Experimental Comparison of Different Methods for Numerical Identification

Valis, Jaroslav

1969

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

Link to publication

Citation for published version (APA):
Valis, J. (1969). *Experimental Comparison of Different Methods for Numerical Identification*. (Research Reports TFRT-3015). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

EXPERIMENTAL COMPARISON OF DIFFERENT METHODS FOR NUMERICAL IDENTIFICATION

JAROSLAV VALIS

REPORT 6914 JUNE 1969
LUND INSTITUTE OF TECHNOLOGY
DIVISION OF AUTOMATIC CONTROL

EXPERIMENTAL COMPARISON OF DIFFERENT METHODS FOR NUMERICAL IDENTIFICATION †

J. Valis

ABSTRACT

This report illustrates typical properties of some of the most known methods for numerical identification of linear systems (Least squares, generalized least squares, Levin's method) on two test examples of 1st order. A universal test program in FORTRAN IV is included.

---

† This work has been supported by the Swedish Board for Technical Development under Contract 68-336-f

TABLE OF CONTENTS

1. INTRODUCTION

In last years many different methods for identification of linear stochastic single input-single output discrete time systems from experimental data were developed. From the user's point of view a comparison of their efficiency is a matter of natural interest.

In [3, 4] K.J. Åström discusses and compares the most known of them:

a) least squares,

b) generalized least squares,

c) Levin's estimate.

This brief report attempts to show the computional results achieved by using these methods (completed also by results of a „special method").

A computer program, which simulates the run of a given system for given input signal and then computes all these estimates, is also described here.

2. OUTLINE OF PROBLEM

The problems of the experimental identification of a system mentioned above can be simply formulated in the following way:

Given a record of input-output data of the length N

$\{u(i),\ y(i);\ i = 1, \ldots, N\}$

which were measured during the run of a linear stochastic system whose input-output relation can be described by Åström's canonical form

$$y(k) + \sum_{i=1}^{n} a_i\, y(k-i) = \sum_{i=1}^{n} b_i u(k-i) + \varepsilon(k) + \sum_{i=1}^{n} c_i\, \varepsilon(k-i) \tag{2.1}$$

(where $\{\varepsilon(k)\}$ is a sequence of normal independent random variables with zero mean and variance λ),

find „reasonably good" estimates $\hat{a}_i$, $\hat{b}_i$ (and possibly $\hat{c}_i$) of the coefficients a_i, b_i, (c_i).

We will now brief review the principle ideas of methods mentioned above.

a) Least squares is the most known method, which chooses such $\hat{a}_i$, $\hat{b}_i$ as estimates of corresponding a_i, b_i, minimizing the loss function

$$V = \sum_{\kappa=1}^{N} \left(y(k) + \sum_{i=1}^{n} \hat{a}_i\, y(k-i) - \sum_{i=1}^{n} \hat{b}_i\, u(k-i) \right)^2 \tag{2.2}$$

Such values of $\hat{a}_i$ and $\hat{b}_i$ can be easily found by solving the following set of linear algebraic equations (s.c. normal equations)

$$\sum_{k=1}^{N} y(k)\, y(k-p) + \sum_{i=1}^{n} \hat{a}_i \sum_{k=1}^{N} y(k-i)\, y(k-p) - \sum_{i=1}^{n} \hat{b}_i \sum_{k=1}^{N} u(k-i)\, y(k-p) = o$$

$$\sum_{k=1}^{N} y(k)\, u(k-p) + \sum_{i=1}^{n} \hat{a}_i \sum_{k=1}^{N} y(k-i)\, u(k-p) - \sum_{i=1}^{n} \hat{b}_i \sum_{k=1}^{N} u(k-i)\, u(k-p) = o \qquad (2.3)$$

for $p = 1, 2, \ldots, n$

Supposing, that the length N is much greater than the order of the system n we can instead of sums like

$$\sum_{k=1}^{N} w(k-i)\, x(k-p)$$

use the estimates of covariance functions defined by

$$r_{wx}^{N}(\tau) = \frac{1}{N-\tau} \sum_{k=\tau+1}^{N-\tau} w(k)\, x(k-\tau) \quad \text{where } \tau = p - i \qquad (2.4)$$

Using this, we can rewrite (2.3) as (for n = 4)

$$\begin{bmatrix}
r_{yy}^N(0) & r_{uy}^N(0) & r_{yy}^N(1) & r_{uy}^N(1) & r_{uy}^N(2) & r_{uy}^N(2) & r_{yy}^N(3) & r_{uy}^N(3) \\
r_{yu}^N(0) & r_{uu}^N(0) & r_{yu}^N(1) & r_{uu}^N(1) & r_{yu}^N(2) & r_{uu}^N(2) & r_{yu}^N(3) & r_{uu}^N(3) \\
r_{yy}^N(1) & r_{uy}^N(1) & r_{yy}^N(0) & r_{uy}^N(0) & r_{yy}^N(1) & r_{uy}^N(1) & r_{yy}^N(2) & r_{uy}^N(2) \\
r_{yu}^N(1) & r_{uu}^N(1) & r_{yu}^N(0) & r_{uu}^N(0) & r_{yu}^N(1) & r_{uu}^N(1) & r_{yu}^N(2) & r_{uu}(2) \\
r_{yy}^N(2) & r_{uy}^N(2) & r_{yy}^N(1) & r_{uy}^N(1) & r_{yy}^N(0) & r_{uy}^N(0) & r_{yy}^N(1) & r_{uy}^N(1) \\
r_{yu}^N(2) & r_{uu}^N(2) & r_{yu}^N(1) & r_{uu}^N(1) & r_{yu}^N(0) & r_{uu}^N(0) & r_{yu}^N(1) & r_{uu}^N(1) \\
r_{yy}^N(3) & r_{uy}^N(3) & r_{yy}^N(2) & r_{uy}(2) & r_{yy}^N(1) & r_{uy}(1) & r_{yy}^N(0) & r_{uy}^N(0) \\
r_{yu}^N(3) & r_{uu}^N(3) & r_{yu}^N(2) & r_{uu}(2) & r_{yu}^N(1) & r_{uu}^N(1) & r_{yu}^N(0) & r_{uu}^N(0)
\end{bmatrix}
*
\begin{bmatrix} \hat{a}_4 \\ -\hat{b}_4 \\ \hat{a}_3 \\ -\hat{b}_3 \\ \hat{a}_2 \\ -\hat{b}_2 \\ \hat{a}_1 \\ -\hat{b}_1 \end{bmatrix}
=
\begin{bmatrix} -r_{yy}^N(4) \\ -r_{yu}^N(4) \\ -r_{yy}^N(3) \\ -r_{yu}^N(3) \\ -r_{yy}^N(2) \\ -r_{yu}^N(2) \\ -r_{yy}^N(1) \\ -r_{yu}(1) \end{bmatrix}
\qquad (2.5)$$

1st order

2nd order

3rd order

4th order

This form is very suitable when we compute the estimates for different orders. (The square matrix in 2.5 is in the program called XTX resp. XTXP.)

The minimal value of the loss function V (2.2) for those $\hat{a}$, $\hat{b}$ is then

$$V = N \cdot \left(r_{yy}(0) + \sum_{i=1}^{n} \hat{a}_i \, r_{yy}(i) - \sum_{i=1}^{n} \hat{b}_i \, r_{yu}(i) \right) \tag{2.6}$$

It is also known, that the method of least squares gives generally biased estimates when used for identification of systems of the type (2.1). This bias can be reduced if we use a model of higher order than the actual one and then find and exclude the (approximate) common factor from $\hat{B}(z^{-1})$ and $\hat{A}(z^{-1})$.

For more detail information see [3,4].

b) Generalized least squares method offers another possibility to remove bias of least squares estimates. From the practical point of view it means nothing else than the use of suitable filters to filter both the input and output data and then to use the ordinary least squares method. Filters must be chosen so that the residuals of L.S. are white.

More exact treatment of this method see also in [1, 3, 4].

c) Levin's method [2] utilizes the fact, that the values $\hat{a}_i$ and $\hat{b}_i$, which minimizes the loss function

$$V_L = \frac{\sum_{k=1}^{N} \left(y(k) + \sum_{i=1}^{n} \hat{a}_i \, y(k-i) - \sum_{i=1}^{n} \hat{b}_i \, u(k-i) \right)^2}{1 + \sum_{i=1}^{n} \hat{a}_i^2} \tag{2.7}$$

are asymptomatically unbiased estimates of a_i and b_i (but only in the case when $a_i = c_i$ in (2.1)).In the general case is the minimization of (2.5) not an easy matter because of nonlinearity of the equations

$$\frac{\partial V_L}{\partial \hat{a}_i} = 0 \text{ for } i=1, \ldots, n$$

and therefore we restrict ourselves to the 1st order case only. The loss function is then

$$V_{L1} = \frac{\sum_{k=1}^{N} \left(y(k) + \hat{a}_1 \, y(k-1) - \hat{b}_1 \, u(k-1) \right)^2}{1 + \hat{a}_1^2} \tag{2.8}$$

The necessary conditions for the minimum are

$$\frac{\partial V_{L1}}{\partial \hat{a}_1} = 0 \ ; \quad \frac{\partial V_{L1}}{\partial \hat{b}_1} = 0$$

$$\frac{\partial V_{L1}}{\partial \hat{a}_1} = 2 \cdot \frac{(1+\hat{a}_i^{\,2}) \cdot \sum\limits_{k=1}^{N} \Big(y(k)+\hat{a}_1\, y(k-1)-\hat{b}_1\, u(k-1)\Big)\, y(k-1) - \hat{a}_i \sum\limits_{k=1}^{N} \Big(y(k)+\hat{a}_1\, y(k-1)-\hat{b}_1\, u(k-1)\Big)^2}{(1+\hat{a}_1^{\,2})^2} = 0 \tag{2.9a}$$

$$\frac{\partial V_{L1}}{\partial \hat{b}_1} = -2 \cdot \frac{\sum\limits_{k=1}^{N} \Big(y(k) + \hat{a}_1\, y(k-1) - \hat{b}_1\, u(k-1)\Big)\, u(k-1)}{1+\hat{a}_1^{\,2}} = 0 \tag{2.9b}$$

When we now use the approximate values of the covariance functions $r_{yy}^{N}(p)$ and $r_{uu}^{N}(p)$ and $r_{yu}^{N}(p)$ like in (2.4), we obtain after straightforward computation the solution of (2.9a, b) as follows

$$\hat{a}_1^{\,2} + \Omega \hat{a}_1 - 1 = 0 \tag{2.10a}$$

$$\text{where } \Omega = \frac{\left(r_{yu}^{N}(0)\right)^2 - \left(r_{yu}^{N}(1)\right)^2}{r_{yy}^{N}(1) \cdot r_{uu}^{N}(0) - r_{yu}^{N}(1) \cdot r_{yu}^{N}(0)} \tag{2.10b}$$

i.e

$$\hat{a}_1 = \frac{-\Omega}{2} - \sqrt{\left(\frac{\Omega}{2}\right)^2 + 1} \tag{2.11a}$$

and

$$\hat{b}_1 = \frac{r_{yu}^{N}(1) + \hat{a}_1 \cdot r_{yu}^{N}(0)}{r_{uu}^{N}(0)} \tag{2.11b}$$

a) <u>„Special estimates"</u>

are based on the following property of the model (2.1):

Let us denote

$$\delta(k) = y(k) + \sum_{i=1}^{n} a_i\, y(k-i) - \sum_{i=1}^{n} b_i\, u(k-i) = \varepsilon(k) + \sum_{i=1}^{n} c_i\, \varepsilon(k-i) \tag{2.12}$$

and take the expectation

$$E\{\delta(k)\, y(k-p)\} = E\Big\{\Big(\varepsilon(k) + \sum_{i=1}^{n} c_i\, \varepsilon(k-i)\Big)\, y(k-p)\Big\} =$$

$$= r_{\varepsilon y}(p) + \sum_{i=1}^{n} c_i\, r_{\varepsilon y}(p-i)$$

But $r_{\varepsilon y}(p) = 0$ for $p \geq 1$ because of the independence of $\varepsilon(k + p)$ on $y(k)$.

Hence $E\{\delta(k)\, y(k - p)\} = 0$ if $p > n$ or

$$r_{yy}(p) + \sum_{i=1}^{n} a_i\, r_{yy}(p - i) - \sum_{i=1}^{n} b_i\, r_{uy}(p - i) = 0 \qquad (2.13)$$

$$\text{for } p > n$$

We can thus obtain the estimates $\hat{a}_i$ and $\hat{b}_i$ using (2.13) by solving the following set of linear equations.

$$\left[\begin{array}{cccc|cccc} r^N_{yy}(n) & r^N_{yy}(n-1) & \cdots & r^N_{yy}(1) & r^N_{uy}(n) & r^N_{uy}(n-1) & \cdots & r^N_{uy}(1) \\ r^N_{yy}(n+1) & r^N_{yy}(n) & \cdots & r^N_{yy}(2) & r^N_{uy}(n+1) & r^N_{uy}(n) & \cdots & r^N_{uy}(2) \\ : & : & : & : & : & : & \cdots & : \\ r^N_{yy}(3n-1) & r^N_{yy}(3n-2) & \cdots & r^N_{yy}(2n) & r^N_{uy}(3n-1) & r^N_{uy}(3n-2) & \cdots & r^N_{uy}(2n) \end{array}\right] * \left[\begin{array}{c} \hat{a}_1 \\ \hat{a}_2 \\ : \\ \hat{a}_n \\ \hline -\hat{b}_1 \\ -\hat{b}_2 \\ : \\ -\hat{b}_n \end{array}\right] = \left[\begin{array}{c} -r_{yy}(n+1) \\ -r_{yy}(n+2) \\ : \\ -r_{yy}(3n) \end{array}\right] \qquad (2.14)$$

Unfortunately, this set of equations (2.14) has a peculiar property: When the input signal is a white sequence, the right hand part of the square matrix contains only zeroes (or very small values) because all $r_{uy}(p) = 0$ for $p > 0$.

3. PROGRAM SIMULATE

Identification of a system from simulated data by means of all methods described in previous sections is carried out automatically by a FORTRAN IV program called SIMULATE.

This program

1) generates an input sequence U(K), K = 1, NN using subroutine RANSS which generates pseudorandom independent normal (0,1) numbers - this sequence may be „coloured" by a „built in" filter if desired.

2) using subroutine SIMDATA simulates the run of a given linear stochastic system, whose output Y(K) is given by (using notation in the program)

$$Y(K) = \sum_{I=1}^{N} B(I)*U(K-I) - \sum_{I=1}^{N} A(I)*Y(K-I) +$$

$$+ LAMBDA*\left(E(K) + \sum_{I=1}^{N} C(I)*E(K-I)\right)$$

where sequence of pseudorandom normal numbers (0,1) is generated by subroutine RANSS (IT, E(K)) with starting value IT = 3.

3) using subroutine COVARF computes covariance functions of input/output data RUU(TAU), RUY(TAU), RYU(TAU), RYY(TAU) for TAU = 0, TAUMAX.

4) using covariance functions (from 3.1) computes least squares estimates of orders N5 = 1, NMAX (in increasing order) and the respective values of loss functions and F-test quantities Fn/n-1.

5) computes „special estimates" for order NS = 1,4.

6) computes the 1st order Levin's estimates.

7) If wanted, the input/output data are filtered by a „moving average" filter FF(I)

$$\tilde{U}(K) = \sum_{I=1}^{NF} FF(I)*U(K-I+1)$$

$$\tilde{Y}(K) = \sum_{I=1}^{NF} FF(I)*Y(K-I+1)$$

and items 2), 3), 4), 5), 6) will be repeated for this filtered data.

Program SIMULATE requires subroutines

RANSS	(pseudorandom number generator)
SIMDATA	(simulation of given system)
COVARF	(computation of covariance functions)
MIART	(for solving of systems of linear algebraic equations in 4 and 5)

Complete list of this program and results for one example follows in Appendix.

4. RESULTS

We run the program SIMULATE for 1st order system

$$y(k) = b_1\ u(k-1) - a_1\ y(k-1) + \lambda\left(e(k) + c_1\ e(k-1)\right)$$

with parameters

b_1 (= B(1)) = 1.0

a_1 (= A(1)) = -0.5

c_1 (= C(1)) = -0.7

λ (= LAMBDA)= 1.0

The input sequence U of length NN = 500 was generated using first order autoregressive filter to obtain „coloured" input signal

$$U(K) = U^1(K) + 0.5 * U(K-I)$$

where $U^1(K)$ was the original input sequence generated by RANSS (as described in section 3).

Computed covariance functions of input/output signal are given in Tab. 1.

The least squares estimates of different order are summarized in Tab. 2.

The F-test shows that we should admit that the system is of the 5th order, but we can confirm ourselves that the polynomials $B(z^{-1})$ and $A(z^{-1})$ have an approximate common factor which is just equal to $z^{-1}B(z^{-1})$:

$$\frac{1.000 + 0.203\, z^{-1} + 0.107\, z^{-2} + 0.084\, z^{-3} + 0.032\, z^{-4} - 0.07\, z^{-5}}{0.960 + 0.739\, z^{-1} + 0.423\, z^{-2} + 0.256\, z^{-3} + 0.220\, z^{-4}} =$$

$$= 1.04\, z^{-1}(1 - 0.584\, z^{-1})$$

The rest is only $z^{-2}(0.104 + 0.067\, z^{-1} - 0.046\, z^{-2} + 0.059\, z^{-3})$

This means that having excluded the common factor we obtain 1st order estimates

$$\hat{b}_1 = 0.960,\ \hat{a}_1 = -0.584$$

which are better than those obtained directly.

According to [4] we may improve the estimates using the common factor (i.e. the $B(z^{-1})$ polynomial for n = 5) as a moving average filter for both input and output data (that means $FF(i) = \hat{b}_i$, i = 1, 2, ..., 5).

$$\tilde{u}_{(k)} = \sum_{i=1}^{5} \hat{b}_i \cdot u(k - i + 1)$$

$$\tilde{y}_{(k)} = \sum_{i=1}^{5} \hat{b}_i\, y(k - i + 1)$$

and then apply the estimation procedures on the filtered input/output data $\{\tilde{u}(k), \tilde{y}(k)\}$.

The results of these trials are summarized in tables Tab. 3 and Tab. 4.

We observe that the results are considerably better estimates than in the foregoing case. Particularly, we must point out that the F-test quantity shows, that the system is of the first order as no significant decrease of the loss function for higher order models are obtained. Note also the small values of estimates $\hat{a}_i$, $\hat{b}_i$ for $i > 1$ for all estimated models. This fact confirms, that the model of the 1st order is entirely appropriate. Table Tab. 5 summarizes results obtained by means of all tested methods.

As noted before, the Levin's estimate should give unbiased estimates only in the case when $a_i = c_i$. To illustrate this, we tried another example when the simulated system was the same as above except for $c_1 = -0.5$ ($= a_1$). The results are summarized in Tab. 6, 7, 8. We can see, that the Levin's estimates are better than the least squares ones only in the case of unfiltered data.

In this example we have to notice another interesting property of the obtained least squares models of higher order: The estimates of coefficients $\hat{a}_i$ for $i \geq 1$ have very small values, in other words we get the models in the form:

$$y_{(k)} = \sum_{i=1}^{n_\ell} \hat{b}_i \, u(k - i)$$

which is nothing else than the description of the system by means of its approximate impulse response.

BIBLIOGRAPHY:

[1] Clarke, D.W.: Generalized-Least-Squares Estimation of the Parameters of a Dynamic Model, Preprints of the IFAC Symposium „Identification in Automatic Control Systems", in Prague, Part 1, Paper 3.17.

[2] Levin, M.J.: Estimation of System Pulse Transfer Function in the Presence of Noise, 4th JACC Minneapolis, June 1963, Preprints, p. 452 - 458.

[3] Åström, K.J.: Lectures on the Identification Problem - The Least Squares Method, Report 6806, Inst. f. regleringsteknik, LTH, Sept., 1968.

[4] Åström, K.J.: Lectures on the Identification Problem - Correlated Residuals, Report LTH, Oct., 1968

Tab. 1

Estimate of Covariance Functions of Input/Output Signals.

τ	$r_{uu}(\tau)$	$r_{uy}(\tau)$	$r_{yu}(\tau)$	$r_{yy}(\tau)$
0	1.2821	1.0508	1.0508	3.5041
1	0.6519	0.5957	1.6156	2.1699
2	0.3463	0.2862	1.4832	1.6639
3	0.1825	0.0727	1.0381	1.0804
4	0.0676	-0.0198	0.6808	0.6163
5	0.0153	-0.0635	0.4423	0.3334
6	0.0073	0.0061	0.2029	0.0465
7	0.0292	-0.0401	0.1562	0.0774
8	0.0323	-0.0165	0.1316	-0.0461
9	-0.0576	-0.0841	0.0587	-0.0408
10	-0.0414	-0.1499	-0.0063	-0.1503
11	-0.0998	-0.1758	-0.1220	-0.0963
12	-0.1161	-0.0478	-0.1428	-0.2946
13	-0.0249	-0.0836	-0.0929	-0.2165
14	0.0111	-0.0555	-0.1604	-0.2308
15	0.0471	-0.1234	-0.0603	-0.2194
16	-0.0561	-0.1014	-0.0307	-0.1656
17	-0.0684	-0.1470	-0.0558	-0.1967
18	-0.0823	-0.2225	-0.0701	-0.3354
19	-0.1166	-0.1499	-0.1929	-0.2925

Tab. 2

Least Squares Estimates from N = 500 Input/Output Data Pairs.

True values: $a_1 = -0.5$, $b_1 = 1.0$, $c_1 = -0.7$, $\lambda = 1.0$

n	a_0	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{a}_4$	$\hat{a}_5$	$\hat{a}_6$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$	$\hat{b}_5$	$\hat{b}_6$	Vn	Fn/n-1
1	1.000	-0.320						0.998						1.1976+000	
2	1.000	0.027	-0.157					0.902	0.603					9.4841-001	65.15530
3	1.000	0.137	-0.016	-0.089				0.935	0.651	0.291				8.9915-001	13.53103
4	1.000	0.175	0.059	-0.001	-0.066			0.951	0.700	0.348	0.183			8.7874-001	5.71397
5	1.000	0.203	0.107	0.084	0.032	-0.070		0.960	0.739	0.423	0.256	0.220		8.5245-001	7.55693
6	1.000	0.208	0.112	0.092	0.046	-0.055	-0.009	0.961	0.745	0.433	0.270	0.232	0.038	8.5179-001	0.18692

Tab. 3

Estimate of Covariance Functions of Filtered I/O Signals.

τ	$r_{uu}(\tau)$	$r_{uy}(\tau)$	$r_{yu}(\tau)$	$r_{yy}(\tau)$
0	4.4552	6.2753	6.2753	14.7263
1	3.8688	4.6035	7.4766	13.4108
2	2.8828	2.9629	7.4771	11.0039
3	1.9491	1.5835	6.4779	8.2533
4	1.1592	0.6296	5.0863	5.5817
5	0.5467	0.1177	3.6584	3.2721
6	0.2534	-0.1088	2.3793	1.6988
7	0.1002	-0.2722	1.4826	0.7238
8	-0.0421	-0.3853	0.8646	0.0374
9	-0.2055	-0.5400	0.3653	-0.4172
10	-0.3111	-0.6749	-0.0719	-0.7816
11	-0.3845	-0.7207	-0.4207	-1.0352
12	-0.3812	-0.6404	-0.6145	-1.3156
13	-0.2701	-0.6256	-0.6763	-1.4175
14	-0.1662	-0.6555	-0.6812	-1.4806
15	-0.1597	-0.7391	-0.5940	-1.4901
16	-0.2745	-0.8001	-0.5323	-1.5046
17	-0.3855	-0.8617	-0.5707	-1.5721
18	-0.4670	-0.8211	-0.6759	-1.6750
19	-0.4958	-0.5550	-0.8149	-1.5753

Tab. 4

Generalized Least Squares Estimates from N = 500 Input/Output Data Pairs

B-Polynomial from Tab. 2 for N = 5 is used as filter to original I/O data

True values: $a_1 = -0.5$, $b_1 = 1.0$, $c_1 = -0.7$, $\lambda = 1.0$

n	a_0	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{a}_4$	$\hat{a}_5$	$\hat{a}_6$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$	$\hat{b}_5$	$\hat{b}_6$	Vn	Fn/n-1
1	1.000	-0.489						0.939						7.7063-001	
2	1.000	-0.574	0.062					0.964	-0.035					7.6356-001	2.29591
3	1.000	-0.579	0.122	-0.042				0.970	-0.030	0.027				7.6036-001	1.03877
4	1.000	-0.578	0.123	-0.000	-0.014			0.966	-0.008	-0.015	0.083			7.5778-001	0.83710
5	1.000	-0.575	0.118	0.009	-0.030	0.029		0.970	-0.016	0.005	0.028	0.068		7.5418-001	1.16922
6	1.000	-0.574	0.125	0.010	-0.034	0.0166	-0.066	0.967	-0.016	0.009	0.077	-0.005	0.184	7.3695-001	5.70463

Tab. 5

Comparison of Different Types of Estimates

True values $a_1 = -0.5$, $b_1 = 1.0$, $c_1 = -0.7$, $\lambda = 1$

Type of Estimate	a_0	$\hat{a}_1$	$\hat{b}_1$	I/O Data
Least Squares	1.000	-0.320	0.998	Original
„Special"	1.000	-0.452	1.146	Original
Levin's	1.000	-0.523	0.831	Original
Gen. L.S.	1.000	-0.489	0.939	Filtered
„Special	1.000	-0.494	0.953	Filtered
Levin's	1.000	-0.5455	0.9098	Filtered

Tab. 6

Least Squares Estimates from N = 500 Input/Output Data Pairs

True values: $a_1 = -0.5$, $b_1 = 1.0$, $c_1 = -0.5$, $\lambda = 1.0$

n	a_0	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{a}_4$	$\hat{a}_5$	$\hat{a}_6$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$	$\hat{b}_5$	$\hat{b}_6$	Vn	Fn/n-1
1	1.000	-0.369						0.992						1.0315+000	
2	1.000	-0.055	-0.145					0.934	0.500					8.7334-001	44.90221
3	1.000	0.004	-0.052	-0.057				0.953	0.531	0.177				8.5586-001	5.04401
4	1.000	0.016	-0.021	-0.009	-0.037			0.958	0.549	0.201	0.083			8.5115-001	1.36033
5	1.000	0.024	-0.005	0.033	0.027	-0.039		0.959	0.562	0.230	0.113	0.131		8.4231-001	2.57192
6	1.000	0.021	-0.007	0.029	0.018	-0.053	0.013	0.959	0.559	0.225	0.106	0.123	-0.024	8.4185-001	0.13419

Tab.7

Generalized Least Squares Estimates from N = 500 Input/Output Data Pairs.

B-Polynomial from Tab. 6 for n = 5 is used as filter to original I/O data.

True values: $a_1 = -0.5$, $b_1 = 1.0$, $c_1 = -0.5$, $\lambda = 1.0$

n	a_0	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{a}_4$	$\hat{a}_5$	$\hat{a}_6$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{b}_4$	$\hat{b}_5$	$\hat{b}_6$	Vn	Fn/n-1
1	1.000	-0.494						0.986						7.6376-001	
2	1.000	-0.574	0.063					0.962	-0.030					7.5696-001	2.22834
3	1.000	-0.578	0.098	-0.028				0.963	-0.021	0.005				7.5561-001	0.44093
4	1.000	-0.577	0.097	-0.006	-0.011			0.962	-0.015	-0.004	0.029			7.5517-001	0.14520
5	1.000	-0.578	0.094	0.000	-0.050	0.036		0.963	-0.020	-0.002	0.010	0.007		7.5298-001	0.71337
6	1.000	-0.579	0.097	0.000	-0.052	0.094	-0.029	0.962	-0.023	-0.002	0.025	-0.017	0.076	7.4988-001	1.00572

Tab. 8

Comparison of Different Types of Estimates

True values: $a_1 = c_1 = -0.5$, $b_1 = 1.0$, $\lambda = 1.0$

Type of Estimate	a_0	$\hat{a}_1$	$\hat{b}_1$	I/O Data
Least Squares	1.000	-0.369	0.992	Original
„Special"	1.000	-0.441	1.165	Original
Levin's	1.000	-0.545	0.843	Original
Gen. L.S.	1.000	-0.494	0.986	Filtered
„Special"	1.000	-0.488	0.965	Filtered
Levin's	1.000	-0.567	0.887	Filtered

Appendix 1.

Complete List of the Program SIMULATE

Description in section 3 of this report.

Subroutine required: RANSS
SIMDATA
COVARF
MIART

```
      PROGRAM SIMULATE
      DIMENSION U(500),Y(500),E(500),A(10),B(10),C(10),RUU(20),RUY(20)
     1,RYU(20),RYY(20)
      DIMENSION XTX(20,20),CI(20),CJ(20),HELP1(10),HELP2(10)
      DIMENSION XTXP(20,20)
      DIMENSION FF(10)
      DIMENSION UF(10),YF(10)
      DIMENSION LSF(10)
       REAL LSF
      REAL LAMBDA
      INTEGER TAUMAX
C
C  DEFINE THE SIMULATAD SYSTEM
C
C     NN  IS THE LENGTH OF THE INPUT/OUPUT DATA SEQUENCES
      NN=500
C
C     LAMBDA IS THE STANDARD DEVIATION OF THE NOISE
      LAMBDA = 1.
C
C     N  IS THE ORDER OF THE SIMULATED SYSTEM
C     A,B,C  ARE THE VECTORS OF COEFFICIENTS
      N=1 $ A(1)=-0.5 $ B(1)=1. $ C(1)=-0.7
C
C     TAUMAX IS MAX. ARG. FOR WHICH COVAR. FUNCT. ARE COMPUTED
      TAUMAX=20
C
C     IC  IF  IC.LT. 0 THE INPUT SEQUENCE IS WHITE,OTHERWISE COLORED
      IC=1
C
C     NMAX IS THE MAXIMAL ESTIMATED ORDER USING LEAST SQUARES
      NMAX=6
C
C  PRINT OUT THE PARAMETERS OF THE SIMULATED SYSTEM
      PRINT 100,N
  100 FORMAT(//*LINEAR DISCRETE TIME DYNAMIC SYSTEM SIMULATION*/*
     1ORDER N= *I3)
      PRINT 101, (A(I),I=1,N)
  101 FORMAT(*AA=  1.000*10F7.3)
      PRINT 102,(B(I),I=1,N)
  102 FORMAT(*BB=        *10F7.3)
      PRINT 103,(C(I),I=1,N)
  103 FORMAT(*CC=  1.000*10F7.3)
      PRINT 104,LAMBDA
  104 FORMAT(*LAMBDA=*F7.3)
C
C  GENERATE THE INPUT SEQUENCE
      IT1=1
      DO 1 I=1,NN
    1 CALL RANSS(IT1,U(I))
C
C  IF IC IS NOTNEGATIV THE INPUT NOISE IS COLORED
      IF(IC)29,30,30
   30 DO 28 I=2,NN
   28 U(I)=0.5*U(I-1)+U(I)
   29 CONTINUE
```

```
C
C  SIMULATE THE RUN OF THE SYSTEM AND COMPUTE THE COVARIANCE FUNCTIONS
      CALL SIMDATA(N,A,B,C,LAMBDA,NN,U,Y,E)
C
C  SET FLAG FOR FILTER
  509 IFL = 1
      NF=5
  508 CONTINUE
      CALL COVARF(NN,U,Y,RUU,RUY,RYU,RYY,TAUMAX)
C
C  PRINT OUT INPUT/OUTPUT DATA
      PRINT 105
  105 FORMAT(//*INPUT - OUTPUT DATA*)
      DO 2 K=1,100
      K1=K+100$K2=K+200$K3=K+300$K4=K+400
    2 PRINT 106,K,U(K),Y(K),K1,U(K1),Y(K1),K2,U(K2),Y(K2),K3,U(K3),Y(K3)
     1,K4,U(K4),Y(K4)
  106 FORMAT(I6,2F7.3,I6,2F7.3,I6,2F7.3,I6,2F7.3,I6,2F7.3)
C
C  PRINT OUT THE COVARIANCE FUNCTIONS
      PRINT 107
  107 FORMAT(//*COVARINCE FUNCTIONS*/
     1*TAU   RUU        RUY       RYU       RYY*)
      DO 3 K=1,TAUMAX
      K1=K-1
    3 PRINT 108,K1,RUU(K),RUY(K),RYU(K),RYY(K)
  108 FORMAT(I4,4F8.4)
C
C  FILL THE XTX MATRIX
      DO 4 I=1,NMAX
      DO 5 J=I,NMAX
      I2=2*I  $  J2=2*J
      XTX(I2-1,J2-1)= XTX(J2-1,I2-1)= RYY(J-I+1)
      XTX(I2,J2)= XTX(J2,I2)=RUU(J-I+1)
      XTX(I2-1,J2)= XTX(J2,I2-1)= RUY(J-I+1)
    5 XTX(I2,J2-1)= XTX(J2-1,I2) = RYU(J-I+1)
      XTX(I2-1,2*NMAX+1) = RYY(NMAX-I+2)
    4 XTX(I2,2*NMAX+1) = RYU(NMAX-I+2)
C
C  COMPUTE AND PRINT OUT THE LEAST SQUARES ESIMATES
      PRINT 109
  109 FORMAT(///*LEAST SQUARES ESTIMATES*)
      DO 8 N5=1,NMAX
      NNN=2*N5
      N8=NNN+1
      DO 24 I10=1,NNN
      DO 25 I11=I10,NNN
   25 XTXP(I10,I11)=XTXP(I11,I10)=XTX(I10,I11)
   24 XTXP(I10,N8)=XTX(I10,N8)
      NS=1 $ INVRT=0 $ IRANK=0 $ IPS=0 $ DELTA=1E-6
      CALL MIART(XTXP,NNN,NNN,NS,INVRT,IRANK,IPS,DELTA,20,20,CI,CJ)
      PRINT 110,N5
  110 FORMAT(//*ORDER OF THE MODEL N=*I3)
      IF(INVRT)6,7,7
    6 PRINT 111
  111 FORMAT(/*MATRIX XTX SINGULAR - NO ESTIMATE*)
```

```
      GO TO 8
    7 DO 88 I=1,N5
      IP=N5-I+1
      HELP1(I)=XTXP(2*IP,NNN+1)
      HELP2(I)=-XTXP(2*IP-1,NNN+1)
   88 CONTINUE
      PRINT 113,(HELP2(I4),I4=1,N5)
  113 FORMAT(/*AA  1.000*10F7.3)
      PRINT 112,(HELP1(I4),I4=1,N5)
  112 FORMAT(/*BB          *10F7.3)
C
C  SET FILTER
      IF(N5-NF) 507,505,507
  505 DO 506 I4=1,N5
  506 FF(I4)=HELP1(I4)
  507 CONTINUE
C
C  COMPUTE THE LOSS FUNCTION
      VAB=RYY(1)
      DO 26 I4=1,N5
      I5=I4+1
   26 VAB=VAB+HELP2(I4)*RYY(I5)-HELP1(I4)*RYU(I5)
      PRINT 119,VAB
  119 FORMAT(*LOSS FUNCTION  V=*E12.4)
C
C  F - TEST OF ORDER
      LSF(N5)=VAB
      IF(N5-1) 702,702,700
  700 F=(LSF(N5-1)-LSF(N5))*(NN-2*N5)/(2.*LSF(N5))
      N51=N5-1
      PRINT 701,N5,N51,F
  701 FORMAT(/*F - TEST(*I2,*/*I2,*) =*F9.5)
  702 CONTINUE
    8 CONTINUE
C
C  COMPUTE THE *SPECIAL* ESTIMATES
      DO 11 NS=1,4
      PRINT 115,NS
  115 FORMAT(///*SPECIAL ESTIMATE OF ORDER NS=*I3)
      NNN=2*NS
       DO 21 IR = 1,NNN
      DO 22 I = 1,NS
      I1=NS+IR-I+1  $  IN=I+NS
      XTX(IR,I) = RYY(I1)
   22 XTX(IR,IN) = -RUY(I1)
      I2=NS+IR+1
   21 XTX(IR,NNN+1) = -RYY(I2)
      INS=1 $ INVRT=0 $ IRANK=0 $ IPS=0 $ DELTA=1E-6
      CALL MIART(XTX,NNN,NNN,INS,INVRT,IRANK,IPS,DELTA,20,20,CI,CJ)
      IF(INVRT)9,10,10
C
C  PRINT OUT THE SPECIAL ESTIMATES
    9 PRINT 114
  114 FORMAT(//*MATRIX XTXS IS SINGULAR - NO ESTIMATE*)
      GO TO 11
   10 I2=NNN+1
```

```
      PRINT 113,(XTX(I4,I2),I4=1,NS)
      N3=NS+1
      PRINT 112,(XTX(I4,I2),I4=N3,NNN)
   11 CONTINUE
C
C  TRY LEVINS ESTIMATE OF THE FIRST ORDER
      GAMA=RYY(2)*RUU(1)-RYU(2)*RYU(1)
      IF(ABS(GAMA).LE. 1E-6) GO TO 13
      OMEGA=(RYU(1)**2-RYU(2)**2)/(GAMA*2)
      DISCR=OMEGA**2+1.
      IF(DISCR) 13,12,12
   12 A1=-OMEGA-SQRT(DISCR)
      B1=(RYU(2)+A1*RYU(1))/RUU(1)
      PRINT 118,A1,B1
  118 FORMAT(////*LEVINS ESTIMATE OF THE FIRST ORDER  A(1)=*
     1F7.4*  B(1)=*F7.4)
      GO TO 14
C
C  LEVINS ESTIMATE CAN NOT BE COMPUTED
   13 PRINT 116
  116 FORMAT(////*THERE IS SOMETHING WRONG WITH COVARIANCE FUNCTIONS-*
     1*LEVINS ESTIMATE CANNOT BE COMPUTED*)
   14 CONTINUE
C
C  FILTER DATA IF FLAG IS SET
      IF(IFL) 510,510,500
  500 IFL=-2
      DO 511 I=1,NF
  511 UF(I)=YF(I)=0.
      DO 501 K=1,NN
      UU=YY=0.
      DO 512 I=2,NF
      UF(I-1)=UF(I)
  512 YF(I-1)=YF(I)
      UF(NF)=U(K)
      YF(NF)=Y(K)
      DO 502 I=1,NF
      J=NF-I+1
      UU=UU+FF(I)*UF(J)
  502 YY=YY+FF(I)*YF(J)
  503 U(K)=UU
  501 Y(K)=YY
      PRINT 600,NF
  600 FORMAT(////*THE BB-POLYNOMIAL FOR N=*I3*  IS USED AS FILTER TO *
     1*ORIGINAL I/O DATA*///*FILTERED*)
      GO TO 508
  510 CONTINUE
      CALL EXIT
      END
```

```
      SUBROUTINE SIMDATA(N,A,B,C,LAMBDA,NN,U,Y,E)
      DIMENSION A(N),B(N),C(N),U(NN),Y(NN),E(NN)
      REAL LAMBDA
      IT2=3
      DO 1 I=1,NN
      CALL RANSS(IT2,EE)
      E(I)=EE*LAMBDA
    1 CONTINUE
      DO 2 K=1,NN
      EE=E(K)
      DO 4 I=1,N
      J=K-I
      IF(J)2,2,4
    4 EE=EE+B(I)*U(J)-A(I)*Y(J)+C(I)*E(J)
    2 Y(K)=EE
C   END OF SIMULATION
      RETURN
      END
```