td>
<td>-1.95</td>
<td>-0.831</td>
<td>-0.281</td>
</tr>
<tr>
<td>800</td>
<td>0.996</td>
<td>1.322</td>
<td>-1.99</td>
<td>-0.846</td>
<td>-0.288</td>
</tr>
<tr>
<td>900</td>
<td>0.998</td>
<td>1.314</td>
<td>-2.03</td>
<td>-0.840</td>
<td>-0.283</td>
</tr>
</tbody>
</table>

Table 4.1: $B = .5$; one series $\{u_k, y_k\}^{900+50}$ (no filter at input); L.S. estimator on the first 50 input and output samples. ($\mathbf{\theta}_0$)

The I.V. algorithm I has been used on 900 input-output pairs. The results agree with the results found by Wong (18). As can be seen, the estimate converges to the true parameter values, which indicates its consistency in this example.
4.4 Closed-loop process with only one forward parameter

![Diagram of closed-loop process](image)

**Fig. 4.3:** Simple closed-loop process

We take the same process already described in 4.3, and consider the case where the disturbed output ($y_k$) is fed back to the input to produce the error signal $f_k$ as the new control term e.g.:

$$f_k = \sum_{i=0}^{q} a_i x_{k-i}$$  \hspace{1cm} (4.11)

As: $f_k = u_k - y_k$, $y_k = x_k + n_k$,

we come to the following relation between $u_k$ and $y_k$:

$$u_k = y_k + \sum_{i=0}^{q} a_i y_{k-i} - \sum_{i=0}^{q} a_i n_{k-i}.$$ \hspace{1cm} (4.11a)

Even in the case where $n_k$ is a white noise signal ($\xi_k$), the resulting equivalent noise component in (4.11a) will be noise, colored with the backward parameters of the open-loop process. If $n_k$ is the output of a filter described by:

$$-\xi_k = a_0 n_k + a_1 n_{k-1} + \ldots + a_q n_{k-q}$$  \hspace{1cm} (4.11b)

where ($\xi_k$) is a white noise input, then the equivalent noise term in (4.11a) reduces to a white noise component $\xi_k$, e.g. eq. (4.11a) becomes:

$$u_k = y_k + \sum_{i=0}^{q} a_i y_{k-i} + \xi_k$$  \hspace{1cm} (4.12)

**Remark 2:** Even in this case, if we use algorithm 1, the matrix $Y$ and the vector $\xi$ are correlated.

Eq. (4.11a) shows that we have to estimate (using algorithm 1) the parameter vector:

$$\beta_c = (1, a_0, a_1, \ldots, a_q)$$  \hspace{1cm} (4.13)
instead of for the open-loop process of the preceding paragraph

\[ \mathbf{b}_o^T = (a_0, a_1, \ldots, a_q). \]

This means that the only correction necessary for the estimated (closed-loop) parameters is the subtraction of one from the first estimated parameter \( a_0 \), where:

\[ \mathbf{b}_c^T = (a_0, a_1, \ldots, a_q). \]

This is a charming property of algorithm I (and, as a matter of fact, the only good one).

The results obtained using algorithm I on the closed-loop D.E. given in eq. (4.11a) are given in tables 4.2a,b; 4.3 and 4.4. The input signal was a white noise sequence (no filter applied).

Table 4.2a gives the individual results for nine input-output sequences \( \{u_k, y_k\}_1^{1050} \). Each series had different starting values for the initialisation of the computer random generator. The noise sequence \( \{n_k\} \) was a white noise sequence \( \{\xi_k\} \).

<table>
<thead>
<tr>
<th>k</th>
<th>( a_0 = 1.000 )</th>
<th>( a_1 = 1.300 )</th>
<th>( a_2 = -0.220 )</th>
<th>( a_3 = -0.832 )</th>
<th>( a_4 = -0.269 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>.869</td>
<td>1.127</td>
<td>-.216</td>
<td>-.648</td>
<td>-.110</td>
</tr>
<tr>
<td></td>
<td>.763</td>
<td>1.282</td>
<td>+.094</td>
<td>-.491</td>
<td>-.203</td>
</tr>
<tr>
<td></td>
<td>.910</td>
<td>1.114</td>
<td>-.264</td>
<td>-.611</td>
<td>-.067</td>
</tr>
<tr>
<td></td>
<td>.675</td>
<td>.912</td>
<td>-.150</td>
<td>-.409</td>
<td>-.055</td>
</tr>
<tr>
<td></td>
<td>.789</td>
<td>1.079</td>
<td>-.062</td>
<td>-.513</td>
<td>-.174</td>
</tr>
<tr>
<td></td>
<td>.766</td>
<td>1.013</td>
<td>-.223</td>
<td>-.663</td>
<td>-.176</td>
</tr>
<tr>
<td></td>
<td>.669</td>
<td>1.055</td>
<td>-.094</td>
<td>-.518</td>
<td>-.175</td>
</tr>
<tr>
<td></td>
<td>.771</td>
<td>1.023</td>
<td>-.262</td>
<td>-.795</td>
<td>-.348</td>
</tr>
<tr>
<td></td>
<td>.771</td>
<td>.930</td>
<td>-.251</td>
<td>-.517</td>
<td>-.016</td>
</tr>
<tr>
<td>500</td>
<td>.915</td>
<td>1.109</td>
<td>-.308</td>
<td>-.721</td>
<td>-.121</td>
</tr>
<tr>
<td></td>
<td>.875</td>
<td>1.263</td>
<td>-.128</td>
<td>-.721</td>
<td>-.262</td>
</tr>
<tr>
<td></td>
<td>.912</td>
<td>1.159</td>
<td>-.220</td>
<td>-.639</td>
<td>-.131</td>
</tr>
<tr>
<td></td>
<td>.793</td>
<td>1.123</td>
<td>-.087</td>
<td>-.592</td>
<td>-.215</td>
</tr>
<tr>
<td></td>
<td>.906</td>
<td>1.198</td>
<td>-.090</td>
<td>-.587</td>
<td>-.165</td>
</tr>
<tr>
<td></td>
<td>.908</td>
<td>1.144</td>
<td>-.269</td>
<td>-.761</td>
<td>-.193</td>
</tr>
<tr>
<td></td>
<td>.737</td>
<td>1.089</td>
<td>-.125</td>
<td>-.574</td>
<td>-.178</td>
</tr>
<tr>
<td></td>
<td>.920</td>
<td>1.116</td>
<td>-.325</td>
<td>-.863</td>
<td>-.321</td>
</tr>
<tr>
<td></td>
<td>.833</td>
<td>1.100</td>
<td>-.204</td>
<td>-.620</td>
<td>-.110</td>
</tr>
<tr>
<td>1000</td>
<td>.948</td>
<td>1.223</td>
<td>-.244</td>
<td>-.789</td>
<td>-.205</td>
</tr>
<tr>
<td></td>
<td>.882</td>
<td>1.202</td>
<td>-.152</td>
<td>-.650</td>
<td>-.178</td>
</tr>
<tr>
<td></td>
<td>.930</td>
<td>1.176</td>
<td>-.265</td>
<td>-.745</td>
<td>-.177</td>
</tr>
<tr>
<td></td>
<td>.902</td>
<td>1.189</td>
<td>-.199</td>
<td>-.752</td>
<td>-.266</td>
</tr>
</tbody>
</table>
Table 4.2a: $B = .5$; nine series $\{u_k, y_k\}_{k=1}^{1000+50}$ (no filter at the input). L.S. estimator on the first 50 input and output samples ($\theta_o$).

Table 4.2b gives the means of the estimates and their standard deviations (9 series).

<table>
<thead>
<tr>
<th>k</th>
<th>$a_0 = 1.000$</th>
<th>$a_1 = 1.300$</th>
<th>$a_2 = -.220$</th>
<th>$a_3 = -.832$</th>
<th>$a_4 = -.269$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.778 ±.045</td>
<td>.790 ±.140</td>
<td>-.205 ±.121</td>
<td>-.429 ±.101</td>
<td>-.052 ±.147</td>
</tr>
<tr>
<td>200</td>
<td>.859 ±.046</td>
<td>.990 ±.120</td>
<td>-.186 ±.130</td>
<td>-.547 ±.107</td>
<td>-.120 ±.104</td>
</tr>
<tr>
<td>300</td>
<td>.901 ±.037</td>
<td>1.092 ±.091</td>
<td>-.161 ±.115</td>
<td>-.603 ±.111</td>
<td>-.157 ±.090</td>
</tr>
<tr>
<td>400</td>
<td>.926 ±.033</td>
<td>1.129 ±.068</td>
<td>-.180 ±.096</td>
<td>-.644 ±.107</td>
<td>-.174 ±.093</td>
</tr>
<tr>
<td>500</td>
<td>.934 ±.033</td>
<td>1.145 ±.056</td>
<td>-.195 ±.092</td>
<td>-.675 ±.098</td>
<td>-.188 ±.069</td>
</tr>
<tr>
<td>600</td>
<td>.939 ±.030</td>
<td>1.150 ±.062</td>
<td>-.208 ±.081</td>
<td>-.692 ±.083</td>
<td>-.192 ±.058</td>
</tr>
<tr>
<td>700</td>
<td>.950 ±.024</td>
<td>1.180 ±.049</td>
<td>-.210 ±.063</td>
<td>-.718 ±.077</td>
<td>-.205 ±.067</td>
</tr>
<tr>
<td>800</td>
<td>.956 ±.022</td>
<td>1.192 ±.042</td>
<td>-.209 ±.059</td>
<td>-.732 ±.070</td>
<td>-.212 ±.063</td>
</tr>
<tr>
<td>900</td>
<td>.957 ±.016</td>
<td>1.195 ±.037</td>
<td>-.212 ±.048</td>
<td>-.734 ±.067</td>
<td>-.208 ±.066</td>
</tr>
<tr>
<td>1000</td>
<td>.956 ±.017</td>
<td>1.194 ±.040</td>
<td>-.212 ±.051</td>
<td>-.736 ±.076</td>
<td>-.211 ±.067</td>
</tr>
</tbody>
</table>

Table 4.2b: The means (and calculated standard deviations) of the estimates obtained using algorithm 1.

Table 4.3 is obtained by taking 4 series of input-output sequences $\{u_k, y_k\}_{k=1}^{1050}$, and computing the mean of the estimates for this 4 series. For the remaining examples the input will be colored noise (input filter).

<table>
<thead>
<tr>
<th>k</th>
<th>$a_0 = 1.000$</th>
<th>$a_1 = 1.300$</th>
<th>$a_2 = -.220$</th>
<th>$a_3 = -.832$</th>
<th>$a_4 = -.269$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.720</td>
<td>1.042</td>
<td>-.111</td>
<td>-.548</td>
<td>-.121</td>
</tr>
<tr>
<td>200</td>
<td>.801</td>
<td>1.094</td>
<td>-.153</td>
<td>-.619</td>
<td>-.152</td>
</tr>
<tr>
<td>300</td>
<td>.861</td>
<td>1.156</td>
<td>-.158</td>
<td>-.681</td>
<td>-.205</td>
</tr>
<tr>
<td>400</td>
<td>.887</td>
<td>1.207</td>
<td>-.152</td>
<td>-.723</td>
<td>-.219</td>
</tr>
<tr>
<td>500</td>
<td>.890</td>
<td>1.216</td>
<td>-.148</td>
<td>-.719</td>
<td>-.225</td>
</tr>
<tr>
<td>600</td>
<td>.922</td>
<td>1.246</td>
<td>-.147</td>
<td>-.737</td>
<td>-.245</td>
</tr>
<tr>
<td>700</td>
<td>.942</td>
<td>1.266</td>
<td>-.160</td>
<td>-.765</td>
<td>-.258</td>
</tr>
<tr>
<td>800</td>
<td>.951</td>
<td>1.272</td>
<td>-.166</td>
<td>-.768</td>
<td>-.253</td>
</tr>
<tr>
<td>900</td>
<td>.938</td>
<td>1.265</td>
<td>-.162</td>
<td>-.760</td>
<td>-.253</td>
</tr>
<tr>
<td>1000</td>
<td>.941</td>
<td>1.265</td>
<td>-.168</td>
<td>-.762</td>
<td>-.247</td>
</tr>
</tbody>
</table>

Table 4.3: Closed-loop process, $B = .5$, filter at the input, white additive noise; L.S. initial estimate, the mean estimates over 4 series are given.
Graph 4.1 gives a plot of the results already given in table 4.2b. Comparing the results given in tables 4.2b and 4.3, it can be seen that there are no significant deviations in the mean of the estimates taken over 9 series and over 4 series. So, all results and simulations to be given in the remainder of this (and the following) paragraphs are based on results from 4 series and filtered input values. The mean over these 4 series will be given.

Table 4.4 shows the results over 4 series, when the noise power is increased to four times that of Table 4.3. Comparing this two tables it can be seen that higher noise levels weaken the convergence of the algorithm, although we can still observe an improvement of the estimate as \( k \) increases.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( a_0 = 1.000 )</th>
<th>( a_1 = 1.300 )</th>
<th>( a_2 = -0.220 )</th>
<th>( a_3 = -0.832 )</th>
<th>( a_4 = -0.269 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.454</td>
<td>0.777</td>
<td>-0.037</td>
<td>-0.266</td>
<td>-0.044</td>
</tr>
<tr>
<td>200</td>
<td>0.581</td>
<td>0.820</td>
<td>-0.075</td>
<td>-0.377</td>
<td>-0.009</td>
</tr>
<tr>
<td>300</td>
<td>0.693</td>
<td>0.967</td>
<td>-0.077</td>
<td>-0.502</td>
<td>-0.123</td>
</tr>
<tr>
<td>400</td>
<td>0.753</td>
<td>1.035</td>
<td>-0.102</td>
<td>-0.577</td>
<td>-0.119</td>
</tr>
<tr>
<td>500</td>
<td>0.761</td>
<td>1.072</td>
<td>-0.086</td>
<td>-0.565</td>
<td>-0.147</td>
</tr>
<tr>
<td>600</td>
<td>0.817</td>
<td>1.135</td>
<td>-0.084</td>
<td>-0.601</td>
<td>-0.184</td>
</tr>
<tr>
<td>700</td>
<td>0.845</td>
<td>1.151</td>
<td>-0.105</td>
<td>-0.635</td>
<td>-0.200</td>
</tr>
<tr>
<td>800</td>
<td>0.863</td>
<td>1.181</td>
<td>-0.113</td>
<td>-0.165</td>
<td>-0.192</td>
</tr>
<tr>
<td>900</td>
<td>0.843</td>
<td>1.159</td>
<td>-0.110</td>
<td>-0.635</td>
<td>-0.192</td>
</tr>
<tr>
<td>1000</td>
<td>0.853</td>
<td>1.172</td>
<td>-0.117</td>
<td>-0.642</td>
<td>-0.188</td>
</tr>
</tbody>
</table>

Table 4.4: Closed-loop process, \( B = 1 \); filter at input, white additive noise, L.S. initial estimate; 4 series; algorithm 1.

Table 4.5 shows the results over 4 series when using the L.S. estimator on all samples e.g.:

\[ \hat{B}_{L.S.} = \left( \bar{\Omega}^T(u, y) \bar{\Omega}(u, y) \right)^{-1} \bar{\Omega}^T(u, y) \bar{u} \]  

(4.14)

The input-output sequences used are the same as those giving the results of table 4.4.

Conclusion: It is clear from inspection of table 4.5 that the L.S. estimator is, in this (colored noise) situation inferior as compared to the I.V. estimator given by algorithm 1. Note however that the (strongly biased) estimates are quickly obtained using (4.14).
Table 4.5: Closed-loop process, $B = 1$, filter at input, L.S. estimator on 1050 input-output samples; algorithm 1.

4.5 The open-loop process with two forward parameters

The D.E. considered in sections 4.3 and 4.4 now has been extended with a term $b_1 u_{k-1}$ e.g.:

$$u_k = -b_1 u_{k-1} + \sum_{i=0}^q a_i y_{k-i} - \sum_{i=0}^q a_i n_{k-i}$$

$$= -b_1 u_{k-1} + \sum_{i=0}^q a_i y_{k-i} - e_k.$$  \hfill (4.15)

All results given in this paragraph have been obtained with:

1. 4 series input-output sequences;
2. $B = .5$;
3. $\sum_{i=0}^q a_i n_{k-i} = \xi_k$;
4. algorithm 1,

unless otherwise specified.

Conclusion: The results given in table 4.6 and table 4.7 indicate a (small) loss in efficiency of the I.V. algorithm, as a pole-zero cancellation occurs.
Table 4.6: \( b_1 = -0.800 \); pole-zero cancellation occurs; open-loop process; \( B = 0.5 \).

<table>
<thead>
<tr>
<th>k</th>
<th>( a_0 = 1.000 )</th>
<th>( a_1 = 1.300 )</th>
<th>( a_2 = -0.220 )</th>
<th>( a_3 = -0.832 )</th>
<th>( a_4 = -0.269 )</th>
<th>( b_1 = -0.800 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>( 0.934 )</td>
<td>( 1.176 )</td>
<td>( -0.309 )</td>
<td>( -0.875 )</td>
<td>( -0.282 )</td>
<td>( -0.837 )</td>
</tr>
<tr>
<td>200</td>
<td>( 0.960 )</td>
<td>( 1.183 )</td>
<td>( -0.354 )</td>
<td>( -0.899 )</td>
<td>( -0.277 )</td>
<td>( -0.844 )</td>
</tr>
<tr>
<td>300</td>
<td>( 0.966 )</td>
<td>( 1.178 )</td>
<td>( -0.377 )</td>
<td>( -0.921 )</td>
<td>( -0.284 )</td>
<td>( -0.866 )</td>
</tr>
<tr>
<td>400</td>
<td>( 0.971 )</td>
<td>( 1.184 )</td>
<td>( -0.367 )</td>
<td>( -0.905 )</td>
<td>( -0.277 )</td>
<td>( -0.860 )</td>
</tr>
<tr>
<td>500</td>
<td>( 0.973 )</td>
<td>( 1.193 )</td>
<td>( -0.362 )</td>
<td>( -0.904 )</td>
<td>( -0.276 )</td>
<td>( -0.860 )</td>
</tr>
<tr>
<td>600</td>
<td>( 0.973 )</td>
<td>( 1.170 )</td>
<td>( -0.401 )</td>
<td>( -0.922 )</td>
<td>( -0.277 )</td>
<td>( -0.882 )</td>
</tr>
<tr>
<td>700</td>
<td>( 0.977 )</td>
<td>( 1.188 )</td>
<td>( -0.383 )</td>
<td>( -0.919 )</td>
<td>( -0.279 )</td>
<td>( -0.875 )</td>
</tr>
<tr>
<td>800</td>
<td>( 0.978 )</td>
<td>( 1.190 )</td>
<td>( -0.391 )</td>
<td>( -0.935 )</td>
<td>( -0.287 )</td>
<td>( -0.882 )</td>
</tr>
<tr>
<td>900</td>
<td>( 0.981 )</td>
<td>( 1.194 )</td>
<td>( -0.394 )</td>
<td>( -0.945 )</td>
<td>( -0.290 )</td>
<td>( -0.886 )</td>
</tr>
<tr>
<td>1000</td>
<td>( 0.980 )</td>
<td>( 1.181 )</td>
<td>( -0.416 )</td>
<td>( -0.956 )</td>
<td>( -0.294 )</td>
<td>( -0.897 )</td>
</tr>
</tbody>
</table>

Table 4.7: \( b_1 = 0.800 \); pole-zero cancellation occurs; open-loop process; \( B = 0.5 \).

<table>
<thead>
<tr>
<th>k</th>
<th>( a_0 = 1.000 )</th>
<th>( a_1 = 1.300 )</th>
<th>( a_2 = -0.220 )</th>
<th>( a_3 = -0.832 )</th>
<th>( a_4 = -0.269 )</th>
<th>( b_1 = 0.800 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>( 0.946 )</td>
<td>( 1.181 )</td>
<td>( -0.234 )</td>
<td>( -0.768 )</td>
<td>( -0.228 )</td>
<td>( 0.689 )</td>
</tr>
<tr>
<td>200</td>
<td>( 0.973 )</td>
<td>( 1.208 )</td>
<td>( -0.247 )</td>
<td>( -0.785 )</td>
<td>( -0.236 )</td>
<td>( 0.718 )</td>
</tr>
<tr>
<td>300</td>
<td>( 0.968 )</td>
<td>( 1.198 )</td>
<td>( -0.249 )</td>
<td>( -0.775 )</td>
<td>( -0.230 )</td>
<td>( 0.705 )</td>
</tr>
<tr>
<td>400</td>
<td>( 0.972 )</td>
<td>( 1.193 )</td>
<td>( -0.242 )</td>
<td>( -0.769 )</td>
<td>( -0.234 )</td>
<td>( 0.705 )</td>
</tr>
<tr>
<td>500</td>
<td>( 0.972 )</td>
<td>( 1.178 )</td>
<td>( -0.256 )</td>
<td>( -0.759 )</td>
<td>( -0.226 )</td>
<td>( 0.682 )</td>
</tr>
<tr>
<td>600</td>
<td>( 0.975 )</td>
<td>( 1.174 )</td>
<td>( -0.255 )</td>
<td>( -0.753 )</td>
<td>( -0.225 )</td>
<td>( 0.679 )</td>
</tr>
<tr>
<td>700</td>
<td>( 0.977 )</td>
<td>( 1.178 )</td>
<td>( -0.261 )</td>
<td>( -0.757 )</td>
<td>( -0.220 )</td>
<td>( 0.677 )</td>
</tr>
<tr>
<td>800</td>
<td>( 0.976 )</td>
<td>( 1.161 )</td>
<td>( -0.281 )</td>
<td>( -0.742 )</td>
<td>( -0.210 )</td>
<td>( 0.653 )</td>
</tr>
<tr>
<td>900</td>
<td>( 0.977 )</td>
<td>( 1.167 )</td>
<td>( -0.283 )</td>
<td>( -0.747 )</td>
<td>( -0.212 )</td>
<td>( 0.654 )</td>
</tr>
<tr>
<td>980</td>
<td>( 0.976 )</td>
<td>( 1.165 )</td>
<td>( -0.279 )</td>
<td>( -0.746 )</td>
<td>( -0.213 )</td>
<td>( 0.654 )</td>
</tr>
</tbody>
</table>
### Table 4.8

<table>
<thead>
<tr>
<th>$k$</th>
<th>$a_0 = 1.000$</th>
<th>$a_1 = 1.300$</th>
<th>$a_2 = -0.220$</th>
<th>$a_3 = -0.832$</th>
<th>$a_4 = -0.269$</th>
<th>$b_1 = 0.300$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.942</td>
<td>1.168</td>
<td>-.254</td>
<td>-.762</td>
<td>-.224</td>
<td>.204</td>
</tr>
<tr>
<td>200</td>
<td>.970</td>
<td>1.182</td>
<td>-.291</td>
<td>-.776</td>
<td>-.211</td>
<td>.208</td>
</tr>
<tr>
<td>300</td>
<td>.968</td>
<td>1.181</td>
<td>-.292</td>
<td>-.770</td>
<td>-.205</td>
<td>.205</td>
</tr>
<tr>
<td>400</td>
<td>.969</td>
<td>1.175</td>
<td>-.291</td>
<td>-.759</td>
<td>-.205</td>
<td>.201</td>
</tr>
<tr>
<td>500</td>
<td>.970</td>
<td>1.185</td>
<td>-.289</td>
<td>-.765</td>
<td>-.207</td>
<td>.202</td>
</tr>
<tr>
<td>600</td>
<td>.972</td>
<td>1.185</td>
<td>-.289</td>
<td>-.757</td>
<td>-.205</td>
<td>.202</td>
</tr>
<tr>
<td>700</td>
<td>.976</td>
<td>1.188</td>
<td>-.293</td>
<td>-.761</td>
<td>-.203</td>
<td>.199</td>
</tr>
<tr>
<td>800</td>
<td>.978</td>
<td>1.215</td>
<td>-.274</td>
<td>-.777</td>
<td>-.215</td>
<td>.222</td>
</tr>
<tr>
<td>900</td>
<td>.980</td>
<td>1.223</td>
<td>-.273</td>
<td>-.785</td>
<td>-.220</td>
<td>.225</td>
</tr>
<tr>
<td>980</td>
<td>.979</td>
<td>1.224</td>
<td>-.267</td>
<td>-.783</td>
<td>-.223</td>
<td>.227</td>
</tr>
</tbody>
</table>

Table 4.8: $b_1 = 0.300$; no pole-zero cancellation; open-loop process; $B = 0.5$.

### Table 4.9

<table>
<thead>
<tr>
<th>$k$</th>
<th>$a_0 = 1.000$</th>
<th>$a_1 = 1.300$</th>
<th>$a_2 = -0.220$</th>
<th>$a_3 = -0.832$</th>
<th>$a_4 = -0.269$</th>
<th>$b_1 = 0.300$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.803</td>
<td>1.014</td>
<td>-.209</td>
<td>-.660</td>
<td>-.200</td>
<td>.094</td>
</tr>
<tr>
<td>200</td>
<td>.807</td>
<td>1.007</td>
<td>-.224</td>
<td>-.655</td>
<td>-.193</td>
<td>.091</td>
</tr>
<tr>
<td>300</td>
<td>.799</td>
<td>1.012</td>
<td>-.205</td>
<td>-.653</td>
<td>-.196</td>
<td>.098</td>
</tr>
<tr>
<td>400</td>
<td>.796</td>
<td>1.017</td>
<td>-.195</td>
<td>-.653</td>
<td>-.199</td>
<td>.104</td>
</tr>
<tr>
<td>500</td>
<td>.797</td>
<td>1.011</td>
<td>-.206</td>
<td>-.652</td>
<td>-.192</td>
<td>.091</td>
</tr>
<tr>
<td>600</td>
<td>.797</td>
<td>1.014</td>
<td>-.200</td>
<td>-.646</td>
<td>-.193</td>
<td>.096</td>
</tr>
<tr>
<td>700</td>
<td>.796</td>
<td>1.018</td>
<td>-.199</td>
<td>-.649</td>
<td>-.193</td>
<td>.097</td>
</tr>
<tr>
<td>800</td>
<td>.795</td>
<td>1.026</td>
<td>-.190</td>
<td>-.655</td>
<td>-.197</td>
<td>.103</td>
</tr>
<tr>
<td>900</td>
<td>.796</td>
<td>1.026</td>
<td>-.190</td>
<td>-.658</td>
<td>-.201</td>
<td>.100</td>
</tr>
<tr>
<td>980</td>
<td>.795</td>
<td>1.026</td>
<td>-.187</td>
<td>-.657</td>
<td>-.202</td>
<td>.102</td>
</tr>
</tbody>
</table>

Table 4.9: $b_1 = 0.300$; no pole-zero cancellation; open-loop process; $B = 0.5$; least-squares estimation on all input-output samples.
<table>
<thead>
<tr>
<th>k</th>
<th>$a_0=1.000$</th>
<th>$a_1=1.300$</th>
<th>$a_2=-.220$</th>
<th>$a_3=-.832$</th>
<th>$a_4=-.269$</th>
<th>$b_1=-.300$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.939</td>
<td>1.192</td>
<td>-.248</td>
<td>-.793</td>
<td>-.249</td>
<td>-.340</td>
</tr>
<tr>
<td>200</td>
<td>.968</td>
<td>1.214</td>
<td>-.281</td>
<td>-.820</td>
<td>-.244</td>
<td>-.347</td>
</tr>
<tr>
<td>300</td>
<td>.966</td>
<td>1.225</td>
<td>-.259</td>
<td>-.811</td>
<td>-.245</td>
<td>-.332</td>
</tr>
<tr>
<td>400</td>
<td>.967</td>
<td>1.219</td>
<td>-.264</td>
<td>-.801</td>
<td>-.239</td>
<td>-.336</td>
</tr>
<tr>
<td>500</td>
<td>.969</td>
<td>1.227</td>
<td>-.259</td>
<td>-.805</td>
<td>-.240</td>
<td>-.335</td>
</tr>
<tr>
<td>600</td>
<td>.972</td>
<td>1.237</td>
<td>-.245</td>
<td>-.795</td>
<td>-.239</td>
<td>-.328</td>
</tr>
<tr>
<td>700</td>
<td>.976</td>
<td>1.251</td>
<td>-.236</td>
<td>-.803</td>
<td>-.246</td>
<td>-.323</td>
</tr>
<tr>
<td>800</td>
<td>.978</td>
<td>1.271</td>
<td>-.218</td>
<td>-.810</td>
<td>-.255</td>
<td>-.313</td>
</tr>
<tr>
<td>900</td>
<td>.980</td>
<td>1.273</td>
<td>-.224</td>
<td>-.818</td>
<td>-.257</td>
<td>-.315</td>
</tr>
<tr>
<td>980</td>
<td>.979</td>
<td>1.271</td>
<td>-.220</td>
<td>-.813</td>
<td>-.257</td>
<td>-.315</td>
</tr>
</tbody>
</table>

Table 4.10: $b_1=-.300$; open-loop process; no pole-zero cancellation; $B=.5$.

<table>
<thead>
<tr>
<th>k</th>
<th>$a_0=1.000$</th>
<th>$a_1=1.300$</th>
<th>$a_2=-.220$</th>
<th>$a_3=-.832$</th>
<th>$a_4=-.269$</th>
<th>$b_1=-.300$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.870</td>
<td>.855</td>
<td>-.520</td>
<td>-.623</td>
<td>-.063</td>
<td>-.546</td>
</tr>
<tr>
<td>200</td>
<td>.915</td>
<td>.980</td>
<td>-.464</td>
<td>-.709</td>
<td>-.128</td>
<td>-.473</td>
</tr>
<tr>
<td>300</td>
<td>.946</td>
<td>1.090</td>
<td>-.384</td>
<td>-.751</td>
<td>-.174</td>
<td>-.408</td>
</tr>
<tr>
<td>400</td>
<td>.951</td>
<td>1.084</td>
<td>-.399</td>
<td>-.755</td>
<td>-.173</td>
<td>-.424</td>
</tr>
<tr>
<td>500</td>
<td>.956</td>
<td>1.110</td>
<td>-.373</td>
<td>-.761</td>
<td>-.185</td>
<td>-.415</td>
</tr>
<tr>
<td>600</td>
<td>.961</td>
<td>1.129</td>
<td>-.358</td>
<td>-.767</td>
<td>-.193</td>
<td>-.404</td>
</tr>
<tr>
<td>700</td>
<td>.967</td>
<td>1.164</td>
<td>-.329</td>
<td>-.784</td>
<td>-.211</td>
<td>-.381</td>
</tr>
<tr>
<td>800</td>
<td>.978</td>
<td>1.188</td>
<td>-.323</td>
<td>-.800</td>
<td>-.221</td>
<td>-.369</td>
</tr>
<tr>
<td>890</td>
<td>.981</td>
<td>1.209</td>
<td>-.301</td>
<td>-.805</td>
<td>-.231</td>
<td>-.354</td>
</tr>
</tbody>
</table>

Table 4.11: $b_1=-.300$; open-loop process; no pole-zero cancellation; $B=.5$; white noise disturbance.
However, the estimates are still reasonably good.

Tables 4.8 and 4.9 compare the results when no pole-zero cancellation occurs, and when respectively algorithm I and the L.S. estimator given in eq. (4.14) are applied. It is seen that the I.V. algorithm is superior over the L.S. algorithm.

Table 4.10 gives the results when $b_1 = -0.300$. The results (using algorithm I) are better than those given in table 4.8.

Table 4.11 gives an indication of the influence of the noise parameters on the estimates. It is seen (table 4.11) that white noise disturbing the process output ($n_k = \xi_k$) gives results slightly inferior compared with the case where $e_k = \xi_k$ (table 4.10).

This is mainly due to the fact that in the latter case the initial estimate (on 60 input-output pairs in this case) gives better results. This better initial model parameter vector $\hat{e}_0^M$ has a great influence on the convergence of the I.V. algorithm.

The results given in table 4.12 can be explained in an analogous way. The noise power has been increased to four times that in table 4.11, and the L.S. (initial) estimate gives poorer results to start from.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$a_0 = 1.000$</th>
<th>$a_1 = 1.300$</th>
<th>$a_2 = -0.220$</th>
<th>$a_3 = -0.832$</th>
<th>$a_4 = -0.269$</th>
<th>$b_1 = -0.300$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.665 ± 0.054</td>
<td>0.638 ± 0.128</td>
<td>-0.306 ± 0.169</td>
<td>-0.314 ± 0.120</td>
<td>0.009 ± 0.084</td>
<td>-0.525 ± 0.114</td>
</tr>
<tr>
<td>200</td>
<td>0.774 ± 0.047</td>
<td>0.748 ± 0.151</td>
<td>-0.337 ± 0.194</td>
<td>-0.411 ± 0.115</td>
<td>-0.024 ± 0.087</td>
<td>-0.507 ± 0.143</td>
</tr>
<tr>
<td>300</td>
<td>0.793 ± 0.051</td>
<td>0.823 ± 0.152</td>
<td>-0.356 ± 0.093</td>
<td>-0.487 ± 0.115</td>
<td>-0.053 ± 0.089</td>
<td>-0.498 ± 0.141</td>
</tr>
<tr>
<td>400</td>
<td>0.820 ± 0.052</td>
<td>0.860 ± 0.161</td>
<td>-0.373 ± 0.185</td>
<td>-0.531 ± 0.104</td>
<td>-0.070 ± 0.093</td>
<td>-0.501 ± 0.136</td>
</tr>
<tr>
<td>500</td>
<td>0.842 ± 0.049</td>
<td>0.898 ± 0.159</td>
<td>-0.377 ± 0.189</td>
<td>-0.567 ± 0.087</td>
<td>-0.086 ± 0.090</td>
<td>-0.494 ± 0.138</td>
</tr>
<tr>
<td>600</td>
<td>0.856 ± 0.045</td>
<td>0.919 ± 0.153</td>
<td>-0.382 ± 0.186</td>
<td>-0.589 ± 0.085</td>
<td>-0.096 ± 0.090</td>
<td>-0.490 ± 0.132</td>
</tr>
<tr>
<td>700</td>
<td>0.868 ± 0.042</td>
<td>0.942 ± 0.153</td>
<td>-0.379 ± 0.176</td>
<td>-0.607 ± 0.078</td>
<td>-0.105 ± 0.089</td>
<td>-0.479 ± 0.130</td>
</tr>
<tr>
<td>800</td>
<td>0.878 ± 0.043</td>
<td>0.959 ± 0.149</td>
<td>-0.380 ± 0.176</td>
<td>-0.622 ± 0.079</td>
<td>-0.112 ± 0.086</td>
<td>-0.473 ± 0.123</td>
</tr>
<tr>
<td>900</td>
<td>0.887 ± 0.044</td>
<td>0.979 ± 0.149</td>
<td>-0.376 ± 0.173</td>
<td>-0.636 ± 0.078</td>
<td>-0.119 ± 0.082</td>
<td>-0.463 ± 0.123</td>
</tr>
<tr>
<td>1000</td>
<td>0.896 ± 0.043</td>
<td>0.991 ± 0.146</td>
<td>-0.380 ± 0.164</td>
<td>-0.650 ± 0.076</td>
<td>-0.125 ± 0.081</td>
<td>-0.464 ± 0.118</td>
</tr>
</tbody>
</table>

Table 4.12: $b_1 = -0.300$, open-loop process; $B = 1$; white additive noise; 50 series; algorithm I.
4.6 The process description introduced by Åström.

In the literature a process is often described by a D.E. of the following type:

\[ q \sum_0^q a_i y_{k-i} = \sum_0^p b_i u_{k-i} + \sum_0^s c_i e_{k-i}, \quad (a_0 = c_0 = 1) \]  \hspace{1cm} (4.16)

If \( q = p = s, \) (and \( b_0 = 0 \)) then eq. (4.16) gives the D.E. introduced by Åström, Bohlin & Wensmark (9).

The open-loop process description given in fig. 4.5a as well as the closed-loop process description given in fig. 4.5b can be put in the form of eq. (4.16).

![Fig. 4.5a: Open-loop process](image)

![Fig. 4.5b: Closed-loop process](image)

1) Open-loop process

It is seen from fig. 4.5a that the following relation holds:

\[ q \sum_0^q a_i x_{k-i} = \sum_0^p b_i u_{k-i} \]

and \( y_k = n_k + x_k \)

so:

\[ q \sum_0^q a_i y_{k-i} = \sum_0^p b_i u_{k-i} + \sum_0^q a_i n_{k-i}. \]

In z-notation:

\[ (1+A(z^{-1}))y_k = (b_o + B(z^{-1}))u_k + (1+A(z^{-1}))n_k \]  \hspace{1cm} (4.17)
As:
\[ n_k = \frac{1+F}{1+G} \xi_k \]
(4.17) can be rewritten:
\[ (1+A(-1))y_k = (b_o + B(z^{-1}))u_k + (1+C(z^{-1})) \xi_k, \]
with:
\[ 1+C(z^{-1}) = \frac{(1+A(z^{-1}))1+F(z^{-1})}{1+G(z^{-1})} \]
(4.18)

2) Closed-loop process

It is seen from fig. 4.5b that the following relation holds:
\[ (1+A)(y_k - n_k) = (b_o + B)(u_k - y_k) \]
or
\[ ((1+b_o)+A+B)y_k = (b_o + B)u_k + (1+C)\xi_k \]
(4.19)

We can normalize eq. (4.1a) with respect to \( y_k \) or:
\[ (1+A^*)y_k = (b_o * + B^*)u_k + \lambda(1+C)\xi_k \]
where
\[ A^* = A+B \]
\[ b_o^* = \frac{b_o}{1+b_o} \]
\[ B^* = \frac{B}{1+b_o} \]
and \( \lambda = \frac{1}{1+b_o} \).

An open-loop or a closed-loop process can thus be described by a D.E. of the type (4.16).

To compare the results obtained with the I.V. method, with those obtained by other authors (e.g. Åström (10), Peterka & Halouskova (20), Gustavsson (21), Hastings-James & Sage (15), Clarke (13)) we will describe (and simulate) our process in this and the following paragraphs by the D.E. (4.16). All estimations will be done with 10 or 50 series of input-output sequences \( \{u_k, y_k\}^{1000+10m} \), where \( m \) equals the total number of parameters to be estimated. The first 10m input-output samples will again be used to obtain a L.S. solution (initial model parameter vector \( b_o \)).

Conclusion:

1) The results given in tables 4.13, 4.14 and 4.15 clearly indicate the need for a "good" initial model parameter vector. The conclusion of the previous paragraphs concerning the convergence as a function of this initial estimate holds here also.

Graph 4.2 gives a quick impression of the results shown in those tables.
Tables 4.16 and 4.17 show that if the noiseterm \((1+C(z^{-1}))e_k\) has \(c_1 \neq 0\) (i.e., 0), the estimates are not as good as when \(c_1 = 0\) (i.e., 0) (cfr. Åström (10)). However, the I.V. algorithm still converges to the true parameters.

2) The results given in tables 4.18, 4.19 and 4.20 are obtained using algorithm 2, which proves to be superior over algorithm 1 in this case. A comparison of the results in tables 4.18 (alg. 2) and 4.15 (alg. 1) clearly shows that the convergence with algorithm 2 is more pronounced than with algorithm 1. Again, this is due to the fact that the L.S. initial estimate using algorithm 2 is better than the one using algorithm 1.

<table>
<thead>
<tr>
<th>(k)</th>
<th>(a_0 = 1.000)</th>
<th>(a_1 = -1.500)</th>
<th>(a_2 = 0.700)</th>
<th>(b_2 = 0.500)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.956 ±0.028</td>
<td>-1.432 ±0.037</td>
<td>0.664 ±0.019</td>
<td>0.459 ±0.060</td>
</tr>
<tr>
<td>200</td>
<td>0.976 ±0.033</td>
<td>-1.460 ±0.048</td>
<td>0.678 ±0.024</td>
<td>0.481 ±0.047</td>
</tr>
<tr>
<td>300</td>
<td>0.979 ±0.019</td>
<td>-1.465 ±0.026</td>
<td>0.681 ±0.014</td>
<td>0.487 ±0.034</td>
</tr>
<tr>
<td>400</td>
<td>0.981 ±0.026</td>
<td>-1.465 ±0.036</td>
<td>0.680 ±0.016</td>
<td>0.486 ±0.027</td>
</tr>
<tr>
<td>500</td>
<td>0.985 ±0.022</td>
<td>-1.473 ±0.034</td>
<td>0.684 ±0.017</td>
<td>0.486 ±0.025</td>
</tr>
<tr>
<td>600</td>
<td>0.988 ±0.019</td>
<td>-1.478 ±0.030</td>
<td>0.687 ±0.016</td>
<td>0.489 ±0.023</td>
</tr>
<tr>
<td>700</td>
<td>0.990 ±0.017</td>
<td>-1.484 ±0.025</td>
<td>0.691 ±0.013</td>
<td>0.488 ±0.019</td>
</tr>
<tr>
<td>800</td>
<td>0.996 ±0.020</td>
<td>-1.491 ±0.028</td>
<td>0.694 ±0.013</td>
<td>0.495 ±0.025</td>
</tr>
<tr>
<td>900</td>
<td>0.995 ±0.017</td>
<td>-1.491 ±0.025</td>
<td>0.695 ±0.012</td>
<td>0.494 ±0.023</td>
</tr>
<tr>
<td>1000</td>
<td>0.997 ±0.016</td>
<td>-1.494 ±0.024</td>
<td>0.696 ±0.012</td>
<td>0.497 ±0.020</td>
</tr>
</tbody>
</table>

Table 4.13: \(B = 0.5, c_0 = 1, c_1 = c_2 = 0, 10\) series, algorithm 1.

<table>
<thead>
<tr>
<th>(k)</th>
<th>(a_0 = 1.000)</th>
<th>(a_1 = -1.500)</th>
<th>(a_2 = 0.700)</th>
<th>(b_2 = 0.500)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.841 ±0.051</td>
<td>-1.260 ±0.070</td>
<td>0.586 ±0.036</td>
<td>0.325 ±0.110</td>
</tr>
<tr>
<td>200</td>
<td>0.898 ±0.048</td>
<td>-1.350 ±0.073</td>
<td>0.630 ±0.036</td>
<td>0.376 ±0.084</td>
</tr>
<tr>
<td>300</td>
<td>0.921 ±0.048</td>
<td>-1.383 ±0.065</td>
<td>0.646 ±0.033</td>
<td>0.404 ±0.070</td>
</tr>
<tr>
<td>400</td>
<td>0.939 ±0.041</td>
<td>-1.409 ±0.062</td>
<td>0.657 ±0.031</td>
<td>0.427 ±0.071</td>
</tr>
<tr>
<td>500</td>
<td>0.950 ±0.038</td>
<td>-1.424 ±0.058</td>
<td>0.664 ±0.029</td>
<td>0.439 ±0.062</td>
</tr>
<tr>
<td>600</td>
<td>0.957 ±0.039</td>
<td>-1.436 ±0.061</td>
<td>0.670 ±0.031</td>
<td>0.445 ±0.059</td>
</tr>
<tr>
<td>700</td>
<td>0.961 ±0.035</td>
<td>-1.442 ±0.054</td>
<td>0.673 ±0.027</td>
<td>0.447 ±0.058</td>
</tr>
<tr>
<td>800</td>
<td>0.967 ±0.035</td>
<td>-1.451 ±0.053</td>
<td>0.677 ±0.027</td>
<td>0.458 ±0.058</td>
</tr>
<tr>
<td>900</td>
<td>0.971 ±0.032</td>
<td>-1.458 ±0.049</td>
<td>0.680 ±0.024</td>
<td>0.465 ±0.053</td>
</tr>
<tr>
<td>1000</td>
<td>0.976 ±0.029</td>
<td>-1.464 ±0.044</td>
<td>0.683 ±0.022</td>
<td>0.471 ±0.048</td>
</tr>
</tbody>
</table>

Table 4.14: \(B = 1, c_0 = 1, c_1 = c_2 = 0, 50\) series, algorithm 1.
Table 4.15: $B = 2$, $c_0 = 1$, $c_1 = c_2 = 0$, 50 series, algorithm 1.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$a_0 = 1.000$</th>
<th>$a_1 = -1.500$</th>
<th>$a_2 = .700$</th>
<th>$+b_2 = .500$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.664 ±0.076</td>
<td>- .978 ±0.094</td>
<td>.446 ±0.055</td>
<td>+.155 ±1.173</td>
</tr>
<tr>
<td>200</td>
<td>.754 ±0.070</td>
<td>-1.127 ±0.103</td>
<td>.522 ±0.053</td>
<td>+.227 ±1.128</td>
</tr>
<tr>
<td>300</td>
<td>.797 ±0.063</td>
<td>-1.192 ±0.095</td>
<td>.555 ±0.049</td>
<td>+.272 ±1.077</td>
</tr>
<tr>
<td>400</td>
<td>.834 ±0.065</td>
<td>-1.246 ±0.097</td>
<td>.578 ±0.048</td>
<td>+.316 ±1.115</td>
</tr>
<tr>
<td>500</td>
<td>.857 ±0.062</td>
<td>-1.282 ±0.094</td>
<td>.595 ±0.047</td>
<td>+.339 ±1.102</td>
</tr>
<tr>
<td>600</td>
<td>.874 ±0.065</td>
<td>-1.308 ±0.101</td>
<td>.608 ±0.051</td>
<td>+.356 ±1.102</td>
</tr>
<tr>
<td>700</td>
<td>.885 ±0.061</td>
<td>-1.327 ±0.092</td>
<td>.617 ±0.046</td>
<td>+.362 ±1.101</td>
</tr>
<tr>
<td>800</td>
<td>.899 ±0.062</td>
<td>-1.347 ±0.092</td>
<td>.626 ±0.046</td>
<td>+.384 ±1.103</td>
</tr>
<tr>
<td>900</td>
<td>.910 ±0.058</td>
<td>-1.362 ±0.086</td>
<td>.634 ±0.043</td>
<td>+.399 ±1.095</td>
</tr>
<tr>
<td>1000</td>
<td>.919 ±0.053</td>
<td>-1.376 ±0.076</td>
<td>.641 ±0.039</td>
<td>+.412 ±1.089</td>
</tr>
</tbody>
</table>

Table 4.16: $B = 2$, $c_0 = 1$, $c_1 = -1$, $c_2 = .2$, 48 series, algorithm 1.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$a_0 = 1.000$</th>
<th>$a_1 = -1.500$</th>
<th>$a_2 = .700$</th>
<th>$+b_2 = .500$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.560</td>
<td>- .717</td>
<td>.280</td>
<td>+.247</td>
</tr>
<tr>
<td>200</td>
<td>.650</td>
<td>- .887</td>
<td>.376</td>
<td>+.278</td>
</tr>
<tr>
<td>300</td>
<td>.704</td>
<td>- .984</td>
<td>.432</td>
<td>+.281</td>
</tr>
<tr>
<td>400</td>
<td>.738</td>
<td>-1.052</td>
<td>.468</td>
<td>+.306</td>
</tr>
<tr>
<td>500</td>
<td>.767</td>
<td>-1.107</td>
<td>.497</td>
<td>+.328</td>
</tr>
<tr>
<td>600</td>
<td>.790</td>
<td>-1.146</td>
<td>.518</td>
<td>+.338</td>
</tr>
<tr>
<td>700</td>
<td>.804</td>
<td>-1.175</td>
<td>.535</td>
<td>+.341</td>
</tr>
<tr>
<td>800</td>
<td>.824</td>
<td>-1.205</td>
<td>.550</td>
<td>+.366</td>
</tr>
<tr>
<td>900</td>
<td>.833</td>
<td>-1.220</td>
<td>.558</td>
<td>+.366</td>
</tr>
<tr>
<td>1000</td>
<td>.865</td>
<td>-1.237</td>
<td>.566</td>
<td>+.381</td>
</tr>
<tr>
<td></td>
<td>$a_a = 1.000$</td>
<td>$a_1 = -1.500$</td>
<td>$a_2 = .700$</td>
<td>$b_2 = .500$</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>100</td>
<td>.660 ±066</td>
<td>- .914 ±103</td>
<td>.393 ±059</td>
<td>.263 ±166</td>
</tr>
<tr>
<td>200</td>
<td>.753 ±072</td>
<td>-1.078 ±111</td>
<td>.481 ±058</td>
<td>.326 ±144</td>
</tr>
<tr>
<td>300</td>
<td>.800 ±068</td>
<td>-1.166 ±104</td>
<td>.529 ±052</td>
<td>.341 ±119</td>
</tr>
<tr>
<td>400</td>
<td>.832 ±068</td>
<td>-1.221 ±102</td>
<td>.558 ±050</td>
<td>.367 ±110</td>
</tr>
<tr>
<td>500</td>
<td>.859 ±064</td>
<td>-1.266 ±096</td>
<td>.581 ±047</td>
<td>.390 ±109</td>
</tr>
<tr>
<td>600</td>
<td>.876 ±067</td>
<td>-1.295 ±096</td>
<td>.596 ±046</td>
<td>.402 ±121</td>
</tr>
<tr>
<td>700</td>
<td>.887 ±062</td>
<td>-1.314 ±090</td>
<td>.606 ±043</td>
<td>.406 ±110</td>
</tr>
<tr>
<td>800</td>
<td>.901 ±064</td>
<td>-1.336 ±092</td>
<td>.617 ±043</td>
<td>.423 ±115</td>
</tr>
<tr>
<td>900</td>
<td>.907 ±060</td>
<td>-1.345 ±087</td>
<td>.622 ±041</td>
<td>.427 ±104</td>
</tr>
<tr>
<td>1000</td>
<td>.914 ±057</td>
<td>-1.357 ±083</td>
<td>.628 ±039</td>
<td>.435 ±094</td>
</tr>
</tbody>
</table>

Table 4.17: $B = 1$, $c_o = 1$, $c_1 = -1.5$, $c_2 = .7$, 50 series, algorithm 1.

<table>
<thead>
<tr>
<th></th>
<th>$b_1 = 1.000$</th>
<th>$b_2 = .500$</th>
<th>$a_1 = -1.500$</th>
<th>$a_2 = .700$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.967 ±120</td>
<td>.529 ±238</td>
<td>-1.499 ±062</td>
<td>.690 ±051</td>
</tr>
<tr>
<td>200</td>
<td>.969 ±141</td>
<td>.543 ±182</td>
<td>-1.494 ±044</td>
<td>.689 ±036</td>
</tr>
<tr>
<td>300</td>
<td>.996 ±073</td>
<td>.540 ±137</td>
<td>-1.491 ±036</td>
<td>.690 ±029</td>
</tr>
<tr>
<td>400</td>
<td>1.011 ±098</td>
<td>.516 ±090</td>
<td>-1.481 ±033</td>
<td>.679 ±030</td>
</tr>
<tr>
<td>500</td>
<td>1.006 ±085</td>
<td>.502 ±081</td>
<td>-1.487 ±028</td>
<td>.683 ±026</td>
</tr>
<tr>
<td>600</td>
<td>.999 ±075</td>
<td>.504 ±076</td>
<td>-1.488 ±032</td>
<td>.686 ±030</td>
</tr>
<tr>
<td>700</td>
<td>.998 ±065</td>
<td>.495 ±064</td>
<td>-1.494 ±028</td>
<td>.692 ±027</td>
</tr>
<tr>
<td>800</td>
<td>.982 ±084</td>
<td>.510 ±077</td>
<td>-1.493 ±029</td>
<td>.690 ±028</td>
</tr>
<tr>
<td>900</td>
<td>.988 ±071</td>
<td>.507 ±073</td>
<td>-1.495 ±032</td>
<td>.693 ±029</td>
</tr>
<tr>
<td>1000</td>
<td>.983 ±065</td>
<td>.513 ±062</td>
<td>-1.495 ±028</td>
<td>.694 ±025</td>
</tr>
</tbody>
</table>

Table 4.18: $B = 2$, $c_o = 1$, $c_1 = c_2 = 0$, 10 series, algorithm 2.
<table>
<thead>
<tr>
<th>k</th>
<th>$b_1 = 1.000$</th>
<th>$b_2 = .500$</th>
<th>$a_1 = -1.500$</th>
<th>$a_2 = .700$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.977 ±180</td>
<td>.835 ±326</td>
<td>-1.327 ±111</td>
<td>.549 ±088</td>
</tr>
<tr>
<td>200</td>
<td>.964 ±179</td>
<td>.742 ±281</td>
<td>-1.388 ±076</td>
<td>.602 ±062</td>
</tr>
<tr>
<td>300</td>
<td>.970 ±136</td>
<td>.682 ±180</td>
<td>-1.417 ±050</td>
<td>.628 ±043</td>
</tr>
<tr>
<td>400</td>
<td>.992 ±101</td>
<td>.591 ±123</td>
<td>-1.443 ±037</td>
<td>.648 ±033</td>
</tr>
<tr>
<td>500</td>
<td>.994 ±093</td>
<td>.569 ±123</td>
<td>-1.459 ±034</td>
<td>.661 ±028</td>
</tr>
<tr>
<td>600</td>
<td>.992 ±100</td>
<td>.565 ±113</td>
<td>-1.463 ±029</td>
<td>.666 ±025</td>
</tr>
<tr>
<td>700</td>
<td>.992 ±085</td>
<td>.552 ±090</td>
<td>-1.471 ±023</td>
<td>.674 ±020</td>
</tr>
<tr>
<td>800</td>
<td>.972 ±115</td>
<td>.575 ±134</td>
<td>-1.471 ±025</td>
<td>.674 ±022</td>
</tr>
<tr>
<td>900</td>
<td>.977 ±097</td>
<td>.566 ±112</td>
<td>-1.475 ±023</td>
<td>.678 ±020</td>
</tr>
<tr>
<td>1000</td>
<td>.968 ±085</td>
<td>.575 ±095</td>
<td>-1.475 ±022</td>
<td>.678 ±019</td>
</tr>
</tbody>
</table>

Table 4.19: $B = 2$, $c_0 = 1$, $c_1 = -1$, $c_2 = .2$, 10 series, algorithm 2.

<table>
<thead>
<tr>
<th>k</th>
<th>$b_1 = 1.000$</th>
<th>$b_2 = .500$</th>
<th>$a_1 = -1.500$</th>
<th>$a_2 = .700$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.984 ±141</td>
<td>.685 ±228</td>
<td>-1.404 ±063</td>
<td>.615 ±051</td>
</tr>
<tr>
<td>200</td>
<td>.981 ±123</td>
<td>.627 ±192</td>
<td>-1.442 ±044</td>
<td>.649 ±053</td>
</tr>
<tr>
<td>300</td>
<td>.969 ±095</td>
<td>.599 ±128</td>
<td>-1.460 ±025</td>
<td>.665 ±020</td>
</tr>
<tr>
<td>400</td>
<td>.993 ±056</td>
<td>.544 ±080</td>
<td>-1.475 ±016</td>
<td>.677 ±013</td>
</tr>
<tr>
<td>500</td>
<td>.990 ±053</td>
<td>.537 ±079</td>
<td>-1.481 ±015</td>
<td>.683 ±012</td>
</tr>
<tr>
<td>600</td>
<td>.991 ±062</td>
<td>.534 ±082</td>
<td>-1.483 ±013</td>
<td>.685 ±010</td>
</tr>
<tr>
<td>700</td>
<td>.993 ±057</td>
<td>.528 ±065</td>
<td>-1.487 ±011</td>
<td>.688 ±009</td>
</tr>
<tr>
<td>800</td>
<td>.983 ±071</td>
<td>.545 ±095</td>
<td>-1.485 ±014</td>
<td>.687 ±011</td>
</tr>
<tr>
<td>900</td>
<td>.986 ±063</td>
<td>.539 ±081</td>
<td>-1.487 ±012</td>
<td>.689 ±009</td>
</tr>
<tr>
<td>1000</td>
<td>.980 ±055</td>
<td>.545 ±067</td>
<td>-1.487 ±010</td>
<td>.689 ±008</td>
</tr>
</tbody>
</table>

Table 4.20: $B = 1$, $c_0 = 1$, $c_1 = -1.5$, $c_2 = .7$, 10 series, algorithm 2.
Table 4.21 given below, compares our results with those obtained by Åström (10) (using L.S. on all samples). It is seen that the I.V. algorithm 2 gives results which are about the same as those obtained by L.S. estimation (although L.S. estimation only works well in this special case).

Remark 1: \( F = \frac{\sigma_n^2}{\sigma_0^2} \) = ratio of noise power disturbing the output and input-signal power. (Keep in mind that the first 10*\( m \) sample pairs \((u_k, y_k)\) are used for an initial L.S. estimation.)

<table>
<thead>
<tr>
<th>( F )</th>
<th>( b_1 = 1.000 )</th>
<th>( b_2 = .500 )</th>
<th>( a_1 = -1.500 )</th>
<th>( a_2 = .700 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.6</td>
<td>.56 ± .24</td>
<td>-1.49 ± .09</td>
<td>.68 ± .05</td>
</tr>
<tr>
<td>2</td>
<td>2.6</td>
<td>.53 ± .24</td>
<td>-1.50 ± .06</td>
<td>.69 ± .05</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>.46 ± .14</td>
<td>-1.47 ± .06</td>
<td>.66 ± .06</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>.41 ± .61</td>
<td>-1.48 ± .07</td>
<td>.74 ± .08</td>
</tr>
</tbody>
</table>

Table 4.21: A comparison between I.V. algorithm 2 and the L.S. estimator.
1. 50 sample pairs I.V. alg. 2;
2. 100 sample pairs I.V. alg. 2;
3. 100 sample pairs L.S. (Åström);
4. 100 sample pairs L.S. (Åström);

A comparison of the results in table 4.19 (alg. 2) with those in table 4.16 (alg. 1) leads to the same conclusions already given above. If we compare the results in table 4.19 with those of Gustavsson (21) concerning the maximum likelihood method (M.L.) and the prior knowledge fitting method (P.K.F.), and with those of Peterka & Halousková (20) concerning the so-called Tally estimate (T.E.), we can see that the I.V. method gives results that are quite close to those of the other methods. This is clearly shown in table 4.22. The M.L. method gives better results, but it is an explicit method which is very time-consuming.

The results in table 4.20 (alg. 2) when compared with the results in table 4.17 (alg. 1), lead to the same conclusions concerning the initial L.S. estimate's importance on the convergence of the I.V. algorithm. Note that in this case the output disturbance is white noise.

Graph 4.3 gives the results of table 4.15 and 4.18. Here it is clearly seen that algorithm 2 is superior over algorithm 1.
Graph 4.2: The influence of the noise power factor $B$ on the estimates.

Graph 4.3: The influence of the estimator chosen (algorithms 1 and 2) on the estimates ($B=2$).
Remark 2: In the following paragraphs only algorithm 2 will be applied. The process description will be the one corresponding with the D.E. (4.16).

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
F & b_1 = 1.000 & b_2 = .500 & a_1 = -1.500 & a_2 = .700 & c_1 = -1.000 & c_2 = .200 \\
\hline
1 & 2.6 & .994 \pm .085 & .569 \pm .123 & -1.459 \pm .034 & .661 \pm .028 & - \\
2 & 2.6 & .968 \pm .085 & .575 \pm .093 & -1.475 \pm .022 & .678 \pm .019 & - \\
3 & 1 & .957 \pm .055 & .461 \pm .067 & -1.493 \pm .018 & .695 \pm .017 & - .702 \pm .152 & .198 \pm .072 \\
4 & 1 & .949 & .529 & -1.503 & .706 & -.966 & .205 \\
5 & 100 & .594 & .780 & -1.477 & .696 & -.981 & .212 \\
6 & 1 & .959 & .505 & -1.513 & .715 & - & - \\
7 & 100 & .601 & .718 & -1.421 & .661 & - & - \\
\hline
\end{array}
\]

Table 4.22: The results using different estimators on a process with the noise parameters \( c_o = 1; c_1 = -1.000; c_2 = .200. \)
1. I.V. algorithm 2: 10 series; 500 input- output data; \( F = 2.6. \)
2. I.V. algorithm 2: 10 series; 1000 input- output data; \( F = 2.6. \)
3. Tally estimate: 49 series, 500 input- output data; \( F = 1. \)
4. Maximum Likelihood: 1 series; 1000 input- output data; \( F = 1. \)
5. Maximum Likelihood: 1 series, 1000 input- output data; \( F = 100. \)
6. Prior knowledge fitting: 1 series; 1000 input- output data; \( F = 1. \)
7. Prior knowledge fitting: 1 series; 1000 input- output data; \( F = 100. \)

4.7 The generalized extended matrix method

4.7.1 As has been discussed in chapter 2, section 2.3, we can use the following algorithm to obtain an (asymptotically) unbiased estimate of \( b: \)

\[
\hat{b}' = \left\{\Omega^T(u,y,e)\Omega(u,y,e)\right\}^{-1}\Omega^T(u,y,e)y \tag{4.20}
\]

where: \( \hat{b}' = (\hat{e}_0, \hat{e}_1, \ldots, \hat{e}_p, -\alpha_1, \ldots, -\alpha_q, -\gamma_1, \ldots, -\gamma_r) \)

and \( y \) and \( \Omega(u,y,e) \) have been defined in section 2.3.

The sequence \( \{\hat{e}_k\} \) is derived from the relation:

\[
\hat{e}_k = y_k + \sum_{i=1}^{q} \alpha_i y_{k-i} - \sum_{i=0}^{p} \beta_i u_{k-i} \tag{4.21}
\]
When we solve the algorithm in an iterative way, there is a close analogy between this method and that of Hastings-James & Sage (15). Indeed, the results obtained with this generalized extended matrix (L.S.) do correspond extremely well with the results obtained using the method of Hastings-James & Sage. (This method has been extensively used by Talmon (22) using the D.E. \( y_k - 1.5y_{k-1} + .7y_{k-2} = u_{k-1} + .5u_{k-2} + \lambda e_k \), where \( e_k = \xi_k - e_{k-1} - 2e_{k-2} \).)

One result obtained with this generalized extended L.S. matrix is given in Table 4.23. In Table 4.24 we can observe the influence of the truncation error (\( r \) is chosen to be equal to 3 and 2 respectively in the estimation scheme, while the true value – in the simulation program – is about 8). From this latter result it can be seen that the apriori knowledge (or estimate) of \( r \) is of extreme importance for a good estimate of \( 2' \). For low noise power, the \( a_i \) and \( b_j \) are estimated quite well, whether \( r \) is chosen well or poorly.

### Table 4.23: \( B = 4 \) (\( F = 10.3 \)); 10 series; \( \alpha_f = .9913 \) (exponential weighting = \( \alpha^2 \); see appendix III) all \( d_i \) (\( i \geq 3 \)) are zero.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( b_1 = 1.000 )</th>
<th>( b_2 = .500 )</th>
<th>( a_1 = -1.500 )</th>
<th>( a_2 = .700 )</th>
<th>( d_1 = -1.000 )</th>
<th>( d_2 = .200 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>1.007 ± .357</td>
<td>.167 ± .668</td>
<td>-1.645 ± .048</td>
<td>.775 ± .041</td>
<td>- .863 ± 1.08</td>
<td>.219 ± .058</td>
</tr>
<tr>
<td>500</td>
<td>1.125 ± .230</td>
<td>.312 ± .427</td>
<td>-1.591 ± .068</td>
<td>.733 ± .057</td>
<td>- .914 ± .059</td>
<td>.230 ± .110</td>
</tr>
<tr>
<td>1000</td>
<td>.980 ± .275</td>
<td>.340 ± .282</td>
<td>-1.553 ± .061</td>
<td>.716 ± .053</td>
<td>- .943 ± .103</td>
<td>.204 ± .063</td>
</tr>
</tbody>
</table>

### Table 4.24: \( B = .5 \) (\( F = .16 \)); 10 series; \( \alpha_f = .9913 \); \( d_4 = .44 \); \( d_5 = .32 \); \( d_6 = .232 \); \( d_7 = .168 \).

<table>
<thead>
<tr>
<th>( k )</th>
<th>( b_1 = 1.000 )</th>
<th>( b_2 = .500 )</th>
<th>( a_1 = -1.500 )</th>
<th>( a_2 = .700 )</th>
<th>( d_1 = 1.000 )</th>
<th>( d_2 = .800 )</th>
<th>( d_3 = .600 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>.993 ± .048</td>
<td>.491 ± .092</td>
<td>-1.506 ± .018</td>
<td>.705 ± .013</td>
<td>.867 ± .054</td>
<td>.555 ± .069</td>
<td>.228 ± .091</td>
</tr>
<tr>
<td>500</td>
<td>1.020 ± .028</td>
<td>.494 ± .044</td>
<td>-1.493 ± .012</td>
<td>.694 ± .010</td>
<td>.863 ± .069</td>
<td>.521 ± .067</td>
<td>.244 ± .027</td>
</tr>
<tr>
<td>1000</td>
<td>.996 ± .031</td>
<td>.499 ± .049</td>
<td>-1.500 ± .012</td>
<td>.700 ± .010</td>
<td>.912 ± .065</td>
<td>.584 ± .068</td>
<td>.256 ± .076</td>
</tr>
<tr>
<td>250</td>
<td>.998 ± .049</td>
<td>.484 ± .093</td>
<td>-1.507 ± .018</td>
<td>.705 ± .013</td>
<td>.788 ± .057</td>
<td>.381 ± .052</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>1.018 ± .032</td>
<td>.491 ± .046</td>
<td>-1.495 ± .010</td>
<td>.696 ± .009</td>
<td>.784 ± .060</td>
<td>.331 ± .052</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>.999 ± .030</td>
<td>.498 ± .049</td>
<td>-1.500 ± .012</td>
<td>.700 ± .010</td>
<td>.822 ± .057</td>
<td>.377 ± .048</td>
<td></td>
</tr>
</tbody>
</table>
4.7.2 If we use now $\hat{\Sigma}^{T}(u, w, \varepsilon)$ instead of $\hat{\Sigma}^{T}(u, y, \varepsilon)$ in eq. (4.20), where the elements $w_k$ are corresponding outputs of the model (cfr. Fig. 4.1), we obtain results that combine the properties of the I.V. estimator with those of a generalized L.S. estimator (e.g. the estimation of the $d_i$). Table 4.25 gives the results using this algorithm i.e.:

$$\hat{\theta}' = (\hat{\Sigma}^{T}(u, w, \varepsilon)\hat{\Sigma}(u, y, \varepsilon))^{-1}\hat{\Sigma}^{T}(u, w, \varepsilon)\gamma,$$

(4.22)

applied to the same process that produced the results shown in table 4.23 (e.g. all $d_i = 0$ (i > 3)).

<table>
<thead>
<tr>
<th>F</th>
<th>B</th>
<th>k</th>
<th>$b_1=1.000$</th>
<th>$b_2=0.500$</th>
<th>$a_1=-1.500$</th>
<th>$a_2=.700$</th>
<th>$d_1=-1.000$</th>
<th>$d_2=.200$</th>
</tr>
</thead>
<tbody>
<tr>
<td>.01</td>
<td>.125</td>
<td>250</td>
<td>1.000 +012</td>
<td>.489 +016</td>
<td>-1.505 +013</td>
<td>.702 +011</td>
<td>-1.007 +090</td>
<td>.237 +071</td>
</tr>
<tr>
<td>500</td>
<td>1.003 +008</td>
<td>.502 +015</td>
<td>-1.499 +008</td>
<td>.697 +008</td>
<td>-1.013 +088</td>
<td>.237 +071</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1.000 +008</td>
<td>.499 +022</td>
<td>-1.499 +018</td>
<td>.699 +014</td>
<td>- .992 +065</td>
<td>.194 +074</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.04</td>
<td>.25</td>
<td>250</td>
<td>1.000 +024</td>
<td>.474 +030</td>
<td>-1.510 +025</td>
<td>.702 +020</td>
<td>- .998 +100</td>
<td>.227 +080</td>
</tr>
<tr>
<td>500</td>
<td>1.006 +015</td>
<td>.505 +029</td>
<td>-1.496 +014</td>
<td>.692 +013</td>
<td>-1.016 +083</td>
<td>.237 +068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1.000 +017</td>
<td>.500 +042</td>
<td>-1.497 +026</td>
<td>.698 +028</td>
<td>- .995 +065</td>
<td>.197 +074</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.16</td>
<td>.50</td>
<td>250</td>
<td>1.002 +049</td>
<td>.426 +060</td>
<td>-1.528 +043</td>
<td>.708 +034</td>
<td>- .923 +106</td>
<td>.174 +076</td>
</tr>
<tr>
<td>500</td>
<td>1.013 +030</td>
<td>.505 +053</td>
<td>-1.493 +028</td>
<td>.683 +023</td>
<td>- .980 +085</td>
<td>.206 +066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1.000 +033</td>
<td>.502 +070</td>
<td>-1.492 +049</td>
<td>.692 +055</td>
<td>- .970 +073</td>
<td>.180 +079</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.64</td>
<td>1.0</td>
<td>250</td>
<td>1.007 +098</td>
<td>.310 +154</td>
<td>-1.581 +050</td>
<td>.740 +044</td>
<td>- .902 +103</td>
<td>.182 +055</td>
</tr>
<tr>
<td>500</td>
<td>1.028 +057</td>
<td>.477 +094</td>
<td>-1.503 +051</td>
<td>.681 +045</td>
<td>- .994 +083</td>
<td>.210 +071</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>.999 +068</td>
<td>.491 +136</td>
<td>-1.487 +088</td>
<td>.684 +102</td>
<td>-1.014 +114</td>
<td>.221 +114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.6</td>
<td>2.0</td>
<td>250</td>
<td>1.007 +187</td>
<td>.152 +332</td>
<td>-1.669 +061</td>
<td>.809 +060</td>
<td>- .838 +095</td>
<td>.198 +034</td>
</tr>
<tr>
<td>500</td>
<td>1.048 +116</td>
<td>.386 +158</td>
<td>-1.540 +104</td>
<td>.698 +101</td>
<td>- .941 +106</td>
<td>.179 +179</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>.993 +142</td>
<td>.422 +241</td>
<td>-1.499 +151</td>
<td>.684 +100</td>
<td>-1.011 +170</td>
<td>.256 +184</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.25: $B$ variable (F); 10 series; alf = .9913; $d_i = 0$ (i > 3).

As can be seen from these results, the standard deviation (calculated over 10 series) grows linearly with $B$ (the noise factor) for the $a_i$ as well as for the $\beta_j$. This standard deviation is about two times as large for the $\beta_j$ as for the $a_i$ (why?), while the standard deviation of the noise parameters $\gamma_i$ is about the same for all $B$. 
This latter fact can be explained in the following way (fig. 4.6):

$$\gamma = -(E^TE)^{-1}E^T \xi, \quad (4.23)$$

**Fig. 4.6:** Equivalent noise filter

and:

$$d = \gamma + (E^TE)^{-1}E^T \xi, \quad (4.24)$$

So:

$$\Delta \gamma = d - \gamma = (E^TE)^{-1}E^T \xi. \quad (4.25)$$

As the magnitude of $e_k$ is directly related to the magnitude of $\xi_k$, an increase of the input of the filter to $\xi_k$ will result in an output

$$\Delta \gamma = \lambda^{-2}(E^TE)^{-1}\lambda E^T \xi = (E^TE)^{-1}E^T \xi$$

Thus:

$$\text{cov} \gamma = E(\Delta \gamma \Delta \gamma^T) = (E^TE)^{-1}, \quad \text{independent of the value of} \ \lambda. \quad (4.26)$$

If however the sequence $\{e_k\}^N_{k=1}$ is not (approximately) equal to the sequence $\{e_k\}^N_{k=1}$, then the $\gamma$ will depend on $\lambda$ (large values of $B$).

From table 4.25 it can be seen that all estimated parameters do agree quite well with the true parameters (including the noise parameters). If we compare these results with table 4.22, we can say that the generalized I.V. method is at least as good as the results from the Tally estimate and as good as the M.L. method and the P.K.F. method.

This comparison can only give an "impression" about the quality of the generalized I.V. method, because the $d_i$ in our process are the same as the $c_i$ in the process considered by Gustavsson (21) and Peterka & Halousková (20).

The results given in table 4.25 for $B = .5$ do agree quite well with those obtained by Hastings-James & Sage (15).

4.7.3 We will now use the algorithm

$$\hat{B} = (\Omega^T(u, y, \xi) \Omega(u, y, \xi))^{-1} \Omega^T(u, y, \xi) \chi \quad (4.26)$$

which has been proposed in chapter 2, section 2.3 (remark 2).

The results obtained are given in table 4.26.

If we compare these results with those given in table 4.22 (M.L., Tally, P.K.F.) we see that they are about the same. It should be noticed however that this method gives extremely good results even for a noise power to input signal power ratio of 42. This latter case once again shows the necessity of taking more than one series input-output values.
Table 4.26: The results using the algorithm given by eq. (4.26) (5 series); B variable, $\alpha = 0.9913$.

The individual series give quite different results concerning $\theta_2$ especially. Taking the mean results in an estimate of this parameter which is reasonably good.

4.7.4 The algorithm

$$\theta'' = \left(\Omega^T(u, w, \xi)\Omega(u, y, \xi)\right)^{-1}u^T(u, w, \xi)x$$  \hspace{1cm} (4.27)

has been used to obtain the results given in table 4.27.

Table 4.27: The results obtained using the algorithm given by eq. (4.27) (5 series); B variable, $\alpha = 0.9913$. 

<table>
<thead>
<tr>
<th>k</th>
<th>$b_1=1.000$</th>
<th>$b_2=.500$</th>
<th>$a_1=-1.500$</th>
<th>$a_2=.700$</th>
<th>$c_1=-1.000$</th>
<th>$c_2=.200$</th>
</tr>
</thead>
<tbody>
<tr>
<td>B=1</td>
<td>250</td>
<td>.957 ± 1.04</td>
<td>.504 ± 1.97</td>
<td>-1.514 ± 0.031</td>
<td>.711 ± 0.024</td>
<td>- .915 ± 0.049</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1.047 ± 0.48</td>
<td>.550 ± 0.96</td>
<td>-1.476 ± 0.32</td>
<td>.684 ± 0.31</td>
<td>-1.014 ± 0.083</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1.031 ± 0.044</td>
<td>.446 ± 0.98</td>
<td>-1.512 ± 0.031</td>
<td>.709 ± 0.027</td>
<td>- .995 ± 0.058</td>
</tr>
<tr>
<td>B=2</td>
<td>250</td>
<td>.902 ± 2.19</td>
<td>.530 ± 0.92</td>
<td>-1.517 ± 0.074</td>
<td>.712 ± 0.062</td>
<td>- .895 ± 0.075</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1.091 ± 1.02</td>
<td>.613 ± 1.92</td>
<td>-1.449 ± 0.060</td>
<td>.665 ± 0.058</td>
<td>- .984 ± 0.107</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1.057 ± 0.088</td>
<td>.406 ± 1.92</td>
<td>-1.519 ± 0.060</td>
<td>.714 ± 0.049</td>
<td>-1.001 ± 0.090</td>
</tr>
<tr>
<td>B=4</td>
<td>250</td>
<td>.787 ± 4.68</td>
<td>.813 ± 0.645</td>
<td>-1.402 ± 1.85</td>
<td>.615 ± 1.77</td>
<td>- .762 ± 1.78</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1.167 ± 2.22</td>
<td>.754 ± 3.77</td>
<td>-1.377 ± 1.02</td>
<td>.634 ± 0.98</td>
<td>- .932 ± 1.16</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1.098 ± 1.68</td>
<td>.370 ± 3.43</td>
<td>-1.514 ± 1.04</td>
<td>.711 ± 0.79</td>
<td>- .994 ± 1.07</td>
</tr>
<tr>
<td>B=8</td>
<td>250</td>
<td>.597 ± 8.95</td>
<td>1.386 ± 1.032</td>
<td>-1.192 ± 2.94</td>
<td>.470 ± 2.53</td>
<td>- .578 ± 2.59</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1.306 ± 4.71</td>
<td>.990 ± 7.00</td>
<td>-1.325 ± 1.33</td>
<td>.600 ± 1.27</td>
<td>- .856 ± 1.79</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1.164 ± 2.98</td>
<td>.383 ± 5.71</td>
<td>-1.462 ± 1.71</td>
<td>.677 ± 1.09</td>
<td>- .940 ± 1.71</td>
</tr>
</tbody>
</table>
The results are about the same as those given in table 4.26, although the standard deviation here is a little bit higher than that obtained using the previous algorithm (eq. (4.26); B = 4.). The explanation for this is quite obvious. For the case where B = 4 the poor initial L.S. estimate \( \hat{\theta}_0 \) causes a bad adjustment of the model-parameters in the beginning (small N). The results for large N however are much better, which shows once again the strong convergence of the sequence \( \{ w_k \}_{k=1}^{N+q} \) to the real process output sequence \( \{ x_k \}_{k=1}^{N+q} \).

### 4.8 The sub-optimal I.V. method

This method has been described in detail in chapter 3. It makes use of the following algorithm:

\[
\hat{\theta} = (\Omega(u^*, w^*)\Omega(u^*, y^*))^{-1}\Omega(u^*, w^*)y^*
\]  

(4.28)

where:

\[
u^*_k = u^*_k + \gamma_i u_{k-i}, \\
y^*_k = y^*_k + \gamma_i y_{k-i}, \\
w^*_k = w^*_k + \gamma_i w_{k-i}.
\]

The \( \gamma_i \) are obtained using the L.S. algorithm:

\[
y = -(E^TE)^{-1}ETe,
\]

and \( E \) and \( e \) have been defined in chapter 3, eq. (3.49). The components of \( E \) and \( e \) will be calculated using eq. (3.51).

If we compare the results (B=.5; B=2) given in table 4.28 with those of table 4.25 (B=.5; B=2) we see that the calculated standard deviation is a bit lower for the results in table 4.28, although the difference is not pronounced.

<table>
<thead>
<tr>
<th>B</th>
<th>k</th>
<th>( b_1 = 1.000 )</th>
<th>( b_2 = 0.500 )</th>
<th>( a_1 = -1.500 )</th>
<th>( a_2 = 0.700 )</th>
<th>( d_1 = -1.000 )</th>
<th>( d_2 = 0.200 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5</td>
<td>250</td>
<td>1.000 ± 0.045</td>
<td>4.88 ± 0.035</td>
<td>-1.503 ± 0.051</td>
<td>0.695 ± 0.044</td>
<td>-0.989 ± 0.098</td>
<td>0.233 ± 0.074</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1.011 ± 0.030</td>
<td>5.09 ± 0.042</td>
<td>-1.503 ± 0.025</td>
<td>0.701 ± 0.016</td>
<td>-1.002 ± 0.099</td>
<td>0.236 ± 0.073</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0.998 ± 0.035</td>
<td>5.07 ± 0.048</td>
<td>-1.491 ± 0.040</td>
<td>0.694 ± 0.037</td>
<td>-1.002 ± 0.071</td>
<td>0.174 ± 0.076</td>
</tr>
<tr>
<td>2</td>
<td>250</td>
<td>1.977 ± 0.185</td>
<td>3.66 ± 1.61</td>
<td>-1.573 ± 1.36</td>
<td>0.743 ± 1.12</td>
<td>-0.908 ± 1.37</td>
<td>0.232 ± 0.89</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1.028 ± 0.140</td>
<td>5.02 ± 0.209</td>
<td>-1.531 ± 0.085</td>
<td>0.727 ± 0.074</td>
<td>-0.981 ± 1.40</td>
<td>0.260 ± 0.098</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1.008 ± 0.144</td>
<td>5.18 ± 1.97</td>
<td>-1.478 ± 1.60</td>
<td>0.688 ± 1.49</td>
<td>-1.023 ± 1.50</td>
<td>0.216 ± 1.39</td>
</tr>
</tbody>
</table>

Table 4.28: B variable (F); alf = .9913; \( d_i = 0 \) (i > 3); 10 series.
<table>
<thead>
<tr>
<th>k</th>
<th>$b_0=0.000$</th>
<th>$b_1=1.000$</th>
<th>$b_2=.500$</th>
<th>$a_1=-1.500$</th>
<th>$a_2=.700$</th>
<th>$d_1=-1.000$</th>
<th>$d_2=.200$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.998±255</td>
<td>.449±310</td>
<td>-1.619±101</td>
<td>.774±096</td>
<td>-1.857±188</td>
<td>.269±142</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>.956±178</td>
<td>.370±232</td>
<td>-1.582±130</td>
<td>.750±100</td>
<td>-1.882±138</td>
<td>.211±117</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>.961±167</td>
<td>.426±199</td>
<td>-1.555±137</td>
<td>.716±097</td>
<td>-1.919±131</td>
<td>.230±081</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>.998±125</td>
<td>.476±166</td>
<td>-1.548±140</td>
<td>.714±097</td>
<td>-1.946±133</td>
<td>.263±114</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>1.028±140</td>
<td>.502±209</td>
<td>-1.531±085</td>
<td>.727±074</td>
<td>-1.981±140</td>
<td>.260±098</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>.990±124</td>
<td>.428±155</td>
<td>-1.537±081</td>
<td>.753±086</td>
<td>-1.997±084</td>
<td>.213±066</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>1.006±121</td>
<td>.547±131</td>
<td>-1.497±092</td>
<td>.707±096</td>
<td>-1.992±139</td>
<td>.183±133</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>.998±190</td>
<td>.495±090</td>
<td>-1.502±106</td>
<td>.711±099</td>
<td>-1.970±108</td>
<td>.156±104</td>
<td></td>
</tr>
<tr>
<td>900</td>
<td>.979±162</td>
<td>.502±169</td>
<td>-1.462±171</td>
<td>.662±138</td>
<td>-1.995±139</td>
<td>.186±111</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1.008±144</td>
<td>.518±197</td>
<td>-1.478±160</td>
<td>.688±149</td>
<td>1.003±150</td>
<td>.216±139</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.29: $B = 2$; 10 series; $b_0$ is assumed to be zero (not-estimated); $\alpha = .9913$.

<table>
<thead>
<tr>
<th>k</th>
<th>$b_0=0.000$</th>
<th>$b_1=1.000$</th>
<th>$b_2=.500$</th>
<th>$a_1=-1.500$</th>
<th>$a_2=.700$</th>
<th>$d_1=-1.000$</th>
<th>$d_2=.200$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.030±161</td>
<td>.529±227</td>
<td>-1.618±136</td>
<td>.759±125</td>
<td>-1.887±207</td>
<td>.312±156</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>.990±181</td>
<td>.422±183</td>
<td>-1.582±151</td>
<td>.741±113</td>
<td>-1.898±173</td>
<td>.258±107</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>.991±195</td>
<td>.515±166</td>
<td>-1.501±157</td>
<td>.661±130</td>
<td>-1.979±179</td>
<td>.299±094</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>1.024±122</td>
<td>.544±122</td>
<td>-1.515±148</td>
<td>.665±128</td>
<td>-1.1007±179</td>
<td>.328±129</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>1.039±129</td>
<td>.504±159</td>
<td>-1.556±140</td>
<td>.760±114</td>
<td>-1.031±197</td>
<td>.325±124</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>.992±131</td>
<td>.442±128</td>
<td>-1.514±152</td>
<td>.753±161</td>
<td>-1.036±171</td>
<td>.249±121</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>1.014±120</td>
<td>.539±160</td>
<td>-1.509±162</td>
<td>.728±154</td>
<td>-1.046±202</td>
<td>.231±174</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>.999±197</td>
<td>.476±103</td>
<td>-1.515±195</td>
<td>.750±158</td>
<td>-1.028±186</td>
<td>.204±164</td>
<td></td>
</tr>
<tr>
<td>900</td>
<td>.995</td>
<td>.502</td>
<td>-1.493</td>
<td>.690</td>
<td>-1.008</td>
<td>.209</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>.998</td>
<td>.575</td>
<td>-1.463</td>
<td>.665</td>
<td>-1.054</td>
<td>.266</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.30: $B = 2$; 10 series; $b_0$ is assumed to be zero (not-estimated); no filter at input; corrections on the mean for $k = 900$ and 1000; (one series gave very bad results here); $\alpha = .9913$. 
### Table 4.31: B = 2; 10 series; b has been estimated; alf = .9913.

<table>
<thead>
<tr>
<th>k</th>
<th>-b = 0.000</th>
<th>b = 1.000</th>
<th>b = .500</th>
<th>b = -1.500</th>
<th>a = .700</th>
<th>d = -1.000</th>
<th>d = .200</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.026 ± .222</td>
<td>.889 ± .272</td>
<td>.458 ± .253</td>
<td>-1.595 ± .131</td>
<td>.747 ± .124</td>
<td>-.935 ± .175</td>
<td>.322 ± .129</td>
</tr>
<tr>
<td>200</td>
<td>-.033 ± .175</td>
<td>1.010 ± .141</td>
<td>.441 ± .174</td>
<td>-1.572 ± .136</td>
<td>.743 ± .123</td>
<td>-.977 ± .139</td>
<td>.302 ± .120</td>
</tr>
<tr>
<td>300</td>
<td>-.074 ± .112</td>
<td>.958 ± .109</td>
<td>.508 ± .143</td>
<td>-1.530 ± .119</td>
<td>.720 ± .122</td>
<td>-.962 ± .157</td>
<td>.235 ± .144</td>
</tr>
<tr>
<td>400</td>
<td>.003 ± .140</td>
<td>.908 ± .126</td>
<td>.448 ± .141</td>
<td>-1.486 ± .125</td>
<td>.688 ± .089</td>
<td>-.969 ± .123</td>
<td>.215 ± .133</td>
</tr>
<tr>
<td>500</td>
<td>.024 ± .115</td>
<td>.968 ± .113</td>
<td>.526 ± .144</td>
<td>-1.465 ± .121</td>
<td>.673 ± .130</td>
<td>-1.028 ± .130</td>
<td>.250 ± .109</td>
</tr>
<tr>
<td>600</td>
<td>.104 ± .157</td>
<td>.937 ± .130</td>
<td>.467 ± .125</td>
<td>-1.437 ± .148</td>
<td>.655 ± .132</td>
<td>-1.012 ± .103</td>
<td>.211 ± .055</td>
</tr>
<tr>
<td>700</td>
<td>.032 ± .123</td>
<td>.981 ± .153</td>
<td>.501 ± .123</td>
<td>-1.486 ± .082</td>
<td>.689 ± .081</td>
<td>-1.039 ± .131</td>
<td>.253 ± .099</td>
</tr>
<tr>
<td>800</td>
<td>.097 ± .140</td>
<td>.947 ± .158</td>
<td>.439 ± .224</td>
<td>-1.535 ± .141</td>
<td>.755 ± .126</td>
<td>-.977 ± .085</td>
<td>.186 ± .045</td>
</tr>
<tr>
<td>900</td>
<td>.059 ± .151</td>
<td>.966 ± .177</td>
<td>.491 ± .138</td>
<td>-1.498 ± .104</td>
<td>.669 ± .136</td>
<td>-1.009 ± .127</td>
<td>.222 ± .150</td>
</tr>
<tr>
<td>1000</td>
<td>.094 ± .141</td>
<td>.943 ± .123</td>
<td>.459 ± .119</td>
<td>-1.476 ± .129</td>
<td>.671 ± .135</td>
<td>-1.046 ± .112</td>
<td>.257 ± .140</td>
</tr>
</tbody>
</table>

### Table 4.32: B = 2; 10 series; b has been estimated; no filter at input; corrections on the mean values for all k; alf = .9913.

<table>
<thead>
<tr>
<th>k</th>
<th>-b = 0.000</th>
<th>b = 1.000</th>
<th>b = .500</th>
<th>b = -1.500</th>
<th>a = .700</th>
<th>d = -1.000</th>
<th>d = .200</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.037</td>
<td>1.010</td>
<td>.526</td>
<td>-1.680</td>
<td>.824</td>
<td>-.865</td>
<td>.276</td>
</tr>
<tr>
<td>200</td>
<td>.080</td>
<td>1.148</td>
<td>.503</td>
<td>-1.663</td>
<td>.708</td>
<td>-.893</td>
<td>.261</td>
</tr>
<tr>
<td>300</td>
<td>.102</td>
<td>1.070</td>
<td>.538</td>
<td>-1.603</td>
<td>.778</td>
<td>-.880</td>
<td>.195</td>
</tr>
<tr>
<td>400</td>
<td>.041</td>
<td>.993</td>
<td>.476</td>
<td>-1.548</td>
<td>.736</td>
<td>-.911</td>
<td>.195</td>
</tr>
<tr>
<td>500</td>
<td>-.010</td>
<td>.990</td>
<td>.518</td>
<td>-1.520</td>
<td>.712</td>
<td>-.984</td>
<td>.236</td>
</tr>
<tr>
<td>600</td>
<td>-.109</td>
<td>.870</td>
<td>.442</td>
<td>-1.413</td>
<td>.641</td>
<td>-1.042</td>
<td>.232</td>
</tr>
<tr>
<td>700</td>
<td>-.063</td>
<td>.892</td>
<td>.441</td>
<td>-1.440</td>
<td>.691</td>
<td>-1.098</td>
<td>.292</td>
</tr>
<tr>
<td>800</td>
<td>-.086</td>
<td>.904</td>
<td>.429</td>
<td>-1.500</td>
<td>.721</td>
<td>-1.053</td>
<td>.243</td>
</tr>
<tr>
<td>900</td>
<td>-.074</td>
<td>.875</td>
<td>.461</td>
<td>-1.429</td>
<td>.594</td>
<td>-1.076</td>
<td>.275</td>
</tr>
<tr>
<td>1000</td>
<td>-.127</td>
<td>.832</td>
<td>.428</td>
<td>-1.390</td>
<td>.610</td>
<td>-1.168</td>
<td>.354</td>
</tr>
</tbody>
</table>

Table 4.32: B = 2; 10 series; b has been estimated; no filter at input; corrections on the mean values for all k; alf = .9913.
The influence of the estimation of a parameter, which is equal to zero (e.g. $b_0$) on the algorithm is clearly shown comparing the results of table 4.29 with those in table 4.31, resp. table 4.30 with those in table 4.32. As can be seen the estimate of $b_1$ is now not as good as in the previous cases, while the influence on the other parameters can be neglected. The results given in tables 4.30 and 4.32 have been obtained without the application of the input filter mentioned in 4.3, remark 1. This means that the noise power to signal power ratio now is 4 ($B = 2$), while in tables 4.29 and 4.31 this ratio was about 2.6. As a consequence, the initial L.S. estimate $\hat{p}_0$ is not as good in the case where $F = 4$ as in the case where $F = 2.6$, and thus the convergence of our algorithm is slower for the results in the tables 4.30 and 4.32.

Remark 1: The results shown in table 4.32 have been corrected. One of the ten series gave such bad results, that it has been omitted. So the average over 9 series has been taken.

Remark 2: A consequence of the weighting of input-, output- and model-output data is that we have a lower bound on the standard deviations of the estimates.

4.9 Conclusion

The I.V. method has great advantages over normal L.S. When the noise level is not too high, it is as good as the other methods stated (M.L. Tally, P.K.F., generalized L.S., extended matrix methods). When the noise level is too high, we have to incorporate a stability test for the model in order to obtain results that have any meaning at all (Wong (18)). The results using the extended matrix methods indicate that these methods also give good results (even in the case of high noise levels, 4.7.3). They are a sort of I.V. approach.

More work on the optimized I.V. has to be done in order to show that it leads to minimal standard deviation in the estimated parameters.
5. Additional remarks and suggestions

It has been shown by the results given in chapter 4, that the instrumental variable estimators quickly converge to the true parameter values (practical consistency; finite sequence length N). By I.V. estimator is meant a method that reduces the equivalent noise term \( e_{eq} \) in the process D.E. to zero in probability, i.e.:

\[
\lim_{N \to \infty} \frac{1}{N} Z^T e_{eq} = 0
\]  (5.1)

This definition includes the estimators given in chapter 3 and those given in chapter 2, section 2.3, or:

1) \( \hat{y} = \Omega(u,y,e)b' + \xi \)
   \( \Omega(u,y,e) \) is an I.V. matrix as here \( e_{eq} = \xi \);

2) \( \hat{y} = \Omega(u,y,\xi)b'' + \xi \)
   \( \Omega(u,y,\xi) \) is an I.V. matrix as here \( e_{eq} = \xi \);

3) \( \hat{y} = \Omega(u,\xi)b + e \)
   \( \Omega(u,w) \) is an I.V. matrix;

4) \( \hat{y} = \Omega(u,\xi)b + e \)
   \( \Omega(u^*,\xi^*) \) is the optimal I.V. matrix, combined with the eq.
   \( \hat{y}^* = \Omega(u^*,y^*)b + e^* \).

So it is seen that the only inherent physical idea behind the I.V. method is the elimination of the equivalent noise term in one way or another. It does not make use explicitly of a quadratic criterion or other integral criteria, which are in some cases difficult to interpret, and even give results that are not consistent at all.

In this way, we see that the I.V. method can be applied with success in cases where we do not know very much about the process itself, except for its structure.

If one is interested in an estimate which has a small standard deviation, the optimized I.V. algorithm described in section 3.4 can be used. In this case we will introduce after say \( N_1 \) iterations an abrupt (or gradually) change in alfa (weighting factor) to one, so as to make sure that the standard deviation is allowed to decrease (remark 2, section 4.8). More work has to be done in order to show asymptotically small standard deviations using this method in practice.
The weighting, applied to obtain an estimate of the noise parameters, puts a lower bound on the standard deviations of the estimates of the process parameters. It is interesting to note that this weighting allows tracking of (slow) variations in the parameters. In this way, we have made an algorithm for time-varying systems.

We have seen in chapters 2 and 3 that the noise parameters obtained are the $c_i$ or the $d_j$ defined in chapter 2. We know that the following relation exists:

$$\frac{(1+A(z^{-1}))1+C(z^{-1})}{1+P(z^{-1})} = 1+C(z^{-1}).$$ (3.8)

So in order to find the parameters of the "noise-filter" itself, we have to take the ratio of the polynomials $1+e_i z^{-1} + \ldots + e_s z^{-s}$ over $1+a_1 z^{-1} + \ldots + a_q z^{-q}$. By making use of the modeloutput sequence $\{w_k\}^{N+q}_{N+1}$, we have an estimate of $\{n_{k}^{N+q}\}^{N+1}$ available as:

$$\{n_{k}^{N+q}\}^{N+1} = \{y_k^{N+q}\}^{N+1} - \{w_k^{N+q}\}^{N+1}.$$ 

So we can use a L.S. estimator on the sequence $\{n_k^{N+q}\}^{N+1}$ to obtain the coefficients of the noise-filter itself $\frac{1+G}{1+F}$. If one is interested in the noise-filter itself, it is expected that this latter method will yield better estimates than the former one. Indeed, the consistency of the I.V. method ensures the condition that $w_k \rightarrow x_k$ (k large).

In the literature, more and more "mathematical" models are mentioned for solution of the estimation problem. Especially in parameter estimation of non-linear systems it is believed that the I.V. approach could yield consistent estimates too.

**Remark 1:** In agreement with other publications we have used in chapter 4 the factor $F = \sigma_z^2 / \sigma_x^2$. This factor gives misleading figures. What one really wants to talk about is the additive noise power at the output divided by the output power of the process or $\sigma_n^2 / \sigma_x^2$ (fig. 5.1).

![Fig. 5.1: The meaning of power ratios.](image-url)
Calculations show that when the D.E. of the process is:

\[ y_{k-1} - 1.5y_{k-1} + 0.7v_{k-2} = u_{k-1} + 0.5u_{k-2} + \xi_{k-1} - \xi_{k-2} + 2\xi_{k-2} \]  

the following relation holds:

\[ \frac{\sigma_n^2}{\sigma_x^2} = \frac{1.5}{19} \quad \frac{\sigma_\xi^2}{\sigma_u^2} = \frac{1.5}{19} \quad F < .1F. \]

So it is seen that, even for B = 3 (B = noise factor), the noise power at the output does not exceed the process power output.

Remark 2: All estimation and simulation programs are written in Algol 60 and are available at the Eindhoven University of Technology, Department of Electrical Engineering, Group Measurement and Control, Eindhoven, the Netherlands.
List of symbols

\( u, y, e \) : vectors containing samples of input signal, output signal, 

\( a, b, c, d \) : vectors containing the parameter values to be estimated.

\( \beta, \gamma \) : vectors containing the estimates.

\( A, B, C, E, \) : matrices

\( A(z^{-1}), B(z^{-1}) \) : polynomials in \( z \), which have the following structure:

\[
A(z^{-1}) = a_1 z^{-1} + \ldots + a_q z^{-q}.
\]

\( x_k, u_k, y_k, n_k \ldots \) : shorthand notation to represent the sampled values at time \( t = kT \) of \( x, u, y, n, \ldots \)

\( \tau \) : sample period

\( E\{ \} \) : expectation operator

\( \Omega(u, y, e) \) : matrix having the structure \( (U; Y; E) \)

\( \xi \) : white noise

\( n \) : additive output noise

\( u \) : input signal

\( x \) : output signal

\( y \) : disturbed output signal

\( w \) : model output signal

\( z \) : shift-operator

\( \Psi \) : correlation function

\( f \) : input signal of the system in the case of a closed-loop situation (i.e. \( f = u - y \))

Remark: Most symbols are defined in the text itself.

When working with a polynomial such as \( A(z^{-1}) \), we write it simply as \( A(z^{-1}) \) whenever this cannot be confused with matrix notation.
Appendix I: Definitions, conventions

1. Definitions
   a) if \((x, x_1, x_2, \ldots)\) is a stochastic process such that for any \(\varepsilon > 0\) (arbitrarily chosen) \(\lim_{N \to \infty} P(|x - x| \geq \varepsilon) = 0\), then we say that \((x, x_1, x_2, \ldots)\) has the probability limit \(x\) and we write:

   \[
   \text{plim} \ x_n = x,
   \]

   b) Slutzky's theorem: suppose that the stochastic process \((x, x_1, x_2, \ldots)\) in \(\mathbb{R}^p\) converges in probability towards \(x\). Suppose the function \(g\), which gives the transformation from \(\mathbb{R}^p\) to \(\mathbb{R}^q\), to be continuous. Then it holds:

   \[
   \text{plim} \ g(x_n) = g(\text{plim} \ x_n) = g(x);
   \]

   c) if the elements of the matrices \(A_i\) and \(B_i\) have a probability limit, then the following holds:

   \[
   \text{plim} \ (A_i^{-1}B_i) = (\text{plim} \ A_i)^{-1} (\text{plim} \ B_i)
   \]

   (which can be deduced from b)).


2. Conventions
   - a polynomial \(b_0 u_k + b_1 u_{k-1} + \ldots + b_p u_{k-p}\) will be represented in most cases by \([b_0 + B(z^{-1})] u_k\) or \([B_0 + B] u_k\), by making use of the shift-operator \(z^{-1} u_k = u_{k-1}\).
   - the "index" variable beneath a summation sign will be omitted, if it is clear from inspection which variable is meant in the formula i.e.

   \[
   \sum_{i=0}^{p} b_i z^{-i} \rightarrow \sum_{i=0}^{p} b_i z^{-i}
   \]

   - the representation of a general filter, consisting of a forward as well as a backward part, will be indicated by the following symbol:

   \[
   u_k \quad \begin{array}{c|c|c}
   b & \quad a \quad \rightarrow \quad x_k
   \end{array}
   \]

   If \(b = (1, 0, \ldots, 0) = 1\) the filter only consists of a backward part; if \(a = (1, 0, \ldots, 0) = 1\) the filter only consists of a forward part.
Appendix II: Recursive formulae

We give a recursive form of the mean and variance of a sequence of observations \( \{z_k\}_{k+1} \).

1. The recursive form for the mean

By definition: 
\[
\bar{z}_k = \frac{1}{k} \sum_{i=1}^{k} z_i
\]

(k observations \( z_i \))

and 
\[
\bar{z}_{k+1} = \frac{1}{k+1} \sum_{i=1}^{k+1} z_i
\]

(k+1 observations \( z_i \))

So: 
\[
\bar{z}_{k+1} = \frac{1}{k+1} \left[ k \bar{z}_k + z_{k+1} \right]
\]

or:
\[
\bar{z}_{k+1} = \bar{z}_k + \frac{1}{k+1} \left( z_{k+1} - \bar{z}_k \right)
\]  

(I)

2. The recursive form for the variance

By definition: 
\[
s^2 = \frac{1}{k-1} \sum_{i=1}^{k} (z_i - \bar{z}_k)^2
\]

(k observations \( z_i \))

and:
\[
s^2_{k+1} = \frac{1}{k+1} \sum_{i=1}^{k+1} (z_i - \bar{z}_{k+1})^2
\]

(k+1 observations \( z_i \))

Substituting eq. (I) in the last equation we get:
\[
k \cdot s^2_{k+1} = \sum_{i=1}^{k+1} \left\{ (z_i - \bar{z}_k)^2 - \frac{z_{k+1} - \bar{z}_k}{k+1} \right\}
\]

\[
= \sum_{i=1}^{k+1} \left\{ (z_i - \bar{z}_k)^2 - \frac{2}{k+1} (z_i - \bar{z}_k) (z_{k+1} - \bar{z}_k) + \frac{1}{(k+1)^2} (z_{k+1} - \bar{z}_k)^2 \right\}
\]

\[
= (k-1)s^2_k + (z_{k+1} - \bar{z}_k)^2 - \frac{2}{k+1} (z_{k+1} - \bar{z}_k)^2 + \frac{1}{k+1} (z_{k+1} - \bar{z}_k)^2
\]

\[
= (k-1)s^2_k + \frac{k}{k+1} (z_{k+1} - \bar{z}_k)^2
\]

or:
\[
s^2_{k+1} = \frac{k-1}{k} s^2_k + \frac{1}{k+1} (z_{k+1} - \bar{z}_k)^2
\]  

(II)

These formulae have been used in the estimation programs.
Appendix III: Exponential weighting of input and output data

Consider the process \( P \) described by the D.E.:

\[
x_k = \frac{p}{0} b_k u_{k-i} - \frac{q}{1} a_i x_{k-i}
\]

\( q \geq p; \text{this is no restriction). The sequences } \{u_k\}^{N+q} \text{ and } \{x_k\}^{N+q} \text{ are available.}

The following set of equations can be obtained:

\[
\begin{align*}
x_{q+1} &= b_0 u_{q+1} + b_1 u_{q+2} + \ldots + b_p u_{q+p} - a_1 x_{q+2} - \ldots - a_q x_1 \\
x_{q+2} &= b_0 u_{q+2} + b_1 u_{q+3} + \ldots + b_p u_{q+p+1} - a_1 x_{q+3} - \ldots - a_q x_2 \\
&\vdots \\
x_{q+N} &= b_0 u_{q+N} + b_1 u_{q+N-1} + \ldots + b_p u_{q+N-p} - a_1 x_{q+N-1} - \ldots - a_q x_N
\end{align*}
\]

Now, suppose that for some reason, we want to give more weight to the set \( \{u_k, x_k\} \) than to the set \( \{u_{k-1}, x_{k-1}\} \). This can be done by weighting the set of equations (2) e.g. exponentially. Multiply eq. (2(N-1)) with the factor \( a \), eq. (2(N-2)) with a factor \( a^2 \), ..., eq. (2(1)) with a factor \( a^{N-1} \), and define:

\[
\begin{align*}
&\mathbf{w}_T x_N = (x_{q+N}, a x_{q+N-1}, \ldots, a^{N-1} x_{q+1}) \\
&\mathbf{w}_N(u, x) =
\begin{bmatrix}
&\mathbf{u} & \cdots & \mathbf{u} \\
&\mathbf{u} & \cdots & \mathbf{u} \\
&\mathbf{a} u_{q+N-1} & \cdots & \mathbf{a} u_{q+N-p} \\
&\mathbf{a} u_{q+N-1} & \cdots & \mathbf{a} u_{q+N-p-1} \\
&\mathbf{a} x_{q+N-2} & \cdots & \mathbf{a} x_{q+N-1} \\
&\mathbf{a} x_{q+N-2} & \cdots & \mathbf{a} x_{q+N-1} \\
&\mathbf{a} x_{q+N-2} & \cdots & \mathbf{a} x_{q+N-1} \\
&\mathbf{a} x_{q+N-2} & \cdots & \mathbf{a} x_{q+N-1} \\
\end{bmatrix}
\end{align*}
\]

\[
\mathbf{b} = (b_0, b_1, \ldots, b_p, -a_1, -a_2, \ldots, -a_q)
\]

We can then write the following matrix equations, resulting from the manipulations proposed on set (2):

\[
\mathbf{w}_N x_N = \mathbf{w}_N(u, x) \mathbf{b}
\]

and so:

\[
\mathbf{D}_N = \left[ \mathbf{w}_N^T(u, x) \mathbf{w}_N(u, x) \right]^{-1} \mathbf{w}_N^T(u, x) \mathbf{w}_N
\]
Define:

\[ \left( \Omega_{N}^{T}(u,x) \right)^{-1} = P_{N} \]  \hspace{1cm} (5)

and:

\[ \Omega_{N}^{T}(u,x) \omega_{N} = q_{N} \]  \hspace{1cm} (6)

Suppose a new set of observations \((u_{q+1}^{N+1}, x_{q+1}^{N+1})\) becomes available. This produces an extra equation in set (2). All of the other equations, which have been multiplied by the factors mentioned above, will now be multiplied by the factor \(a\).

So eq. (3) now becomes:

\[ \begin{bmatrix} x_{N+1}^{q+1} \\ \vdots \\ x_{N} \end{bmatrix} = \begin{bmatrix} u_{N+1}^{T} \\ \vdots \\ \omega_{N}(u,x) \end{bmatrix} b = \omega_{N+1}^{T}(u,x) \mathbf{D}_{N+1} \]  \hspace{1cm} (7)

and:

\[ \begin{bmatrix} u_{N+1}^{T} \\ \vdots \\ \omega_{N}(u,x) \end{bmatrix} = (u_{N+1}^{q+1}, u_{q+1}, \ldots, u_{q+1}^{N+1}, p, x_{N+1}^{q+1}, \ldots, x_{N+1}) \]

Eq. (7) and eq. (5), (6) give rise to:

\[ (P_{N+1})^{-1} = \omega_{N+1}^{T}(u,x) \omega_{N+1}(u,x) = (u_{N+1}^{T} ; \omega_{N+1}^{T}(u,x)) \begin{bmatrix} u_{N+1}^{T} \\ \vdots \\ \omega_{N}(u,x) \end{bmatrix} \]

or:

\[ (P_{N+1})^{-1} = \frac{u_{N+1}^{T}}{u_{N+1}} + \alpha^{2} (P_{N})^{-1} ; \]

and:

\[ \omega_{N+1}^{T}(u,x) \omega_{N+1} = (u_{N+1}^{T} ; \omega_{N+1}^{T}(u,x)) \begin{bmatrix} x_{N+1}^{q+1} \\ \vdots \\ x_{N} \end{bmatrix} \]

or:

\[ \omega_{N+1}^{T}(u,x) \omega_{N+1} = x_{N+1}^{q+1} + \alpha^{2} \omega_{N+1}^{T}(u,x) x_{N} \]

Using Plackett's formula (Westenberg (19)) on eq. (8) we obtain:

\[ P_{N+1} = \frac{1}{\alpha^{2}} \left[ P_{N} - \frac{u_{N+1}^{T}}{u_{N+1}} \frac{u_{N+1}}{u_{N+1}} P_{N} / (\alpha^{2} u_{N+1}^{T} P_{N} u_{N+1}) \right] \]  \hspace{1cm} (10)
and combining eq. (7) with eq. (9) we obtain:

\[
    b_{N+1} = P_{N+1} \cdot q_{N+1} \\
    \quad = P_{N+1}(\alpha^2 q_{N} + x_{N+q+1} - N+1) \\
\]

This reduces, after some manipulations also to the form

\[
    b_{N+1} = b_{N} - P_{N} \cdot \frac{(u_{N+1}^T b_{N} - x_{N+q+1})}{\alpha^2 + u_{N+1}^T P_{N-1} u_{N+1}} \\
    \quad (11) \\
\]

Remark 1: by substituting \( y_k \) for \( x_k \) and \( \beta \) for \( b \), we obtain the recursive form of the L.S. estimator.

Remark 2: by substituting one for \( \alpha \), we obtain the "usual" recursive formulae given by Westenberg (19).
Appendix IV: Closed-loop parameter estimation

We will apply the definitions and formulae already given in chapter 4, section 4.1.

Consider the process $P$, in a closed-loop situation as given in fig. 1.

![Diagram of a closed-loop system](image)

Fig. 1: Simple closed-loop situation ($q \geq p$).

The D.E. of the closed-loop process is, considering eq. (4.9a):

$$
\frac{q}{0} a_i y_{k-i} = \frac{p}{0} b_i u_{k-i} - \frac{p}{0} b_i y_{k-i} + e_k
$$

and thus with (4.5):

$$
u_k = (u_{k-1}, \ldots, u_{k-p}, y_k, \ldots, y_{k-q})^T b_c + e_k
$$

where:

$$
\begin{align*}
1b_c^T &= (-1b_1, -1b_2, \ldots, -1b_p, 1+1a_0, 1b_1+1a_1, \ldots, 1b_p+1a_p, 1a_{p+1}, \ldots, 1a_q) \\
1\beta_c^T &= (-1\beta_1, -1\beta_2, \ldots, -1\beta_p, 1\alpha_0, 1\alpha_1, \ldots, 1\alpha_p, 1\alpha_{p+1}, \ldots, 1\alpha_q)
\end{align*}
$$

So, using algorithm 1, defined by eq. (4.4), we obtain a consistent estimate of $b_c^T$, by taking $\lim_{N \to \infty} \beta_c^T$.

If we suppose the standard deviation (for finite but large length of observations $N$) to be $\sigma_c^T = (\sigma_\beta_1, \sigma_\beta_2, \ldots, \sigma_\alpha_0, \ldots, \sigma_\alpha_q)$, we can obtain the open-loop parameters, with their upper bound standard deviations in the following way:

$$
\begin{align*}
1b_1 &\rightarrow 1\beta_1 \pm \sigma_\beta_1 \\
1b_p &\rightarrow 1\beta_p \pm \sigma_\beta_p \\
\ldots \\
1a_0 &\rightarrow 1\alpha_0 \pm 1 \pm \sigma_\alpha_0 \\
1a_1 &\rightarrow 1\alpha_1 \cdot 1\beta_1 \pm (\sigma_\alpha_1 + \sigma_\beta_1) \\
\ldots \\
1a_p &\rightarrow 1\alpha_p \cdot 1\beta_p \pm (\sigma_\alpha_p + \sigma_\beta_p) \\
1a_{p+1} &\rightarrow 1\alpha_{p+1} \pm \sigma_\alpha_{p+1} \\
\ldots \\
1a_q &\rightarrow 1\alpha_q \pm \sigma_\alpha_q
\end{align*}
$$
If, however, we write eq. (1) in the following way:

\[ y_k = (u_k, u_{k-1}, \ldots, u_{k-p}, y_{k-1}, y_{k-2}, \ldots, y_{k-q})^{2b \frac{1}{c}} + \epsilon_k \]  

(4)

where now:

\[
2b^T = \left( \frac{2b_0}{1+2b_0}, \ldots, \frac{2b_p}{1+2b_0}, \ldots, -\frac{2a_1+2b_1}{1+2b_0}, \ldots, -\frac{2a_p+2b_p}{a+2b_0}, \ldots, \frac{2a_q}{1+2b_0} \right)
\]

and:

\[
2b_c^T = (2b_0, \ldots, 2b_p, -2a_1, \ldots, -2a_p, \ldots, -2a_q)
\]

the use of algorithm 2 will also give a consistent estimate of \( \frac{b}{c} \).

The difference is obvious. If we want to determine the open-loop parameters of the process from these latter estimates, the standard deviations of these could be larger, than in the previous case.

So, the use of algorithm 1 on closed-loop processes, where we want to know the open-loop parameters, can offer advantages over algorithm 2, even though its convergence is slower than that of the latter.
Literature

1. Woodside, C.M.,
   Estimation of the order of linear systems.
   IFAC Symposium Prague, June 1970 (paper 1.7).

2. Deutsch, R.,
   Estimation Theory.

3. Goldberger, A.S.,
   Econometric Theory.

4. Levin, M.J.,
   Optimum estimation of impulse response in the presence of noise.

   A comparison of some process parameter estimating schemes.

7. Shaw, L.,

8. Evers, V.H.C.M.,
   The use of a generalized model for least-squares estimation of closed-loop systems (in Dutch).

9. Åström, K.J., T. Bohlin & S. Wensmark,
   Automatic construction of linear stochastic dynamic models for stationary industrial processes with random disturbances using operating records.
   IBM Nordic Laboratory Sweden, Technical paper TP 18.150, June 1965.

10. Åström, K.J.,
    Lectures on the identification problem - The least-squares method.

11. Valis, J.,
    Experimental comparison of different methods for numerical identification.
    Lund Institute of Technology - Division of Automatic Control, Report 6914, June 1969.
12. Åström, K.J. & P. Eykhoff,
   System identification, a survey.
   IFAC Symposium Prague, June 1970 (paper 0.1).

13. Clarke, D.W.,
   Generalized least-squares estimation of the parameters of a dynamic model.

14. Eykhoff, P.,
   Process parameter and state estimation.
   IFAC Symposium Prague, 1967 (survey paper 2).

15. Hastings-James, R. & M.W. Sage,
   Recursive generalized least-squares procedure for on-line identification
   of process-parameters.

16. Young, P.C.,
   The use of linear regression and related procedures for the identification
   of dynamic processes.

17. Reiersoll, O.,
   Confluence analysis by means of lag moments and other methods of confluence
   analysis.

18. Wong, K.Y.,
   Estimation of parameters of linear systems using instrumental variable.
   University of California, Berkeley, Ph.D., 1966.

19. Westenberg, J.Z.,
   Some identification schemes for linear processes.

20. Peterka, V. & A. Halousková,
   Tally estimate of Åström-model for stochastic systems.
   IFAC Symposium Prague, June 1970 (paper 2.3).

21. Gustavsson, I.,
   Comparison of different methods for identification of linear models for
   industrial processes.
   IFAC Symposium Prague, June 1970 (paper 11.4).

22. Talmon, J.L.,
TH-Reports:

1) Dijk, J., M. Jeuken and E.J. Maanders

2) Veefkind, A., J.H. Blom and L.H.Th. Rietjens

3) Boom, A.J.W. van den and J.H.A.M. Melis

4) Eykhoff, P., P.J.M. Ophey, J. Severs and J.O.M. Oome

5) Vermij, L. and J.E. Daalder

6) Houben, J.W.M.A. and P. Massee

7) Heuvel, W.M.C. van den and W.F.J. Kersten

8) Vermij, L.

9) Westenberg, J.Z.
10) Koop, H.E.M., J. Dijk and E.J. Maanders

11) Veefkind, A.
NON-EQUILIBRIUM PHENOMENA IN A DISC-SHAPED MAGNETOHYDRODYNAMIC GENERATOR.


13) Teuling, D.J.A.
ELECTRONIC IMAGE MOTION COMPENSATION IN A PORTABLE TELEVISION CAMERA.

14) Lorencin, M.
AUTOMATIC METEOR REFLECTIONS RECORDING EQUIPMENT.