Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations

Read

NARENDRA K. GUPTA, MEMBER, IEEE, AND RAMAN K. MEHRA, MEMBER, IEEE

Abstract—**This paper discusses numerical aspects of computing maximum likelihood estimates for linear dynamical systems in state-vector form. Different gradient-based nonlinear programming methods are discussed in a unified framework and their applicability to maximum likelihood estimation is examined. The problems due to singular Hessian or singular information matrix that are common in practice are discussed in detail and methods for their solution are proposed. New results on the calculation of state sensitivity functions via reduced order models are given. Several methods for speeding convergence and reducing computation time are also discussed.**

I. Introduction

THE use of maximum likelihood estimation in practice leads to difficult nonlinear programming problems. It is not uncommon to find situations where the likelihood surface has multiple maxima, saddle-points, discontinuities, and singular Hessian in the parameter space. The application of the steepest descent method leads to extremely slow convergence rate and the straightforward application of the Newton–Raphson and the Gauss-Newton methods may lead either to no convergence or convergence to wrong stationary points. From a statistical viewpoint, only the absolute maximum of the likelihood function provides an unbiased, consistent, and efficient estimate. Thus, it is important to locate the true maximum of the likelihood function. It should be pointed out that the difficulties in obtaining the absolute maximum of the likelihood function do not invalidate the likelihood principle, as has been convincingly argued by Bernard [1] and Edwards [2]. The maxima of the function and its partial derivatives are evaluates which should be looked upon as only summary information about the function. In cases where it is difficult to obtain these evaluates, one may still use the values of the likelihood function at different points in the parameter space to make relative statements about the likelihood of one parameter value versus the other.

Inadequate model specification and parameterization may cause anomalies in the likelihood function. A common problem in parameter identification is overparameterization leading to singular or nearly singular information matrix, since it is difficult to determine *a priori* whether or not a given sample has adequate information for estimating all parameters in the model. Underparameterization, on the other hand, may lead to spurious local maxima and saddle points. Sometimes the likelihood function increases up to a maximum value and then immediately falls to minus infinity, see Edwards [2]. These problems can often be avoided by changing the model or by changing the parameterization. These problems have also been considered by Åström and Bohlin [3] and Bohlin [4].

The likelihood function is often maximized using a gradient approach starting from some *a priori* values. Good starting values are important to ensure convergence to the absolute maximum.

The computation of the gradients of the likelihood function for parameters of a dynamic system may require the determination of the sensitivities of the system state to unknown parameters. This is usually the most time consuming step in the computation. From a practical viewpoint the computation of state sensitivities using reduced-order models is important in parameter estimation particularly in high-order systems with many unknown parameters. In linear systems these techniques allow a considerable saving in computation time which makes the estimation of parameters from data feasible for practical systems. Åström and Bohlin [3] have given some techniques for carrying out this reduction for single input-single output systems in a canonical form. However, for multiinput-multioutput systems, most efforts to date [5]–[7] have concentrated on finding bounds on the order of the model which can generate states sensitivities for all system parameters. Very little attention has been paid to the formulation of practical techniques leading to these lowest order models for general state vector representations. Formulations by Wilkie and Perkins [5], Denery [6], and Neuman and Sood [7] involve state transformations and do not fully exploit the characteristics of the system in most cases.

A practical method for obtaining lowest order models for sensitivity functions computations is outlined here. The technique makes full use of special system characteristics and has general application to high-order systems with a large number of unknown parameters.

The paper is organized as follows. Section II gives the likelihood function for dynamic systems. Section III de-

Manuscript received January 23, 1974; revised July 30, 1974. This work was supported in part by NASA, Edwards Flight Research Center under Contract NAS4-2068 and in part by the Office of Naval Research under Contract N00014-72-C-0328.

N. K. Gupta is with Systems Control, Inc., Palo Alto, Calif.
R. K. Mehra is with the Division of Engineering and Applied Physics, Harvard University, Cambridge, Mass.

tails the nonlinear programming methods which can be used to maximize the likelihood function. Various modifications to handle nearly singular information matrices are given in Section IV. Section V deals with the computation of gradient and Hessian of the cost function. The sensitivity function reduction technique for single-input and multi-input systems are given in Section VI. Section VII gives some approximation techniques and Section VIII states the conclusions.

II. Likelihood Function for Dynamic Systems

Consider the linear representation of a dynamic system, in which the state x ($n \times 1$ vector) obeys the differential equation

$$\dot{x}(t) = Fx(t) + Gu(t) + \Gamma w(t), \qquad 0 \le t \le T$$
$$E\{x(0)\} = x_0 \quad \text{and} \quad E\{(x(0) - x_0)(x(0) - x_0)^T\} = P_0 \quad (1)$$

and measurements of p linear functions of the state variables are taken at discrete times t_k

$$y(t_k) = Hx(t_k) + v(t_k), \qquad k = 1,2,\cdots,N \quad (2)$$

where $u(t)$ is a $q \times 1$ vector of deterministic input and $w(t)$ and $v(t_k)$ are uncorrelated Gaussian white noise sources. The power spectral density of w is Q and the covariance matrix of $v(t_k)$ is R. θ is the vector of m unknown parameters in F, G, H, Γ, Q, R, x_0, and P_0.

In the parameter estimation problems when the maximum likelihood method is used, it is usually more convenient to work with the negative of the logarithm of the likelihood function. It is possible to do so because the logarithm is a monotonic function. It can be shown [9] that the negative log-likelihood function $J(\theta)$ is

$$J(\theta) = \tfrac{1}{2} \sum_{i=1}^{N} \{\nu^T(t_i,\theta)B^{-1}(t_i,\theta)\nu(t_i,\theta) + \log |B(t_i,\theta)|\} \quad (3)$$

where

$$\nu(t_i,\theta) = y(t_i) - E\{y(t_i)|y(t_{i-1}), y(t_{i-2}) \cdots y(t_1)\} \quad (4)$$

and

$$B(t_i,\theta) = E\{\nu(t_i,\theta)\nu^T(t_i,\theta)\}. \quad (5)$$

Here $\nu(t_i,\theta)$ and $B(t_i,\theta)$ denotes the innovations and their covariances which may be obtained from the Kalman filter equations for the system (1) and (2), given here.

Prediction

$$\frac{d}{dt}\hat{x}(t|t_{j-1}) = F\hat{x}(t|t_{j-1}) + Gu(t), \qquad \hat{x}(t_0|t_0) = x_0$$

$$\frac{d}{dt}P(t|t_{j-1}) = FP(t|t_{j-1}) + P(t|t_{j-1})F^T + \Gamma Q\Gamma^T,$$
$$P(t_0|t_0) = P_0, \; t_{j-1} \le t \le t_j. \quad (6)$$

Measurement Update

$$K(t_j) = P(t_j|t_{j-1})H^T(HP(t_j|t_{j-1})H^T + R)^{-1}$$
$$\hat{x}(t_j|t_j) = \hat{x}(t_j|t_{j-1}) + K(t_j)\{y(t_j) - H\hat{x}(t_j|t_{j-1})\}$$
$$P(t_j|t_j) = \{I - K(t_j)H\}P(t_j|t_{j-1})$$
$$\nu(t_j) = y(t_j) - H\hat{x}(t_j|t_{j-1})$$
$$B(t_j) = HP(t_j|t_{j-1})H^T + R. \quad (7)$$

We then minimize $J(\theta)$ with respect to θ subject to the Kalman filter constraints (6) and (7). It is a nonlinear programming problem and we discuss below various methods for solving it.

Remark: It is important to have good starting values of the parameters since this considerably improves the probability of convergence and of locating the absolute maximum of the likelihood function. It would also be useful to determine if an appropriate model is being used and all parameters are identifiable from the data. The start-up procedure should provide unbiased and consistent estimates of the parameters. Generally, this can be done by using a least-squares type of method. Then the likelihood function is maximized by using an iterative scheme that keeps the estimate unbiased and consistent. An example of this procedure is contained in Dhrymes, Klein, and Stieglitz [11].

III. Nonlinear Programming Methods

This section and the next one are a summary of the gradient-based nonlinear programming methods[1] that have been used for computing maximum likelihood estimates.

The basic iteration in gradient-type nonlinear programming methods is

$$\theta_{i+1} = \theta_i - \rho_i R_i g_i \quad (8)$$

where θ_i is the parameter vector at the ith iteration, g_i is a vector of gradients of the negative log-likelihood function $J(\theta)$, i.e.,

$$g_i = \left.\frac{\partial J}{\partial \theta}\right|_{\theta=\theta_i}.$$

R_i is an approximation to the second partial matrix $(\theta^2 J/\partial\theta^2)^{-1}|_{\theta=\theta_i}$ and ρ_i is a scalar step size parameter chosen to ensure that $J(\theta_{i+1}) < J(\theta_i) - \epsilon$ where, ϵ is a positive number that can be chosen in a variety of ways (see Polak [13]). The class of nonlinear programming methods to be discussed here differ mainly in their selection of R_i, and in some cases ρ_i and g_i. It is shown in Luenberger [14] that the convergence rate near the minimum for algorithm (8) with ρ_i chosen by a one-dimensional search is

$$J(\theta_{i+1}) \le \left(\frac{\mu_{\max} - \mu_{\min}}{\mu_{\max} + \mu_{\min}}\right)^2 J(\theta_i) \quad (9)$$

where $\mu_{\max}$ and $\mu_{\min}$ are the maximum and minimum

[1] Nonlinear programming methods based on function evaluation alone [38], [39] are not described since their use for parameter estimation has been limited and they do not provide easy checks for identifiability. However, for systems with large numbers of parameters, these methods, including random search methods, may prove quite effective [12].

eigenvalues of $R_i(\partial^2 J/\partial\theta^2)$. It is clear from (9) that best convergence is achieved by making R_i as nearly as possible equal to $(\partial^2 J/\partial\theta^2)^{-1}$.

A. Newton–Raphson (NR) Method

In this method, R_i is chosen as $(\partial^2 J/\partial\theta^2)^{-1}|_{\theta=\theta_i}$ and $\rho_i = 1$ except when this choice of ρ_i gives an increase in cost. When this method converges, the convergence is quadratic; however, the method has the following drawbacks:

1) It fails to converge to the desired optimum whenever $(\partial^2 J/\partial\theta^2)$ has some negative eigenvalues.

2) If $(\partial^2 J/\partial\theta^2)$ is nearly singular, there are numerical problems in inverting it. This may result in slow or no convergence at all.

3) Generally, the computation of $(\partial^2 J/\partial\theta^2)$ is very expensive. The Gauss–Newton method described below which uses an approximation to $(\partial^2 J/\partial\theta^2)$ is much more efficient.

For the above reasons, the NR method is generally not used in parameter estimation problems. The drawbacks 1) and 2) may be remedied by the methods described in Section V for the Gauss–Newton method, but the extra computation involved in computing the exact Hessian may not pay off in terms of the improved convergence rate, particularly when the parameter values are far from true values.

B. Gauss–Newton (GN) Method ✓

In this method one chooses R_i as the inverse of the Fisher information matrix M_i where

$$M_i = E\left[\frac{\partial^2 J}{\partial\theta^2}\bigg|_{\theta=\theta_i}\right] = E\left[\left(\frac{\partial J}{\partial\theta}\right)\left(\frac{\partial J}{\partial\theta}\right)^T_{\theta=\theta_i}\right]. \quad (10)$$

The expectation is taken over the whole sample space. M_i is a nonnegative definite symmetrix matrix. In statistical literature, the above technique is known as the "Method of Scoring" [15] and in the control literature it has been called modified Newton–Raphson, quasilinearization, and differential corrections in somewhat different contexts.

Using (3), we can write

$$\frac{\partial J}{\partial\theta(k)}\bigg|_{\theta=\theta_i} = \sum_{t=1}^{N}\left\{\nu^T B^{-1}\frac{\partial\nu}{\partial\theta(k)} - \tfrac{1}{2}\nu^T B^{-1}\frac{\partial B}{\partial\theta(k)}B^{-1}\nu + \tfrac{1}{2}\operatorname{tr}\left(B^{-1}\frac{\partial B}{\partial\theta(k)}\right)\right\}\bigg|_{\theta=\theta_i} \quad (11)$$

$$\begin{aligned} M_i(j,k) &= E\left\{\frac{\partial J}{(\partial\theta(j))}\left(\frac{\partial J}{\partial\theta(k)}\right)^T\bigg|_{\theta=\theta_i}\right\} \\ &= \sum_{t=1}^{N} E\left\{\left[\left(\frac{\partial\nu}{\partial\theta(j)}\right)^T B^{-1}\frac{\partial\nu}{\partial\theta(k)}\right] + \tfrac{1}{2}\operatorname{tr}\left[B^{-1}\frac{\partial B}{\partial\theta(j)}B^{-1}\frac{\partial B}{\partial\theta(k)}\right] + \tfrac{1}{4}\operatorname{tr}\left(B^{-1}\frac{\partial B}{\partial\theta(j)}\right)\operatorname{tr}\left(B^{-1}\frac{\partial B}{\partial\theta(k)}\right)\right\}_{\theta=\theta_i} \end{aligned} \quad (12)$$

where the arguments of ν and B are not written explicitly and $\theta(j)$ is the jth component of θ vector. In the Gauss–Newton method M_i is generally estimated from the sample as

$$\begin{aligned} \hat{M}_i(j,k) = \sum_{t=1}^{N}\Bigg\{&\left(\frac{\partial\nu}{\partial\theta(j)}\right)^T B^{-1}\frac{\partial\nu}{\partial\theta(k)} \\ &+ \tfrac{1}{2}\operatorname{tr}\left[B^{-1}\frac{\partial B}{\partial\theta(j)}B^{-1}\frac{\partial B}{\partial\theta(k)}\right] \\ &+ \tfrac{1}{4}\operatorname{tr}\left(B^{-1}\frac{\partial B}{\partial\theta(j)}\right)\operatorname{tr}\left(B^{-1}\frac{\partial B}{\partial\theta(k)}\right)\Bigg\}\bigg|_{\theta=\theta_i}. \end{aligned} \quad (13)$$

An exact expression for M_i derived in [16] may also be used. Moreover, it can be precomputed for a given value of θ. In (11) and (13), $(\partial\nu/\partial\theta(j))$ and $(\partial B/\partial\theta(j))$ are calculated by solving linear differential equations, also known as "sensitivity equations" (cf. Section VI). Notice that $\hat{M}$ does not require calculation of $(\partial^2\nu/\partial\theta(j)\partial\theta(k))$ and $(\partial^2 B/\partial\theta(j)\partial\theta(k))$.

The Gauss–Newton method is probably the most commonly used method for maximum likelihod estimation. Since $\hat{M}_i$ is nonnegative definite, one can always find a ρ_i such that $J(\theta_{i+1}) < J(\theta_i)$. The method, however, does run into problems when $\hat{M}_i$ is singular or nearly singular. In this case, two problems arise.

1) The computed $R_i = \hat{M}_i^{-1}$ may turn out to be indefinite or negative-definite, so that $J(\theta_{i+1}) > J(\theta_i)$ for all $\rho_i > 0$.

2) The step size $\Delta\theta_i = \rho_i R_i g_i$ is very large in singular directions. This can be seen by a singular value decomposition of $\hat{M}_i$ and R_i

$$\begin{aligned} \hat{M}_i &= \sum_{j=1}^{m}\lambda_j v_j v_j^T \\ R_i &= \sum_{j=1}^{m}\frac{1}{\lambda_j}v_j v_j^T \\ \Delta\theta_i &= \sum_{j=1}^{m}\frac{\rho_i}{\lambda_j}(v_j^T g_i)v_j \end{aligned} \quad (14)$$

where λ_j is an eigenvalue of $\hat{M}_i$ corresponding to the eigenvector v_j. The step size in direction v_j is $(\rho_j)/(\lambda_j)(v_j^T g_i)$ which may be very large for small λ_j. Thus, the algorithm takes large steps in those parameter directions about which least information is available. This causes the convergence rate to be very poor. The techniques for handling near singular information matrixes are discussed in Section V.

C. Variable Metric Methods

Since the original work of Davidon [17] and Fletcher and Powell [18] the number of variable metric methods has proliferated. The main advantage of these methods is that they do not require analytical computation of the Hessian. Instead, these methods update the Hession or the inverse Hessian numerically from gradient information gained during the search procedure. A comparison of different variable metric methods on nonlinear least squares

problems by Bard [19] shows that the rank one correction methods are better than the Davidon–Fletcher-Powell method.[2] We describe here a rank one correction (ROC) method and an algorithm due to Jacobson and Oksman [20]. These methods are to be used in conjunction with the computation of the gradient by the Lagrange multiplier method as described in Section VI below.

Rank-One-Correction (ROC) Method: Let

$$\Delta\theta_i = \theta_{i+1} - \theta_i \tag{15}$$

$$\eta_i = g_{i+1} - g_i. \tag{16}$$

From Taylor series expansion of $g_{i+1} = (\partial J/\partial\theta)_{\theta=\theta_{i+1}}$

$$\eta_i = H_i \Delta\theta_i \tag{17}$$

where

$$H_i = \left.\frac{\partial^2 J}{\partial\theta^2}\right|_{\theta=\theta_i} \tag{18}$$

$$\Delta\theta_i = H_i^{-1}\eta_i. \tag{19}$$

Rank one correction consists of modifying an estimate of the Hessian A_i by a rank-one matrix B_i such that $A_{i+1} = A_i + B_i$ converges to H_i^{-1} in m steps for a quadratic. If we require in addition that (18) hold for A_{i+1}, the only solution for B_i turns out to be

$$B_i = \frac{1}{p_i^T \eta_i} p_i p_i^T \tag{20}$$

where

$$p_i = \Delta\theta_i - A_i\eta_i. \tag{21}$$

Notice that no one-dimensional search is required and $A_{m+1} = H^{-1}$ for a quadratic cost function $J(\theta)$.

The matrices A_i are not guaranteed to be positive-definite. One way to handle this problem is to compute the eigenvalues of A_i and replace the negative ones by their absolute values. This is analogous to the Greenstadt procedure [21] for the Newton–Raphson method, discussed in Section V below. The ROC method has been extended by Jacobson and Oksman [20] to finite-step convergence for homogeneous functions.

D. Choice of ρ_i

Several schemes are available for selecting ρ_i. Since, in dynamic systems, the calculation of $J(\theta)$ is very expensive, methods which require several trial values of ρ_i during each iteration are not desirable. A simple procedure is to use $\rho_i = \rho^{(0)}$ (in Gauss–Newton, $\rho^{(0)} = 1$ is satisfactory) in those cases where

$$J(\theta_{i+1}) - J(\theta_i) \geq \epsilon\rho_i g_i^T R_i g_i \tag{22}$$

and

$$0 < \epsilon < \tfrac{1}{2}.$$

[2] The comparison by Bard [19] of Gauss–Newton and rank one correction methods neglects the fact that the gradient alone can be calculated with less computation using adjoint equations (see Section VI).

If condition (22) is not satisfied, $J(\theta_i)$, g_i, $J(\theta_{i+1})$, and g_{i+1} are used to obtain a quadratic fit in ρ and the stationary value $\rho^{(1)}$ that minimizes the quadratic fit is computed and $\rho_i = \rho^{(1)}$. A low value of ϵ is recommended with this procedure.

IV. Singular or Nearly Singular Information Matrices

Based on a review of nonlinear programming literature and computational experience of [10], [24], four methods have been proposed for singular or nearly singular information matrices (in the Gauss–Newton or Newton–Raphson approach). In the numerical procedure it is better to obtain the parameter step by solving the following linear equations:

$$M_i(\theta_{i+1} - \theta_i) = -g_i. \tag{23}$$

The accuracy of this calculation depends upon the conditioning of M_i (ratio of maximum to minimum eigenvalue) but positive definiteness of M_i is not lost through inversion.

A. Levenberg–Marquardt Procedure [22], [23]

In this method, R_i is chosen as

$$R_i = (M_i + \alpha_i A_i)^{-1} \tag{24}$$

where A_i is a positive-definite matrix and $\alpha_i > 0$ is a scalar parameter. Generally, $A_i = I$ and α_i is chosen large enough so that the eigenvalues of $(M_i + \alpha_i A_i)$ are all positive and above a threshold value. Rules for the selection of α_i are given by Marquadt [23] and Bard [19]. The method has been used successfully in solving nonlinear least-squares problems.

B. Modified Gauss–Newton

Since the problem arises basically due to the eigenvalues λ_j that are nearly zero (in some cases, negative due to round-off errors), one sets all eigenvalues less that a certain threshold, say 10^{-6}, at a positive value. The method is applied more easily to the normalized information matrix M^* defined as

$$M^*(jk) = \frac{M(j,k)}{\sqrt{M(j,j)M(k,k)}}. \tag{25}$$

Furthermore, there are generally less numerical errors in inverting M^* as compared to M. The above method has been used quite successfully by Bard [19].

C. Combined Gradient and Rank-Deficient Gauss–Newton

The rank deficient Gauss–Newton has been used quite extensively especially in aircraft parameter identification [10], [24]. It can be used in many different forms. The simplest procedure is to use a pseudo-inverse of M for R as follows:

$$R = \sum_{j=1}^{m-k} \frac{1}{\lambda_j} v_j v_j^T \tag{26}$$

See (14)

where λ_j $(j = 1, 2, \cdots, m)$ are the eigenvalues of M and v_j are the corresponding eigenvectors such that $\lambda_1 > \lambda_2 \cdots \lambda_{m-k} > b > \lambda_{m-k+1} > \lambda_m$ and b is a suitably chosen threshold value. It has been shown by Ben-Israel [25] that the method converges to a point where $Rg = 0$. To avoid this problem, it is proposed here that the rank-deficient search which is basically a search in the subspace of dominant eigenvalues be followed by a gradient search ($R_i = I$) or a one-dimensional search in the subspace of remaining eigenvalues, i.e., with R_i chosen as

$$R_i = \alpha \sum_{j=m-k+1}^{m} \frac{1}{\lambda_j} v_j v_j^T, \qquad \alpha \leq 1. \tag{27}$$

Luenberger [12] proposes a similar approach for penalty methods in which the Newton–Raphson and gradient steps are alternated. The rank deficient method can also be used with normalized information matrix M^* defined by (25).

Numerically, better accuracy is obtained by computing the parameter step in each eigenvector directions and then adding them to determine the total step, i.e.,

$$\theta_{i+1} = \theta_i - \rho \sum_{j=1}^{m-k} \frac{v_j^T g_i}{\lambda_j} v_j. \tag{28}$$

Usually $v_j^T g_i$ is small for small eigenvalues λ_j. Since the elements of R_i are dominated by small eigenvalues, the components of the parameter step in large eigenvalue directions may be inaccurate if R_i and θ_{i+1} are computed using (26) and (8).

D. Iterative Stepwise Regression

Each iteration of the Gauss–Newton method is basically a solution of a linear regression problem. Thus, in cases where the significance of parameters in the model is not known *a priori*, one may use stepwise regression to select the most significant parameters during each iteration. The procedure has been used by Jennrich and Sampson [26] who also show how parameters may be added or taken out in a very convenient fashion by using matrix sweep or Gauss–Jordan pivoting. This method is also useful for *handling constraints on the parameters*, since the parameters can be taken in and out of the model by simple pivoting. The direct projection of the Gauss–Newton step on the constraint boundary is not satisfactory since it can lead to nonstationary points [26]. It may, however, require more iterations than other methods as shown by Bard [19] on a test example of Jennerich and Sampson [26].

V. Computation of the Gradient and Hessian of the Cost Function

The cost function of (3) depends on $\nu(t_i,\theta)$ and $B(t_i,\theta)$, which are obtained by solving a set of difference-differential equations (6) and (7). There are two different techniques for computing the gradient and Hessian of the negative log-likelihood function. In the dynamic programming formulation it is necessary to solve only n differential equations and n quadrature equations to find the gradient of $J(\theta)$ and additional $(n/2)(n + 1 + 2m)$ differential equations and $(m(m + 1))/2$ quadrature equations to determine the second derivative. The sensitivity equation method gives both the gradient and the information matrix in terms of the sensitivity functions.

For the Gauss–Newton method, it is better to use the sensitivity function approach, whereas for the variable metric method and the Newton–Raphson method, it is better to use the dynamic programming formulation. Furthermore, for ascertaining that a stationary point is indeed a local minimum (and not a saddle-point), it is recommended that the Hessian $(\partial^2 J/\partial\theta^2)$ be computed by the above method at the converged solution and its eigenvalues checked for positiveness.

A. Dynamic Programming Formulation [8], [37]

The gradient of $J(\theta)$ can be computed most efficiently by introducing Lagrange multiplier functions $(\partial V/\partial\hat{x})$ where $V(\hat{x},\theta,t)$ is the return function of dynamic programming, defined as the cost of going from state $\hat{x}$ at time t to final time T along the Kalman filter state trajectory generated by the constant "control" θ, i.e.,

$$V(\hat{x},\theta,t) = \tfrac{1}{2}\int_t^T [\nu^T(s,\theta)\, B^{-1}(s,\theta)\, \nu(s,\theta) + \ln |B(s,\theta)|], \delta(s - t_i)\, ds \tag{29}$$

where $\delta(s - t_i)$ is the delta function. Clearly

$$V(\hat{x},\theta,T) = 0$$
$$V(\hat{x}_0,\theta,0) = J(\theta).$$

Define

$$L(\hat{x},\theta,t) = \tfrac{1}{2}[\nu^T(t,\theta)B^{-1}(t,\theta)\nu(t,\theta) + \ln |B(t,\theta)|]\delta(t - t_i). \tag{30}$$

Considering a transition from t to $t + dt$ and $\hat{x}$ to $\hat{x} + dx$, we can write

$$V(\hat{x},\theta,t) = V(\hat{x} + dx,\theta,t + dt) + L(\hat{x},\theta,t)\, dt. \tag{31}$$

Expanding $V(\hat{x} + dx,\theta,t + dt)$ to first order, in (31) and considering the limit $dt \to 0$, we obtain

$$\frac{\partial V}{\partial t} = -\left(\frac{\partial V}{\partial \hat{x}}\right)^T f - L \tag{32}$$

where

$$f(\hat{x},\theta,t) = \dot{\hat{x}} = (F - KH)\hat{x} + Gu + Ky$$
$$K = K(t)\delta(t - t_i). \tag{33}$$

Equation (32) may now be differentiated with respect to θ to obtain differential equations for $(\partial V/\partial\theta)$ and $(\partial^2 V/\partial\theta^2)$. We give the final expressions using the operator relation

$$(\dot{}) \triangleq \frac{\partial}{\partial t}(\,) + \frac{\partial}{\partial \hat{x}}(\,) f \tag{34}$$

$$\frac{\partial \dot{V}}{\partial\theta(j)} = -\left(\frac{\partial V}{\partial \hat{x}}\right)^T \frac{\partial f}{\partial\theta(j)} - \frac{\partial L}{\partial\theta(j)} \tag{35}$$

$$\left(\frac{\partial^2 \dot{V}}{\partial\theta(j)\partial\theta(k)}\right) = -2\left(\frac{\partial^2 V}{\partial\hat{x}\partial\theta(k)}\right)^T \frac{\partial f}{\partial\theta(j)} - \left(\frac{\partial V}{\partial\hat{x}}\right)^T \frac{\partial^2 f}{\partial\theta(j)\partial\theta(k)} - \frac{\partial^2 L}{\partial\theta(j)\partial\theta(k)}. \quad (36)$$

The boundary conditions at terminal time $t = T$ are zero and $(\partial f/\partial\theta(j))$ is evaluated assuming $\hat{x}$ given, i.e.,

$$\frac{\partial f}{\partial\theta(j)} = \frac{\partial}{\partial\theta(j)}\{(F - KH)\}\hat{x} + \frac{\partial G}{\partial\theta(j)}u + \frac{\partial K}{\partial\theta(j)}y. \quad (37)$$

The partial derivatives of J are

$$\frac{\partial J}{\partial\theta} = \frac{\partial V}{\partial\theta}(\hat{x}_0,\theta,0)$$

$$\frac{\partial^2 J}{\partial\theta^2} = \frac{\partial^2 V}{\partial\theta^2}(\hat{x}_0,\theta,0). \quad (38)$$

Differential equations for $(\partial V/\partial\hat{x})$, $(\partial^2 V/\partial\hat{x}\partial\theta(k))$ are obtained from (32) and (34).

$$\frac{\partial\dot{V}}{\partial\hat{x}} = -\left(\frac{\partial V}{\partial\hat{x}}\right)^T \frac{\partial f}{\partial\hat{x}} - \frac{\partial L}{\partial\hat{x}} \quad (39)$$

$$\frac{\partial^2\dot{V}}{\partial\hat{x}\partial\theta(k)} = -\left(\frac{\partial^2 V}{\partial\hat{x}\partial\theta(k)}\right)^T \frac{\partial f}{\partial\hat{x}} - \frac{\partial^2 V}{\partial\hat{x}^2}\frac{\partial f}{\partial\theta(k)} - \left(\frac{\partial V}{\partial\hat{x}}\right)^T \frac{\partial^2 f}{\partial\hat{x}\partial\theta(k)} - \frac{\partial^2 L}{\partial\hat{x}\partial\theta(k)} \quad (40)$$

$$\frac{\partial^2\dot{V}}{\partial\hat{x}^2} = -\frac{\partial^2 V}{\partial\hat{x}^2}\frac{\partial f}{\partial\hat{x}} - \left(\frac{\partial f}{\partial\hat{x}}\right)^T \frac{\partial^2 V}{\partial\hat{x}^2} - \frac{\partial^2 L}{\partial\hat{x}^2}. \quad (41)$$

The boundary conditions for (39)–(41) are zero at final time $t = T$. Notice that $(\partial f/\partial\hat{x}) = (F - KH)$ and higher order partials of f with respect to $\hat{x}$ are zero.

B. Sensitivity Functions Method

It is clear from (11)–(13) that the first and second gradients of the cost can be computed in terms of the first gradient of the innovation and its covariance. For linear systems in statistical steady state the covariance of the innovations and its derivatives with respect to various parameters are constant. The innovations and its gradients can be computed in an efficient manner by the sensitivity functions reductions technique.

Consider, first, the following system with no process noise:

$$\dot{x} = Fx + Gu, \qquad x(0) = x_0. \quad (42)$$

The system starts from the initial state x_0. An alternate representation of the system is obtained by adding one more input and converting the initial condition to zero, i.e.,

$$\dot{x} = Fx + Gu + x_0 u_{q+1} \triangleq Fx + G'u', \qquad x(0) = 0 \quad (43)$$

and

$$u_{q+1} = \delta(t) \quad (44)$$

where δ is the dirac delta function. This shows that the sensitivities for the initial conditions can be computed in much the same way as the sensitivities for other parameters in G. Since x_0 is not different from parameters in G in this representation, the primes will be removed in (43). Thus, (42) with zero initial condition can be considered without loss of generality.

The following system properties which depend on the parameters θ, are important in sensitivity computation.

Definition 1—Structural Controllability:[3] A system is said to be structurally controllable if it is controllable for almost all values of parameters. The system may be uncontrollable if certain relations hold among the parameters.

Definition 2—Structural Linear Dependence: A set of vectors have structural linear dependence if a linear combination of these vectors is zero for almost all values of parameters. The particular linear combination may depend on the values of the parameters.

Example 1: Consider the system

$$\dot{x} = \begin{pmatrix}\theta_1 & \theta_2\\ \theta_3 & \theta_4\end{pmatrix}x + \begin{pmatrix}1\\1\end{pmatrix}u \quad (45)$$

the controllability matrix is

$$\begin{pmatrix}1 & \theta_1+\theta_2\\ 1 & \theta_3+\theta_4\end{pmatrix}. \quad (46)$$

The system is controllable unless $\theta_1 + \theta_2 = \theta_3 + \theta_4$. Thus, if $\theta_1 = \theta_4 = -1$ and $\theta_2 = \theta_3 = -5$, the system is uncontrollable in the classical sense but structurally controllable.

Initially, the following simplifications can be made:

1) The system is made structurally controllable (including x_0 in G) by dropping uncontrollable states. Since the initial condition is zero, the system never moves into the uncontrollable subspace. This reduces the order of the system. Note that the states which are uncontrollable only for the given values of the parameters but which are structurally controllable should not be dropped.

2) All structurally linearly dependent columns of G matrix are lumped with other columns. This reduces the number of effective controls.

These two simplifications reduce computations later. However, the order of the system required to generate sensitivity functions will be the same, even if these simplifications are not made.

The state sensitivity for parameter $\theta(j)$ follows the differential equation

$$\frac{d}{dt}\frac{\partial x}{\partial\theta(j)} = F\frac{\partial x}{\partial\theta(j)} + \frac{\partial F}{\partial\theta(j)}x + \frac{\partial G}{\partial\theta(j)}u \quad (47)$$

$$\frac{\partial x}{\partial\theta(j)}(0) = 0.$$

The state sensitivities for all parameters θ can be written as

[3] A similar definition has been given by C.-T. Lin, *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 201–208, June 1974.

$$\dot{x}_\theta = F_\theta x_\theta + G_\theta u \qquad (48)$$
$$x_\theta(0) = 0$$

where

$$x_\theta = \begin{bmatrix} x \\ \dfrac{\partial x}{\partial \theta(1)} \\ \cdot \\ \cdot \\ \cdot \\ \dfrac{\partial x}{\partial \theta(m)} \end{bmatrix} \quad n(m+1) \times 1 \qquad (49)$$

$$F_\theta = \begin{bmatrix} F & 0 & \cdots & 0 \\ \dfrac{\partial F}{\partial \theta(1)} & F & \cdots & 0 \\ \cdot & 0 & F & \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \dfrac{\partial F}{\partial \theta(m)} & 0 & 0 & F \end{bmatrix} \quad G_1 = \begin{bmatrix} G \\ \dfrac{\partial G}{\partial \theta(1)} \\ \cdot \\ \cdot \\ \cdot \\ \dfrac{\partial G}{\partial \theta(m)} \end{bmatrix}. \qquad (50)$$

$n(m+1) \times n(m+1) \qquad n(m+1) \times q$

If (F_θ, G_θ) is uncontrollable, the corresponding controllability matrix is of rank less than $(m+1)n$, say r. Let Q_1 be the set of r independent columns in the controllability matrix. If Q_2 is such that Q_1 and Q_2 form a set of $n(m+1)$ linearly independent vectors, then

$$x_\theta' \triangleq (Q_1 : Q_2)^{-1} x_\theta \triangleq \begin{pmatrix} Q_1^\dagger \\ \cdots \\ Q_2^\dagger \end{pmatrix} x_\theta \qquad (51)$$

follows the differential equation (see Chen [27])

$$\dot{x}_\theta' = \begin{pmatrix} F_{11}' & F_{12}' \\ 0 & F_{22}' \end{pmatrix} x_\theta' + \begin{pmatrix} G_1' \\ 0 \end{pmatrix} u. \qquad (52)$$

Since the initial condition in (52) is zero, the last $(m+1)n - r$ uncontrollable states remain zero throughout. The remaining r states, x_c follow the differential equation

$$\dot{x}_c = F_c x_c + G_c u$$
$$x_c(0) = 0 \qquad (53)$$

where

$$F_c \triangleq F_{11}' = Q_1^\dagger F_\theta Q_1$$
$$G_c \triangleq G_1' = Q_1^\dagger G_\theta. \qquad (54)$$

Also

$$x_\theta = (Q_1 : Q_2) x_\theta'$$
$$= Q_1 x_c \qquad (55)$$

since other states in x_θ' are zero. Note that $Q_1^\dagger$ is a pseudoinverse of Q_1 and depends on Q_2. The transformation from F_θ, G_θ to F_c, G_c and from x_c to x_θ does not involve Q_2 explicitly. Therefore, $Q_1^\dagger$ can be chosen to be *any* pseudoinverse of Q_1, for example,

$$Q_1^\dagger = (Q_1^T Q_1)^{-1} Q_1^T. \qquad (56)$$

Process Noise: The Kalman filter representation of a system with process noise is

$$\dot{\hat{x}} = F\hat{x} + Gu + K\nu$$
$$= (F - KH)\hat{x} + Gu + Ky \qquad (57)$$

and

$$\nu = y - H\hat{x} \qquad (58)$$

where x is the best estimate of state given the past observations. Clearly,

$$\frac{\partial \nu}{\partial \theta(i)} = -H \frac{\partial \hat{x}}{\partial \theta(i)} - \frac{\partial H}{\partial \theta(i)} x. \qquad (59)$$

Thus, it is necessary to compute $\hat{x}(t)$ and its gradients to find the innovation and its gradients. In the steady state K is a constant and, therefore, (57) is like (42) except that u and y are inputs and $[G:K]$ is the input distribution matrix. With this modification, all the results derived for the no process noise case hold also for the system with process noise.

The next section investigates the nature and dimension of the controllable subspace of (F_θ, G_θ) and explores efficient methods for finding Q_1.

VI. Practical Techniques for Sensitivity Functions Reductions

A. Single-Input Systems

In single-input systems, F_θ is a $(m+1)n \times (m+1)n$ matrix and G_θ is a $(m+1)n$ vector. The following holds.

Statement: For a single input system, the rank of the controllability matrix of (F_θ, G_θ) is less than or equal to $2n$.

Proof: The controllability matrix of (F_θ, G_θ) is

$$C_\theta = [G_\theta : F_\theta G_\theta : \cdots F_\theta^{(m+1)n-1} G_\theta]. \qquad (60)$$

It is easy to show that

$$F_\theta^p G_\theta = \begin{bmatrix} F^p G \\ \cdots\cdots \\ \dfrac{\partial}{\partial \theta(1)} F^p G \\ \cdots\cdots \\ \cdot \\ \cdot \\ \cdot \\ \cdots\cdots \\ \dfrac{\partial}{\partial \theta(m)} F^p G \end{bmatrix} \triangleq D(F^p G) \triangleq \begin{pmatrix} F^p G \\ \overline{0} \end{pmatrix} + D^*(F^p G). \qquad (61)$$

If

$$F^n = \alpha_0 I + \alpha_1 F \cdots \alpha_{n-1} F^{n-1}, \qquad (62)$$

the $(n+k)$th column of C_θ is

$$D(F^{n+k-1} G) = \sum_{i=0}^{n-1} \{ D^*(\alpha_i I) F^{k+i-1} G + \alpha_i D(F^{k+i-1} G) \}. \qquad (63)$$

The second term is a linear combination of n preceding columns of C_θ. Thus,

$$\operatorname{rank} C_\theta = \operatorname{rank}\Big\{D(G)\vdots\cdots\vdots D(F^{n-1}G)\vdots \sum_{i=0}^{n-1} D^*(\alpha_i I)F^iG\vdots\cdots \vdots \sum_{i=0}^{n-1} D^*(\alpha_i I)F^{i+n-1}G\vdots \sum_{i=0}^{n-1} D^*(\alpha_i I)F^{n+i}G\vdots\cdots. \quad (64)$$

The $(2n+k)$th column of the right-hand side matrix is

$$\sum_{i=0}^{n-1} D^*(\alpha_i I)F^{n+i+k-1}G = \sum_{i=0}^{n-1} D^*(\alpha_i I)\sum_{j=0}^{n-1}\alpha_j F^{i+j+k-1}G = \sum_{j=0}^{n-1}\alpha_j \sum_{i=0}^{n-1} D^*(\alpha_i I)F^{i+(j+k)-1}G \quad (65)$$

which is a linear combination of n previous columns for $k \geq 0$. Therefore,

$$\operatorname{rank} C = \operatorname{rank}\,[G_\theta\vdots F_\theta G_\theta\vdots\cdots F_\theta{}^{2n-1}G_\theta] \leq 2n. \quad (66)$$

Thus, the order of the system required to compute all state sensitivities for a single-input system cannot exceed $2n$. In many practical cases, it is smaller as shown in Example 2.

Corollary 1: If the structurally controllable subspace of (F,G) is of the order p, the maximal order of the controllable subspace of (F_θ,G_θ) is $2p$.

Example 2: Consider the following system:

$$\dot{x} = \begin{pmatrix}\theta_1 & \theta_2\\ 0 & -1\end{pmatrix}x + \begin{pmatrix}0\\1\end{pmatrix}u, \qquad x(0) = 0. \quad (67)$$

The state vector and its sensitivities for θ_1 and θ_2 form a set of six differential equations. Since the number of states is two and the number of controls is one, only the first four columns can be independent in the controllability matrix of (F_θ,G_θ). These columns are

$$\operatorname{rank}(C_\theta) = \operatorname{rank}\left[D\begin{pmatrix}0\\1\end{pmatrix}, D\begin{pmatrix}\theta_2\\-1\end{pmatrix}, D\begin{pmatrix}\theta_1\theta_2-\theta_2\\1\end{pmatrix}, D\begin{pmatrix}\theta_1{}^2\theta_2-\theta_1\theta_2+\theta_2\\-1\end{pmatrix}\right] \quad (68)$$

$$= \operatorname{rank}\begin{bmatrix}0 & \theta_2 & \theta_1\theta_2-\theta_2 & \theta_1{}^2\theta_2-\theta_1\theta_2+\theta_2\\ 1 & -1 & 1 & -1\\ 0 & 0 & \theta_2 & 2\theta_1\theta_2-\theta_2\\ 0 & 0 & 0 & 0\\ 0 & 1 & \theta_1-1 & \theta_1{}^2-\theta_1+1\\ 0 & 0 & 0 & 0\end{bmatrix}. \quad (69)$$

The first three columns are independent for $\theta_2 \neq 0$. If θ_2 is zero, only the first two columns are independent and the required model is of second order.

B. Multiinput Systems

Statement: The rank of the controllability matrix of (F_θ,G_θ) cannot exceed $(q+1)n$ for a q inputs system.

Proof: In multiinput systems, G_θ is a matrix with q columns. The controllability matrix of (F_θ,G_θ) has $(m+1)nq$ columns which can be written

$$C_\theta = [D(G_1,FG_1,\cdots,F^{n(m+1)-1}G_1)\vdots D(G_2,FG_2,\cdots, F^{n(m+1)-1}G_2)\vdots\cdots\vdots D(G_q,FG_q,\cdots,F^{n(m+1)-1}G_q)] \quad (70)$$

where G_i is the ith column of G. From Section VII-A, it is clear that the last $(m-1)n$ columns involving any of vectors G_i's are linearly dependent on the first $2n$ columns for that vector. From (64) and (65),

$$\operatorname{rank}(C_\theta) \triangleq \rho = \operatorname{rank}[\{D(G_1,FG_1,\cdots,F^{n-1}G_1), \Sigma D^*(\alpha_i I)F^iG_1,\cdots,\Sigma D^*(\alpha_i I)F^{i+n-1}G_1\}\vdots\cdots\{D(G_q,FG_q,\cdots,F^{n-1}G_q),\Sigma D^*(\alpha_i I)F^iG_q,\cdots, \Sigma D^*(\alpha_i I)F^{i+n-1}G_q\}] \quad (71)$$

where all summations are from 0 to $n-1$. Let the structurally independent columns in the controllability matrix of (F,G) be

$$[G_1,\cdots,F^{k_1-1}G_1,G_2,\cdots,F^{k_2-1}G_2,\cdots,G_q,\cdots,F^{k_q-1}G_q] \quad (72)$$

$$\sum_{i=1}^{q} k_i = n. \quad (73)$$

This set of linearly independent vectors in the controllability matrix span the complete n-dimensional space. So any vector can be represented as a linear combination of these vectors. In particular,

$$F^{j-1}G_k = \beta_1 G_1 +,\cdots,+\beta_{k_1}F^{k_1-1}G_1,\cdots,\beta_n F_q{}^{k-1}G_q, \qquad 1\leq j\leq n \quad 1\leq k\leq q. \quad (74)$$

Therefore,

$$\sum_{i=0}^{n-1} D^*(\alpha_i I)F^{i+j-1}G_k = \beta_1\sum_{i=0}^{n-1} D^*(\alpha_i I)F^iG_1 + \beta_2\sum_{i=0}^{n-1} D^*(\alpha_i I)F^{i+1}G_1\cdots, +\beta_n\sum_{i=0}^{n-1} D^*(\alpha_i I)F^{i+k}q^{-1}G_q. \quad (75)$$

This is a linear combination of n vectors in the right-hand side matrix of (74) for all j and k (the values of β_i depend on j and k). Thus,

$$\rho = \operatorname{rank}\,[D(G_1,FG_1,\cdots,F^{n-1}G_1)\vdots D(G_2,FG_2,\cdots,F^{n-1}G_2)\vdots\cdots\vdots D(G_q,FG_q,\cdots,F^{n-1}G_q),\Sigma D^*(\alpha_i I)F^iG_1, \Sigma D^*(\alpha_i I)F^{i+1}G,\cdots,\Sigma D^*(\alpha_i I)F^iG_q,\cdots, \Sigma D^*(\alpha_i I)F^{i+k_q-1}G_q] \leq (q+1)n. \quad (76)$$

Remarks: These sensitivity reduction methods can be extended to discrete system equations with slight modification. For systems in canonical form and for scalar autoregressive moving average models, the transformations of equations (54) and (55) simplify greatly. The sensitivity of the states or outputs to all parameters can be easily computed by using a sensitivity model, which is twice the size of the original model. These methods were used implicity by Åström and Bohlin [3] for scaler systems. Neuman and Sood [7] deal in considerable detail with multiinput-multioutput systems in canonical form, when

the sensitivity of the state vector to all canonical parameters is required. The method presented by us is much more general and applicable.

C. Computation Procedure

A computer program may be used to carry out this sensitivity functions reduction in linear constant-coefficient system by the following procedure.

1) The initial condition, if nonzero, is appended to the input distribution matrix and the number of inputs is increased by one. The linearly dependent columns in G are merged. Then the structurally uncontrollable states in (F,G) are dropped.

2) Matrices F_θ and G_θ are formed. $k_1,k_2,\cdots,k_q$ of (72) are determined and are used to choose $(q+1)n$ appropriate columns from the controllability matrix of (F_θ,G_θ).

3) The dimension k_i' of state space controllable from each input u_i alone is determined. If for any i

$$2k_i' < n + k_i \tag{77}$$

the last $n + k_i - 2k_i'$ columns involving G_i in the right-hand side matrix of (76) are dropped. (This simplification is due to Corollary 1.)

4) The remaining columns are checked for linear independence. Gram–Schmidt procedure is used to drop columns, which are linearly dependent on other columns. The set of remaining columns is Q_1.

5) Any pseudo-inverse of Q_1 is determined. Equation (54) is used to compute F_c and G_c.

6) Equation (53) is solved for $x_c(t)$ and (55) is used to find x_θ at the desired points.

VII. Approximation Methods for Reducing Computation Time

In problems with a large number of parameters and complicated nonlinear differential equations, the computation of the sensitivity equations at each iteration is very time consuming. The following methods are proposed for reducing computation time.

1) Hold the information matrix constant for several iterations and compute the gradient by using (33) and (35). This method has been used by Ben-Israel [25] who shows that even though this method requires more iterations, the computation time can be less than the NR or GN method. For on-line applications, one may precompute the information matrix as shown in [16].

2) When far away from the minimum, use a large step size for integrating the differential equations. Refine the time-step as the minimum is approached. This method is advocated by Kleissig and Polak [28].

3) Generally, $J(\theta)$ is quadratic in some of the parameters, say θ_1, and nonquadratic in others, say θ_2. Golub and Pereyra [29] propose a technique which requires direct search with respect to θ_2 while calculating θ_1 at each step by solving a set of linear equations. In some cases, this method can give improvements in total convergence time. In all cases, one iteration of this method before starting the full NR or GN search is desirable [29].

4) For stationary processes, the spectral techniques given by Hannan [30] and Mehra [31], [32] may be used to reduce computation. The use of Fast Fourier Transform techniques is particularly attractive. Akaike [33] has shown that under certain approximations the information matrix is block Toeplitz. If these assumptions are made, efficient methods for inverting the information matrix [34] or solving the normal equations [35] can be used. Morf and Kailath [36] indicate the relationship of these methods to certain filtering algorithms of "fast square-root" type for state estimation. Furthermore, Akaike [33] has shown that Hannan's procedure [30] is equivalent to one iteration of the Newton–Raphson method on the likelihood function.

IX. Conclusions

Several computational aspects of maximum likelihood estimation have been discussed with the hope of stimulating further research in this area. From a practical standpoint, this is one of the most important problems in the applicability of system identification techniques to large scale systems. It is clear from our discussions that an extensive amount of work on related problems has been done by numerical analysts. An assimilation of this work and further extensions would lead us towards a routine use of maximum likelihood methods for estimation of parameters in dynamic systems.

References

[1] G. A. Bernard, "The use of the likelihood function in statistical practice," in *Proc. 5th Berkeley Symp. Math. Statist. and Probability*, 1966, pp. 27–40.

[2] A. W. F. Edwards, *Likelihood*. New York: Cambridge, 1972.

[3] K. J. Åström and T. Bohlin, "Numerical identification of linear dynamic systems from normal operating records," in *Proc. IFAC Symp. Theory of Self-Adaptive Processes*, Teddington, 1968.

[4] T. Bohlin, "On the maximum likelihood method of identification," *IBM J. Res. Develop.*, vol. 14, no. 1, pp. 41–51, 1970.

[5] D. F. Wilkie and W. R. Perkins, "Generation of sensitivity functions for linear systems using low-order models," *IEEE Trans. Automat. Contr.*, vol. AC-14, pp. 123–130, Apr. 1969.

[6] D. G. Denery, "Simplification in the computation of the sensitivity functions for constant coefficient linear system," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 348–350, Aug. 1971.

[7] C. P. Newman and A. K. Sood, "Sensitivity functions for multi-input linear time-invariant systems—Part II: Minimum-order models," *Int. J. Contr.*, vol. 15, no. 3, pp. 451–463, 1972.

[8] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*. New York: Elsevier, 1970.

[9] R. K. Mehra, "Maximum likelihood identification of aircraft parameters," in *1970 Joint Automatic Control Conf., Preprints*, Atlanta, Georgia, June 1970.

[10] W. E. Hall, Jr., N. K. Gupta, and R. G. Smith, "Identification of aircraft stability and control derivatives for the high angle-of-attack regime," Systems Control, Inc., ONR Tech. Rep., Mar. 1974.

[11] P. J. Dhrymes, L. R. Klein, and K. Stieglitz, "Estimation of distributed lags," *Int. Economic Rev.*, vol. II, June 1970.

[12] J. Opacic, "A heuristic approach to multimodal function minimization—The global aspect," presented at the 9th Annu. Allerton Conf. Circuit and System Theory, Oct. 1971.

[13] E. Polak, *Computational Methods in Optimization—A Unified Approach*. New York: Academic, 1971.

[14] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Reading, Mass.: Addison–Wesley, 1972.

[15] C. R. Rao, *Linear Statistical Inference and its Applications*. New York: Wiley, 1965.

[16] R. K. Mehra, "Synthesis of optimal inputs for multiinput/multioutput systems with process noise, Parts I and II," Division of Engineering and Applied Physics, Harvard Univ., Cambridge, Mass., Tech. Rep. TR 649, Feb. 1974.

[17] W. C. Davidon, "Variable metric methods for minimization," Argonne Nat. Lab, Argonne, Ill., A.E.C. Res Develop. Rep, ANL-5990, 1959.
[18] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Comput. J.*, vol. 6, 1963.
[19] Y. Bard, "Comparison of gradient methods for the solution of nonlinear parameter estimation problems," *SIAM J. Numer. Anal.*, vol. 7, Mar. 1970.
[20] D. H. Jacobson and W. Oksman, "An algorithm that minimizes homogeneous functions of N variables in $N + 2$ iterations and rapidly minimizes general functions," Div. Eng. Appl. Phys., Harvard Univ., Cambridge, Mass., Tech. Rept. TR 618, October 1970.
[21] J. Greenstadt, "On the relative efficiencies of gradient methods," *Math. Comput.*, vol. 21, 1967.
[22] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *Quart. Appl. Math.*, vol. 2, pp. 164–168, 1944.
[23] D. W. Marquardt, "An algorithm for least squares estimation of nonlinear parameters", *SIAM J. Numer. Anal.*, vol. 11, pp. 431–441, 1963.
[24] D. E. Stepner and R. K. Mehra, "Maximum likelihood identification and optimal input design for identifying aircraft stability and control derivatives," NASA CR-2200, Mar. 1973.
[25] A. Ben-Israel, "A Newton–Raphson method for the solution of systems of equations," *J. Math. Anal. Appl.*, vol. 15, pp. 243–252, 1966.
[26] R. I. Jennrich and P. F. Sampson, "Application of stepwise regression to nonlinear estimation," *Technometrics*, vol. 10, pp. 63–72, 1968.
[27] C. T. Chen, *Introduction to Linear Systems Theory*. New York: Holt, Rinehart and Winston, 1970.
[28] R. Kleissig and E. Polak, "An adaptive precision gradient method for optimal control," *SIAM J. Contr.*, vol. 11, Feb. 1973.
[29] G. H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM J. Numer. Anal.*, vol. 10, Apr. 1973.
[30] E. J. Hannan, *Multiple Time Series Analysis*. New York: Wiley, 1970.
[31] R. K. Mehra, "Identification in control and econometrics, similarities and differences," *Ann. Economic Social Measurement*, Jan. 1974.
[32] R. K. Mehra, "Frequency domain synthesis of optimal inputs for linear system parameter estimation," Div. Eng. Appl. Phys., Harvard Univ., Cambridge, Mass., Tech. Rep. TR 645, July 1973.
[33] H. Akaike, "Maximum likelihood identification of Gaussian autoregressive moving average models," *Biometrika*, vol. 60, no. 2, pp. 255–265, 1973.
[34] H. Akaike, "Block Toeplitz matrix inversion," *SIAM J. Appl. Math.*, 1973.
[35] P. Whittle, "On the fitting of multivariable autoregression, and the approximate canonical factorization of a spectral density matrix," *Biometrika*. vol. 50, pp. 129–134, 1963.
[36] M. Morf and T. Kailath, "Square root algorithms for least squares estimation," in *Proc. 8th Princeton Symp. Information Sciences*, Mar. 1974.
[37] S. R. McReynolds, "The successive sweep method and dynamic programming," *J. Math. Anal. Appl.*, vol. 19, no. 3, 1967.
[38] M. J. D. Powell, "Recent advances in unconstrained optimization," *Math. Programming*, no. 1, pp. 26–57, 1971.
[39] E. Polak, "A modified secant method for unconstrained optimization," Dep. Elec. Eng., Univ. Calif., Berkeley, Rep., Jan. 1973.

Narendra K. Gupta (M'74) was born in Panipat, India, on September 30, 1948. He received the B. Tech. degree in mechanical engineering with highest honors from the Indian Institute of Technology, New Delhi, in 1969. In 1970 he received the M.S. degree in aeronautics from the California Institute of Technology, Pasadena, where he was the recipient of an Earle C. Anthony Fellowship. He received the Ph.D. degree from Stanford University, Stanford, Calif., in 1974, specializing in control theory. He was awarded a Stanford University fellowship during 1970–71.

He has had summer jobs at Godrej Limited, California Institute of Technology, and Stanford University. Currently, he is a Research Engineer at Systems Control, Inc., Palo Alto, Calif., where he is actively involved in system identification problems with applications to aircraft, human operators and chemical processes.

Raman K. Mehra (S'67–M'68) for a photograph and biography see page 768 of this issue.