Real-Time Identification

Part I

Wieslander, Johan

1969

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

REAL TIME IDENTIFICATION - PART I

JOHAN WIESLANDER

REPORT 6908 NOVEMBER 1969
LUND INSTITUTE OF TECHNOLOGY
DIVISION OF AUTOMATIC CONTROL

REAL TIME IDENTIFICATION - Part I [†]

Johan Wieslander

ABSTRACT

In this paper the problem of identifying a time varying system, i.e. tracking the system parameters, is considered. The model structure in this case is the one usually used in Least Squares methods. Three different real time identification schemes are introduced. Similarities and differences between the three methods are discussed. The used system model structure and its limitations are considered and a possible technique to overcome the demonstrated difficulties is outlined. A practical problem concerning bias in the measured signals is recognized and solved.

Finally, three digitally simulated examples are included. They are intended to demonstrate the ability of the algorithms to track rapidly varying parameters. The last one shows an application to non-linear systems.

[†] This work has been partially supported by the Swedish Board for Technical Development under Contract 69-631/U489.

TABLE OF CONTENTS

1. INTRODUCTION

In a practical application of control theory there is always a need for a model of the process under study. There are two fundamental methods to obtain such a model, either from basic physical, mechanical and thermodynamical equations or from measured process input-output data. The latter method is called identification. Before choosing the identification method, there are two questions to answer, viz."What is the intended use of the obtained model?" and "What are the most important properties of the system, that are to be accounted for?"

If the model is to be used for automatic design of control laws, a parametric model would be adequate, and if the system can be assumed to be linear and time invariant, identification schemes such as the maximum likelihood method,can be used. If, on the other hand, an important property of the system is that, it is time varying, other methods must be used. Methods capable of tracking time varying parameters are known as real time identification methods. These are extremely important for adaptive control algorithms, which can be considered to consist of two parts, one that computes the actual values of the system parameters, and another one that uses this information to compute the control signal.

In section 2 of this report the model to be used throughout is presented. In chapter 3 - 5 three different real time identification methods are shown and these are compared in chapter 6. Chapter 7 contains a further discussion of the model. Chapter 8 introduces an idea of how the limitations found in the previous chapter may be overcome. In chapter 9 some practical problems are discussed, and in chapter 10 results from digital simulations are shown.

2. THE MODEL

The system model to be used throughout is:

$$A(q^{-1})\, y(t) = B(q^{-1})\, u(t) + e(t) \qquad (2.1)$$

where y is the output signal and u is the input signal of the system. e(t) is the misfit between the model and the available measurements at time t. We assume that e(t) is independent gaussian with E e(t) = 0 and

$$E\, e(t)\, e(s) = R_2\, \delta_{ts}$$

q is the forward shift operator, i.e.

$$q\, x(t) = x(t+1)$$

The polynomials A and B are defined as follows:

$$A(q^{-1}) = 1 + a_1\, q^{-1} + a_2\, q^{-2} + \ldots + a_n\, q^{-n}$$

$$B(q^{-1}) = b_1\, q^{-1} + b_2\, q^{-2} + \ldots + b_n\, q^{-n}$$

Rewrite (2.1) to obtain:

$$y(t) = \varphi(t-1)\, \theta(t) + e(t) \qquad (2.2)$$

where

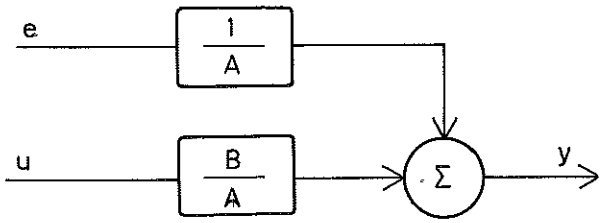
$$\varphi(t-1) = \left[-\, y(t-1) \quad u(t-1) \quad -\, y(t-2) \quad \ldots \quad -\, y(t-n) \quad u(t-n)\right]$$

$$\theta(t) = \left[a_1(t) \quad b_1(t) \quad a_2(t) \quad \ldots \quad a_n(t) \quad b_n(t)\right]^T$$

In the equation (2.2) φ is a row vector containing old input and output values while θ contains the parameter of the polynomials A and B. The vector θ may be varying with time.

The model describes a system, whose pulse transfer function is:

$$H(q^{-1}) = \frac{B(q^{-1})}{A(q^{-1})}$$



It may seem unnatural to use a model that assumes the very special type of noise disturbance indicated in the figure above. The reason why the model is used in spite of this is that it is sufficiently simple so that real time identification schemes can be derived to track the parameters of it and that it actually describes both the system and the noise to a certain extent. Refer to sections 7 and 8 for a more detailed discussion on the matter.

3. IDENTIFICATION USING STOCHASTIC APPROXIMATION

3.1. General Principles.

Let the aim be to find the minimum of a function F(x).

An often used iterative algorithm is:

$$x_{n+1} = x_n - \gamma[n] \text{ grad } F(x_n) \qquad \gamma[n] > 0 \tag{3.11}$$

If in each step the value of the gradient is determined by measurements and these are corrupted by noise, certain restrictions must be imposed on the function F and the step size $\gamma[n]$ in order for the algorithm to converge. The step size must e.g. tend to zero as $n \to \infty$ but not too fast. For details, see e.g. ref. [9] or [7]. If the step size γ tends to a small positive value the algorithm will not converge but x_N will be close to the true value if N is large. See ref. [2].

3.2. Stochastic Approximation Used for Identification

Assume that the system can be described by the model (2.2):

$$y(t) = \varphi(t-1) \quad \theta(t) \quad + e(t)$$

The aim is to determine the parameter vector θ. Define the loss function:

$$F(\theta) = \frac{1}{2}\big(y(t) - \varphi(t-1) \quad \theta(t)\big)^2 \quad \forall t. \tag{3.21}$$

Hence we have:

$$\text{grad } F(\theta) = -\varphi(t-1)^T \{y(t) - \varphi(t-1) \quad \theta(t)\} \tag{3.22}$$

The algorithm for estimating θ then is:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma(t)\varphi(t-1)^T \{y(t) - \varphi(t-1) \quad \hat{\theta}(t)\} \tag{3.23}$$

The behaviour of the algorithm is determined by the properties of the step length $\gamma(t)$. For normalization, the denominator of

$\gamma(t)$ is chosen to be

$$\left(\varepsilon + \varphi(t-1)\,\varphi(t-1)^T\right)$$

where ε is a small number. (ε is so that the denominator never will be zero.)

In the case of time invariant parameters θ will converge to the true value if the numerator is chosen as $1/t$. See ref. [9]. We thus have:

$$\gamma(t) = \frac{1}{t\left(\varepsilon + \varphi(t-1)\,\varphi^T(t-1)\right)} \tag{3.24}$$

If, on the other hand, the numerator is chosen to be constant $= \gamma_0$, the algorithm will no longer converge but according to ref. [2] the estimates will stay close to the true value if γ_0 is sufficiently small. Furthermore, as the step size does not decrease the algorithm will have the ability to track slow parameter variations. This will be illustrated in example I.

4. REAL TIME LEAST SQUARES

The least squares method works by minimizing

$$\sum_{t=n}^{N} e(t)^2$$

where $e(t)$ is defined by equation (2.1). As is well known this method can be used in a recursive fashion (see ref. [4]). To obtain an algorithm with real time properties we should use a loss function that puts more weight on recent observations. This can be done by using the loss function

$$V(N) = \sum_{t=n}^{N} \lambda^{N-t}\, e(t)^2 \qquad \lambda < 1 \tag{4.1}$$

Using the notation of section 2, an algorithm minimizing (4.1) is given by:

$$\begin{cases} \hat{\theta}(t+1) = \hat{\theta}(t) + K(t)\{y(t) - \varphi(t-1)\ \hat{\theta}(t)\} \\ K(t) = P(t)\,\varphi(t-1)^T\{1 + \varphi(t-1)\ P(t)\,\varphi(t-1)^T\}^{-1} \\ P(t+1) = \frac{1}{\lambda}\left[P(t) - K(t)\{1 + \varphi(t-1)\ P(t)\,\varphi(t-1)^T\}K(t)^T\right] \end{cases} \tag{4.2}$$

A detailed derivation of this algorithm is given in appendix B. The algorithm is discussed in section 6.

5. THE KALMAN FILTER APPROACH.

We again use the model of section 2:

$$y(t) = \varphi(t-1)\ \theta(t) + e(t) \tag{5.1}$$

where $e(t)$ is assumed to be gaussian with

$$E\ e(t)\ e(s) = R_2\ \delta_{ts}$$

In order to account for the time variance of the parameters assume that they may be described by

$$\theta(t+1) = \theta(t) + v(t) \tag{5.2}$$

where $v(t)$ is independent gaussian variables

$$E\ v(t) = 0,\ E\ v(t)\ v(s)^T = R_1\ \delta_{ts}$$

Equation (5.2) together with (5.1) is the equation of a linear time varying dynamical system driven by white noise where θ, containing the parameters of the model, is the state vector. According to the assumptions made, a Kalman filter may be used to estimate the state θ. We obtain:

$$\begin{cases} \hat{\theta}(t+1) = \hat{\theta}(t) + K(t)\{y(t) - \varphi(t-1)\,\hat{\theta}(t)\} \\ K(t) = P(t)\,\varphi(t-1)^T\{R_2 + \varphi(t-1)\,P(t)\,\varphi^T(t-1)\}^{-1} \\ P(t+1) = P(t) + R_1 - K(t)\{R_2 + \varphi(t-1)\,P(t)\,\varphi(t-1)^T\}K(t)^T \end{cases} \tag{5.3}$$

These recursive equations constitute an algorithm for real time identification.

When implementing the algorithm, the P equation can be scaled so that $R_2 = 1$. This means that R_1, the covariance matrix of the parameter noise, is given with the measurement error variance R_2 as unit. The value of the R_1 matrix must be passed to the algorithm by the user. In general the characteristics of the parameter variations are not known. In this case good practice is to set $R_1 = r \cdot I$, where $r = 0.01 \ldots 0.0001$ typically. A small value will make the estimations less sensitive to the measurement noise while the ability to track fast parameter variations is degraded. A more thorough discussion of the properties of the algorithm and the relation to the previously mentioned algorithms can be found in section 6. Methods to determine the R1 matrix are discussed in ref. [1] and [8]. See also example III.

6. A COMPARISON BETWEEN THE DIFFERENT ALGORITHMS.

The equations (3.23), (4.2) and (5.3) possess great similarities. The factor:

$$res(t) = y(t) - \varphi(t-1)\,\hat{\theta}(t) \tag{6.1}$$

the residual, is common to all of them and is the difference between the observed output from the system and the expected output. This difference is used to modify the parameter estimates. The sign and magnitude of the change is given by:

$$\gamma_0\,\varphi(t-1)\left(\varepsilon + \varphi(t-1)\,\varphi(t-1)^T\right)^{-1} \quad \text{or} \tag{6.2}$$

$$P(t)\,\varphi(t-1)\left(1 + \varphi(t-1)\,P(t)\,\varphi(t-1)^T\right)^{-1}$$

The only difference is that in the case of least squares or Kalman estimates additional information from the P matrix is included thus presumably providing more accurate estimates. It can be shown that if $\lambda = 1$ or $R1 = 0$ (as used for identification of time invariant systems) the P matrix tends to zero as $1/t$. This corresponds to the values used for γ in the time invariant stochastic approximation case. Cf eq. (3.24). The difference between the least squares and the Kalman method lies in the P equation. In order to obtain real time estimates one should prevent the P matrix from converging to zero in resemblance with the choice of a constant γ_0. This is done in the least squares case by dividing the elements in P by $\lambda < 1$ in each step, while in the Kalman case the same thing is accomplished by adding the constant matrix R1. The main difference is that the R1 matrix has a physical interpretation as the covariance matrix of the noise driving the parameters. This means that if a priori knowledge of the behaviour of the parameters is available it is easily incorporated in the algorithm. For an example of this situation refer to the discussion in 10.2.

The simplicity of the stochastic approximation algorithm is interesting. This property could be used in a case where the Kalman indentification algorithm is used in adaptive control, ref. [10]. When there is evidence that the system parameters happen to change little with time for a while one could switch over to the simpler stochastic app-

roximation algorithm, thus saving computation time. When one detects that the parameters start varying again, switch back to the more sophisticated algorithm.

7. DISCUSSION OF THE STRUCTURE OF THE MODEL.

The model used is:

$$A(q^{-1})\, y(t) = B(q^{-1})\, u(t) + e(t) \tag{7.1}$$

where e(t) was assumed to be white noise. In practice e(t) is likely to be correlated, hence a more appropriate model would be

$$A(q^{-1})\, y(t) = B(q^{-1})\, u(t) + C(q^{-1})\, e(t). \tag{7.2}$$

As in the case of least squares estimation, the parameters of the model (7.1) will be biased estimates of the true values if the system is given by equation (7.2). Furthermore, a statistical F test of the model order will indicate a too high value. A remedy of this situation would be:

a) to use prefiltering with the filter:

$$H(q^{-1}) = \frac{1}{C(q^{-1})}$$

Define $\bar{y} = Hy$ and $\bar{u} = Hu$ where y and u are the measured variables. Perform the identification on the filtered variables to obtain the model:

$\bar{A}\bar{y} = \bar{B}\bar{u} + \bar{e}$ $\qquad$ $\bar{e}$ uncorrelated

We thus obtain the model for the unfiltered variables

$$\bar{A}y = \bar{B}u + C\bar{e} \tag{7.3}$$

which is of the required structure. The controller could then be designed using an adaptive minimum variance strategy. A drawback of this method is that it requires a priori knowledge

of the polynomial C in (7.2), a not very likely situation.

b) to use the model (7.1) with the order indicated by the test mentioned above in order to derive a control algorithm. This may seem to be a rude method, but it will be shown in section 8 that a controller obtained in this way can be expected to have some nice properties.

c) to use a combination of a) and b). The reason why the somewhat unprejudiced method in b) should work is that the model is to be used to design controllers, and in this case the only requirement is that the model does describe behaviour of the system up to a certain extent, while it is immaterial whether the model has a physical interpretation or not.

The reason to use the model (7.1) and not (7.2) is due to the fact that the estimates of the C polynomial coefficients would be non-linear functions of the input-output data, and it has for a long time been impossible to derive recursive or real time algorithms in this case. Lately results on a real time identification method for the model (7.2) has been reported, [5] and [3], but the convergence is slow.

8. A CONJECTURE
- IDENTIFICATION WITH TOO HIGH MODEL ORDER.

As stated earlier, the statistical F test will indicate a too high model order when the measurement noise e(t) in (7.1) is correlated. It will now be made plausible that a dead-beat strategy, derived from such a model, will have nice properties. Let the obtained model be:

$$A(q^{-1})\, y(t) = B(q^{-1})\, u(t-k) + e(t) \quad \text{or} \tag{8.1}$$

$$y(t) = \frac{B}{A}\, u(t-k) + \frac{1}{A}\, e(t) \tag{8.2}$$

Assume, that the system could be described deterministically by

$$A_1\, y(t) = B_1\, u(t-k) \tag{8.3}$$

It can then be expected that $B = B_1 B_2$ and $A = A_1 A_2$ with $B_2 \approx A_2$. (8.2) gives then:

$$y(t) = \frac{B_1 B_2}{A_1 A_2}\, u(t-k) + \frac{1}{A_1 A_2}\, e(t) = \frac{B_1}{A_1}\, u(t-k) + \frac{A_2^{-1}}{A_1}\, e(t) \tag{8.4}$$

(8.4) is of the same structure as (7.2). From (8.4) a minimum variance strategy can be derived:

$$A_2^{-1} = A_1 F + q^{-k} G \tag{8.5}$$

which gives:

$$u = -\frac{G}{F B_1}\, y = H_{mv}\, y \tag{8.6}$$

From (8.1) a dead-beat strategy can be found:

$$1 = A_1 A_2 F' + q^{-k} G' \tag{8.7}$$

$$u = -\frac{G'}{F'B_1B_2}\,y \approx -\frac{G'}{F'B_1A_2} = H_{db}\,y \qquad (8.8)$$

A comparison of (8.5) with (8.7) shows that $F' \approx F$ and $G' \approx A_2\,G$. We thus find:

$$H_{db} = -\frac{G'}{F'B_1A_2} \approx -\frac{G}{FB_1} = H_{mv}$$

There is no need to stress that this argumentation in no way is intended to prove that a dead-beat regulator achieved in the described manner always will be close to optimum. It is felt, however, that a model of the form (8.1) does contain enough useful information of the system behaviour to allow close-to-optimum controllers to be designed even in the case of correlated noise.

9. THE PROBLEM OF BIASED MEASUREMENTS IN REAL TIME IDENTIFICATION.

A problem which often arises when identifying on measured plant data is what to do with biased signals. When there is a bias in one or both of the measured signals they will not fit in a linear model. This problem has been solved by different methods before, but as will be shown these methods cannot be used in the case of real time identification.

9.1. Subtract the Mean.

A method often used in off-line methods is to compute and subtract the mean from the measurements. It is obvious that this trick cannot be used in a real time environment.

9.2. Prefiltering with a High-Pass Digital Filter.

Also this method is often used in off-line calculations. A simple high-pass digital filter is $H(q^{-1}) = (1 - q^{-1})$. Prefiltering with this filter is not feasible if the input signal is a PRBS type signal, which is often the case in practice. The reason is that the parameter correction term in all of the three methods discussed earlier is essentially the vector φ of old input-output data with proper weighting. In this case the filtered input signal will be zero except when there is a change in the input. The correction of the b parameter estimates will thus be close to zero save from those instants mentioned above. This behaviour is drastically illustrated in figure 1. The problem can be solved to a certain extent by introducing more dynamic in the filter, but this means to incorporate a lot of arbitrary constants in the filter design, a not very expedient solution.

9.3. Incorporating an Unknown Constant in the Model.

Let us use the model:

$$A(q^{-1})\ y(t) = B(q^{-1})\ u(t) + k + e(t) \tag{9.31}$$

The part of the bias in the signals that does not fit in the linear model is accounted for by the parameter k. This method has been used in conjunction with the maximum likelihood method and good results are reported. However, attempts to use the model (9.31) for real time identification algorithms have failed to converge correctly (see results below). The reason is probably that the value of k is a function of the other parameters. The minimization problem thus is too complicated to be handled by a recursive method as in the case of real time identification. A modification of the method described above has proved successful. We use the following model:

$$y(t) + a_1\ y^*(t-1) + \ldots + a_n\ y^*(t-n) = B(q^{-1})\ u(t) + k(t) + e(t) \tag{9.32}$$

where

$$y^*(t) = y(t) - \hat{k}(t)$$

To illustrate the success of this method results will be shown of an identification of a system, given by:

$$y(t) = 0.5\ y(t-1) + 1.0\ u(t-1) + 0.1\ e(t)$$

$$y_m(t) = y(t) + 50$$

$$u_m(t) = u(t)$$

$$e(t) \in N(0.1)$$

The initial values of $\hat{\theta}$ and P was $\hat{\theta}(0) = 0$ and $P(0) = 100 * I$.

Results:

t	$\hat{a}_1$	$\hat{b}_1$	$\hat{k}$
50	-0.489	1.013	49.870
100	-0.515	1.010	49.955
150	-0.491	1.005	49.977
200	-0.487	0.993	50.000
*	-0.019	0.856	49.102

The identification scheme used was a Kalman filter with R1 = 10^{-3} × I. The starred estimates are obtained at time t = 200 using the model (9.31).

10. EXAMPLES.

In this section three examples will be discussed. The first one is a simple example of the stochastic approximation method used in real time application. The second one is intended to illustrate the ability of the algorithm (5.3) to track fast parameter variations. Also the influence of the choice of the R1 matrix is illustrated. The third example, finally, illustrates a principle that can be used to determine the R1 matrix in the Kalman identification algorithm.

10.1. Real Time Stochastic Approximation Identification.

The system to be identified was given by:

$$y(t) + a(t)\, y(t-1) = b(t)\, u(t-1) + 0.1\, e(t)$$

where $e(t) \in N(0,1)$ and $E\, e(t)\, e(s) = \delta_{ts}$ and

$$a(t) = \begin{cases} -0.5 & t \leq 300 \\ -0.5 - 0.0045(t-300) & 300 < t \leq 400 \\ -0.95 + 0.0045(t-400) & 400 < t \leq 500 \end{cases}$$

$$b(t) = \begin{cases} 1.0 & t \leq 200 \\ 1.0 + 0.02(t-200) & 200 < t \leq 300 \\ 3.0 & 300 < t \leq 400 \\ 3.0 - 0.02(t-400) & 400 < t \leq 500 \end{cases}$$

The identification algorithm used was that given in (3.23) with $\gamma(t)$ given by:

$$\gamma(t) = \frac{GAM}{GAM + \varphi(t-1)\,\varphi^T(t-1)} \qquad (10.11)$$

where:

$$GAM = \begin{cases} 1.0 & t \leq 10 \\ 1.0 - \frac{(GAM - GAMZ)}{40}(t-10) & 10 < t \leq 50 \\ GAMZ & t > 50 \end{cases}$$

The choice of the values of GAM may need some discussion. GAM is given a relatively high value at the first 10 sampling points in order to allow a fast convergence to estimated parameter values in the vicinity of the true ones. For time instants greater than 50 GAM assumes the stationary value GAMZ. In between GAM decreases linearly towards GAMZ.

In figures 2 and 3 the results of identification with GAMZ equal to 1.0 resp. 0.1 are shown. Note in figure 2 the ability of the algorithm to track the parameter variations. Compare the performance in the time interval 0 - 200 where both parameters are held constant with that shown in figure 3. As can be seen, the algorithm with GAMZ = 0.1 is much less sensitive to noise but on the other hand the tracking ability as shown in figure 3 is not as good as that shown previously. The illustrated contrast between tracking ability and noise rejection is a common (and very natural) factor to take into account in all implementations of real time identification algorithms.

10.2. The Kalman Identification Applied to Rapidly Varying Parameters.

This example is intended to demonstrate:

- the ability of the algorithm to track fast parameter variations,
- the influence of the choice of the R1 matrix.

A continuous system was simulated digitally. The system had a transfer function of the form:

$$G(s) = \frac{(T_1 s + 1)}{(T_2 s + 1)(T_3 s + 1)} \qquad (10.21)$$

T_2 and T_3 was held constant; T_2 = 1 sec and T_3 = 0.1 sec. T_1 was varying randomly from $T_1(t_s) \approx 2.0$ sec to $T_1(t_f) \approx 0.1$ sec. (t_s = starting time; t_f = final time.)

The continuous system was transformed to its discrete time equivalent:

$$y(t) + a_1\, y(t-1) + a_2\, y(t-2) = b_1\, u(t-1) + b_2\, u(t-2) + e(t) \qquad (10.22)$$

The sampling interval was chosen as 0.05 sec. In this representation a_1 and a_2 were constant while b_1 and b_2 were varying with time in such a manner that $b_1(t) - b_1(t-1) = -\left(b_2(t) - b_2(t-1)\right)$. u(t) was gaussian noise with band width $f_B \approx 5$ Hz and σ_u = 10.0. e(t) was white noise with E e(t) e(s) = δ_{ts}. In figure 4 is shown the amplitude characteristics for the system (10.21) at t = 0, 200 and 500. The input and output signals are shown in figure 5. Note the decrease of the system band width.

In order to compare different identification runs, one need a goodness criterion and in this case:

$$V = \sum_{t=1}^{N} res(t)^2 \qquad (10.23)$$

was used.

The identification scheme used was the Kalman type one described in section 5. In the first run the R1 matrix was chosen to be R1 = 0.01 * I. The tracking of the parameters a_1 and b_2 can be studied in figure 6. As can be seen the estimates follow the general trend of the parameters but fail to find the details. The loss function is in this case V = 3605.

In this example a priori knowledge of the R1 matrix is available. The theoretical value calculated from knowledge of the standard deviation of T_1 is:

$$R1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & -0.01 \\ 0 & 0 & -0.01 & 0.01 \end{pmatrix} \qquad (10.24)$$

If this value is used in the identification the results shown in figure 7 are obtained.

Note, that in this case the estimate of the parameter a_1 tends to a constant value (as the estimate of a_2) in accordance with the a priori information used when setting $R1_{11} = 0$. Note also the improved tracking of the parameter b_2. This is due to the more accurate knowledge of the two constant parameters and the a priori information that the covariance of the two b parameters is -0.01 (the element $R1_{34}$). The conclusion of this example is thus, that accurate a priori information is of great value, if accessible. On the other hand false a priori information is likely to do great harm if used in that it will impose a misleading restriction on the algorithm. In other words, R1 matrices not of the form R1 = r · I should be used restrictively. The problem of how to optimize the choice of R1 matrix will be discussed in ref. [8].

10.3. Segerståhl's Example

The real time identification algorithm (5.3) has been applied to example 1 of ref. [6]. The system in this example is given by:

$$y(t) = a(t)\, u(t) + b(t)\, u(t-1) + 1.225\, e(t) \tag{10.31}$$

where e(t) are uncorrelated N(0,1) random variables and:

$$\begin{cases} a(t) = 2 \exp \{ - 0.2\, u(t) + 0.9 \} \\ b(t) = 0.5 + 2 \exp \{ - 0.15\, u(t) + 0.7 \} \end{cases} \tag{10.32}$$

Note, that the parameters in this case are non-linear functions of the input. The non-linear effects are in this case described as a time dependence of the parameters. The model used was of the same structure as the system. In this experiment the R1 matrix was chosen as the one that minimized the squared sum of the last thirty residuals, i.e. the loss function:

$$V = \sum_{t=171}^{200} res(t)^2$$

This procedure will yield a value of the R1 matrix that minimizes the mean square difference between the predicted and the measured output. The reason to choose the thirty last points is that the parameters vary most rapidly in this interval. The result of a search for minimum is shown in the table below.

A plot of the estimated parameter values with the R1 matrix:

$$R1 = \begin{pmatrix} 3.5 & 2.9 \\ 2.9 & 2.5 \end{pmatrix} \times 10^{-3}$$

is shown in figure 8. In figure 9 the measured and predicted output signals are given. A comparison will show that the results obtained with this method are superior to those of ref. [6].

R1	V
$\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} \times 10^{-3}$	131.91
$\begin{pmatrix} 1.5 & 0 \\ 0 & 4.5 \end{pmatrix}$	132.41
$\begin{pmatrix} 4.5 & 0 \\ 0 & 1.5 \end{pmatrix}$	131.88
$\begin{pmatrix} 4.5 & 2.5 \\ 2.5 & 1.5 \end{pmatrix}$	126.01
$\begin{pmatrix} 3.5 & 2.9 \\ 2.9 & 2.5 \end{pmatrix}$	125.64

Values of the loss $V = \sum_{171}^{200} res(t)^2$ for different R1.

APPENDIX A

REFERENCES

[1] T. Bohlin: Real Time Estimation of Time-Variable Process Characteristics, TP 18.190, IBM Nordic Laboratory, Lidingö, Sweden.

[2] J.P. Comer: Some Stochastic Approximation Procedures for Use in Process Control, Annals of Mathematical Statistics, Vol. 35, No. 3, 1964.

[3] I. Gustavsson: Unpublished Material

[4] R.C.K. Lee: Optimal Estimation, Identification and Control, MIT Press, Cambridge, Mass., 1964.

[5] Panuska: Conference on Adaptive Systems, Pennsylvania State University, Oct., 1969.

[6] B. Segerståhl: A Theory of Parameter Tracking Applied to Slowly Varying Nonlinear Systems, Technical Session 5, IFAC, Warsaw, 1969.

[7] Ya.Z. Tsypkin: Adaption, Learning and Self-Learning in Control Systems, Survey Paper, IFAC, London, 1966.

[8] J. Wieslander: Real Time Identification, Part II, Report 6909, Lund Institute of Technology, Division of Automatic Control, to appear.

[9] D.J. Wilde: Optimum Seeking Methods, Prentice Hall, Englewood, Cliffs., N.J., 1964.

[10] B. Wittenmark: On Adaptive Control of Low Order Systems, Report 6918, Lund Institute of Technology, Division of Automatic Control, 1969.

[11] K.J. Åström: Lectures on the Identification Problem - The Least Squares Method, Report 6806, Lund Institute of Technology, Division of Automatic Control, 1968.

11. CONCLUSION.

In this report three similar algorithms for real time identification have been presented. Through experiments they have been shown to possess encouraging performance as to precision and ability to track fast parameter variations. A more thorough comparison of the methods as well as a study on the behaviour of a controller of the type discussed in section 8 are currently being made by two students as master thesis work. Results should be available early 1970. In a forthcoming second part of this report among other things a discussion of methods to estimate the R1 matrix from measured data and an example of identification on data from an industrial process will appear.

APPENDIX B

DERIVATION OF THE REAL TIME LEAST SQUARES IDENTIFICATION ALGORITHM.

As a rule whenever it is possible to derive a recursive method for the identification of a system model from input-output data one can obtain a real time algorithm by preventing the estimate correction step-length from decreasing to zero (cf the discussion in chapter 6). The only problem in how this should be done is to retain an attractive physical interpretation.

B 1. The Least Squares Method.

Consider the system model:

$$y(t) = - a_1 y(t-1) - \ldots - a_n y(t-n) + b_1 u(t-1) + \ldots$$

$$\ldots + b_n u(t-n) + e(t) \qquad \text{(B 11)}$$

Introduce:

$$Y = \begin{bmatrix} y(n) \\ y(n+1) \\ \vdots \\ y(N) \end{bmatrix} \quad \phi = \begin{bmatrix} -y(n-1) & u(n-1) & \ldots & -y(0) & u(0) \\ & & & & \\ & & & & \\ -y(N-1) & u(N-1) & & -y(N-n) & u(N-n) \end{bmatrix}$$

$$\theta = \begin{bmatrix} a_1 \\ b_1 \\ a_2 \\ \vdots \\ b_n \end{bmatrix} \qquad \text{and} \qquad E = \begin{bmatrix} e(n) \\ \vdots \\ e(N) \end{bmatrix}$$

The result of measurements at $t = 0, 1, \ldots, N$ can now be expressed as:

$$Y = \phi\theta + E \qquad \text{(B 12)}$$

As is commonly known (see eq. (2.5) of [11]) the estimate of Θ that minimizes:

$$V = \sum_{n}^{N} e(t)^2 = E^T E \qquad \text{(B 13)}$$

is given by:

$$\hat{\Theta} = (\phi^T \phi)^{-1} \phi^T Y \qquad \text{(B 14)}$$

B 2. Real Time Least Squares.

Intuitively what to do when identifying a system model in a time varying case is to put more weight on recent data. This can be done by using the loss function:

$$V(N) = \sum_{t=n}^{N} \lambda^{N-t} e(t)^2 \qquad \lambda < 1 \qquad \text{(B 21)}$$

This will give an exponential weighting to the squared deviation from a deterministic model. In deriving an algorithm that minimizes (B 21) we follow closely the method of ref. [11] and [4] in deriving a recursive least squares algorithm.

Consider the situation at time N. At this instant the estimate

$$\hat{\Theta}(N) = (\phi_N^T \phi_N)^{-1} \phi_N^T Y_N$$

is available. A new measurement is made at time N + 1. Find the estimate $\hat{\Theta}(N+1)$ that minimizes:

$$V(N+1) = \sum_{n}^{N+1} \lambda^{N-t+1} e(t)^2 = \lambda \sum_{n}^{N} \lambda^{N-t} e(t)^2 + e(N+1)^2 =$$

$$= \lambda E_N^T E_N + e(N+1)^2$$

Introduce the notation:

$$y = y(N+1) , \qquad \mu = \sqrt{\lambda}$$

$$\varphi = | - y(N)\ u(N)\ - y(N-1) \ldots\ u(N-n+1) |$$

The results of measurements up to t = N + 1 can be written as:

$$\begin{bmatrix} \mu\ Y_N \\ y \end{bmatrix} = \begin{bmatrix} \mu\ \phi_N \\ \varphi \end{bmatrix} \theta + \begin{bmatrix} \mu\ E_N \\ e(N+1) \end{bmatrix} \tag{B 22}$$

Hence we have:

$$V(N+1) = \lambda\ E_N^T\ E_N + e(N+1)^2 = \begin{bmatrix} \mu\ E_N \\ e(N+1) \end{bmatrix}^T \begin{bmatrix} \mu\ E_N \\ e(N+1) \end{bmatrix} \tag{B 23}$$

As in the time invariant case the least squares estimate is given by:

$$\hat{\theta}(N+1) = \left(\begin{bmatrix} \mu\ \phi_N \\ \varphi \end{bmatrix}^T \begin{bmatrix} \mu\ \phi_N \\ \varphi \end{bmatrix} \right)^{-1} \begin{bmatrix} \mu\ \phi_N \\ \varphi \end{bmatrix}^T \begin{bmatrix} \mu Y \\ y \end{bmatrix} = \tag{B 24}$$

$$= (\lambda\ \phi_N^T\ \phi_N + \varphi^T\ \varphi)^{-1}\ (\lambda\ \phi_N^T\ Y + \varphi^T\ y)$$

By means of the matrix lemma:

$$(A + BC)^{-1} = A^{-1} - A^{-1}\ B\{1 + CA^{-1}B\}^{-1}\ CA^{-1} \tag{B 25}$$

the first factor can be rewritten as:

$$(\lambda\ \phi_N^T\ \phi_N + \varphi^T\ \varphi)^{-1} = \frac{1}{\lambda}\ (\phi_N^T\ \phi_N)^{-1} - \frac{1}{\lambda}\ (\phi_N^T\ \phi_N)^{-1}\ \varphi^T\ x$$

$$x\{1 + \varphi \frac{1}{\lambda}\ (\phi_N^T\ \phi_N)^{-1}\ \varphi^T\}^{-1} \varphi \frac{1}{\lambda}\ (\phi_N^T\ \phi_N)^{-1}$$

Now introduce for simplicity $\phi = \phi_N$, then

$$\hat{\theta}(N+1) = (\phi^T\phi)^{-1}\,\phi^T y + \frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T y -$$

$$- \frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\{1 + \varphi\,\frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\}^{-1}\varphi(\phi^T\phi)^{-1}\,\phi^T y -$$

$$- \frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\{1 + \varphi\frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\}^{-1}\varphi\frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T y =$$

$$= A + B - C - D$$

We find:

$$A = \hat{\theta}(N)$$

$$C = \frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\{1 + \varphi\frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\}^{-1}\varphi\hat{\theta}(N) = K(N)\varphi\hat{\theta}(N)$$

$$B - D = \frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\{1 + \varphi\frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\}^{-1}\{1 + \varphi\frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T -$$

$$- \varphi\frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\}y = \frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\{1 + \varphi\frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\}^{-1}\,y = K(N)y$$

That is:

$$\hat{\theta}(N+1) = \hat{\theta}(N) + K(N)\big(y - \varphi\hat{\theta}(N)\big) \qquad \text{(B 26)}$$

where

$$K(N) = \frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\{1 + \varphi\frac{1}{\lambda}\,(\phi^T\phi)^{-1}\varphi^T\}^{-1}$$

Introducing

$$P(N) = \frac{1}{\lambda}\,(\phi_N^T\,\phi_N)^{-1}$$

we have

$$K(N) = P(N)\,\varphi^T\{1 + \varphi P(N)\,\varphi^T\}^{-1} \qquad \text{(B 27)}$$

What remains is to find a recursive equation for P. Using (B 25) we have:

$$P(N+1) = \frac{1}{\lambda}\,(\phi_{N+1}^{T}\,\phi_{N+1})^{-1} = \frac{1}{\lambda}\,(\lambda\phi_{N}^{T}\,\phi_{N} + \varphi^{T}\varphi)^{-1} =$$

$$= \frac{1}{\lambda}\,\{\frac{1}{\lambda}\,(\phi_{N}^{T}\,\phi_{N})^{-1} - \frac{1}{\lambda}\,(\phi_{N}^{T}\,\phi_{N})^{-1}\,\varphi^{T}\{1 + \varphi\frac{1}{\lambda}\,(\phi_{N}^{T}\,\phi_{N})^{-1}\,\varphi^{T}\}^{-1}\,x$$

$$x\,\varphi\frac{1}{\lambda}\,(\phi_{N}^{T}\,\phi_{N})^{-1}\} = \frac{1}{\lambda}\left[P(N) - K(N)\{1 + \varphi P(N)\,\varphi^{T}\}\quad K(N)^{T}\right] \qquad \text{(B 28)}$$

Summarizing we thus have the equations:

$$\begin{cases} \hat{\theta}(N+1) = \hat{\theta}(N) + K(N)\left[y - \varphi\hat{\theta}(N)\right] \\ K(N) = P(N)\,\varphi^{T}\{1 + \varphi P(N)\,\varphi^{T}\}^{-1} \\ P(N+1) = \frac{1}{\lambda}\left[P(N) - K(N)\{1 + \varphi P(N)\,\varphi^{T}\}\quad K(N)^{T}\right] \end{cases} \qquad \text{(B 29)}$$

APPENDIX C

FIGURES

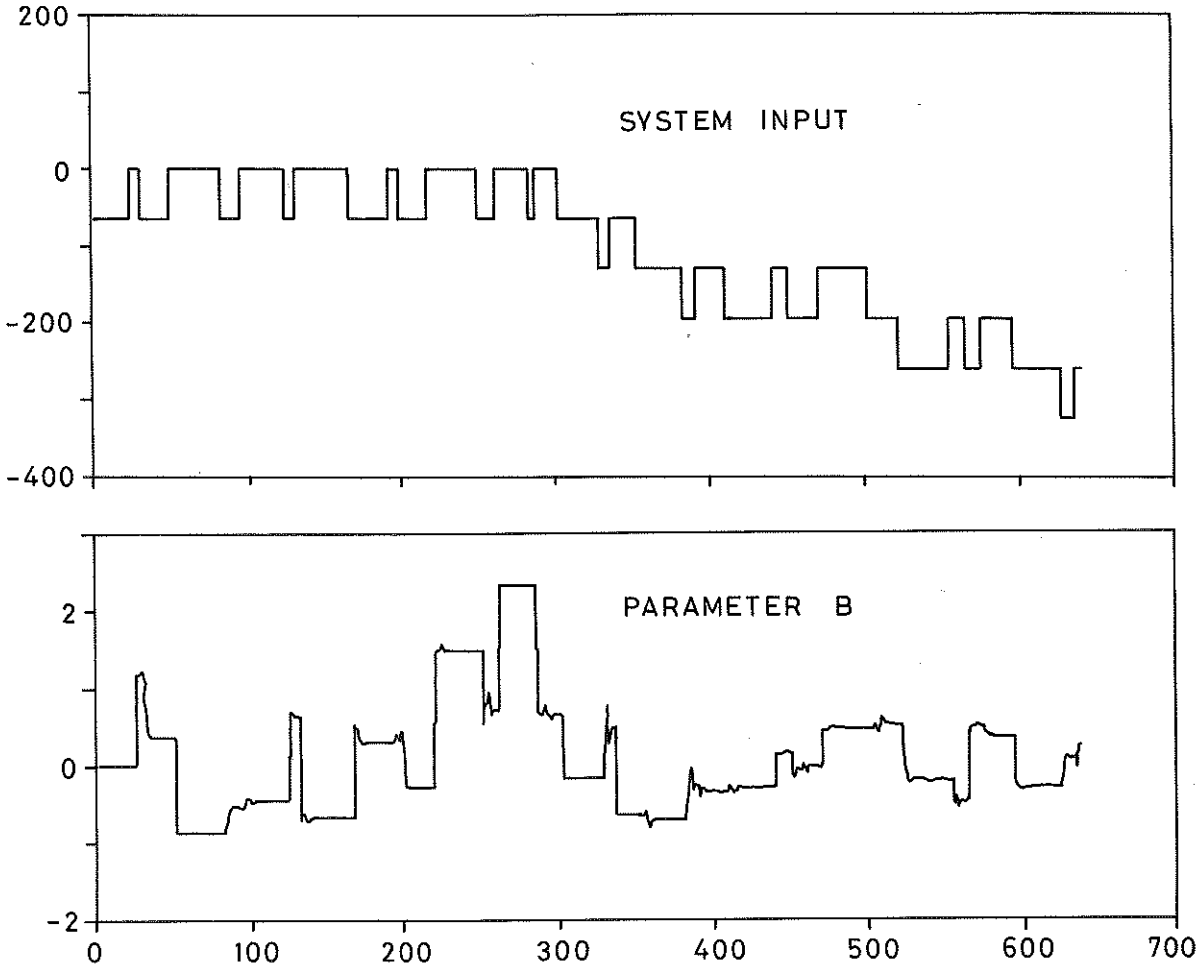


Fig. 1 - The effect of prefiltering with $(1-q^{-1})$ when the input is a PRBS signal. Note the sudden changes in the estimated b-parameter when there is a change in the input.

The input signal is the steam pressure in the cylinders of a paper mill drying section. The output is the paper moisture.

Measurements are from Billerud AB, Gruvön, Sweden.

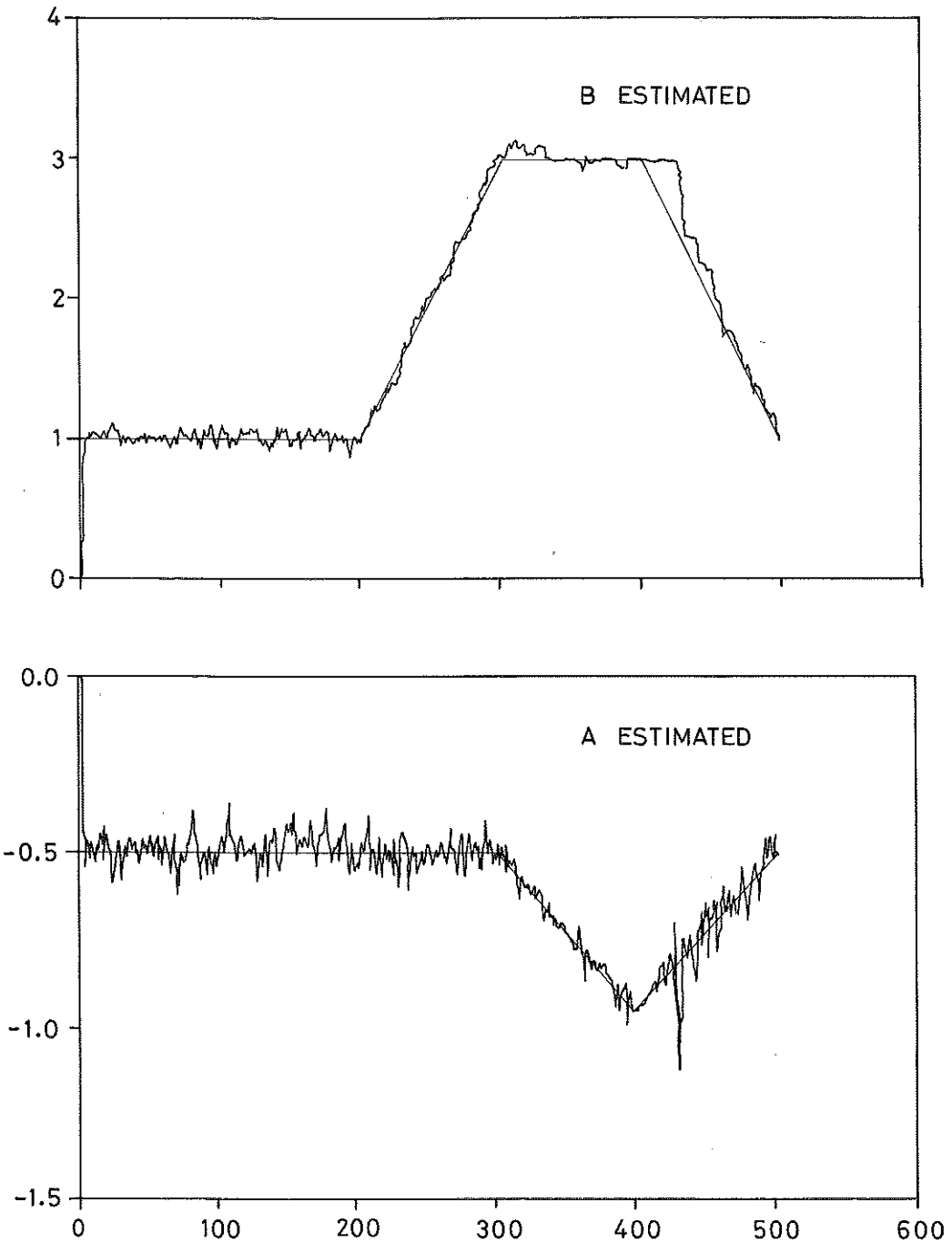


Fig. 2 - Stochastic approximation real time identification with GAMZ = 1.0
Heavy line = Estimated values
Thin line = True values

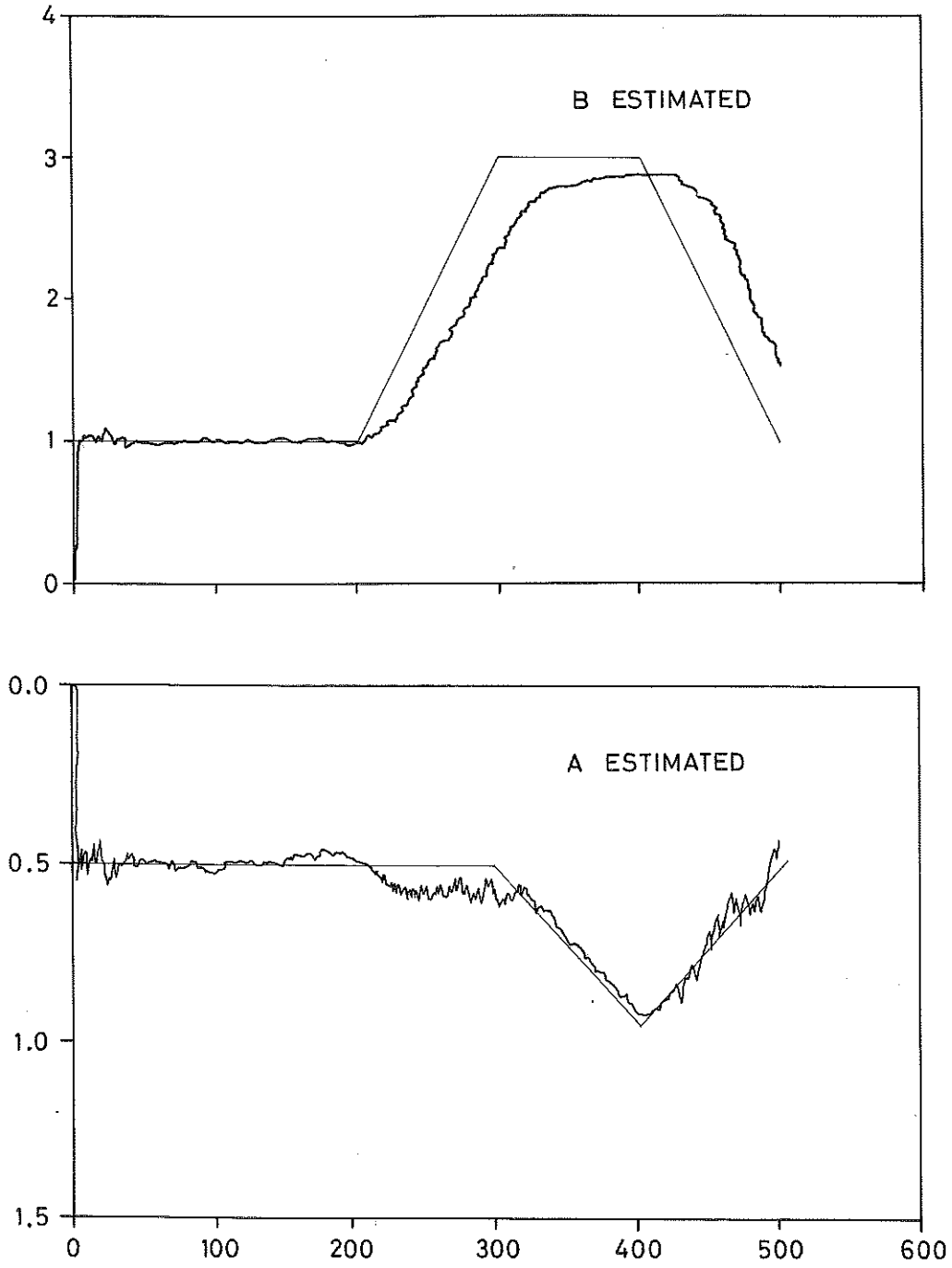


Fig. 3 - Stochastic approximation real time identification with GAMZ = 0.1
Heavy line = Estimated value
Thin line = True values

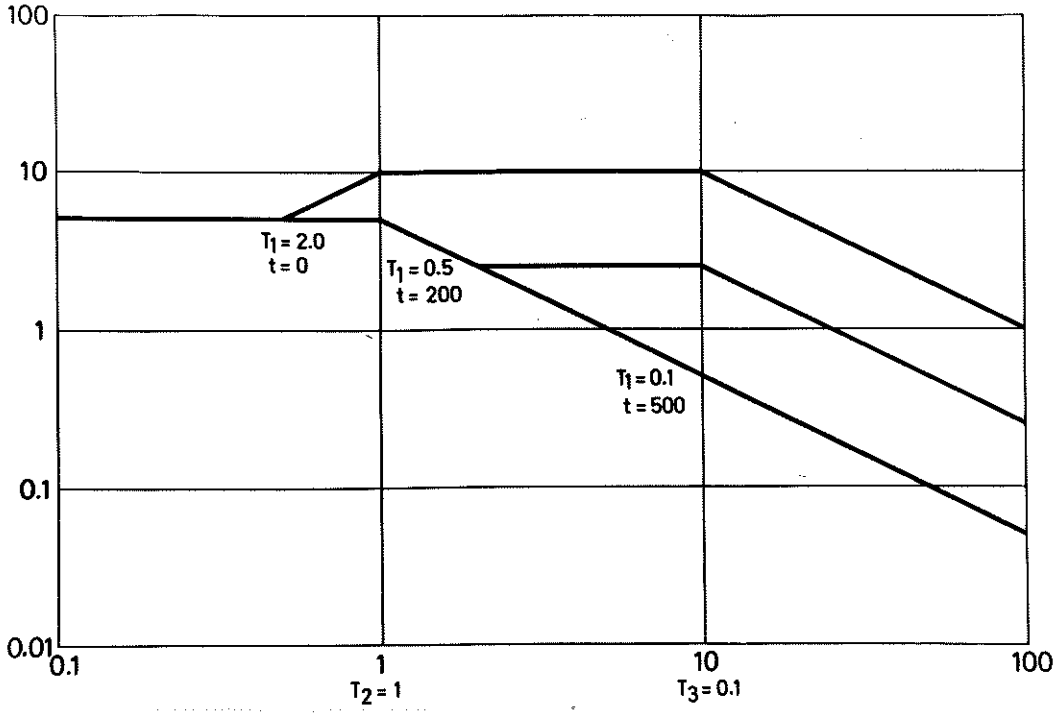


Fig. 4 - The amplitude characteristic of the system in Example 2

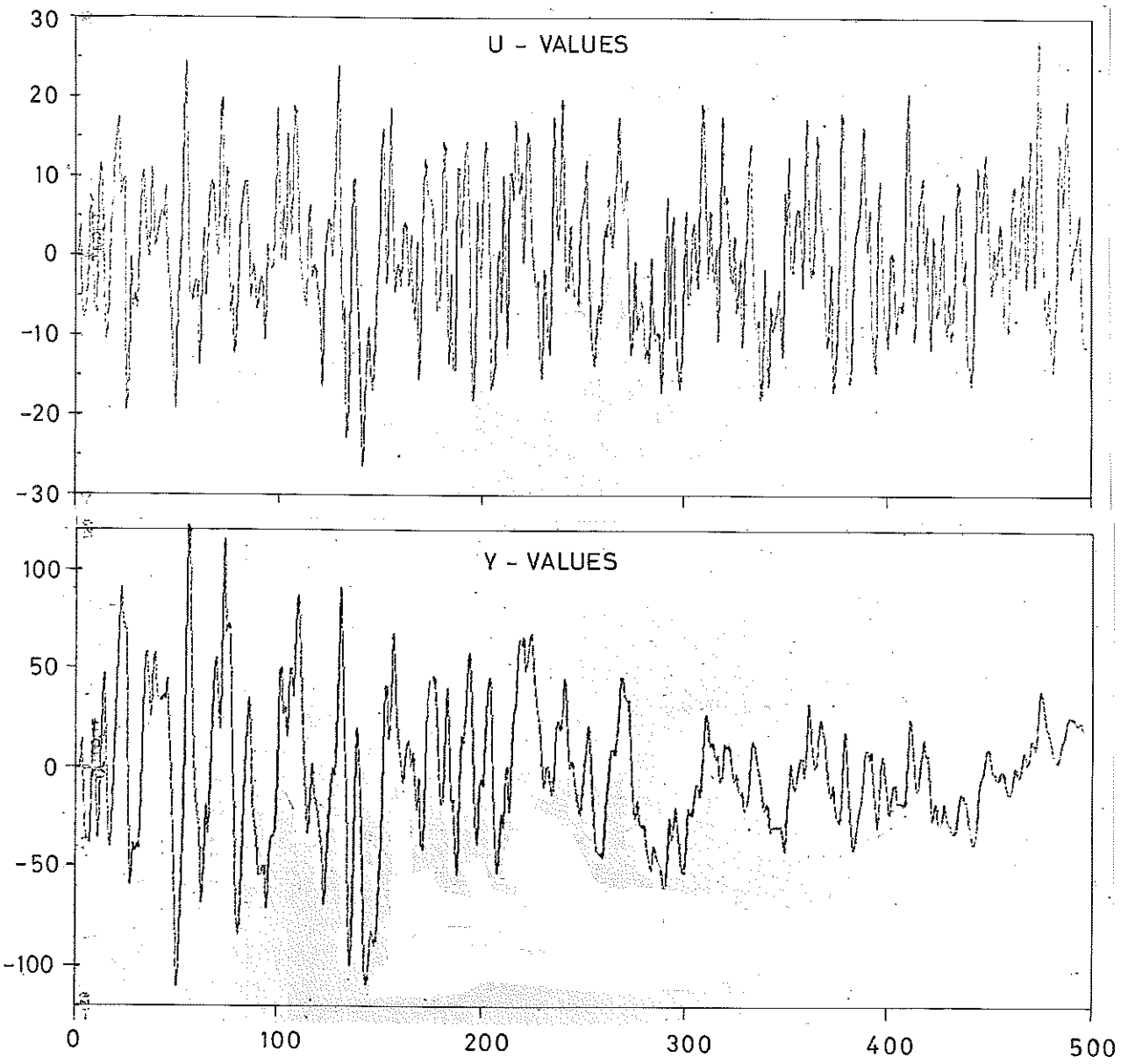


Fig. 5 - The input and output signals from Example 2

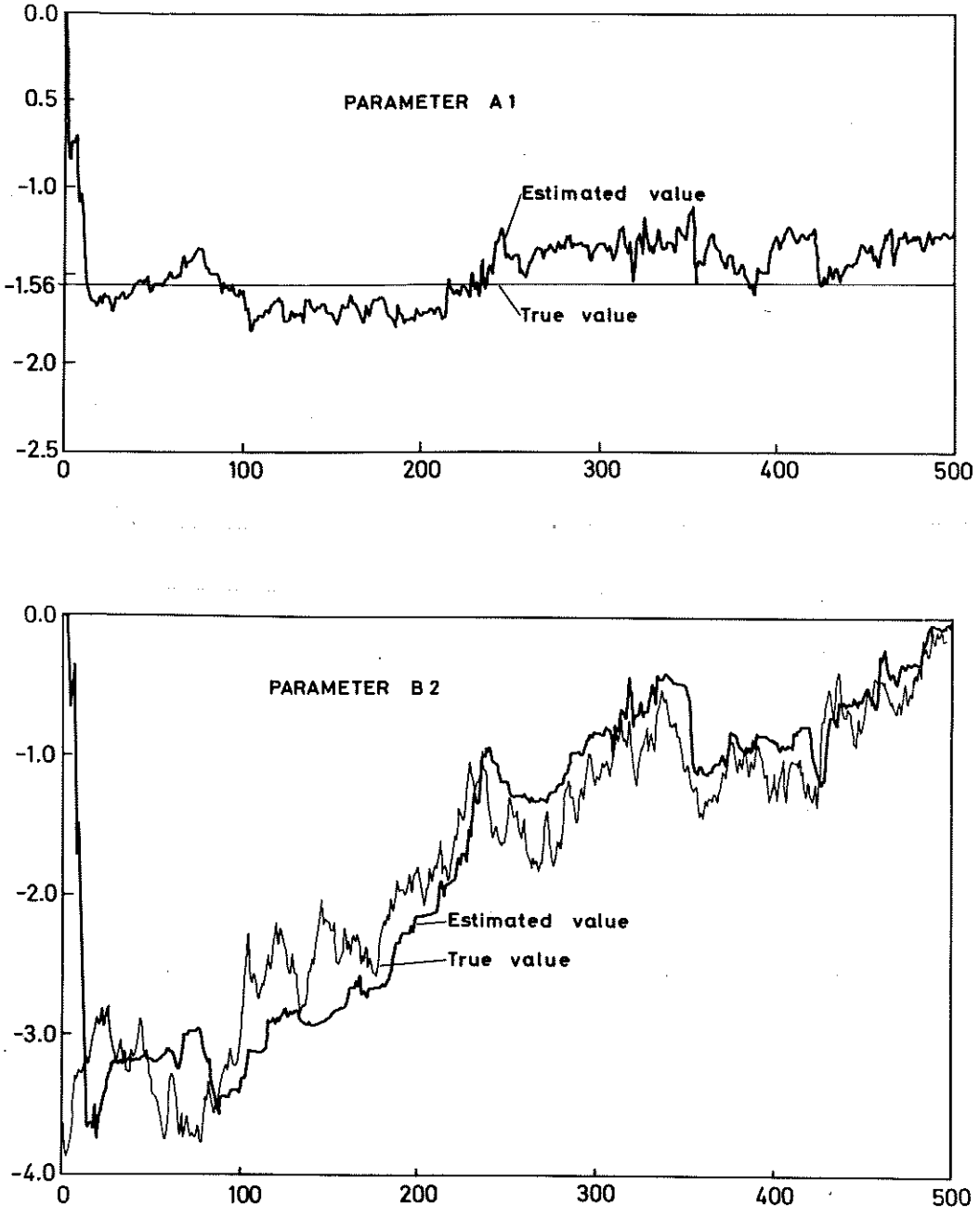


Fig. 6 - Identification of the system (10.22) using Kalman approach. R1-matrix equal to 0.01 × I

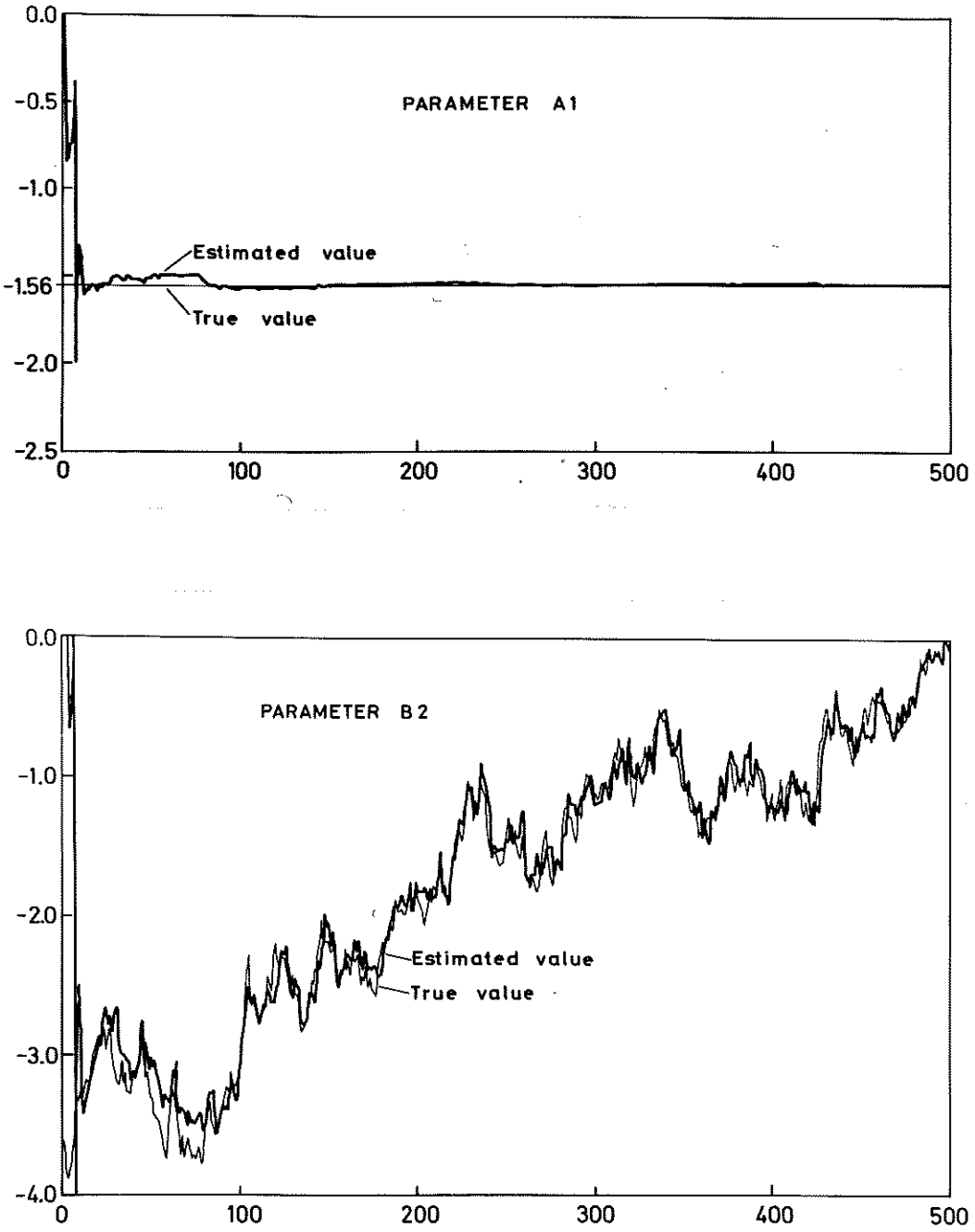


Fig. 7 - Identification of the system (10.22) using Kalman approach. R1-matrix is given by (10.24)

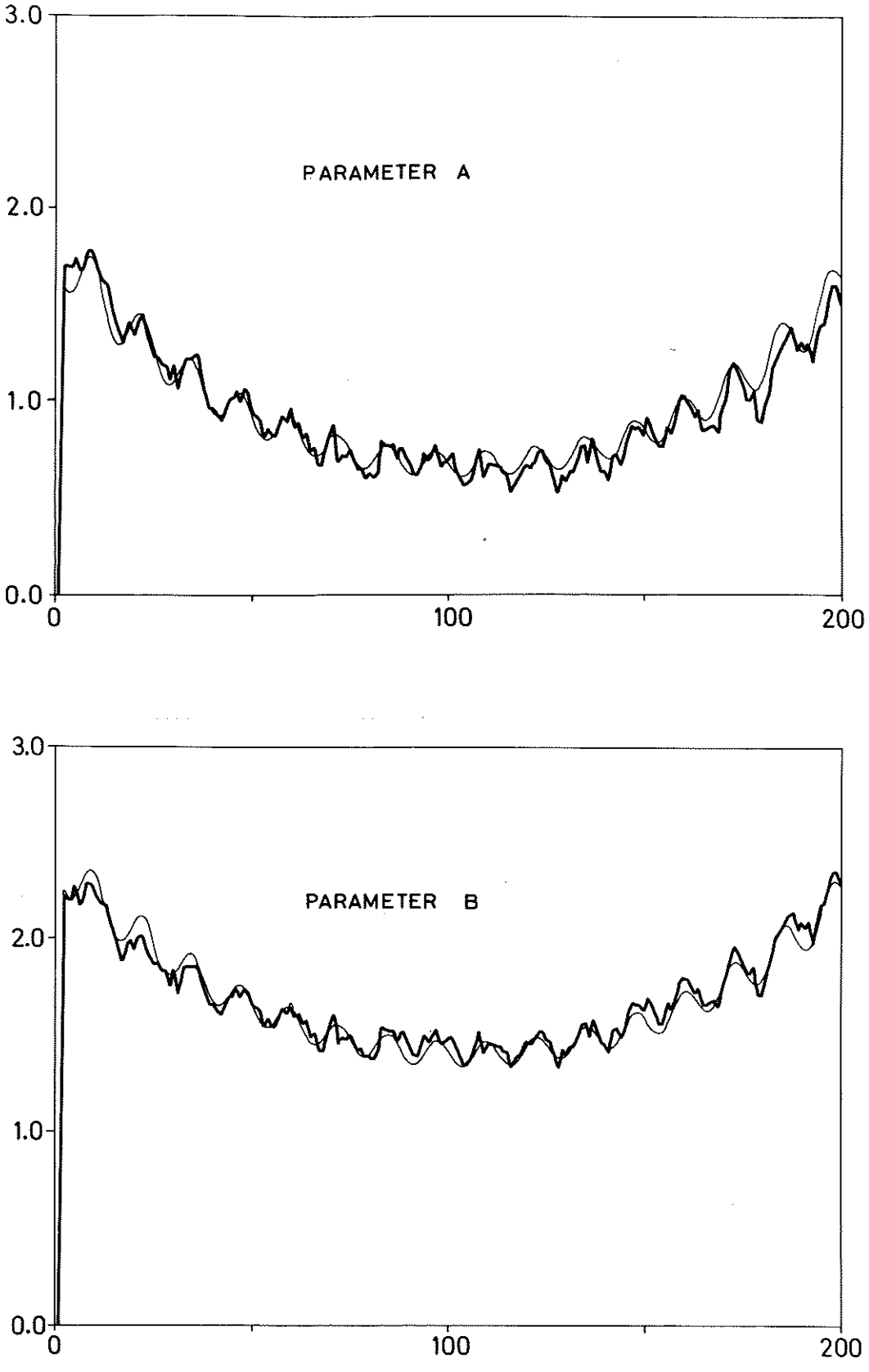


Fig. 8 - Estimated parameters of system (10.31)
Heavy line : Estimated values
Thin line : True values

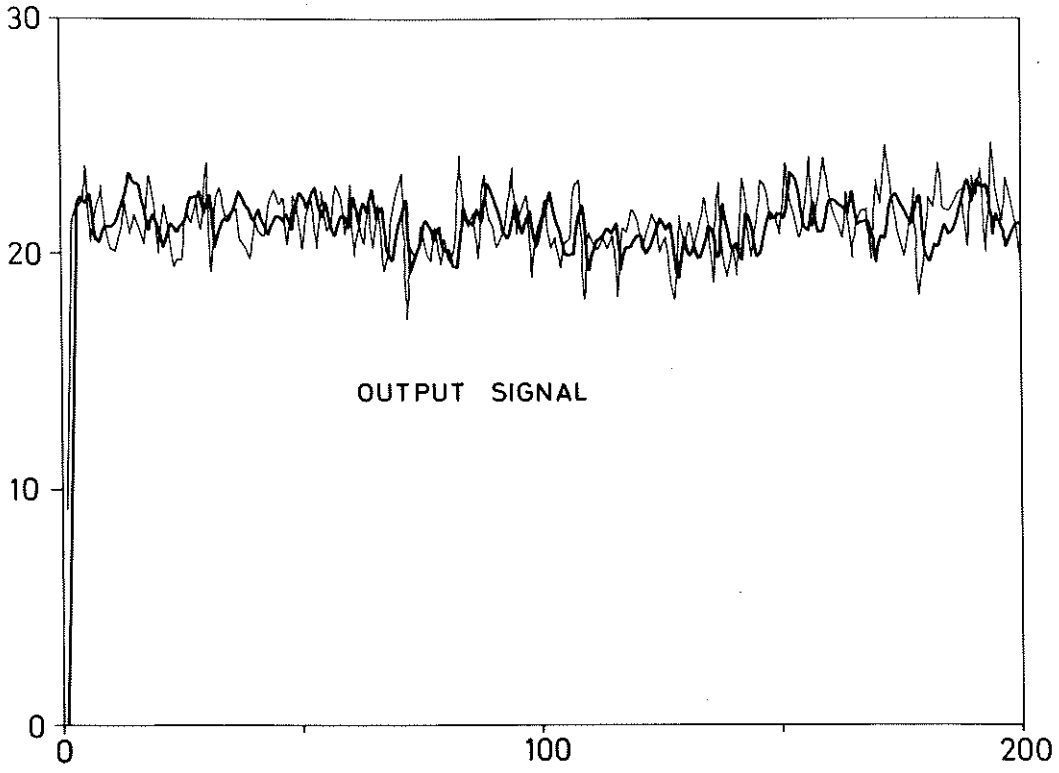


Fig. 9 - The predicted (heavy line) and actual (thin line) output from system (10.31) identified with

$$R1 = \begin{bmatrix} 3.5 & 2.9 \\ 2.9 & 1.5 \end{bmatrix} \times 10^{-3}$$