JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 16, No. 1, January-February 1993

Newton Algorithm for Fitting Transfer Functions to Frequency Response Measurements

J. T. Spanos*
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109
and
D. L. Mingori†
University of California, Los Angeles, Los Angeles, California 90024

In this paper the problem of synthesizing transfer functions from frequency response measurements is considered. Given a complex vector representing the measured frequency response of a physical system, a transfer function of specified order is determined that minimizes the sum of the magnitude-squared of the frequency response errors. This nonlinear least squares minimization problem is solved by an iterative global descent algorithm of the Newton type that converges quadratically near the minimum. The unknown transfer function is expressed as a sum of second-order rational polynomials, a parameterization that facilitates a numerically robust computer implementation. The algorithm is developed for single-input, single-output, causal, stable transfer functions. Two numerical examples demonstrate the effectiveness of the algorithm.

I. Introduction

THE synthesis of transfer functions from frequency response data has received considerable attention in the literature due to its importance in the area of linear system identification. Typically, the frequency response of a physical system is measured with the aid of actuator-sensor hardware and a data acquisition system. The measured response is then used to synthesize a linear system that is subsequently used for analysis and control system design. Not addressing the issues associated with the measurement, the present paper focuses on the problem of finding a transfer function whose frequency response best approximates the measured response in the least squares sense.

Also known as "complex curve fitting,"[1] early work on this problem concentrated on fitting the frequency response data by a transfer function that minimizes a *linear* least squares error function.[1-6] Levy[1] expressed the unknown transfer function as a ratio of two complex polynomials (i.e., numerator and denominator) and proposed to minimize the magnitude-squared of the error weighted by the magnitude-squared of the denominator. He showed that the solution satisfies a linear least squares problem. Sanathanan and Koerner[2] extended Levy's approach by proposing an iterative technique whereby the weighting used at iteration k is determined from the least squares solution at iteration $k-1$. Numerical case studies and further analysis indicate that the Sanathanan-Koerner iteration is superior to the noniterative scheme of Levy and often yields good fits to the data.[2,6] More recently, numerically robust computer implementations of these approaches have been reported by Richardson and Formenti,[3] Adcock,[4] and Dailey and Lukich,[5] who express the numerator and denominator of the unknown transfer function as sums of orthogonal polynomials.

However, there is no guarantee that the Sanathanan-Koerner iteration will converge. In fact, the conditions under which it converges are not known. Furthermore, Whitfield[6] showed that even when this linear least squares iteration converges, it does not converge to the minimum of the *nonlinear* least squares cost function that is defined by the sum of the magnitude squared of the frequency response errors. Yet a more significant drawback of the Sanathanan-Koerner algorithm is that it may converge to an unstable transfer function in situations where the given frequency response data represent a perfectly stable system.

In the present paper we develop an iterative global descent algorithm that searches for a minimum of the nonlinear least squares cost function. The unknown transfer function is expressed as a sum of second-order rational polynomials, a parameterization resulting in a numerically robust computer implementation. Analytical expressions for the first and second derivatives of the cost function with respect to the transfer function parameters are obtained and used to set up a Newton iteration. Near the minimum the algorithm converges quadratically. Far from the minimum a strategy is adopted to restrict the size of the Newton step and to effectively deal with a possibly nonpositive definite Hessian matrix. As a result, each iteration guarantees a decrease in the cost function, and all iterates are restricted to have stable poles. Initialization of the algorithm is addressed, and two examples representing lightly damped flexible structures are presented.

II. Problem Statement

We begin by assuming that the sequence of frequency response pairs

$$\omega_i, \quad G(j\omega_i); \qquad i = 1, 2, \ldots, p \tag{1}$$

is known. The complex number $G(j\omega_i)$ will be referred to as the *measurement* at frequency ω_i. We consider the problem of finding a transfer function $\hat{G}(j\omega, x)$ such that the cost function

$$J(x) = \sum_{i=1}^{p} w_i \left| G(j\omega_i) - \hat{G}(j\omega_i, x) \right|^2 \tag{2}$$

is minimized. The positive constants w_i allow the measurements $G(j\omega_i)$ to be weighted. Numerous parameterizations of the transfer function exist (i.e., ratio of two polynomials, product of poles and zeroes, partial fraction expansion). In this paper we adopt the following parameterization:

$$\hat{G}(j\omega, x) = \sum_{k=1}^{m} \frac{a_k + j\omega b_k}{\alpha_k + j\omega\beta_k - \omega^2} + d \tag{3}$$

Received Oct. 18, 1990; revision received Jan. 27, 1992; accepted for publication May 8, 1992.

*Member of Technical Staff, Guidance and Control Section. Member AIAA.
†Professor, Mechanical, Aerospace, and Nuclear Engineering Department. Member AIAA.

where

$$x^T = [a^T \; b^T \; \alpha^T \; \beta^T \; d] \tag{4}$$

and

$$a^T = [a_1 \quad a_2 \quad \cdots \quad a_m] \tag{5a}$$

$$b^T = [b_1 \quad b_2 \quad \cdots \quad b_m] \tag{5b}$$

$$\alpha^T = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_m] \tag{5c}$$

$$\beta^T = [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_m] \tag{5d}$$

The parameter vector x has dimension $n = 4m + 1$, and it completely defines the transfer function $\hat{G}(j\omega, x)$. As given in Eq. (3), the order of $\hat{G}(j\omega, x)$ will always be even; however, the term $a_{m+1}/(\alpha_{m+1} + j\omega)$ can easily be added on the right side of Eq. (3) to allow for odd order transfer functions as well.

III. Derivatives of the Cost Function

The cost function defined in Eq. (2) can be written as

$$J(x) = \sum_{i=1}^{p} w_i [G(j\omega_i) - \hat{G}(j\omega_i, x)]^* [G(j\omega_i) - \hat{G}(j\omega_i, x)] \tag{6}$$

where the asterisk denotes the complex conjugate transpose. Direct differentiation yields the first and second derivatives of $J(x)$ in terms of the first and second derivatives of $\hat{G}(j\omega, x)$:

$$g_r(x) \triangleq \frac{\partial J(x)}{\partial x_r} = 2\mathrm{Re} \sum_{i=1}^{p} w_i \left[\frac{\partial \hat{G}(j\omega_i, x)}{\partial x_r}\right]^* [\hat{G}(j\omega_i, x) - G(j\omega_i)] \tag{7}$$

$$H_{sr}(x) \triangleq \frac{\partial^2 J(x)}{\partial x_s \partial x_r} = 2\mathrm{Re} \sum_{i=1}^{p} w_i \left\{ \left[\frac{\partial \hat{G}(j\omega_i, x)}{\partial x_s}\right]^* \left[\frac{\partial \hat{G}(j\omega_i, x)}{\partial x_r}\right] + \left[\frac{\partial^2 \hat{G}(j\omega_i, x)}{\partial x_s \partial x_r}\right]^* [\hat{G}(j\omega_i, x) - G(j\omega_i)] \right\} \tag{8}$$

where $r, s = 1, 2, \ldots, n$ and Re denotes the real part of the expression that follows it. The vector $g(x)$ and matrix $H(x)$ are the *gradient* and *Hessian* of the cost function, respectively. The first derivatives of the transfer function with respect to the parameters are obtained by direct differentiation of Eq. (3):

$$\frac{\partial \hat{G}(j\omega, x)}{\partial a_r} = \frac{1}{\alpha_r + j\omega\beta_r - \omega^2} \tag{9a}$$

$$\frac{\partial \hat{G}(j\omega, x)}{\partial b_r} = \frac{j\omega}{\alpha_r + j\omega\beta_r - \omega^2} \tag{9b}$$

$$\frac{\partial \hat{G}(j\omega, x)}{\partial \alpha_r} = \frac{-(a_r + j\omega b_r)}{(\alpha_r + j\omega\beta_r - \omega^2)^2} \tag{9c}$$

$$\frac{\partial \hat{G}(j\omega, x)}{\partial \beta_r} = \frac{-j\omega(a_r + j\omega b_r)}{(\alpha_r + j\omega\beta_r - \omega^2)^2} \tag{9d}$$

$$\frac{\partial \hat{G}(j\omega, x)}{\partial d} = 1 \tag{9e}$$

where $r = 1, 2, \ldots, m$. Similarly, for $r, s = 1, 2, \ldots, m$, the second derivatives of $\hat{G}(j\omega, x)$ are

$$\frac{\partial^2 \hat{G}(j\omega, x)}{\partial a_s \partial a_r} = \frac{\partial^2 \hat{G}(j\omega, x)}{\partial b_s \partial a_r} = \frac{\partial^2 \hat{G}(j\omega, x)}{\partial b_s \partial b_r} = 0 \tag{10a}$$

$$\frac{\partial^2 \hat{G}(j\omega, x)}{\partial \alpha_s \partial a_r} = \frac{-1}{(\alpha_r + j\omega\beta_r - \omega^2)^2} \delta_{sr} \tag{10b}$$

$$\frac{\partial^2 \hat{G}(j\omega, x)}{\partial \beta_s \partial a_r} = \frac{\partial^2 \hat{G}(j\omega, x)}{\partial \alpha_s \partial b_r} = \frac{-j\omega}{(\alpha_r + j\omega\beta_r - \omega^2)^2} \delta_{sr} \tag{10c}$$

$$\frac{\partial^2 \hat{G}(j\omega, x)}{\partial \beta_s \partial b_r} = \frac{\omega^2}{(\alpha_r + j\omega\beta_r - \omega^2)^2} \delta_{sr} \tag{10d}$$

$$\frac{\partial^2 \hat{G}(j\omega, x)}{\partial \alpha_s \partial \alpha_r} = 2\frac{(a_r + j\omega b_r)}{(\alpha_r + j\omega\beta_r - \omega^2)^3} \delta_{sr} \tag{10e}$$

$$\frac{\partial^2 \hat{G}(j\omega, x)}{\partial \beta_s \partial \alpha_r} = 2\frac{j\omega(a_r + j\omega b_r)}{(\alpha_r + j\omega\beta_r - \omega^2)^3} \delta_{sr} \tag{10f}$$

$$\frac{\partial^2 \hat{G}(j\omega, x)}{\partial \beta_s \partial \beta_r} = 2\frac{-\omega^2(a_r + j\omega b_r)}{(\alpha_r + j\omega\beta_r - \omega^2)^3} \delta_{sr} \tag{10g}$$

where $\delta_{sr} = 1$ if $s = r$, and $\delta_{sr} = 0$ if $s \neq r$. Finally, all second derivatives of $\hat{G}(j\omega, x)$ with respect to the parameter d vanish:

$$\frac{\partial^2 \hat{G}(j\omega, x)}{\partial d \partial a_r} = \frac{\partial^2 \hat{G}(j\omega, x)}{\partial d \partial b_r} = \frac{\partial^2 \hat{G}(j\omega, x)}{\partial d \partial \alpha_r} = \frac{\partial^2 \hat{G}(j\omega, x)}{\partial d \partial \beta_r} = \frac{\partial^2 \hat{G}(j\omega, x)}{\partial d^2} = 0 \tag{10h}$$

The symmetry of the Hessian matrix (i.e., $H_{rs} = H_{sr}$) is implied from the differentiability of the transfer function $\hat{G}(j\omega, x)$. The gradient and Hessian of the cost function can be expressed directly in terms of the parameters by substituting Eqs. (9) and (10) into Eqs. (7) and (8).

IV. Newton Iteration

The power series expansion of the cost function $J(x)$ about x^0 is

$$J(x) = J(x^0) + (x - x^0)^T g(x^0) + \frac{1}{2!}(x - x^0)^T [H(x^0)](x - x^0) + \mathcal{O}(\|x - x^0\|^3) \tag{11}$$

where $\|\cdot\|$ denotes the Euclidean norm. The cost function has a minimum at x^{opt} if its gradient vanishes and its Hessian is positive definite at x^{opt} [i.e., $g(x^{\mathrm{opt}}) = 0$ and $y^T H(x^{\mathrm{opt}}) y > 0$, $\forall\, y \neq 0$]. Now if x^0 is in the neighborhood of x^{opt}, (i.e., $\|x^0 - x^{\mathrm{opt}}\| \leq \epsilon$, where ϵ is a sufficiently small positive constant), the nonlinear minimization problem

$$\min_x J(x) \tag{12}$$

can be approximated by the considerably simpler minimization problem

$$\min_x \mathcal{J}^0(x) \tag{13}$$

where

$$\mathcal{J}^0(x) = J(x^0) + (x - x^0)^T g(x^0) + \frac{1}{2!}(x - x^0)^T [H(x^0)](x - x^0) \tag{14}$$

is the quadratic approximation of $J(x)$ in the neighborhood of x^0. The solution of Eq. (13) is given by $x = x^1$ where

$$x^1 = x^0 - [H(x^0)]^{-1} g(x^0) \tag{15}$$

Clearly, the minimizer of Eq. (13) is only an approximation to the minimizer of Eq. (12) (i.e., $x^1 \approx x^{\mathrm{opt}}$). The procedure can

be repeated by expanding the cost function in a power series about x^1 and minimizing its quadratic approximation to obtain x^2, a better estimate of x^{opt}. The repeated minimization of the quadratic approximation results in the iteration

$$x^{k+1} = x^k - [H(x^k)]^{-1} g(x^k); \qquad k = 0, 1, 2, \ldots \tag{16}$$

which is known as the *Newton iteration*. Under mild conditions on the continuity of the cost function and its derivatives in the neighborhood of x^{opt}, it can be shown[7] that the Newton iterates x^k converge quadratically to x^{opt}, i.e.,

$$\lim_{k\to\infty} x^k = x^{opt} \qquad \text{and} \qquad \lim_{k\to\infty} \frac{\| x^{k+1} - x^{opt}\|}{\| x^k - x^{opt}\|^2} < \infty \tag{17}$$

The convergence of the Newton iteration to a minimum of the cost function is highly sensitive in the initial iterate x^0. Convergence is guaranteed only when $\| x^0 - x^{opt}\| \leq \epsilon$ and, in fact, the iterates generated by Eq. (16) can diverge from x^{opt} if x^0 is not sufficiently close to x^{opt}. Generally, convergence problems occur when the Hessian is not positive definite at some iterate x^k or when the quadratic function $\mathcal{J}^k(x)$ is a poor approximation to the cost function $J(x)$ at $x = x^{k+1}$. To prevent divergence, the Newton iteration is modified such that successive iterates result in a decrease of the cost function.

V. Global Descent Strategy

One of the most effective methods for modifying the Newton iteration is the Levenberg-Marquardt technique[7-9]:

$$x^{k+1} = x^k - [H^k + \mu^k I]^{-1} g^k; \qquad k = 0, 1, 2, \ldots \tag{18}$$

where we have used the simplified notation $H^k = H(x^k)$ and $g^k = g(x^k)$. The scalar μ^k is known as the *Levenberg-Marquardt parameter* and its proper selection in the interval $[0, \infty)$ ensures that x^{k+1} is a descent iterate, i.e., $J(x^{k+1}) \leq J(x^k)$ for some $\mu^k \in [0, \infty)$. This is easily verified by considering the eigen decomposition of the Hessian matrix:

$$H^k = U^k \Lambda^k U^{kT} \tag{19}$$

where $U^k U^{kT} = U^{kT} U^k = I$, $\Lambda^k = \text{diag}(\lambda_1^k, \lambda_2^k, \ldots, \lambda_n^k)$. Substituting Eq. (19) into Eq. (18) yields

$$x^{k+1} = x^k - U^k [\Lambda^k + \mu^k I]^{-1} U^{kT} g^k; \qquad k = 0, 1, 2, \ldots \tag{20}$$

from which it is clear that when $\mu^k \gg \max\{|\lambda_i^k|\}$, $x^{k+1} \approx x^k - g^k/\mu^k$. Consequently, the Levenberg-Marquardt parameter μ^k interpolates between the Newton step $-[H^k]^{-1} g^k$ and an arbitrarily small step along the direction of steepest descent. A conceptually simple descent algorithm can be obtained from the sequence of scalar minimizations:

$$\min_{\mu^k} J[x(\mu^k)]; \qquad k = 0, 1, 2, \ldots \tag{21a}$$

where

$$x(\mu^k) = x^k - [H^k + \mu^k I]^{-1} g^k \tag{21b}$$

Alternatively, the value of μ^k in Eq. (18) can be selected such that

$$\| x^{k+1} - x^k \| = \rho^k \tag{22}$$

where ρ^k is the radius of a "trust" region in parameter space within which the quadratic approximation to the cost function is valid. Substituting Eq. (20) into Eq. (22), we obtain the explicit relation between the trust region parameter ρ^k and the Levenberg-Marquardt parameter μ^k:

$$\left[\sum_{i=1}^{n} \left(\frac{u_i^{kT} g^k}{\lambda_i^k + \mu^k}\right)^2\right]^{1/2} = \rho^k \tag{23}$$

where u_i^k is the ith column of U^k. Thus, for a given value of ρ^k, the value of μ^k is sought that satisfies the previous nonlinear equation. An efficient iterative technique for computing the largest root of Eq. (23) is due to Hebden and is described in Refs. 7 and 9. Consequently, the problem has shifted from selecting μ^k in Eq. (18) to selecting ρ^k in Eq. (23). The initial value of the trust radius ρ^0 is arbitrary, but if the Hessian is positive definite it may be selected such that $\mu^0 = 0$. In subsequent iterations the trust radius is adjusted in accordance with the ratio:

$$r^k = \frac{J(x^k) - J(x^{k+1})}{J(x^k) - \mathcal{J}^k(x^{k+1})} \tag{24}$$

When $r^k \approx 1$, the quadratic approximation $\mathcal{J}^k(x)$ to the cost function $J(x)$ is excellent, and the trust radius ρ^{k+1} may be increased. Conversely, the quadratic approximation is poor and ρ^{k+1} should be decreased. Such strategy has also been adopted by Bryson and Carrier[10] who address the closely related optimal least squares model reduction problem.

Our global descent algorithm is a variation of the prototype algorithms given by Fletcher[7] and Sorensen.[9] It is implemented as follows:

1) Initialize $x = x^0$, $\rho = \rho^0$, $k = 0$.

2) Compute $J(x^k)$, $g(x^k)$, $H(x^k) = U^k \Lambda^k U^{kT}$ and test for convergence; if $\| g(x^k) \|_\infty <$ tolerance and $\min\{\lambda_i^k\} > 0$, stop.

3) Solve Eq. (23) for μ^k and evaluate r^k from Eq. (24).

4) If $r^k < 0.25$, set $\rho^k = 0.25\rho^k$ and go to step 3. If $r^k > 0.75$, set $\rho^k = 2\rho^k$.

5) Set $x^{k+1} = x^k - U^k[\Lambda^k + \mu^k I]^{-1} U^{kT} g^k$, $\rho^{k+1} = \rho^k$, $k = k + 1$, and go to step 2.

The stability of the iterates $\hat{G}(j\omega, x^k)$ depends on the signs of the parameters α^k and β^k in vector x^k. If any of these is negative, the kth iterate reflects an unstable transfer function. Consequently, the trust radius may be decreased until a stable iterate is produced. This modification can be inserted before step 5.

The eigen decomposition of the Hessian in each iteration is not necessary, and the much faster Cholesky decomposition may be used instead. However, when the number of points p is very large (i.e., in the thousands) and the number of parameters n is considerably smaller (i.e., in the tens), the eigen decomposition does not significantly slow down the algorithm since most of the computation time during each iteration is consumed in evaluating the cost function, gradient, and Hessian.

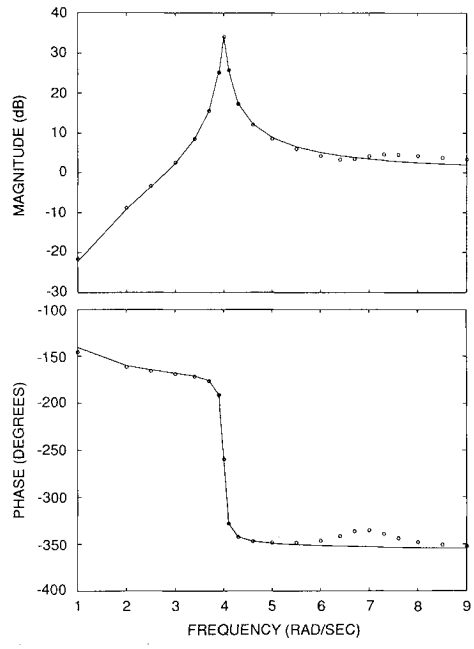

Fig. 1 Frequency response data and second-order optimal fit.

VI. Approximating the Hessian

Here we discuss an approximation to the Hessian that results in reducing the number of computations in each iteration. The cost function of Eq. (6) can be written in the form

$$J(x) = [G - \hat{G}(x)]^* W [G - \hat{G}(x)] \tag{25}$$

where

$$G = \begin{bmatrix} G(j\omega_1) \\ G(j\omega_2) \\ \vdots \\ G(j\omega_p) \end{bmatrix}; \qquad \hat{G}(x) = \begin{bmatrix} \hat{G}(j\omega_1, x) \\ \hat{G}(j\omega_2, x) \\ \vdots \\ \hat{G}(j\omega_p, x) \end{bmatrix} \tag{26}$$

and $W = \text{diag}[w_1, w_2, \ldots, w_n]$ is the diagonal matrix of frequency weights. The nonlinear vector function $\hat{G}(x)$ can be expanded in a power series about x^k and approximated by truncating all but the zeroth and first-order terms:

$$\hat{G}(x) \approx \hat{G}(x^k) + \frac{\partial \hat{G}(x^k)}{\partial x}(x - x^k) \tag{27a}$$

The matrix $\partial \hat{G}(x^k)/\partial x$ is the Jacobian of the error, i.e.,

$$\frac{\partial \hat{G}(x^k)}{\partial x} \triangleq \begin{bmatrix} \frac{\partial \hat{G}(j\omega_1, x)}{\partial x_1} & \frac{\partial \hat{G}(j\omega_1, x)}{\partial x_2} & \cdots & \frac{\partial \hat{G}(j\omega_1, x)}{\partial x_n} \\ \frac{\partial \hat{G}(j\omega_2, x)}{\partial x_1} & \frac{\partial \hat{G}(j\omega_2, x)}{\partial x_2} & \cdots & \frac{\partial \hat{G}(j\omega_2, x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \hat{G}(j\omega_p, x)}{\partial x_1} & \frac{\partial \hat{G}(j\omega_p, x)}{\partial x_2} & \cdots & \frac{\partial \hat{G}(j\omega_p, x)}{\partial x_n} \end{bmatrix}_{x=x^k} \tag{27b}$$

Substitution of Eqs. (27) into Eq. (25) results in a quadratic function that is minimized by $x = x^{k+1}$:

$$x^{k+1} = x^k - [\mathcal{H}(x^k)]^{-1} g(x^k) \tag{28}$$

where

$$g(x^k) = 2\text{Re}\left\{\left[\frac{\partial \hat{G}(x^k)}{\partial x}\right]^* W [\hat{G}(x^k) - G]\right\} \tag{29}$$

and

$$\mathcal{H}(x^k) = 2\text{Re}\left\{\left[\frac{\partial \hat{G}(x^k)}{\partial x}\right]^* W \left[\frac{\partial \hat{G}(x^k)}{\partial x}\right]\right\} \tag{30}$$

Equation (29) gives the gradient of the cost function and is in agreement with Eq. (7). The matrix $\mathcal{H}(x)$ is a positive definite (or positive semidefinite) approximation to the Hessian $H(x)$ given in Eq. (8). Clearly $\mathcal{H}(x)$ corresponds to the first term on the right side of Eq. (8), which depends only on first-order derivatives of the transfer function $\hat{G}(j\omega, x)$.

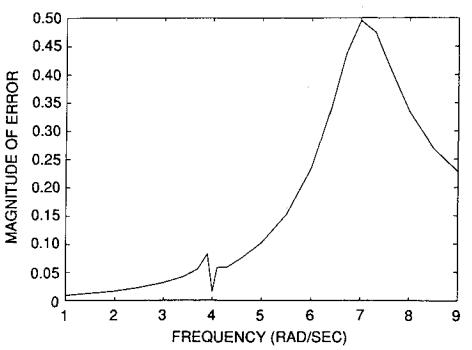

Fig. 2 Magnitude of the approximation error.

Equation (28) defines an iteration (also known as the *Gauss-Newton iteration*[7,8]) that is equivalent to solving a sequence of linear least squares problems. As a result, the formation of the approximate Hessian $\mathcal{H}(x)$ is not necessary, and the orthogonal decomposition of the Jacobian $\partial \hat{G}(x^k)/\partial x$ may be used to solve the implied linear least squares problem in each iteration. If $J(x)$ is minimized at x^{opt} and if the weighted error $\|W^{1/2}[G - \hat{G}(x^{\text{opt}})]\| = \sqrt{J(x^{\text{opt}})}$ is sufficiently small, the Gauss-Newton iteration is effective and converges linearly when initialized in the neighborhood of x^{opt}. Along these lines, $\|W^{1/2}[G - \hat{G}(x^{\text{opt}})]\|$ is considered large if it is comparable to the largest eigenvalue of $\mathcal{H}(x^{\text{opt}})$.[8] On the other hand, if the cost function becomes identically zero at x^{opt} (i.e., an exact fit to the data), it is evident from Eq. (8) that $\mathcal{H}(x^{\text{opt}}) = H(x^{\text{opt}})$ and the convergence rate of the Gauss-Newton iteration is quadratic. So, the smaller the error at the minimum, the faster the Gauss-Newton iteration converges. As with the Newton iteration, when initialized far from the minimum, iteration (28) may not converge at all; however, the Levenberg-Marquardt modification discussed earlier provides for a global descent algorithm.

The Gauss-Newton method is more computationally efficient than its Newton counterpart and should be preferred when it is anticipated that the cost function will be small at the minimum (clearly the case of interest in most transfer function curve-fitting problems). In certain cases the order of the transfer function can be selected sufficiently large to insure that the cost function is indeed small at the minimum. On the other hand, the Newton iteration can be used when the cost function is not expected to be small at the minimum as in cases where the frequency response measurement is contaminated by large levels of noise.

VII. Initialization

The proposed algorithm requires that the parameter vector x be initialized. However, not all of the parameters need be initialized. We will show that the cost function is quadratic in the parameters a, b, and d, and, as a result, only parameters α and β require initial values.

The parameter vector x can be partitioned as follows:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \quad \text{where} \quad x_1 = \begin{bmatrix} a \\ b \\ d \end{bmatrix}, \quad x_2 = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{31}$$

The transfer function defined in Eq. (3) can be written in the form

$$\hat{G}(j\omega, x) = [A(j\omega, x_2)]\, x_1 \tag{32}$$

where

$$A(j\omega, x_2) = [B^T(j\omega, x_2) \;\; j\omega B^T(j\omega, x_2) \;\; 1] \tag{33}$$

and

$$B(j\omega, x_2) = \begin{bmatrix} (\alpha_1 + j\omega\beta_1 - \omega^2)^{-1} \\ (\alpha_2 + j\omega\beta_2 - \omega^2)^{-1} \\ \vdots \\ (\alpha_m + j\omega\beta_m - \omega^2)^{-1} \end{bmatrix} \tag{34}$$

In view of Eq. (32) and the definition of $\hat{G}(x)$ in Eq. (26), the cost function of Eq. (25) takes the form

$$J(x_1, x_2) = [G - A(x_2)\, x_1]^* W [G - A(x_2)\, x_1] \tag{35}$$

where

$$A(x_2) = \begin{bmatrix} A(j\omega_1, x_2) \\ A(j\omega_2, x_2) \\ \vdots \\ A(j\omega_p, x_2) \end{bmatrix} \tag{36}$$

The structure of Eq. (35) reveals that the cost function is quadratic in the parameter vector x_1. We point out that nonlinear least squares problems of the general form of Eq. (35) have been addressed by Golub and Pereyra[11] and Ruhe and Wedin.[12] In their work, such special structure is exploited by algorithms that minimize the cost function with respect to x_1 and x_2 separately.

Consequently, for any initial value of x_2, an initial value for x_1 can be found that satisfies the minimization problem:

$$\min_{x_1} J(x_1, x_2) \tag{37}$$

This is a weighted least squares problem with solution

$$x_1 = \{\mathrm{Re}[A^*(x_2)WA(x_2)]\}^{-1}\,\mathrm{Re}[A^*(x_2)WG] \tag{38}$$

Therefore, only the parameter x_2 needs to be initialized. The initial value of x_1 can be set from Eq. (38). A good initial value of x_2 can often be obtained from the iterative linear least squares algorithm of Sanathanan and Koerner.[2] We have used the numerically superior implementation of Dailey and Lukich[5] to initialize x_2 with much success.

VIII. Examples

The algorithm will now be applied to two examples. First, the frequency response of a fourth-order transfer function is sampled at 22 points and fitted with a second-order transfer function. Second, 1600 frequency response measurements from an experimental flexible structure are fitted with two transfer functions of different order and the results obtained are compared with the data.

Example 1

The purpose of this example is to demonstrate that the proposed algorithm can be used for model reduction as well as for model synthesis. Consider the frequency response given in Table 1. This is obtained by sampling the fourth-order transfer function

$$G(s) = \frac{1.1s^4 + 0.688s^3 + 49.592s^2 - 35.28s}{s^4 + 1.48s^3 + 65.112s^2 + 26.32s + 784}$$

at $s = j\omega_i$, $i = 1, \ldots, 22$, where ω_i are the 22 frequency points in the first column of Table 1. The preceding transfer function represents a system with two flexible modes: the first is at 4 rad/s with 1% damping whereas the second is at 7 rad/s with 10% damping. We seek the second-order transfer function $\hat{G}(s) = (a + bs)/(\alpha + \beta s + s^2) + d$ that represents the best fit to the data in the least squares sense. Using unity weighting (i.e., $w_i = 1$; $i = 1, 2, \ldots, 22$), our algorithm converged to the transfer function

$$\hat{G}(s) = \frac{-15.9937 - 0.8216s}{15.9986 + 0.0800s + s^2} + 1.0071$$

The cost function at the minimum was

$$\sum_{i=1}^{22} \left| G(j\omega_i) - \hat{G}(j\omega_i) \right|^2 = 1.2979$$

The largest element of the gradient at the computed minimum was approximately zero (i.e., $\| g(x^{\mathrm{opt}}) \|_\infty \approx 10^{-12}$), and the Hessian was positive definite (i.e., smallest eigenvalue at +5.5873). The magnitude and phase of the complex curve fit are plotted in Fig. 1. It is shown that the optimal second-order transfer function captures the dominant mode at 4 rad/s. In addition, a plot of the approximation error is shown in Fig. 2. As expected, the magnitude of the error is largest at frequencies near the nondominant mode at 7 rad/s.

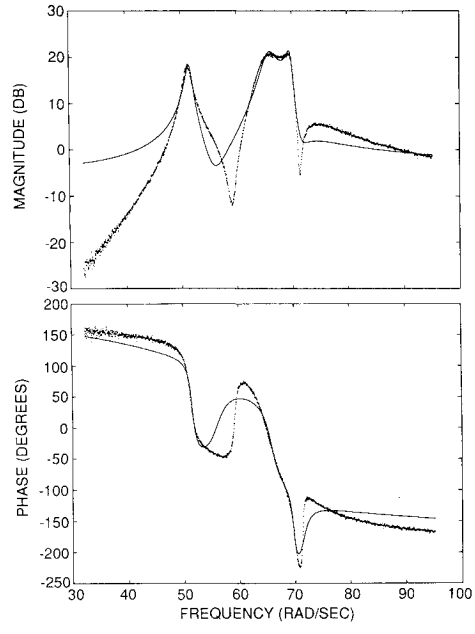

Fig. 3 Experimental data and sixth-order optimal fit.

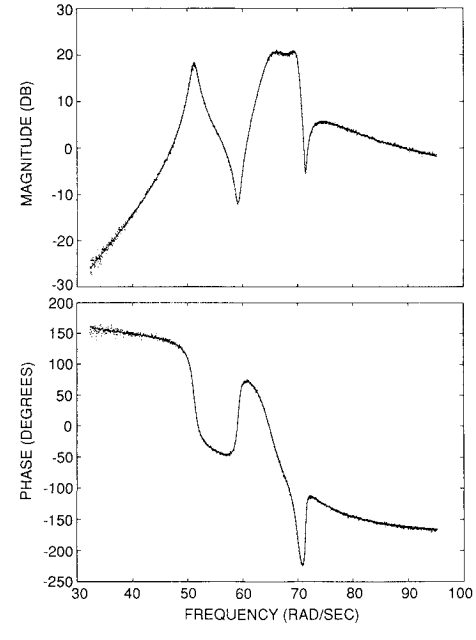

Fig. 4 Experimental data and eighth-order optimal fit.

Example 2

This example illustrates the application of the algorithm in system identification of a lightly damped flexible structure. The problem involved fitting a transfer function to frequency response data taken from the Jet Propulsion Laboratory's precision truss structure.[13] There are 1600 frequency response measurements in the range 30–100 rad/s representing a transfer function from a force actuator to an accelerometer. Direct inspection of the magnitude plot suggested that a transfer function of order six or higher would be necessary to adequately capture the data.

Figure 3 shows the sixth-order optimal fit to the data. The large error incurred indicates that a higher order curve fit should be attempted. Using an eighth-order transfer function, the Newton algorithm converged to the result shown in Fig. 4. Clearly, the quality of the approximation is visually excellent, and it is concluded that, in the frequency range of interest, the dynamics of the flexible structure can be accurately modeled by an eighth-order transfer function.

Table 1 Frequency response data

ω, rad/s	Real $[G(j\omega)]$	Imag $[G(j\omega)]$
1.0	−0.0690	−0.0476
2.0	−0.3437	−0.1150
2.5	−0.6591	−0.1702
3.0	−1.3170	−0.2618
3.4	−2.6579	−0.3865
3.7	−6.0140	−0.3819
3.9	−17.9136	3.5101
4.0	−9.0471	50.0080
4.1	16.5110	10.3599
4.3	7.0597	2.2383
4.6	3.9623	0.9463
5.0	2.6586	0.5504
5.5	1.9742	0.3998
6.0	1.5984	0.3853
6.4	1.4093	0.4714
6.7	1.3743	0.5972
7.0	1.4818	0.6779
7.3	1.6126	0.6066
7.6	1.6473	0.4747
8.0	1.6054	0.3423
8.5	1.5292	0.2503
9.0	1.4640	0.1998

As a final remark we point out that the curve fits in Figs. 3 and 4 were obtained using nonunity frequency weighting. In particular, we found that the choice

$$w_i = \frac{1}{|G(j\omega_i)|^2}; \qquad i = 1, 2, \ldots, 1600$$

produced better quality approximations than the weighting $w_i = 1$. From this and other examples it became evident that, when unity weighting was used, the curve fits often had large errors in the frequency ranges where the magnitude of the measurement $|G(j\omega_i)|$ was small. This was observed particularly in cases where there were large (i.e., better than 60 dB) variations in the magnitude of the response. On the other hand, the inverse-squared weighting tends to emphasize the measurements with small magnitude at the expense of measurements with large magnitude, thereby better capturing the data near the antiresonances while sacrificing some accuracy near the resonances. Naturally, when the fitted transfer function model is to be used for control system design, the frequency weighting w_i should be selected to emphasize the frequency region where closed-loop performance is critical (i.e., near the loop gain cross-over frequency).

IX. Summary and Conclusions

In this paper we have considered the problem of synthesizing single-input, single-output transfer functions from frequency response measurements. The unknown transfer function is parameterized by a sum of second-order rational polynomials. An iterative algorithm of the Newton type is developed to solve the complex curve-fitting problem in the least squares sense. Given an initial estimate of the unknown transfer function poles, the algorithm seeks a local minimum of the cost function using first- and second-order derivative information. Regardless of the initial estimate, the cost function is decreased at each iteration and all iterates are constrained to have stable poles. This descent algorithm does not guarantee convergence to the global minimum of the cost function which may exhibit multiple local minima. However, our experience with numerical examples indicates that, when the cost function has multiple local minima, the approximation is often poor and the order of the transfer function can be safely increased to improve the results. Two numerical examples demonstrate that the algorithm is effective in practice.

Acknowledgment

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with NASA. This work was supported by the JPL Control-Structures Interaction program.

References

[1]Levy, E. C., "Complex-Curve Fitting," *IRE Transactions on Automatic Control,* Vol. AC-4, May 1959, pp. 37–44.

[2]Sanathanan, C. K., and Koerner, J., "Transfer Function Synthesis as a Ratio of Two Complex Polynomials," *IEEE Transactions on Automatic Control,* Vol. AC-8, Jan. 1963, pp. 56–58.

[3]Richardson, M. H., and Formenti, D. L., "Parameter Estimation from Frequency Response Measurements Using Rational Fractional Polynomials," *Proceedings of the First International Modal Analysis Conference* (Orlando FL), 1982, pp. 167–181.

[4]Adcock, J. L., "Curve Fitter for Pole-Zero Analysis," *Hewlett-Packard Journal,* Jan. 1987, pp. 33–36.

[5]Dailey, R. L., and Lukich, M. S., "MIMO Transfer Function Curve Fitting Using Chebyshev Polynomials," SIAM 35th Anniversary Meeting, Denver, CO, 1987.

[6]Whitfield, A. H., "Asymptotic Behaviour of Transfer Function Synthesis Methods," *International Journal of Control,* Vol. 45, No. 3, 1987, pp. 1083–1092.

[7]Fletcher, R., *Practical Methods of Optimization,* 2nd ed., Wiley, New York, 1987.

[8]Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization,* Academic Press, New York, 1981.

[9]Sorensen, D. C., "Newton's Method with a Model Trust Region Modification," *SIAM Journal of Numerical Analysis,* Vol. 19, 1982, pp. 409–426.

[10]Bryson, A. E., and Carrier, A., "A Second Order Convergent Algorithm for Optimal Multi-Input Multi-Output System Order Reduction," *Journal of Guidance, Control, and Dynamics* (to be published).

[11]Golub, G. H., and Pereyra, V., "The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate," *SIAM Journal of Numerical Analysis,* Vol. 10, 1973, pp. 413–432.

[12]Ruhe, A., and Wedin, P., "Algorithms for Separable Nonlinear Least Squares Problems," *SIAM Review,* Vol. 22, 1980, pp. 318–337.

[13]Fanson, J., and Briggs, H., "JPL Phase-0 CSI Experiment Results and Real Time Control Computer," Fourth Annual NASA/DOD Control-Structures Interaction Conference, Orlando, FL, Nov. 1990.