T. Bohlin

On the Maximum Likelihood Method of Identification

Abstract: The maximum likelihood principle of estimation applied to the linear black-box identification problem gives models with theoretically attractive properties. Also, the method has been applied to industrial data (various processes in paper production) and proved able to work in practice.

This paper presents further developments of the method in the case of a single output. The reliability and speed of the identification algorithm have been improved, and the method has been made easier to use. A rather sophisticated computer program, however, was needed. It employs a generalized model structure, an improved hill-climbing algorithm, and an automatic procedure for determining model orders and transport delays. Some statistics from performance tests of the program are presented.

Review of development

The maximum likelihood method of numerical identification of linear dynamics systems for a single output was introduced in 1965 and was shown to have good theoretical properties.[1,2] The actual performance of the method was tested on a number of artificially generated samples of data. This performance agreed with theoretical predictions. At least three different aspects of the performance are essential: 1) How often and how fast the method gives a result, 2) how good the result is, and 3) how easy the method is to use.

It is natural, in designing the method, that the second aspect gets the highest attention, since it depends on the basic principle adopted for the method. The first aspect depends on the algorithm, which can very likely be improved later, and something can always be done about the third aspect by automating, i.e., by writing a good computer program.

The initial investigation of "how good the result is"[1,2] showed that for long samples the method has the highest possible accuracy—it is asymptotically efficient. This result holds theoretically and also experimentally for data artificially generated according to a model consistent with the assumptions of the identification method. For practical applications using industrial data, this result, of course, is not conclusive. How true the result is in that case will depend on how well the industrial process satisfies the assumptions of the identification method. They are:

1) Linearity
2) Normality of disturbances
3) Time-invariance of process characteristics
4) Time-invariance of disturbance characteristics.

The author is at the IBM Nordic Laboratory, Lidingo 9, Sweden.

For a given sample of data, all four assumptions can be checked.

The results also depend on whether the model has the right structure and order. It is virtually impossible to check this; it is doubtful whether the real process has any order at all. Any model may then be termed as "correct" if it predicts correctly and "good" if it is also simple.

The structure, or form of model, was chosen with other requirements in mind than to be "right," namely, to be general enough to fit many processes, to make the computations simple, and to make the theory manageable.

Therefore, the second development phase has been to test the method on industrial data to see if assumptions held, if the structure was good, what orders were reasonable and, generally, to see how the method would work in practice. This was done for $1\frac{1}{2}$ years in 1965-66, using various processes connected with paper production as the sources of data.[3-5]

The result of this test was that the method worked very well in practice, *if* one knew how to handle it. In particular, the ability of the method to tolerate various kinds of random disturbances was encouraging and went beyond what the theory claims. In practice both normality and the requirement that disturbances be a time-invariant stochastic time-series can be relaxed. As a result of some of these tests, however, the model structure was changed.

The "if" clause is significant. It is to indicate that while the second aspect of performance (quality of model) was very good, the other two aspects (speed and ease of use) could be improved. Work on this has taken another year, and the purpose of this paper is to report the improvements that have been made.

ances. In 1), 2), and 3) disturbances enter with the input, in 4) they are added to the output. The form 1) allows white noise only. In 2) the disturbance spectrum must be known; it is specified through the pre-filter F. In 3) and 4) disturbance spectra need not be white and need not be known. Their characteristics are estimated, together with the other parameters.

Of course, it is not possible to appoint one of the structures as the "best" in general. Any one may be best if it happens to fit the particular process studied. However, it is believed that in practical cases of interest there are always at least observation errors added to the output, so that the structure should reflect this fact explicitly. Then the form 4) with a separate disturbance term $\lambda C(z^{-1})D^{-1}(z^{-1})e(t)$ is most adequate. Also, since the spectrum is arbitrary, the structure allows random drift. Drift has been present in most samples of industrial data investigated during the applications.

The form 4) seemingly contains more parameters than the alternatives, since it is most general. However, if it fits the process, the model will in effect contain fewer parameters. If 4) must be written in one of the other forms, one has to pay for this by increasing the order and therefore the number of parameters. For instance, if the order of all polynomials in 4) is n, one gets $4n + 2$ parameters in 4) and $6n + 2$ parameters in 3). In case 1) the equivalent number of parameters (for a fixed accuracy of approximation) depends on the parameter values, but is in any case $\geq 4n + 2$. For quite reasonable parameter values (e.g., zeroes of $C(z^{-1})$ close to the unit circle) it can be much higher. The advantage of 4) over 1), 2), and 3) is accentuated if more than one input (m, say) are acting simultaneously. Then the number of parameters in a general fixed-order structure is $(m + 1)(2n + 1)$ in 4), $(m + 1)(mn + 2n + 1)$ in 3), and $\geq (m + 1)(mn + n + 1)$ in 1).

It is possible to retain the advantage of 1) by deliberately identifying a model of high order and afterwards rewriting it in either of the forms 3) or 4), eliminating redundant parameters.[8] However, the eliminating operation may be cumbersome, especially for more than one input. Whether this alternative is faster would depend on the eliminating routine and the required minimal order of the model of form 1).

The allowance in the case 4) for an arbitrary disturbance spectrum has turned out to have an important practical consequence: If many input variables u influence the output y simultaneously, it is feasible, in practice, to analyze the influence of one input variable at a time. The effects of other inputs are then absorbed in the disturbance term together with all other variables (measurable or unmeasurable) that add up to form the disturbance. This is so because the effects of inputs and disturbances have been separated in the model. Disturbances, and hence other (stationary) inputs do not therefore affect the A and B operators. In cases 1), 2), and 3) they do.

Modeling algorithm

The algorithm for maximum likelihood modeling, described briefly in the following, is a modification and extension of one given in Ref. 2. Since one obviously needs a digital computer to implement it, the algorithm will be described in terms of a computer program.

The complete modeling program comprises a set of subroutines. Some of these are aimed at improving on the third aspect of performance (ease of use), mentioned initially, but are not strictly necessary for the method. Only their functions will be stated. The hill-climbing algorithm will be described in more detail, since it is crucial for the performance of the method. Also, the method used to determine unknown orders and input delays will be outlined.

The subroutines are:

A. A basic identification algorithm: It estimates the unknown polynomials A_i, B_i, C, D and constants λ, κ in the structure

$$y(t) = \sum_{i=1}^{m} \frac{B_i(z^{-1})}{A_i(z^{-1})} u_i(t - \tau_i) + \lambda \frac{C(z^{-1})}{D(z^{-1})} e(t) + \kappa \qquad (1)$$

by maximizing the likelihood function, when polynomial orders and input delays τ_i have been specified. It also calculates the covariance matrix of the estimation errors. Maximizing the likelihood function is equivalent to minimizing a particular loss function $V(\theta)$, where θ is the collection of all unknown constant parameters except λ. The loss $V(\theta)$ is the sum of squared one-step-ahead prediction errors* or model residuals $\lambda e(t|\theta)$, defined by (1) for any given θ (Ref. 1). The routine consists essentially of two parts, alternately executed until little reduction in loss is received:

- A routine for evaluating the loss function and its first- and second-order derivatives at a given point θ^k.
- A hill-climbing routine, which calculates a new trial point θ^{k+1}. The term "hill-climbing" is used in spite of the fact that a minimum is being sought.

B. A routine testing for redundancy: It is executed when **A** has found a solution and decides, by chi-square tests, whether parameters that have been tagged as possibly redundant, are significantly different from

* This loss function may well be taken to define the estimate, if, for some reason, one does not believe in maximum likelihood or does not think that disturbances are normal.

zero. If not, the values are set to zero, and values of nonzero parameters are adjusted accordingly.

C. A routine setting orders and delays: Repeated executions of the sequence **C**, **A**, and **B** make a search routine in the space of order and delay values. Its purpose is to arrive at the lowest number of parameters consistent with a near-minimal loss. The function of **C** is to decide upon new trial settings or to make a final decision on orders and delays, while **A** and **B** determine the outcome of a trial setting.

When the search routine has accepted a model, the following two routines test the validity of certain basic assumptions for the identification:

D. A routine testing time-invariance: It identifies two models based on the first and second halves of the data sample and tests, by chi-square test, whether the two models deviate significantly.

E. A routine for testing disturbances: This routine examines the sequence of computed model residuals $e(t)$. Theoretically, these residuals should be normally distributed and uncorrelated. In practice normality is not required, but individual, large residuals, even if tolerated by the identification method, may indicate that something is wrong with the data at that point. Therefore, the routine checks the magnitudes and points out to the user any residuals greater than four standard deviations. This means tolerating larger errors than is conventional in statistical tests. However, a limit of four standard deviations satisfies the purpose of guarding against large errors in the data, and it gives an added safety against error indications in case the data sample should not quite behave according to theory (e.g., deviate from normality).

The identification routine describes the process behind the data sample by a model of the form 1), which is a system of difference equations. Often more conventional process characteristics are desired. Therefore, the program includes

F. A set of routines for analysis of the model: The routines derive from the model 1):

- A step response for each input
- A Bode diagram for each input
- A power spectrum of random disturbances
- A decomposition of the data sample into effect of inputs and disturbances
- A minimum-variance control law, including feedback and feed-forward terms (if any)
- The closed-loop step response
- The theoretical lower limit for the control error on this particular process.

Thus, the normal sequence of executions in a complete analysis of a data sample is **C**, **A**, **B**, **C**, **A**, **B**, $\cdots$, **D**, **A**, **A**, **E**, **F**. However, each execution of **A** results in a complete model of the form 1) with *specified* order. Therefore, one may use **A** alone in order to save computing and programming.

Determining orders and input delays

The order and (integer) delay parameters in the model 1) are determined in similar ways; "delays" τ_i are defined as the lowest powers in the polynomials $z^{\tau i}B_i(z)$, while "orders," denoted n_i^a, n_i^b, n^c, and n^d, are the highest powers in A_i, B_i, C, and D respectively. Further, the binary variable n^k, is introduced, which is zero if $\kappa = 0$ and unity if $\kappa \neq 0$.

The space of all possible combinations of integer order and delay parameters is separated into $m + 1$ subspaces, which are treated independently. Each subspace is spanned by a triplet (n_i^a, n_i^b, τ_i), $i = 1, \cdots, m$ or (n^c, n^d, n^k). This arbitrary separation is motivated chiefly by computing efficiency, but is further supported by the following reasoning: The concept behind the choice of the structure 1) is one of superposition of effects of a number of known independent variables and one unknown but independent disturbance, to form the observed output. Therefore, it is reasonable that a choice of order and delay parameters associated with a particular input is not influenced by those associated with other inputs or with the disturbance.

As stated, a search method is used to determine ,τ_i n_i^a, n_i^b, n^c, n^d, and n^k, and as such, defined by a measure to judge the outcome of a trial setting of order and delay parameters and a strategy for choosing a new trial setting. It is outlined below.

- *Measure of significance*

Obviously, there is a trade-off between number of parameters and resulting loss; increasing the number of parameters reduces loss. What one needs is to find a minimal number above which further loss reduction is small. This is achieved by formulating the problem in a probabilistic language; the approach has been used elsewhere:[8]

For long samples, and under the null-hypothesis that all orders are at least equal and all delays are at most equal to those of the true process, the reduction in loss received by increasing the total number of parameters from n to n' has a chi-square distribution with $n' - n$ degrees of freedom, if divided by the loss and multiplied by the length N of the sample.[8] Thus the range of probable values of a chi-square variable determines what reduction can be expected by increasing model orders above those of the true process; such a reduction is nonsignificant. Conversely, if the computed reduction is larger than that range, the reduction is significant and observations contradict the null hypothesis.

The range of the tolerance interval depends on the confidence one wishes to have in a possible rejection of the null hypothesis. It may be determined by the user by specifying the risk he is willing to run that the decision might be wrong.[11] This gives the possibility to influence the complexity of the final model; specifying a smaller risk for a decision "order is greater than n" to be wrong results in fewer such decisions and therefore a tendency towards low-order models.

With specified confidence the range is given by the definition of the chi-square variable.[12] In particular, a nonsignificant, relative loss reduction has mean $(n' - n)/N$ and variance $2(n' - n)/N$.

For a given setting (n_i^a, n_i^b, τ_i), $i = 1, \cdots, m$ and (n^c, n^d, n^k) the **B**-routine carries out a number of tests to determine the significance of the setting compared to lower-order alternatives. Thus, losses are compared for the nominal setting and a series of $8(m + 1)$ alternatives, determined by modifying the triplets $(n_i^a, n_i^b + \tau_i, \tau_i)$ $i = 1, \cdots, m$ by the 2^3 possible combinations of $\begin{Bmatrix} -1 & -1 & +1 \\ 0, & 0, & 0 \end{Bmatrix}$ and the triplet (n^c, n^d, n^k) by those of $\begin{Bmatrix} -1 & -1 & -1 \\ 0, & 0, & 0 \end{Bmatrix}$.

In this way the **B**-routine calculates the sensitivity of the loss function with respect to order and delay parameters. The result is obtained in the form of an indicator vector with $3(m + 1)$ components $\{I_1^a, I_1^b, I_1^\tau, \cdots, I_m^a, I_m^b, I_m^\tau, I^c, I^d, I^k\}$ stating whether the sensitivities with respect to individual components of the trial setting $\{n_1^a, n_1^b + \tau_1, \tau_1, \cdots, n_m^a, n_m^b + \tau_m, \tau_m, n^c, n^d, n^k\}$ are significant or not.

• *Search strategy*

This is heuristic and not claimed to be optimal in any sense. It has two functions: viz., to make decisions regarding the true order and delay parameters, and, if needed, to determine a new trial setting.

In order to formulate the search strategy, introduce as state variables for the search integer intervals $(0, r_i^a)$, (r_i^τ, r_i^b), $(0, r^c)$, and $(0, r^d)$, such that coefficients of polynomials A_i, $z^{\tau_i}B_i$, C, and D respectively are zero for powers outside the corresponding intervals. A sequence of decisions, based on information from the **B**-routine, narrow the intervals successively. The following decision rule is applied:

If $I_i^a = I_i^b = 0$, then $r_i^a = n_i^a - 1$
and $r_i^b = n_i^b + \tau_i - 1$.
If $I_i^\tau = 0$, then $r_i^\tau = \tau_i + 1$.
If $I^c = I^d = 0$, then $r^c = n^c - 1$
and $r^d = n^d - 1$.
Otherwise intervals are unchanged.

The rule is supported by the following reasoning: Multiplying by a common factor $(1 - \alpha z^{-1})$ numerator and denominator of a ratio of polynomials in the model 1) raises the order of the model without changing the loss. Conversely, a test result indicating two simultaneous nonsignificant high-order coefficients suggests that a common factor is present and, therefore, that the orders of the associated ratio are excessive. A single nonsignificant high-order coefficient, however, does not imply a common factor and therefore, necessarily, that the order of the associated polynomial is excessive; higher-order coefficients may be nonzero. (The rule may lead to a wrong decision under unfortunate circumstances. Even if the true orders are higher than those of the model, two simultaneous, zero high-order coefficients may occur accidentally for particular, isolated combinations of coefficient values. In such cases the search routine will fail and decide upon a model with too low order.)

The current values of the intervals thus carry the sum of the latest and previous decisions. Normally this sum is not sufficient to define the order and delay parameters uniquely. In that case the main rules for setting new orders and delays are:

n_i^a is incremented if $I_i^a = 1$ and $n_i^a + 1 \leq r_i^a$
decremented if $I_i^a = I_i^b = 0$
unchanged otherwise.
$n_i^b + \tau_i$ is incremented if $I_i^b = 1$ and $n_i^b + \tau_i + 1 \leq r_i^b$
decremented if $I_i^a = I_i^b = 0$
unchanged otherwise.
τ_i is incremented if $I_i^\tau = 0$
decremented if $I_i^\tau = 1$ and $\tau_i - 1 \geq r_i^\tau$
unchanged otherwise.
n^c is incremented if $I^c = 1$ and $n^c + 1 \leq r^c$
decremented if $I^c = I^d = 0$
unchanged otherwise.
n^d is incremented if $I^d = 1$ and $n^d + 1 \leq r^d$
decremented if $I^c = I^d = 0$
unchanged otherwise.
$n^k = 1$.

These rules state that polynomials are expanded, within the limits of the intervals, as long as expansions yield significant decrease in loss. They are reduced when test results indicate that reduction is feasible without significant increase in loss.

The main rule is modified somewhat to treat the cases when $B_i = 0$ or when n_i^b would otherwise become negative, indicating that no effect of the input variable u_i has been detected in the output.

The search is terminated when all order and delay parameters remain unchanged.

The search strategy described will normally result in a process with two phases which can be more or less pronounced, viz., increase of the total number of parameters, while loss decreases significantly, followed by elimination of a number, Δn say, of redundant parameters,

while loss V increases with a small amount $\approx V\ \Delta n/N$ (see also Fig. 2).

It is evident that good start values for the intervals contribute essentially to the efficiency of the search. The method is suited to a case where one has some a priori knowledge of the true order and delay parameters, but where this knowledge is uncertain. It is not suited and therefore inefficient when the process has large and unknown transport delays. In that case the routine will have to search over possible delay values, starting with zero and going upwards. Since a model must be built for each delay value in the search, this will take time.

Hill climbing

The hill-climbing routine is based on a Newton-Raphson algorithm; however, a number of modifications are temporarily employed in situations where the algorithm can otherwise be expected to fail.

The Newton-Raphson algorithm fits a quadratic surface to the function $V(\theta)$ by equating function values V, first order derivatives V_θ, and second order derivatives $V_{\theta\theta}$ at the current point θ^k, and choosing the next point θ^{k+1} as the (nearest) stationary point on the quadratic surface. The algorithm is thus

$$\theta^{k+1} = \theta^k - K(\theta^k)\, V_\theta(\theta^k), \tag{2}$$

where $K(\theta) = V^\dagger_{\theta\theta}(\theta)$. The matrix $V^\dagger_{\theta\theta}$ is the so-called pseudo-inverse of $V_{\theta\theta}$. It is equal to $V^{-1}_{\theta\theta}$, whenever the ordinary inverse exists.

The Newton-Raphson algorithm converges to a minimum of $V(\theta)$, if the start value θ^0 is sufficiently close. It may also converge from a far-off starting point, if the function is sufficiently "well-behaved." A sufficient condition is that $V_{\theta\theta}(\theta^k)$ be non-negative definite, which means that the surface must not curve downwards in any direction. Otherwise (2) may converge to any stationary point, or it may diverge.

Therefore, the routine includes a number of tests in order to detect when $V(\theta)$ is not well behaved, so that (2) must be modified. The following two properties of the algorithm (2) are the base for such modifications:

i) The algorithm (2) converges to a minimum for any K that is non-negative definite and sufficiently small.
ii) The algorithm (2) converges rapidly if K is also close to $V^\dagger_{\theta\theta}$.

The idea is thus to approximate $V_{\theta\theta}$ by a non-negative definite matrix, whenever $V_{\theta\theta}$ is not non-negative definite by itself.

The tests and the modifications of (2), possibly following the tests, are an attempt to automate the actions a "man in the loop" may take to overcome the difficulties met in practice, when hill-climbing on the likelihood surface. The tests correspond to the judgments the man makes, and the various modifications correspond to the set of alternative strategies he can use. Specifically, the routine deviates from the normal course of Newton-Raphson hill-climbing, whenever one or more of the following difficulties are met:

- *Unfavorable curvature*

This is indicated when $V_{\theta\theta}(\theta^k)$ is not non-negative definite. Now, this matrix is the sum of two terms.[2,8]

$$V_{\theta\theta}(\theta) = \lambda^2 \sum_{t=1}^{N} e_\theta(t) e^T_\theta(t) + \lambda^2 \sum_{t=1}^{N} e(t) e_{\theta\theta}(t), \tag{3}$$

of which the first one is always non-negative definite. Further, the sequence of $e(t)e_{\theta\theta}(t)$ is uncorrelated and has zero mean for θ equal to the true parameter vector, i.e., near the minimum. The second term therefore becomes relatively less important as the sample length N increases, and the routine replaces $V_{\theta\theta}$ by its first term $V^*_{\theta\theta}$, when the complete second-derivative matrix is not non-negative definite. $V^*_{\theta\theta}$ is further always employed whenever $(\theta^{k-1} - \theta^k)$ $V_\theta(\theta^{k-1}) >$ a constant, i.e., outside the immediate vicinity of the minimum.

For θ-values far from the minimum neither $V_{\theta\theta}$ nor $V^*_{\theta\theta}$ necessarily gives a good estimate of step length and direction towards the minimum; however, using a non-negative definite matrix guarantees that a step is taken in the direction of decreasing loss. Also, $V^*_{\theta\theta}$ requires less computing per iteration. In fact, the second term of (3) is needed only to ensure a faster-than-linear convergence rate towards the end of the hill-climbing. The value of the constant normally affects the computing time only; a low value reduces the average amount of computing per iteration but tends to increase the number of iterations, a high value has the opposite effects. A compromise, if required, must be determined empirically.

- *Singularity*

The difficulty arises when $V_{\theta\theta}$ (or $V^*_{\theta\theta}$) contains linearly near-dependent columns. This would have two effects on a Newton-Raphson hill-climbing:

i) The routine inverting $V_{\theta\theta}$ would fail due to round-off errors.
ii) Even if $V_{\theta\theta}$ could be inverted the hill-climbing would take a very long step along what is estimated to be a "valley." This would be unfortunate in cases where the valley reflects only a local property of the function $V(\theta)$.[13]

In this case $V_{\theta\theta}$ is approximated by a matrix having exactly dependent columns. This matrix is pseudo-inverted. The difference between the two inverses is, geometrically, that while near a slowly descending valley the true inverse aims at the absolute minimum, essentially stepping along

the valley, the pseudo-inverse locates the relative minimum perpendicular to the valley.

Since the two solutions are very different, the step taken depends critically on what is meant by "near-dependence." In order to motivate the definition used for the present purpose, introduce the diagonal matrix Q of square roots of diagonal elements in $V_{\theta\theta}$. Also, factorize $V_{\theta\theta} = QSS^TQ$, where S is a left-triangular matrix.[9] The rank of $V_{\theta\theta}$ is revealed by the diagonal elements of S. In particular, the rank is n-p if and only if p diagonal elements are zero. The corresponding columns in S are then undetermined and may be set to zero.

Now, suppose the true inverse of a nonsingular or near-singular matrix $V_{\theta\theta}$ were used to calculate a step towards the minimum. Then the step taken would be

$$h = -V_{\theta\theta}^{-1}V_\theta = -Q^{-1}S^{T^{-1}}S^{-1}Q^{-1}V_\theta. \tag{4}$$

If, particularly, θ is equal to the true parameter values, then for large N, $EV_\theta V_\theta^T \sim \lambda^2 V_{\theta\theta}$ (Refs. 2 and 8), and hence $V_\theta = \lambda QSw$, where w is a random vector such that $Ew = 0$ and $Eww^T = I$. The step taken from a hypothetical position defined by the true parameter values would be

$$h = -\lambda Q^{-1}S^{T^{-1}}w. \tag{5}$$

It follows from (4) that when a diagonal element s_{ii} is small, the step h_i taken in the direction i will generally be large. Also, Eq. (5) indicates that this will be the case also if the value of the loss function is near minimum; i.e., the routine will step along a valley. Therefore the following definition suits the purpose of evading the effect of ii):

> The non-negative definite matrix $V_{\theta\theta}$ has p linearly near-dependent columns if p diagonal elements of S are smaller than a constant ρ. A corresponding approximation of $V_{\theta\theta}$ of rank $n - p$ is defined by $QS_pS_p^TQ$, where S_p is obtained during factorization of $Q^{-1}V_{\theta\theta}Q^{-1}$ by zeroing the p columns for which $s_{ii} < \rho$.

The following reasoning serves to assess the effect of using S_p in place of S and thus to give some guidance in assigning a value to the constant ρ. Similar arguments have been used in Ref. 14, but based on an expansion of $V_{\theta\theta}$ in terms of eigenvectors and eigenvalues instead of the present factorization into triangular matrices. Assume that one eigenvalue in $V_{\theta\theta}^{-1}$ dominates. Then one diagonal element in S is essentially smaller than the others. Rearrange rows and columns in $V_{\theta\theta}$ so that in S the smallest diagonal element appears in the last place. Then from (5)

$$h_n = -\lambda q_{nn}^{-1}s_{nn}^{-1}w_n.$$

Now, the product $-\lambda q_{nn}^{-1}w_n$ would be the step taken if only θ_n were allowed to vary. The factor s_{nn}^{-1} may therefore be interpreted as an amplification of the step length due to the influence of near-dependent variables. By limiting diagonal elements in S to those $< \rho$ (and pseudo-inverting) a bound is set to large amplifications of step length in directions nearly perpendicular to the direction of the slope. Thus the value of ρ determines how narrow the (inverted) "hill" is allowed to be before it is treated as a "valley."

The value of ρ is set primarily to eliminate the effect of ii). To ensure that i) is also avoided the routine estimates the effect of round-off errors, and inverts if the effect is tolerable. Otherwise it increases the constant and repeats.

The concept of near-dependence is introduced here to amend computational difficulties in the hill-climbing. Although related to, it must not be confused with the statistical definition of dependence between the components of the estimate θ. The latter dependence is established by chi-square tests (see the preceding section).

• *Instability*

The new point θ^{k+1} calculated from (2) may define an unstable or otherwise unacceptable model. There are two kinds of instability, viz., numerical instability of the algorithm calculating $V(\theta)$, and instability of the model. None appears as long as all zeroes of the polynomials A_i, C, and D fall inside the unit circle. Hence, for each order of a polynomial there is a fixed region of coefficient values inside which a polynomial is acceptable, and the region of admissible θ is the logical product of the regions for all polynomials. The boundaries of the stable regions for polynomials are linear for first and second orders and linear or curved for higher orders.

When a nominal θ^{k+1} has been calculated, the algorithm tests whether it falls inside the admissible region. If not, the boundary first crossed by a straight line connecting θ^k and θ^{k+1} is introduced as a constraint on the approximating quadratic surface, i.e., a minimum is sought on the boundary. For a linear boundary the minimum is calculated explicitly; for a nonlinear boundary an iterative procedure is used, where the nonlinear boundary is substituted by a sequence of tangent hyperplanes.

For θ-values outside the stable region the likelihood function may take on very large values. Since the function is analytic, this means that the derivatives may have large values also on the boundary and immediately inside, in particular that $V_{\theta\theta}$ changes fast with θ. This is a severe condition for a Newton-Raphson algorithm, which, assuming constant $V_{\theta\theta}$, then does not work well. For this reason it is undesirable to have an approximating point θ^k on the boundary (unless it happens to coincide with the minimum), and the routine simply reduces the step taken by a factor of 0.9, whenever a boundary has been hit. Thus a boundary may be approached iteratively but not reached in a single step.

The test on admissibility is repeated for the new, constrained point, and if this is also outside the admissible region, the minimum of the quadratic approximation is further constrained by an additional boundary. Constraints are introduced only when the inadmissible region would otherwise be entered.

Thus, the hill-climbing may converge to an unconstrained point inside the region or to a constrained point on the boundary. In the latter case the algorithm in effect reduces the number of free parameters (by introducing constraints) and this shows up in the covariance matrix of the estimation error, which then becomes singular.

• *Unsatisfactory reduction of loss*

The new loss $V(\theta^{k+1})$ may turn out to be not appreciably lower, or even higher than the previous $V(\theta^k)$. For a truly quadratic function the reduction would be

$$\delta = \tfrac{1}{2}(\theta^k - \theta^{k+1})^T V_\theta(\theta^k).$$

The routine tests whether at least half of this reduction has actually been reached. If this is not the case, then the minimum has been overshot, since for a short enough step the function is always decreasing by 2δ.

In the case of unsatisfactory reduction the Newton-Raphson hill-climbing is temporarily inhibited, and a new point is selected on the straight line connecting θ^k and θ^{k+1}. To do this the routine introduces a scalar parameter x defined on this line:

$$\theta(x) = (1 - x)\,\theta^k + x\theta^{k+1}$$

and fits a four-parameter curve $c_0 + c_1 x + c_2 x^{c_3}$ to values and slopes of $V[\theta(x)]$ in θ^k and θ^{k+1}. It selects the new point θ^{k+2} as the minimum of the approximating curve, provided the value of x falls inside the interval (0.25, 0.9). Otherwise the nearest end point of the interval is used. The strategy is useful in cases where $V_{\theta\theta}$ changes fast, e.g., when the minimum locates near a stability boundary.

• *Start values*

For fixed A_i-, C-, and D-polynomials the loss function is quadratic in the coefficients of the B_i-polynomials, and the so-constrained hill-climbing will find the minimum in one step from any starting point. This is utilized to get an improved start value for the unconstrained hill-climbing. Thus, in the first step of the hill-climbing the coefficients of A_i, C, and D are locked to their initial values (normally zero), and B_i-coefficients only are free to vary.

Performance of the hill-climbing routine

The modeling algorithm has been applied to a number of data samples, some generated artificially, some collected from a paper making process. In most cases the order-determining routine was used so that no additional information was given to the computer program (except number of input variables). This means that several models have been identified on the same sample, and also that the set of models formed on the test data is a mixture of high-, low-, and correct-order models with correct and incorrect transport delays. The high-order models are difficult, since they make the loss functon singular.

Although only the final model in each sequence is generally acceptable, i.e., has the right delays and no redundant parameters, the collection of all models illustrates the performance of the hill-climbing routine. However, it is important to note that, while the starting point θ^0 for generating the first model in a series was zero, those of the later models were often good, since the algorithm automatically sets start values for the hill-climbing utilizing the preceding model.

• *Test data*

The test samples (length: 200 to 500 points) were produced as follows:

1) Artificially generated, using models of the form 1):
   28 models with two input variables.
2) Pure sine wave:
   4 models with no input (spectral analysis)
3) Recorded from a paper making process, drying:
   6 models with one input, 29 models with two inputs,
   7 models with four inputs.
4) Recorded from a paper making process, sheet forming:
   13 models with one input.
5) Recorded from a paper making process, pulp refining:
   7 models with two inputs.
6) The fluctuations of the stock-market value of a share were used as data:
   3 models with no input.
   The total number of models tried was 97.

Of course, there is no guarantee that the test cases are in any sense representative for the set of "industrial data." A common characteristic has been unmeasurable disturbances of the same order of magnitude as the effect of inputs and containing various frequencies. This is the case for which the method was particularly constructed, and where the theoretical advantages are. Large disturbances, however, tend to result in low-order models, since a possible fine structure of the process dynamics is drowned in noise.

• *Test results*

The hill-climbing routine converged to a minimum in all test cases.

The routine occasionally utilized one or more of the modifications of the Newton-Raphson hill-climbing algorithm, designed to treat difficult situations. The

following statistics give the number of times a particular difficulty has been met:*

Unfavorable curvature:	113
Near-singular loss function ("valley"):	116
Inadmissible region entered:	105
Unsatisfactory reduction of loss:	62
Total number of iterations:	720

The number of iterations needed for a hill-climbing is summarized for the test material in Fig. 1. As seen, there is no clear dependence between number of parameters and number of iterations needed. The average number of iterations over the test material has been seven. The average number of parameters has also been seven.

Figure 2 illustrates how the modeling proceeded in a particular case. It shows the loss as a function of number of iterations for two sequences of models: a complicated case with four input variables, and a simpler case with two input variables. The data are the same, so that in the latter case two input variables have been deleted. Any effect of those variables then adds to the disturbance, resulting in a higher final loss. At each break in the loss curve the program has changed the orders and/or delays of the model and therefore the number of parameters. This number is noted below each curve segment. The number of data points is 284.

Comment on Fig. 2: The algorithm began with a model of few parameters and then increased the number, thus the decrease of loss. When increase of order no longer yielded a reduction of loss, the algorithm started reducing the complexity of the model. The loss remained roughly the same and even increased slightly. The curves illustrate a few general properties of the modeling routine, when this has to determine orders and delays. As a rule the later models require few iterations, since the start values for the hill-climbing routine (i.e., parameter values for the preceding model) are good. Models with redundant parameters constitute difficult cases and therefore require many iterations in spite of the fact that little reduction in loss may be received. Early models also require many iterations, partly because of poor start values and partly because the model orders/delays are wrong, and the model therefore cannot be fitted well to the data.

Conclusions

The maximum likelihood method of identification is a general black-box identification method for discrete-time data and, since there are other such methods, it is appropriate to list its relative merits (and drawbacks) for those who may consider using it. The list expresses

* *Note:* During the tests the admissible region of model parameters was restricted further, so that besides unstable solutions certain kinds of stable but unwanted solutions were also excluded. The number of unstable cases is therefore less than 105.

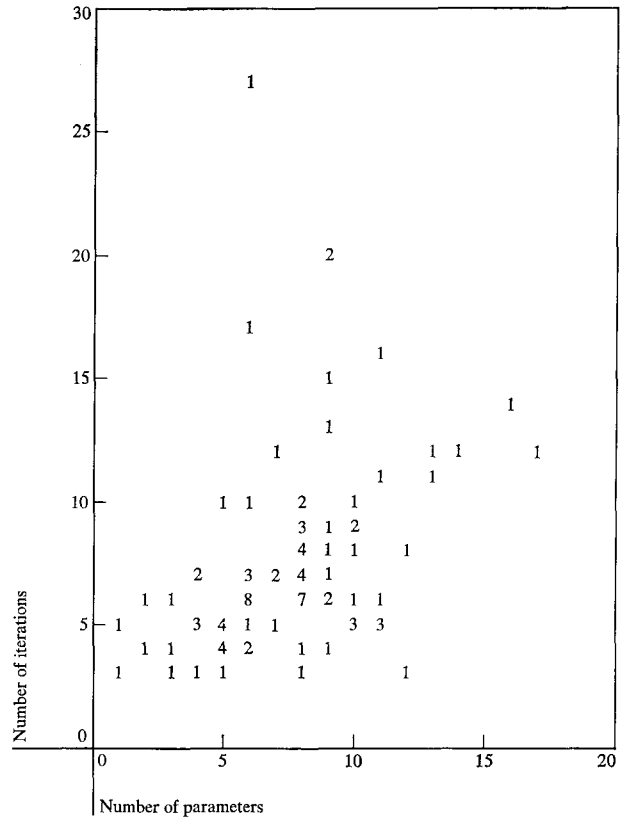

Figure 1 Total number of models identified on the test material distributed over number of parameters employed by the order-setting routine and number of iterations needed by the hill-climbing routine.

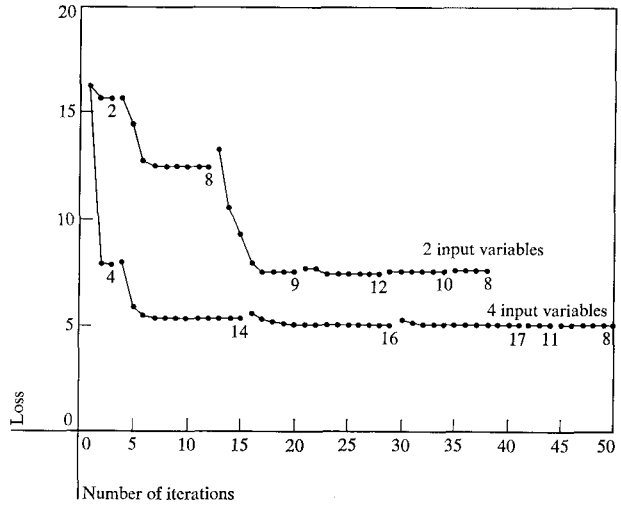

Figure 2 Modeling loss as a function of number of steps of iteration for two test cases.

both theoretical results and experience gained when using the method. The statements are regrettably not precise, since they are brief.

Application is suitable:

In cases where random disturbances are a concern.
When the process has a pronounced dynamic behavior and/or responds slowly.
When the process cannot be subject to experiment at the user's convenience.
When the purpose of identification is for automatic control.

Features of the method are:

It covers all linear, time-invariant and finite-order models.
A few inputs, a single output.
Arbitrary input sequences.
Arbitrary disturbance spectrum.
Disturbance characteristics are estimated.
Model accuracy is estimated.
The minimum-variance control law follows easily from the model.
Order tests are feasible.
Search for transport delays is feasible, although cumbersome.
Basic assumptions are possible to check.
Arbitrary parameter constraints are feasible.

Performance is illustrated by the following properties:

Quality of model:

For long samples the estimates are unbiased and have the theoretically lowest possible variances.
Acceptable results have been obtained also for industrial data corrupted by large and irregular disturbances.
A minimum number of parameters is used.

Reliability and speed of algorithm:

Hill-climbing is used; computing time is unpredictable.
Computing time per step is approximately proportional to nN (for large N), where n is number of parameters and N is length of sample.
Efficiency is believed to be high for a hill-climbing routine, when orders and transport delays are specified.
Efficiency is probably low, when transport delays are large and unspecified.
Reliability is high.

Ease of use:

Identification may be completely computerized.
In its most developed form the modeling needs no a priori specifications.
Diagnostics are feasible to detect improper use.
Conventional model characteristics are feasible for comparison with experience.

"Speed" and "ease of use" may partly be traded against each other. It generally holds for the method that the more skill one acquires in handling it, the less computing one needs. If one is ignorant, one has to pay for this with longer computing times. But this is also an asset; for it may be interesting to note that by automating the method sufficiently, it is possible to a certain extent to substitute money for knowledge.

Acknowledgments

The author wishes to thank those who participated in the various phases of developing the method. The theory and basic algorithm were developed jointly with Prof. K. J. Åström, who also supervised the work until 1966. A major contribution was made by S. Wensmark, who programmed and carried out most of the tests and also wrote approximately half of the 6000 FORTRAN statements of the final modeling program. B. Pehrson contributed by programming and W. Tuel by a formal user's evaluation. Thanks also go to Billerud Paper Company for supplying the process and computer control system (and permission to experiment) needed for the tests, in particular to O. Alsholm, who had the burden of responsibility in case something went wrong.

The work was done as part of the research activity of the IBM Nordic Laboratory, Sweden, for the advancement of computer control in the process industries.

References

1. K. J. Aström and T. Bohlin, "Numerical Identification of Linear Dynamic Systems from Normal Operating Records," *Proc. Theory of Self-Adaptive Control Systems,* Teddington, 1965. Published by Instrument Society of America, P. H. Hammond (editor). Also IBM Nordic Laboratory Report TP 18. 159.
2. K. J. Aström, T. Bohlin, and S. Wensmark, "Automatic Construction of Linear Stochastic Dynamic Models for Stationary Industrial Processes with Random Disturbances Using Operating Records," IBM Nordic Laboratory Report TP 18. 150, 1965.
3. K. J. Aström, "Computer Control of a Paper Machine—an Application of Linear Stochastic Control Theory," *IBM J. Res. Develop.* **11,** 389 (1967).
4. K. J. Aström and T. Bohlin, "Integrated Computer Control of a Kraft Paper Mill—New Methods for Control Strategy Design in Operation," IBM Nordic Laboratory Report TP 18. 172, 1966.
5. T. Bohlin, "Paper Machine Identification for Purposes of Computer Control," Paper- Rubber- and Plastics-Automation Congress, Antwerp, 1966.
6. K. J. Aström, "Notes on the Regulation Problem," IBM Nordic Laboratory, Technical Report CT 211, 1965.
7. P. A. N. Briggs, D. W. Clarke, and P. H. Hammond, "Introduction to Statistical Identification Methods in Control Systems," *Control* **12,** 233 (1968).
8. K. J. Aström, "On the Achievable Accuracy in Iden-

tification Problems," IFAC Symposium on Identification in Automatic Control Systems, Prague, June 1967. Preprints.
9. V. N. Faddeeva, *Computational Methods of Linear Algebra.* Dover, New York, 1959.
10. D. W. Clarke, "Generalized Least Squares Estimation of the Parameters of a Dynamic Model," IFAC Symposium on Identification in Automatic Control Systems, Prague, June 1967. Preprints.
11. M. G. Kendall and S. Stuart, *The Advanced Theory of Statistics,* vol. 2, Griffin, London, 1961.
12. M. G. Kendall and S. Stuart, *The Advanced Theory of Statistics,* vol. 1, Griffin, London, 1958.
13. D. J. Wilde, *Optimum Seeking Methods,* Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
14. Greenstadt, J., "On the Relative Efficiencies of Gradient Methods," *Math. Comp.* **21,** 360 (1967).

Received April 22, 1969