Application of Interval Analysis Techniques to Linear Systems: Part III—Initial Value Problems

EDWARD P. OPPENHEIMER, MEMBER, IEEE, AND ANTHONY N. MICHEL, FELLOW, IEEE

Abstract—**In Part I of the present three-part paper [3] we established new results for continuous and rational interval functions which are of interest in their own right. We used these results in the second part of the present three-part paper [1] to study interval matrix exponential functions and to devise a method of constructing augmented partial sums which approximate interval matrix exponential functions as closely as desired. In the present paper (the third part of the present three-part paper) we use the above results to develop an algorithm which enables us to obtain estimates of bounds for the set of all solutions of initial-value problems of linear systems of autonomous first-order ordinary differential equations that linearly depend on a parameter belonging to an interval. We demonstrate the applicability of our results by considering three specific examples: an *RLC* circuit, an instrument servomechanism, and the design of a minimum plant sensitivity optimal linear regulator.**

I. Introduction

In a companion paper [3] we established new results for continuous and rational interval functions. In a second companion paper [1] we used the results of [3] to study interval exponential functions and to develop an algorithm which enables us to obtain reasonably good estimates for the range of interval matrix exponential functions of an interval (parameter) variable. Since this algorithm is implemented on a finite wordlength digital computer, it is necessary to incorporate machine bounding arithmetic.

In the present paper we apply the results developed in [1] and [3] to initial-value problems described by linear systems of autonomous first-order ordinary differential equations which depend on a parameter belonging to an interval (i.e., a parameter with tolerances). Our aim is to develop algorithmic results which will enable us to obtain interval bound solutions (solution envelopes) for such initial-value problems. We demonstrate the applicability of the method advanced herein by applying our results to three specific examples.

The present paper is organized as follows:

In Section III we consider initial-value problems described by systems of autonomous first-order ordinary differential equations which depend linearly on a perturbation parameter θ which belongs to an interval. We associate with such initial-value problems subproblems which are obtained by dividing the original parameter interval into several subintervals. We then utilize the results of [1, sec. VII] to obtain interval bound solutions (envelopes) for each of these subproblems. Finally, an estimate for the interval solution for the initial-value problem is obtained by taking the union over all of the subproblem interval solution estimates. This procedure of subdividing the parameter interval (of the original initial-value problem) into several subintervals, to reduce conservativeness, is consistent with theorem 14M of [3].

The algorithm to accomplish the above procedure is described in Section IV while the computer program to implement this algorithm is described in Section V.

In Section VI we discuss the relationship of the present approach (which is applicable to large parameter variations) to sensitivity and robustness (which usually are applicable to infinitesimal parameter variations).

We demonstrate the applicability of our results by considering three specific examples: (a) an *RLC* circuit, (b) an instrument servomechanism, and (c) the design of a minimum plant sensitivity optimal linear regulator.

II. Notation

In the subsequent sections, we will make use of the notation established in the first two parts of the present three-part paper [1], [3]. This notation will not be repeated here.

III. Initial-Value Problems

The class of initial-value problems which we consider are described by a system of linear, autonomous, first-order ordinary differential equations,

$$\dot{x} = Ax, \qquad x(0) = x_0 \tag{1}$$

where the constant $n \times n$ matrix $A = ((a_{ij}))$ contains elements which depend linearly on a perturbation parameter θ, where $x \in R^n$, where $\dot{x} = dx/dt$, and where x_0 denotes the initial condition.

In order to facilitate matrix element linear dependence on a negative or decreasing value of the parameter, the algorithm developed in the present section is designed to

Manuscript received January 21, 1987; revised December 2, 1987. The work of A. N. Michel was supported in part by the National Science Foundation under Grant ECS84-19918. This paper was recommended by Associate Editor F. M. A. Salam.

E. P. Oppenheimer is with the Applied Physics Laboratory, Johns Hopkins University, Laurel, MD 20707.

A. N. Michel is with the Department of Electrical and Computer Engineering, University of Notre Dame, Notre Dame, IN 46556.

IEEE Log Number 8822598.

input the matrix elements $a_{ij} = [a_{ij}^L, a_{ij}^R]$ as a set of two single-precision floating-point numbers, not necessarily "interval-ordered," where it is permissible to have $a_{ij}^R < a_{ij}^L$. Then the matrices

$$G_1 = \left(\left(g_{ij}^1\right)\right) \triangleq \left(\left(\left(a_{ij}^R + a_{ij}^L\right)/2\right)\right)$$
$$G_2 = \left(\left(g_{ij}^2\right)\right) \triangleq \left(\left(\left(a_{ij}^R - a_{ij}^L\right)/2\right)\right) \quad (2)$$

are computed employing bounding interval arithmetic (see [1]) and the perturbation parameter dependent differential equation may be formulated as

$$\dot{x} = (G_1 + \theta G_2)x, \qquad x(0) = x_0$$
$$\theta \in [-1.0, +1.0]. \quad (3)$$

In this manner, the coefficient matrix elements $g_{ij}^1 + \theta g_{ij}^2$ accommodate the "signed" dependence on the perturbation parameter $\theta \in [-1.0, +1.0]$ as a result of the sign of g_{ij}^2.

For any value of the perturbation parameter $\theta \in [-1.0, +1.0]$ (assume for this discussion that G_1 and G_2 are real matrices and that $x(0) = x_0$ is a real vector), the unique solution of (3) is given by (see [2])

$$x(t) = \Phi(\theta, t)x_0 \quad (4)$$

where $\Phi(\theta, t)$ is the state transition matrix given by

$$\Phi(\theta, t) \triangleq e^{(G_1 + \theta G_2)t}. \quad (5)$$

Consistent with our results in [1] and [3], the interval solution technique of our algorithm will be to subdivide the $[-1.0, +1.0]$ perturbation interval into M equal width subintervals defined by (single-precision endpoint computations),

$$\theta_i \triangleq \left[\frac{-M+2(i-1)}{M}, \frac{-M+2i}{M}\right], \qquad i = 1, \cdots, M \quad (6)$$

and to compute, in a manner analogous to (4), the interval solutions for each of the *subproblems* given by

$$\dot{x} = (G_1 + \theta G_2)x$$
$$x(0) = x_0$$
$$\theta \in \theta_i, \qquad i = 1, \cdots, M. \quad (7)$$

In the implementation of our algorithm (the details of which may be found in [4]), (7) is evaluated at equally spaced intervals of time. To be more specific, the linear algorithm subproblem solution technique which we advocate, employs the specific solution value of time and the linearity of the problem (see (4)) by computing subproblem "pseudofundamental" interval matrices which set theoretically include the set of all real fundamental matrices for each value of the parameter in the subinterval (see (5) and (6)) and which element-wise satisfy an "inclusion" error bound. Finally, the interval solution for (3) is obtained by taking the union over the subproblem interval solutions to (7), producing interval bounds or envelopes for the set of all solutions associated with the interval vector initial condition and the perturbation parameter, including the effects of algorithmic computer truncation or rounding errors.

IV. Description of the Algorithm

The interval solution algorithm for the present initial-value problem makes use of the results for the nested centered form computation of the interval fundamental matrix which we developed in section VII of [1]. Rather than restate all of these results for the initial-value problem on hand, we rephrase and specialize only those relations from [1] and [3] which we need to describe our algorithm. To this end, the $\theta_{ic} + \eta_i$ centered form interval representation for each subinterval θ_i (given in (6)) is computed in single-precision arithmetic defined by

$$\theta_{i_c} \triangleq [c_i, c_i], \qquad c_i = \frac{-M+2i-1}{M}$$

$\eta_i \triangleq [-w_i, w_i]$, where

$$w_i \triangleq \max\left\{c_i - \left[\frac{-M+2(i-1)}{M}\right], \left[\frac{-M+2i}{M}\right] - c_i\right\}. \quad (8)$$

Now let the following symbols on top of Φ denote:

(no symbol)	an exact interval arithmetic calculation,
$\tilde{(\,)}$	a bounding interval arithmetic calculation,
$\bar{(\,)}$	the united extension (see equation (8) in [3]),
$\hat{(\,)}$	an "augmented" bounding interval arithmetic calculation (see equation (67) in [1]).

Also, let

$$\Phi_K(\Phi_{jc} + \eta_j, t_1) = \text{the nested centered form } K \text{ term truncated series calculation for eq. (57) in [1].} \quad (9)$$

In accordance with section VII in [1], let

$$\Phi(\theta_{jc} + \eta_j, t_1) = \Phi_K(\theta_{jc} + \eta_j, t_1) + R_K(\theta_{jc} + \eta_j, t_1) \quad (10)$$

and let r_{lm_K} denote the elements of the remainder term $R_K(\theta_{jc} + \eta_j, t_1)$.

For the present initial-value problem, the relationship (66) in [1] assumes now the form

$$\left|r_{lm_K}(\theta_{jc} + \eta_j, t_1)\right| \leq \left(\tilde{\tilde{g}}_{l_j}/\tilde{\tilde{g}}_{m_j}\right)\tilde{\tilde{\tau}}_{u_j}$$
$$\leq 16^{-P} \min\left\{\left|\tilde{\phi}_{lm_K}^L(\theta_{jc} + \eta_j, t_1)\right|, \left|\tilde{\phi}_{lm_K}^R(\theta_{jc} + \eta_j, t_1)\right|\right\} \quad (11)$$

where $|\cdot|$ denotes the interval magnitude operation and where the center term in (11) is a double-precision calculation using double-precision computed arguments (denoted by $\approx$) with the result truncated to a single-precision number and the relationship checked in the single precision arithmetic. In (11), the terms $\tilde{\phi}_{lm_K}^L(\cdot)$ and $\tilde{\phi}_{lm_K}^R(\cdot)$ are defined by the matrix

$$\tilde{\Phi}_K = \left(\left(\tilde{\phi}_{lm_K}, \tilde{\phi}_{lm_K}^R\right)\right).$$

The augmented centered form final result $\hat{\Phi}_K(\theta_{jc}+\eta_j, t_1)$ computed by eq. (67) in [1] satisfies the relationship:

$$\begin{aligned}&[1-\epsilon, 1+\epsilon]\cdot\Phi(\theta_{jc}+\eta_j, t_1)\\ &\quad\supset \hat{\Phi}_K(\theta_{jc}+\eta_j, t_1)\\ &\quad\triangleq \tilde{\Phi}_K(\theta_{jc}+\eta_j, t_1)+Z_j \supset \Phi(\theta_{jc}+\eta_j, t_1)\\ &\quad\supset \overline{\Phi}(\theta_{jc}+\eta_j, t_1)\end{aligned} \tag{12}$$

where

$$Z_j = \left(\left(\left[-\left(\tilde{\tilde{g}}l_j/\tilde{\tilde{g}}_{m_j}\right)\tilde{\tilde{\tau}}_{u_j}, \left(\tilde{\tilde{g}}_{l_j}/\tilde{\tilde{g}}_{m_j}\right)\tilde{\tilde{\tau}}_{u_j}\right]\right)\right) \tag{13}$$

and the absolute value of the interval endpoint errors of the computed elements of $\hat{\Phi}_K$ relative to the corresponding endpoints of the elements of the exact interval arithmetic result Φ are bounded by (see [1, eqs. (26), (27)])

$$\epsilon = 2\left(\frac{16^{-P}}{1-16^{-P}}\right). \tag{14}$$

(This formulation for the augmenting matrix Z_j assumes that the $\|\cdot\|_u$ rather than the $\|\cdot\|_{u'}$ optimal Householder matrix norm has been used in computing the bound τ_{u_j} in eq. (65) in [1].)

The results of propositions 12M and 13M in [3] assume in the present case the form

$$\bigcup_{j=1}^{M} \Phi_K(\theta_{jc}+\eta_j, t_1) \supset \bigcup_{j=1}^{M} \overline{\Phi}_K(\theta_{jc}+\eta_j, t_1) \tag{15}$$

and

$$\bigcup_{j=1}^{M} \Phi(\theta_{jc}+\eta_j, t_1) \supset \bigcup_{j=1}^{M} \overline{\Phi}(\theta_{jc}+\eta_j, t_1). \tag{16}$$

Turning now to the subproblem solution method, each numerical subproblem bounding interval arithmetic solution (denoted by $\tilde{x}_j(t_1)$) is computed as a bounding interval arithmetic matrix product (see (4) for the analogy) and provides the set inclusion relationship

$$\begin{aligned}&\left\{[1-\epsilon, 1+\epsilon]\cdot\Phi(\theta_{jc}+\eta_j, t_1)\right\}x(0)\\ &\quad\supset \tilde{x}_j(t_1)\\ &\quad\triangleq \hat{\Phi}(\theta_{jc}+\eta_j, t_1)x(0) \supset \Phi(\theta_{jc}+\eta_j, t_1)x(0)\\ &\quad\supset \overline{\Phi}(\theta_{jc}+\eta_j, t_1)x(0) \supset \bigcup_{\eta\in\eta_j}\bigcup_{x_0\in x(0)}\left\{\Phi(\theta_{jc}+\eta, t_1)x_0\right\}\end{aligned} \tag{17}$$

where x_0 is meant to denote any degenerate interval (real) vector initial condition. Although (12) provides an analytic error bound for the elements of $\hat{\Phi}_K$ with respect to Φ, a similar type of expression is not possible for $\tilde{x}_j(t_1)$ by virtue of the complexity of even the exact interval arithmetic (see [3, eq. (1)]) involved in (17).

Finally, denoting the algorithm interval solution for (3) as $\tilde{x}(t_1)$, we obtain the relationship

$$\begin{aligned}&\bigcup_{j=1}^{M}\left\{[1-\epsilon, 1+\epsilon]\cdot\Phi(\theta_{jc}+\eta_j, t_1)\right\}x(0)\\ &\quad\supset \tilde{x}(t_1)\\ &\quad\triangleq \bigcup_{j=1}^{M}\tilde{x}_j(t_1) \supset \bigcup_{j=1}^{M}\Phi(\theta_{jc}+\eta_j, t_1)x(0)\\ &\quad\supset \bigcup_{j=1}^{M}\overline{\Phi}(\theta_{jc}+\eta_j, t_1)x(0)\\ &\quad\supset \bigcup_{j=1}^{M}\bigcup_{\eta\in\eta_j}\bigcup_{x0\in x(0)}\left\{\overline{\Phi}(\theta_{jc}+\eta_j, t_1)x_0\right\}\end{aligned} \tag{18}$$

where again an analytic error bounding expression is not possible for $\tilde{x}(t_1)$.

We are now in a position to give a general description of the computer program of the algorithm outlined above. This is accomplished in the next section.

We conclude by observing that theorem 14M of [3], applied to the present initial-value problem, gives justification to the philosophy of our algorithm of increasing the number of subintervals in the perturbation parameter interval with the expectation of obtaining less conservative results for the overall solution technique.

V. Description of the Computer Program for the Algorithm

In the present section we provide a brief description of the operation and organization of the computer program for our algorithm. For a complete description of this computer program, refer to [4].

Before going into specifics, we note that in the computer implementation of our algorithm, we make use of interval-bounding arithmetic in order to provide *true* interval solution bounds. Such interval arithmetic implementations were first developed for mainframes (see, e.g., [5]–[7]). More recent interval arithmetic routines include the mainframe ACRITH package [8] and the microcomputer based Pascal-SC language [9].

We now turn to the description of our computer program.

The *inputs* to the program are as follows:

(a) dimension and parameter dependent interval coefficient matrix A, where *inverted* a_{ij} element "interval" ordering is permitted to provide a decreasing signature with increasing parameter value;
(b) integer number of parameter interval subdivisions required;
(c) integer choosing one of the two Householder matrix norms;
(d) integer negative exponent value of the base 16 used in the algorithm truncation bound test (see inequality (11));

(e) integer number of Δt solution steps required;
(f) real value of Δt;
(g) integer starting value number of terms in the series to be initially attempted for each subproblem fundamental matrix term (value run up is automatic for described failures);
(h) integer limit value for number of acceptable failures of inequality (11) (called a "single fault") for any subproblem fundamental matrix term before a failure is declared. (See also [1, p. 38]);
(i) integer limit value for number of acceptable failures of inequality (11) with "max" replacing "min" (called a "double fault") for any subproblem fundamental matrix term before a failure is declared. (See also [1, p. 38]);
(j) integer limit value for number of times the subproblem fundamental matrix term may be computed with error counts exceeding those given in (h) and (i) above and still be accepted before the "rolling storage" mechanism is in effect (explanation follows);
(k) problem interval vector of initial conditions.

The *outputs* of the program include a file of times and total problem interval solution values (i.e., interval union over the subproblem interval solution values), where the first interval solution values are the initial conditions at $t = 0$. Various diagnostics are output to support analysis of the algorithm performance and certain installation dependent plotting files and parameters are produced. Only a moving window of subproblem interval solution values are maintained in the computer during execution to minimize algorithm operating storage, and none are written to output files.

In order to provide the greatest versatility, the algorithm was designed so that the user writes a main controlling program incorporating a specified sequence of calls to the subroutines, additionally providing their required sequence of initializing inputs and program controls.

Following is a brief description of the subroutines called and their functions.

A. NORM

The subroutine inputs the problem dimension, parameter dependent A "interval" coefficient matrix, Householder matrix norm selection index, and number of parameter subdivisions desired. The G_1 and G_2 matrices (equation (2)) are computed in the interval arithmetic, and the $[-1, +1]$ interval is similarly subdivided (equation (6)) for the subproblem computations. The selected Householder matrix norms ([1, eqs. (29) or (31)]) are computed for the subproblems following solution for their maximal eignevalues and corresponding eigenvectors with the EISPAC routines.

B. GCOMP

This subroutine computes the balance of the *complete* sequence of "G" interval matrices $G_3, G_4, \cdots, G_{230}$ [1, eq. (45)]), for the maximal algorithm design storage value of 20 matrix exponential series terms. These are stored with G_1 and G_2 interval matrices for subsequent use in the centered form computations by subroutine PHCOMP.

C. PRCOMP

This subroutine computes the *complete* sequence of interval binomial coefficients $C_1, C_2, \cdots, C_{210}$ (equation (51) in [1]), for the maximal algorithm design storage value of 20 matrix exponential series terms. These are stored for subsequent use in the centered form computations in subroutine PHCOMP. The integer P is input here and the test factor 16^{-P} (equation (11)) is computed and saved. The centered form interval representations for the subintervals (equation (6)) are computed by (8) and stored.

D. DRIVER

This subroutine controls the algorithm's solution technique. Its opening sequence inputs Δt, the number of solution steps desired, and the problem vector interval initial conditions. All subproblems are initialized with these values, completing the opening sequence.

On the first iteration $t_1 = \Delta t$ and subroutine PHCOMP is called which computes the augmented subproblem centered form fundamental matrix approximations $\hat{\Phi}_K(\theta_{jc} + \eta_j, t_1)$ of (12), where K is the number of terms in the series approximation that satisfies the specified relative error bounds. (The initializing value of K is input on the opening call to and within subroutine PHCOMP, where it may be later automatically increased to satisfy the relative error bounds there.) Then the subproblem solutions $\tilde{x}_j(t_1)$ are computed (see (17)) and stored. Finally, the total problem vector interval solution union $\tilde{x}(t_1)$ is computed (see (18)) and output and the next iteration begins with calls to PHCOMP to compute $\hat{\Phi}_K(\theta_{jc} + \eta_j, t_2)$, $t_2 = 2\Delta t$.

Successful computation of the $\hat{\Phi}_K(\theta_{jc} + \eta_j, t_n)$ matrices is determined within PHCOMP, where the algorithm "conservation of storage" technique limits $n \leqslant 8$ (similarly, $K \leqslant 20$). The final acceptable augmented fundamental matrix approximations occurring at t_N (i.e., $\hat{\Phi}_K(\theta_{jc} + \eta_j, t_N)$ are used for the remaining subproblem solutions, *without* further calls to PHCOMP.

To demonstrate, suppose $t_N = 2\Delta t$ (i.e., the final acceptable fundamental matrix approximations occurs at $N = 2$). Then DRIVER functions as

$$\tilde{x}_j(t_1) = \hat{\Phi}_K(\theta_{jc} + \eta_j, t_1)x_{j0}$$
$$\tilde{x}_j(t_2) = \hat{\Phi}_j x_{j0}, \quad \text{where } \hat{\Phi}_j \equiv \hat{\Phi}_K(\theta_{jc} + \eta_j, 2\Delta t)$$
$$\tilde{x}_j(t_3) = \hat{\Phi}_j \tilde{x}_j(t_1)$$
$$\tilde{x}_j(t_4) = \hat{\Phi}_j \tilde{x}_j(t_2)$$
$$\cdots, \text{etc.}$$

For this example where the PHCOMP tests have prematurely shut off at $N = 2$, only three sets of subproblem solutions are maintained in the "rolling storage" scheme (i.e., in storage at any one time), namely $\tilde{x}_j(t_i)$, $\tilde{x}_j(t_{i+1})$, and $\tilde{x}_j(t_{i+2})$, $i = 1, 2, \cdots$. The next set of subproblem solutions $\tilde{x}_j(t_{i+3}) = \hat{\Phi}_j \tilde{x}_j(t_{i+1})$ reuse storage of the $\tilde{x}_j(t_i)$

solutions, and so on. If the $\hat{\Phi}_K$'s had been computed successfully with increasing n (i.e., not shut off at $n=2$ like the example), the automatic algorithm limit of $N=8$ would have enabled the "rolling storage" technique at that point, closing down further calls to PHCOMP with the result that a final set of subproblem transition matrices $\hat{\Phi}_j \equiv \hat{\Phi}_K(\theta_{jc}+\eta_j, 8\Delta t)$ would be used for the balance of the solution execution. In that case the *first* "rolling storage" subproblem solutions would have been $\tilde{x}_j(t_9) = \hat{\Phi}_j \tilde{x}_j(t_1)$.

E. PHCOMP

On the opening call to PHCOMP from DRIVER, the beginning value K_0 is input for the number of terms to be initially attempted in the computation of the subproblem fundamental matrix elements. Limit values are read for the number of "single" and "double" fault errors acceptable on any one set of subproblem fundamental matrices. The limit number for the number of individual subroutine entry failures acceptable before a PHCOMP failure is declared is input (i.e., value at which further calls to PHCOMP are shut off and a final set of $\hat{\Phi}_j$'s are declared) and the limit number for the number of automatic "run-ups" allowed in the increases in the value of K is read.

The interval values of t_i and factorials (for the current K) necessary for the computation of the τ_{u_j}'s of (11) are computed. Then the bulk of the interval computations are performed:

- the nested values of the B terms (equations (52a) and (52b) in [1]);
- the τ_{u_j} terms in equation (65) in [1];
- the nested E terms (equations (55a) and (55b) in [1]);
- the nested centered form for $\hat{\Phi}_K(\theta_{jc}+\eta_j, t_i)$ terms ((12) and [1, eq. (57)];
- the error criteria is tested ([1, eq. (66)] and (11)); and
- the augmented fundamental matrix approximation terms $\hat{\hat{\Phi}}_K(\theta_{jc}+\eta_j, t_i)$ are computed (see [1, eqs. (12), (13), and (67)].

The various algorithmic testings and responses for the K "run-up" and enabling of the "rolling storage" technique are deferred to the reference [4].

F. CURVES

This subroutine is directed by the program input stream and it is called to produce CalComp drum type incremental plotting via the installation dependent SIMPLOT routines. For this reason, it will not be discussed further at this time.

Various diagnostics are provided to assist in the analysis of the algorithm performance and these are also contained in the reference. A possible sequence of calls from the user written main program and the subroutines would be (by letter and number of call), $A1$, $B1$, $C1$, $D1$ ($E1$, $E2$, $E3, \cdots, E6$), $F1$. (The "E" calls shown here in parentheses are meant to depict an automatic algorithm PHCOMP "shut off" at $n=6$, with the last value used in the "rolling storage" fundamental matrices occurring at $t=5\Delta t$.) This then completes the brief description of the algorithm, except to indicate that the program for a five state vector, $K_{\max}=20$, and 25 parameter subdivisions executed in a compact 129K bytes of storage (exclusive of the EISPAC and SIMPLOT routines). Overlay techniques could be employed to further reduce storage requirements. Algorithm's execution speed is greatly enhanced (and storage reduced) by the "rolling storage" technique. The technique can be readily implemented on a microcomputer, given fairly compact eigenvalue–eigenvector routines and Interval Arithmetic implementation.

VI. Relation of Present Results to Sensitivity and Robustness

The determination of changes in system performance due to changes in parameters is of great importance in engineering analysis and design. For example, perturbations of parameters in automatic control systems may model the effect of uncertainty in manufacturing tolerances (necessarily nonzero), the aging of components, and environmental causes, to name a few. Solutions to problems caused by these types of phenomena are traditionally solved via sensitivity analysis and/or results which address robustness aspects.

One common approach to analyzing the sensitivity of the solutions of nonlinear systems of ordinary differential equations with a perturbation parameter is to assume "sufficiently small" variations of the parameter and linearize about the nominal solution. If the nonlinear differential equation is written as

$$\begin{aligned} &\dot{x}(t) = f(x(t), \theta),\ t \in [0, t_1] \\ &x(0) = x_0 = \text{initial condition, where} \\ &\theta = \theta_0 + \delta\theta = \text{scalar perturbation parameter with nominal value } \theta_0 \text{ and perturbation } \delta\theta = \theta - \theta_0 \end{aligned} \tag{19}$$

then the commonly used sensitivity analysis approach is to approximate the solution $x(t)$ by the "first approximation"

$$\begin{aligned} &x(t) \approx x_0(t) = \delta\theta \cdot \delta x(t), \text{ where} \\ &x_0(t) = \text{nominal parameter value solution to (19), and} \\ &\delta x(t) = \text{the "first variation."} \end{aligned} \tag{20}$$

The "first variation" is the solution to the differential equation

$$\begin{aligned} &\delta\dot{x}(t) = f_x(x_0(t), \theta_0)\delta x(t) + f_\theta(x_0(t), \theta_0)\delta\theta \text{ and} \\ &\delta x(0) = 0 \end{aligned} \tag{21}$$

where f_x and f_θ are matrices of partial derivatives. In the analogous linear problem

$$\dot{x}(t) = (A + \theta B)x(t) \tag{22}$$

the "first approximation" solution corresponding to (20) becomes

$$x(t) \cong \left(\Phi_0(t) + \delta\theta \int_0^t \Phi_0(t-\tau) B \Phi_0(\tau)\, d\tau\right) x(0), \text{ where}$$

$$\Phi_0(t) = e^{(A+\theta_0 B)t}. \tag{23}$$

The parenthesized term on the right-hand side of (23) will be recognized as the first-order terms in the series expansion

$$e^{[(A+\theta_0 B)+\delta\theta B]t} = e^{(A+\theta_0 B)t} + \delta\theta \int_0^t e^{(A+\theta_0 B)(t-\tau)} \cdot B e^{(A+\theta_0 B)\tau}\, dt + \cdots.$$

Thus even in the linear case, if the variation is not "sufficiently small," the error in the "first approximation" sensitivity method may necessarily dictate a "higher order" analysis, which in itself suffers from increased complexity.

Toward the goal of determining quantitative bounds for *large* perturbation parameter sensitivity effects on the nominal solution to (22), it must be remarked that it is defective rational to expect that all of the "perturbed" solutions will be bounded between the two solutions associated with the parameter values at the endpoints of the perturbation interval (this is demonstrated in the *RLC* circuit example given in Section VII).

The interval algorithm developed in the present paper then may be considered as a numerical technique which computes bounds (albeit, perhaps conservative) for the *large* perturbation parameter sensitivity effects on the nominal solution to (22), including the effects of algorithmic numerical truncation errors.

VII. Specific Examples

We demonstrate the applicability of the results of the present three part paper by applying the algorithm developed herein to three specific examples. The first of these involves the well-known *RLC* series circuit. The second example deals with an instrument servo, while the third example addresses an optimal control problem.

A. The Parameterized RLC Circuit (a second-order example)

The differential equation of this example describes the transient that arises upon closing a series circuit which consists of a resistor, inductor, capacitor and battery. Using electrical charge, $q(t)$, as the dependent variable, the differential equation is

$$L\ddot{q} + R\dot{q} + \frac{1}{C}q = Eu(t)$$

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0. \end{cases} \tag{24}$$

The initial conditions $q(0)$ and $\dot{q}(0)$ are assumed to be zero. Using a consistent set of units, let us assume, for purposes of discussion, that $L = 1.0$, $C = 1.0$, $E = 1.0$, and $R = 0.4 \pm 0.2$. Substitution of these values into (24) yields the equation

$$\ddot{x} + 2\delta\omega_n \dot{x} + \omega_n^2 x = u(t) \tag{25}$$

with the undamped natural frequency $\omega_n = 1.0$ and with the damping ratio

$$\delta = 0.2 \pm 0.1.$$

Equation (25) can be written in a driven "companion system" form and the driving function can be augmented into a larger homogeneous system by defining the variables $x_1 = q$, $\dot{x}_1 = x_2$, $x_3 = Eu(t)$, and $\dot{x}_3 = 0$. If we let θ represent the value of R in the "tolerance interval" [0.2, 0.6], then we obtain

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & -\theta & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},$$

$$\begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{26}$$

In order to demonstrate that all of the solutions of (24) will not necessarily be bounded between the two solutions associated with the parameter values at the endpoints of the perturbation interval, by employing the analytic solution for $q(t)$,

$$q(t) = x_1(t) = 1 - \frac{e^{-\delta t}}{\sqrt{1-\delta^2}} \cos\left(\sqrt{1-\delta^2}\, t - \phi\right)$$

$$\phi = \tan^{-1} \frac{\delta}{\sqrt{1-\delta^2}}$$

double-precision solution values for the nominal and endpoint values of the damping ratio parameter were calculated in time incremental steps of 0.02 for the small time interval [4.5, 4.7]. The resulting enlarged SIMPLOT results are depicted in Fig. 1.

The present algorithm was applied to the system (26) by partitioning the parameter interval into 25 equal segments (and by solving the 25 resulting subproblems), by using the transpose Householder norm (see [1, eq. (31)]), by using $P = 4$ in inequality (11) (the subproblem pseudo-fundamental matrix element "pass–fail–augment" error test inequality parameter), and by using a time increment Δt of 0.5 s. The resulting plots are shown in Fig. 2 where (a) depicts the interval bound for $x_1(t) = q(t)$ with $x_3(t) = u(t)$ superimposed, while (b) displays the interval bound solution for $x_2(t) = \dot{q}(t)$. A comparison of the interval bound solution of Fig. 2(a) with the transient response curves for the unit-step input second-order system with damping ratio values on the interval [0.1, 0.3] (see, e.g., D'Azzo and Houpis [10]) effectively demonstrates that the present algorithm has successfully produced rather tight response envelopes.

B. A Linearized Servomechanism With Load Inertia Parameter (a fourth-order example)

The present example is encountered in a practical text on instrument servomechanism design by Chubb [11]. Fig. 3 presents the original nonlinear system block diagram for the positional motor–generator instrument

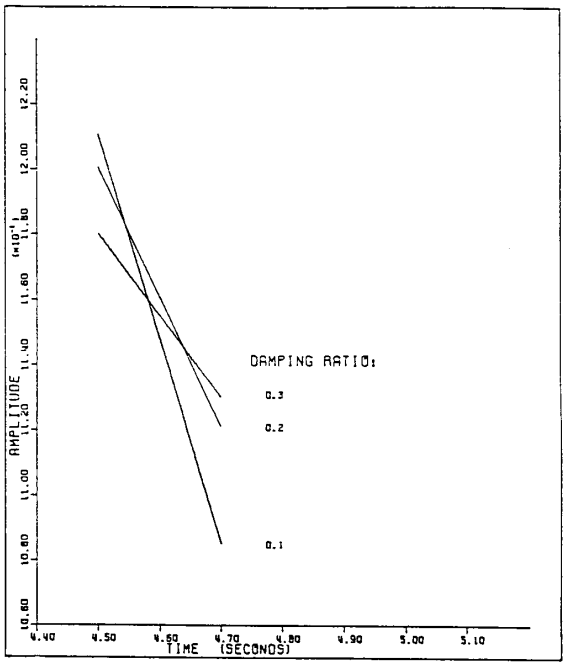


Fig. 1. Solution response for the second-order example (calculated in double-precision) for three damping ratio values.

servomechanism. Since [11] begins analysis with the linearization of the system, the three nonlinear blocks have been drawn omitting the nonlinear "corner" designations. The following list of parameters will assist the perusal of the block diagram, where it will be noted that the motor has been referred through the gear train to the load side:

- K_f follow-up gain effective at the amplifier (V/rad),
- K_g generator gain effective at the amplifier (V/1000 rpm),
- K_a amplifier gain (V/V),
- K_m motor torque gain (oz·in/V),
- ϕ_f followup carrier phase shift effective at the amplifier (deg),
- ϕ_g generator carrier phase shift effective at the amplifier (deg),
- ϕ_a amplifier carrier phase shift (deg),
- N gear ratio (N rev, motor = 1 rev. load),
- B_m motor damping (oz·in/rad/s),
- J_L motor inertia (gm·cm^2),
- K_s gear train stiffness (oz·in/rad),
- θ_L load displacement (rad),
- θ_m motor displacement (rad),
- θ_i positional displacement input (rad),
- T_L lod torque (oz·in),
- T_f load friction torque (oz·in),
- T_s stall torque (oz·in),
- T_d motor damping torque (oz·in),
- T_f gear train torque (oz·in),
- E_f followup voltage (V),
- E_g generator damping voltage (V).

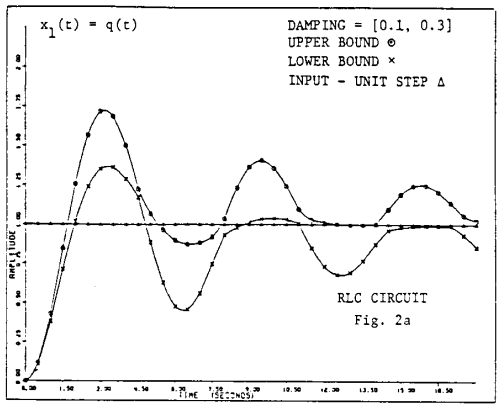


Fig. 2. Interval bound solutions for Example A ($\delta \in [0.1, 0.3]$). (a) Top: $x_1(t) = q(t)$. Bottom: $x_2 = \dot{q}(t)$.

The linearized system is obtained by (i) replacing the coulomb friction by viscous damping f_L (oz·in/rad/s), (ii) setting the motor starting voltage equal to zero, (iii) assuming that the amplifier saturation limits are sufficiently large so that they may be neglected, and (iv) removing the backlash deadzone.

Letting $x_1 = \dot{\theta}_m$, $x_2 = \theta_m$, $x_3 = \dot{\theta}_L$, $x_4 = \theta_L$, $x_5 = \theta_i$, assuming that the followup, generator and amplifier carrier phase shifts are each zero and assuming a unit step positional displacement input, the homogeneous first-order linear differential equation system may be written by inspection of Fig. 3 as

$$\dot{x} = Ax \tag{27}$$

where

$$x^T = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5],$$

$$x(0) = [0 \quad 0 \quad 0 \quad 0 \quad 1]^T,$$

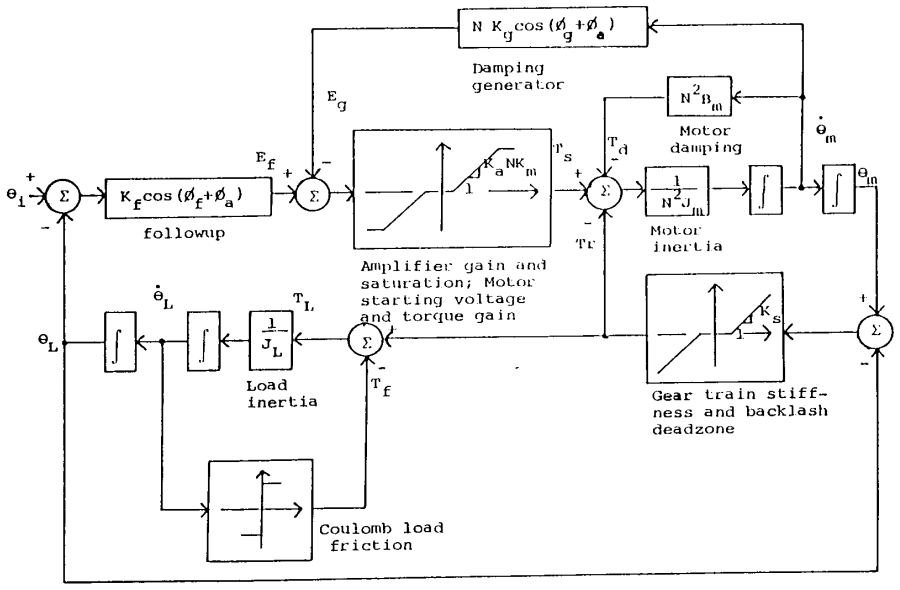


Fig. 3. Nonlinear positional motor–generator instrument servomechanism.

and

$$A = \begin{bmatrix} -\dfrac{B_m + K_g K_a K_m}{J_m} & -\dfrac{K_s}{N^2 J_m} & 0 & \dfrac{K_s - K_f K_a N K_m}{N^2 J_m} & \dfrac{K_f K_a K_m}{N J_m} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{K_s}{J_L} & -\dfrac{f_L}{J_L} & -\dfrac{K_s}{J_L} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{28}$$

The unit system for the calculations will be the oz·in·sec system and the conversion factor for inertia is 1.42×10^{-5} oz·in·sec^2/gm·cm^2. The following coefficient values are given in [11, p. 103] with the exception of the tolerance on the load inertia:

K_f 18.0 V/rad
K_a 240.0 V/V
K_g 0.14 V/1000 rpm
B_m 0.0002 oz·in/rad/s
J_m 1.0 gm·cm^2
K_m 0.0026 oz·in/V
J_L 3000.0 ± 300.00 gm·cm^2
K_s 1500.0 oz·in/rad
N 100.0
f_L 0.0 oz·in/rad/s.

The 10-percent tolerance load inertia arises in the denominator of the expressions for two of the elements in the coefficient matrix A given above. In order to insure the "signed" load inertia parameter dependence in the interval algorithm, the two "interval" coefficients are computed as

$$\left[a_{34}^L, a_{34}^R\right] = \left[-a_{32}^L, -a_{32}^R\right] = \left[-\frac{K_s}{J_L^L}, -\frac{K_s}{J_L^R}\right]$$

where

$$J_L \triangleq \left[J_L^L, J_L^R\right] = [2700.0 \quad 3300.0].$$

The algorithm was applied to the system (27) by partitioning the parameter interval into 25 equal segments, by using the transpose Householder norm (see [1, eq. (31)], by using $P = 3$ in inequality (11), and by using a time increment Δt of 0.0035 s.

The interval algorithm results are shown in Fig. 4 where the discrete-time interval solution bound endpoints are plotted. Superimposed on each plot are continuous solution curves for the nominal and endpoint values of the load inertia parameter which were obtained using the single-precision SYSLIB numerical integration package NODE (which uses a fourth-order predictor corrector technique which initially computes the necessary backpoints by the Runge–Kutta–Gill single-step method (see, e.g. Henrici [12, pp. 122, 154]).

C. A Minimum Plant Sensitivity Optimal Linear Regulator Design (a fifth-order problem)

In the present example we consider an optimal linear state-regulator problem where a feedback compensator is optimally designed to minimize a specified cost functional.

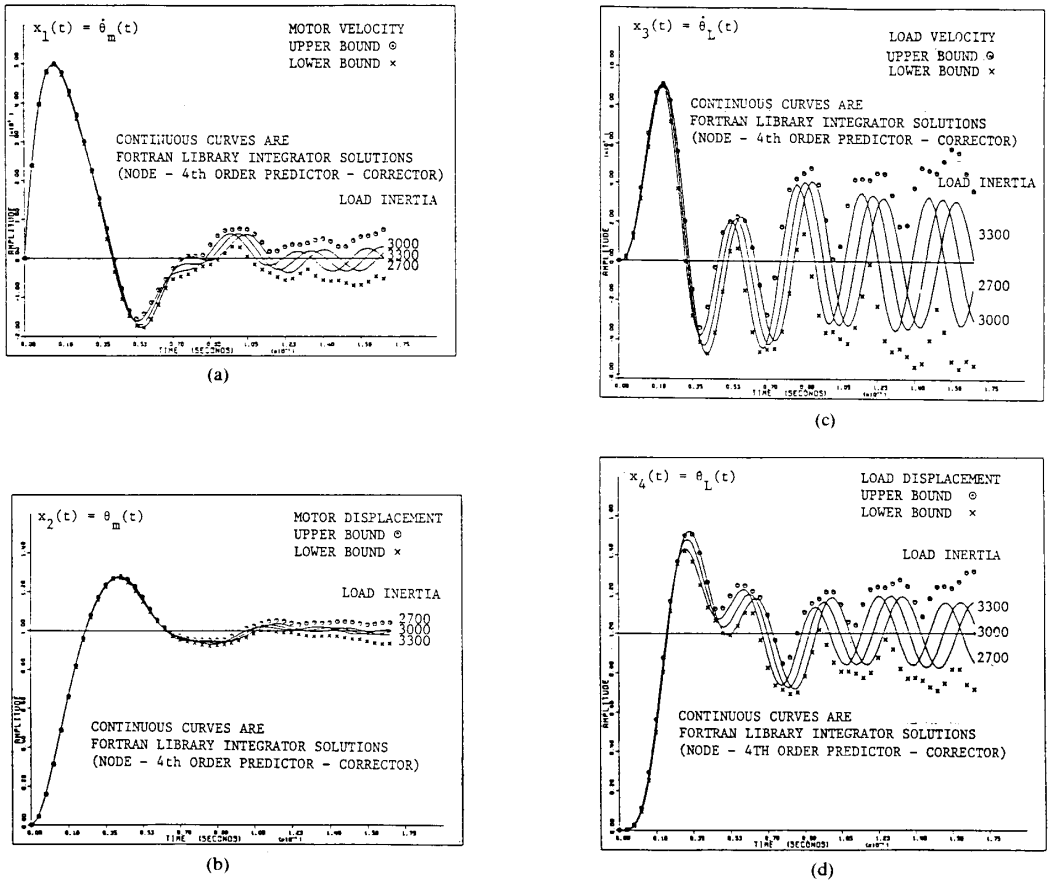


Fig. 4. Interval bound solution endpoints for the fourth-order example ($J_L = [2700.0, 3300.0]$. (a) Top: $x_1(t) = \theta_m(t)$. (b) Bottom: $x_2(t) = \theta_m(t)$. (c) Top: $x_3(t) = \theta_L(t)$. (d) Bottom: $x_4(t) = \theta_L(t)$.

The optimal compensator design additionally includes a first variation cost functional analysis which allows arbitrary parameters in the optimal compensator to be selected so that the design is least sensitive to variation in certain plant parameters. Here we follow along the lines of [13].

Specifically, we consider a linear system given by

$$\dot{x} = Ax + bv$$
$$y = Cx \tag{29}$$

where A is an $n \times n$ matrix with nominal value A_0, C is a constant $m \times n$ matrix ($m < n$), b is a constant n vector, x is the n vector state, v is a scalar control input, and y is the m-vector output. We assume that (A, b) is completely controllable and (A, C) is completely observable.

We consider the linear optimal regulator problem where the control is to minimize the cost functional

$$I = \int_0^\infty \left\{ x^T Q x + \sum_{i=1}^{p} \gamma_i \left[v^{(i)} \right]^2 \right\} dt \tag{30}$$

where Q is a constant positive semidefinite $n \times n$ matrix, where $\gamma_i \geq 0$ $i = 0, \cdots, p-1$, $\gamma_p \triangleq \gamma > 0$, $v^{(i)} \triangleq d^i v / dt^i$, and where p is the smallest integer such that

$$\operatorname{rank}\left[C^T, A_0^T C^T, \cdots, \left(A_0^T \right)^p C^T \right] = n. \tag{31}$$

Let $u_1 = v$, $u_2 = \dot{v}, \cdots, u_{p+1} = \dot{u}_p = U$ and let $u = (u_1, \cdots, u_p)^T$. Let $z = (x^T, u^T)^T$ with $z_0 = z(0)$,

$$\hat{A}_0 = \begin{bmatrix} A_0 & (b:0) \\ 0 & B \end{bmatrix} \tag{32}$$

where B denotes the $p \times p$ matrix

$$B = \begin{bmatrix} 0 & 1 & & & & \\ & 0 & 1 & & & \\ & & 0 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & & 1 \\ & & & & & 0 \end{bmatrix} \tag{33}$$

and let b denote the $(n + p)$ vector

$$\hat{b} = (0, \cdots, 0, 1)^T. \tag{34}$$

Also, let $\hat{Q}$ denote the $(n + p) \times (n + p)$ positive semidefi-

nite matrix

$$\hat{Q} = \begin{bmatrix} Q & 0 \\ 0 & \Gamma \end{bmatrix} \tag{35}$$

where $\Gamma = \text{diag}(\Gamma_i)$, $i = 0,1,\cdots, p-1$. Then the linear regulator problem may be reformulated as

$$\dot{z} = \hat{A}_0 z + \hat{b}U, \qquad z_0 \triangleq z(0) \tag{36}$$

and

$$I = \int_0^\infty \left(z^T \hat{Q} z + \gamma U^2\right) dt. \tag{37}$$

The optimal scalar control U is given by

$$U = -\hat{k}^T z = -\sum_{i=1}^{n} k_i x_i - \sum_{i=1}^{p} k_{n+i} u_i \tag{38}$$

where the $(n+p)$-vector $\hat{k}$ is obtained from

$$\hat{k} = \frac{1}{\gamma} P_\infty \hat{b} \tag{39}$$

and P_∞ is the steady-state solution of the matrix Riccati equation

$$-\dot{P} = P\hat{A}_0 + A_0^T P + \hat{Q} - \frac{1}{\gamma} P\hat{b}\hat{b}^T P$$

$$P(0) = 0. \tag{40}$$

From the original problem, the optimal negative feedback compensating transfer function matrix is given by

$$G_c(s) = [G_1(s), \cdots, G_m(s)] \tag{41}$$

where

$$G_j(s) = -\frac{u(s)}{y_j(s)} = \sum_{i=0}^{p} \beta_i^j s^i \Big/ \left(s^p + \sum_{l=0}^{p-1} \alpha_l s^l\right) \tag{42}$$

$j = 1, \cdots, m$, and thus (38) may also be written as

$$U = -\sum_{i=0}^{p-1} \alpha_i u_{i+1} - \sum_{i=0}^{p} \sum_{j=1}^{m} \beta_i^j y_j^{(i)} \tag{43}$$

where $y_j^{(i)} = c_j^T x^{(i)}$, c_j^T is the jth row of C and $x^{(i)} = d^i x / dt^i$. Then

$$\sum_{i=0}^{p} \sum_{j=1}^{m} \beta_i^j c_j^T A_0^i x = \sum_{i=1}^{n} k_i x_i \tag{44}$$

and

$$\sum_{i=1}^{p} \alpha_{i-1} u_i + \sum_{i=1}^{p} \sum_{j=1}^{m} \beta_i^j c_j^T \sum_{k=1}^{i} A_0^{i-k} b u_k = \sum_{i=1}^{p} k_{i+1} u_i. \tag{45}$$

In matrix form, (45) may be written as

$$\left[C^T, A_0^T C^T, \cdots, \left(A_0^T\right)^p C^T\right] \begin{bmatrix} \beta_0^1 \\ \beta_0^2 \\ \vdots \\ \beta_p^1 \\ \beta_p^m \end{bmatrix} = \begin{bmatrix} k_1 \\ \vdots \\ k_n \end{bmatrix} \tag{46}$$

This represents n equations in $m(p+1)$ unknowns. If $m(p+1) > n$, then $m(p+1) - n = k$ of the β_i^j's are arbitrary and the system is still optimal. If (46) does not have arbitrary β_i^j's, they may be realized by increasing the value of p. Although the compensated system is optimal, it may be very sensitive to plant parameter variations if the arbitrary β_i^j's are not properly chosen. (The p α_i's are defined by (45).

Assume that l plant parameters in A may vary and that k arbitrary β_i^j's are to be determined. Let the l elements of A be denoted by the l-vector a with nominal value a_0. As a result of the first-order variational analysis (with respect to a) of the cost functional, the k arbitrary β_i^j's are assigned values so that the optimal design is least sensitive to the plant parameter variations a.

Let the k-vector $\hat{\beta}$ represent the arbitrary β_i^j's and let the vector β denote the n dependent β_i^j's. Then (46) may be partitioned and rewritten as

$$[SS] \begin{bmatrix} \beta \\ \hat{\beta} \end{bmatrix} = \begin{bmatrix} k_1 \\ \vdots \\ k_n \end{bmatrix} \triangleq k$$

or

$$\beta = (\bar{S})^{-1}(k - S\hat{\beta}) \tag{47}$$

where $\bar{S}$ and S are, respectively, $n \times n$ and $n \times k$ matrices.

Now let A represent the plant with the assumed parameter variation a. Define

$$\hat{A} = \begin{bmatrix} A & b \vdots 0 \\ 0 & B \end{bmatrix}.$$

With $\hat{k}$ (and, therefore, k) determined by (39), on a term by term comparison of (38), (42), (44), and (45) and employing (47), the control U may be written in the form

$$U = -\left[\hat{k}^T + \hat{\gamma}^T\left(\hat{A} - \hat{A}_0\right) + \hat{\beta}^T \hat{K}^T\left(\hat{A} - \hat{A}_0\right)\right] z \tag{48}$$

where $\hat{\gamma}$ is an $(n+p)$ constant vector and $\hat{K}$ is an $(n+p) \times k$ constant matrix. Equation (48) then represents the plant parameter variation dependent optimal control U with the arbitrary constant vector $\hat{\beta}$.

The "sensitivity-optimal" value of $\hat{\beta}$, call it $\hat{\beta}^*$, is selected on the basis of the first-order variational design analysis and may be computed algebraically. Since three-tuple tensor multiplication is sometimes involved, the tensor multiplicative notation will appear following the definitions given below, whenever these occur:

$$\hat{\beta}^* = -(D^T D)^{-1} D^T d, \qquad D^T = \hat{K}^T \left(\frac{\partial \hat{A}}{\partial a}\right)_{a_0} \hat{P}\hat{M}v$$

$$(D^T)_{ij} = (\hat{K}^T)_{ik} \left(\frac{\partial \hat{A}}{\partial a}\right)_{klj} (\hat{P})_{lm} (\tilde{M})_{mn} (v)_n$$

$$\left(\frac{\partial \hat{A}}{\partial a}\right)_{klj} = \frac{\partial (\hat{A})_{kl}}{\partial a_j}, \qquad a_j \in a$$

$$\hat{P}^{-1}\left(\hat{A}_0 - \hat{b}\hat{k}^T\right)\hat{P} = \Lambda = \text{diag}(\lambda_i)$$

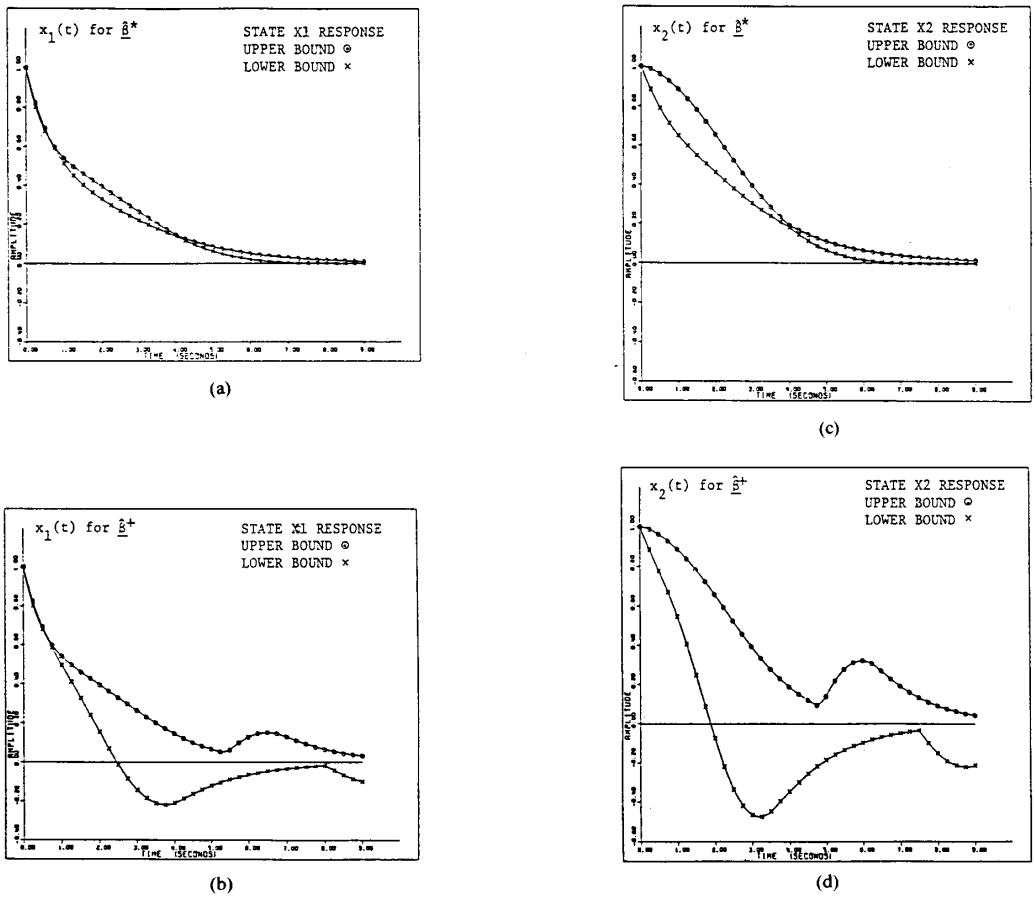


Fig. 5. Interval bound solutions for the fifth-order example ($a_{11} = [2.0, -1.8]$, $a_{22} = [-1.0, -1.5]$). (a) Top: $x_1(t)$ for β^*. (b) Bottom: $x_1(t)$ for β^*. (c) Top: $x_2(t)$ for $\hat{\beta}^*$. (d) Bottom: $x_2(t)$ for $\hat{\beta}^+$. (e) Top: $x_3(t)$ for $\hat{\beta}^*$. (f) Bottom: $x_3(t)$ for $\hat{\beta}^+$. (g) Top: $x_4(t)$ for $\hat{\beta}^*$. (h) Bottom: $x_4(t)$ for $\hat{\beta}^+$. (i) Top $u(t)$ for $\hat{\beta}^*$. (j) Bottom: $u(t)$ for $\hat{\beta}^+$.

(where it is assumed that the λ_i are distinct):

$$\tilde{M} = \operatorname*{diag}_i \left(\sum_{j=1}^{n+p} \frac{v_j w_j}{\lambda_{i+}\lambda_j} \right), \qquad v = \hat{P}^{-1} z_0$$

$$w = E\hat{P}^{-1}\hat{b}, \quad E = \left(\left(\frac{-n_{ij}}{\lambda_{i+}\lambda_j} \right) \right)$$

$$N = \hat{P}^T \hat{K} \hat{P}, \quad \hat{K} = 2(\hat{Q} + \gamma \hat{k} \hat{k}^T)$$

$$d^T = v^T T v, \quad (d^T)_k = (v^T)_i (T)_{ijk} (v)_j$$

$$(T)_{ijk} = \left(\left(\left(\frac{-r_{ijk}}{\lambda_{i+}\lambda_j} \right) \right) \right) \quad \text{and}$$

$$\left(\left((r_{ijk}) \right) \right) = (E)_{il} (\hat{P}^{-1})_{lm} \left(\frac{\partial \hat{A}}{\partial a} \right)_{mnk} (\hat{P})_{nj}. \qquad (49)$$

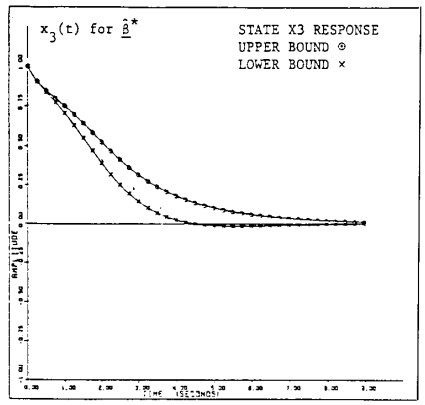
In summary then, the optimal compensator (42), or equivalently, the optimal control (48) solves the linear optimal regulator problem (29), (30), or equivalently, (36), (37), for the nominal plant and if the "sensitivity-optimal" $\hat{\beta}^*$ is computed by (49), the cost functional is least sensitive to plant parameter variations on a first-order variational design basis.

Using the linear interval integration algorithm, we now compare the state variable responses for a design example incorporating the above results for specified plant parameter variations when $\hat{\beta} = \hat{\beta}^*$ and when $\hat{\beta}$ is not selected to satisfy the minimum sensitivity criterion.

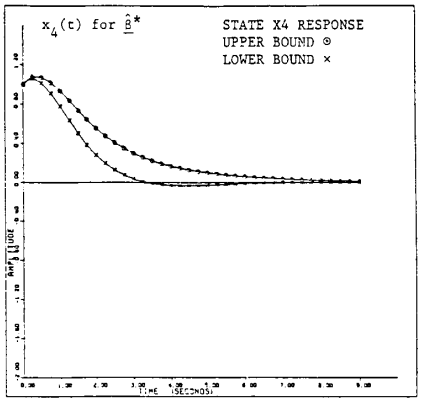
Specifically, the completely controllable and completely observable linear state variable system to be compensated is given by (29), where

$$A = \begin{bmatrix} a_{11} & 1 & 0 & 0 \\ 0 & a_{22} & 1 & 0 \\ 0 & 0 & -1.5 & 1 \\ 0 & 0 & 0 & -0.5 \end{bmatrix},$$

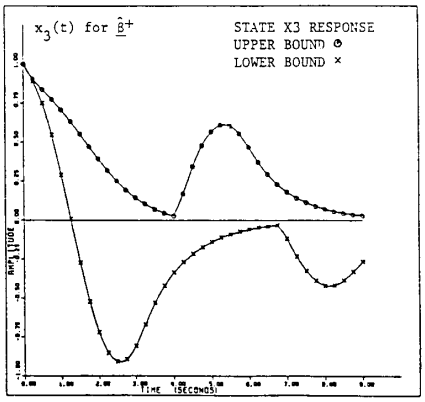
$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \qquad c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



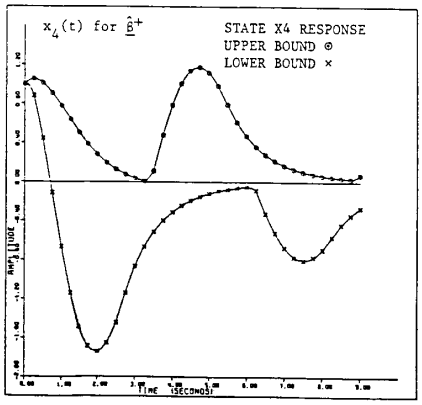
(e)



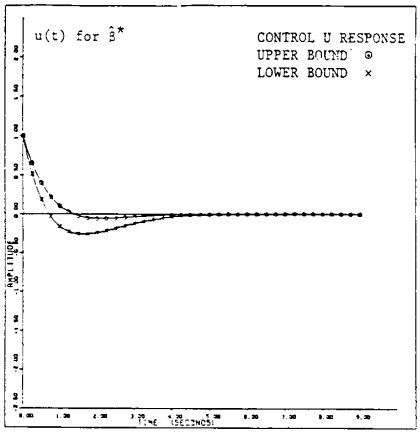
(g)



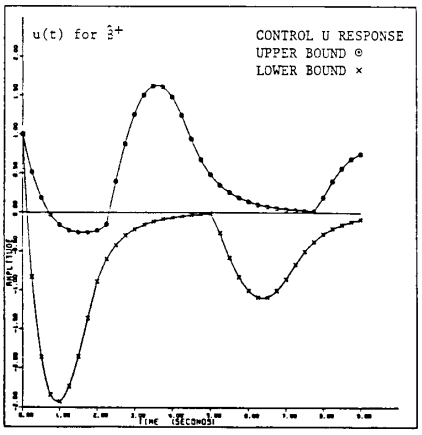
(f)



(h)



(i)



(j)

Fig. 5. (*Continued*)

and the nominal value of the plant matrix is denoted by A_0, where $a_{11} = -2.0$ and $a_{22} = -1.0$. The negative feedback compensated system optimally minimizes the performance index (30) for the nominal plant, where in the present example the minimum value of $p = 1$ satisfies the design criterion (31) and the performance index parameter values are selected as $Q = I$, $\gamma_0 = 1$ and $\gamma_1 = 1$. The reformulated performance index (37), with $\Gamma = \gamma_0$ and $\gamma = \gamma_1$ is the well-known linear state regulator optimization problem.

For the resulting nominal plant optimal compensator design in the present example, examination of (46) indicates that $m(p+1)-n=k=2$ of the compensator β_i^j coefficients are arbitrary. The arbitrary coefficients are selected with $\hat{\beta}=(\beta_1^1,\beta_1^2)^T$ and (46) is reformulated as (47), where the vector k is the result of the optimal linear state-regulator solution.

The perturbed linear system with the nominal plant optimal compensator will be analyzed in the state variable form

$$\dot{z}=\hat{A}z+\hat{b}U, \quad \text{where } \hat{A}=\begin{bmatrix} A & b \\ 0^T & 0 \end{bmatrix} \tag{50}$$

and U is given by (48). This reflects the fixed nominal plant optimal compensator design with the arbitrary design parameter values unspecified and with the perturbed plant. (The actual nominal plant optimal compensator design will not be considered here and would be obtained from (39)–(47).) The formulation for U given in (48) may be realized by substituting the optimal nominal plant design result (47) and the perturbed plant value A in (43), (44) and (45). The matrices in (48) are then found to be

$$\hat{\gamma}^T=[0 \quad 0 \quad k_4 \quad 0 \quad 0]$$

and

$$\hat{k}^T=\begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Substituting these into (48) and the resulting expression for U into (50) produces the homogeneous state variable system (which models the perturbed plant, the fixed nominal plant optimal compensator design and the effects of the arbitrary $\hat{\beta}$)

$$\dot{z}=\begin{bmatrix} a_{11} & 1 & 0 & 0 & 0 \\ 0 & a_{22} & 1 & 0 & 0 \\ 0 & 0 & -1.5 & 1 & 0 \\ 0 & 0 & 0 & -0.5 & 1 \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} z \tag{51}$$

where

$$a_{51}=-\beta_1^1(a_{11}+2)-k_1 \quad a_{52}=-\beta_1^2(a_{22}+1)-k_2$$
$$a_{53}=-k_3 \quad a_{54}=-k_4 \quad \text{and}$$
$$a_{55}=-k_5.$$

In the present example, the initial condition is selected as $z(0)=z_0=[1 \quad 1 \quad 1 \quad 1 \quad 1]^T$.

The value of the arbitrary coefficients vector $\hat{\beta}=(\beta_1^1,\beta_1^2)^T$ which yields the minimum first-order sensitivity of the cost functional (30) with respect to the $l=2$ vector $a=(a_{11},a_{22})^T$ for the optimally compensated nominal plant system is found to be $\hat{\beta}^*=(0.417,1.588)^T$. Although the computation of $\hat{\beta}^*$ is largely algebraic, a procedural listing is given here to provide an understanding of the computations which were required:

1) Equation (40)—determine P_∞.
2) Equation (30)—determine $\hat{k}$.
3) Equations (49)—determine $\hat{P}$, $\hat{P}^{-1}$, and $=\hat{P}^{-1}z_0$, $\hat{\hat{K}}$, N, E, w, the eigenvalues of Λ, $\tilde{M}$, the tensor $(\partial\hat{A}_0/\partial a)_{klj}$, D^T, $(R)_{ijk}$, $(T)_{ijk}$, d^T, and since D^T is not singular, find the minimum sensitivity "arbitrary" coefficient vector value $\hat{\beta}^*=-D^Td=(0.417,1.588)^T$.

Now assume that the two plant parameter variations for the present example are linearly dependent on one scalar parameter. In this sense then, the example values (see [13]) for the algorithm input plant perturbation parameter "intervals" should be $a_{11}=[-2.0,-1.8]$ and $a_{22}=[-1.0,-1.5]$ and the corresponding effects of the perturbation parameter on the homogeneous state variable model (51) should be input as the "intervals":

$$a_{51}=\left[-k_1,-\beta_1^1(-1.8+2.0)-k_1\right]$$
$$a_{52}=\left[-k_2,-\beta_1^2(-1.5+1.0)-k_2\right]. \tag{52}$$

It should be remarked that this forces a "signed" parameter dependence of the algorithm with the effect that for the nominal value of a_{11} and a_{22}, the model will exactly represent the optimally compensated unperturbed nominal system.

In order to appraise the effects on the perturbed system performance of selecting the arbitrary parameters in the nominal plant optimal compensator which minimizes the first-variation of the cost functional, the problem (51) was input to the interval algorithm with the "interval" values a_{51} and a_{52} in (52) computed for $\hat{\beta}=\hat{\beta}^*$ and $\hat{\beta}=\hat{\beta}^+=(10.0,-10.0)^T$. In each case the algorithm operational control parameters were as follows:

(i) The parameter interval was divided into 25 equal segments.
(ii) The Houselolder norm given in (29) of [1] was employed.
(iii) In inequality (11) $P=4$ was used.
(iv) A time increment Δt of 0.5 s was employed.
(v) 36 time increment steps were evaluated.

The resulting algorithm plots of the interval bound solutions for each of the state variables are shown in Fig. 5(a)–5(j) where for each state the $\hat{\beta}^*$ and $\hat{\beta}^+$ results are displayed together for comparison. Examination of the $\hat{\beta}^*$ and $\hat{\beta}^+$ interval bound solutions for each of the state variables vividly defends the feasibility of selecting the minimum sensitivity $\hat{\beta}^*$ for the nominal plant optimal compensator design. Additionally, as a result of the "signed" algorithm dependence on a single perturbation parameter (see (48) and (52)), both the $\hat{\beta}^*$ and $\hat{\beta}^+$ interval bound solutions for the homogeneous state variable system models (51) must contain the unperturbed nominal plant optimally compensated solution trajectory, Utilizing this property, it is possible to investigate the "intersection" of the two interval bound solutions by overlaying the state plots and interestingly observe that, for the following states and time intervals the unperturbed optimal trajectory is precisely indicated since the only intersection is the

lower bound for $\hat{\beta}^*$ and the upper bound for $\hat{\beta}^+$:

$$x_1(t), \quad t = [4.0, 5.25]$$

$$x_2(t), \quad t = [4.0, 4.75]$$

$$x_3(t), \quad t = [0.0, 4.00]$$

$$x_4(t), \quad t = [0.0, 3.25]$$

$$u(t), \quad t = [0.0, 2.00].$$

VIII. Concluding Remarks

In this paper we have used the results of [1] and [3] to establish an algorithm which enables us to generate estimates for solution bounds of initial-value problems described by systems of linear, autonomous, first-order ordinary differential equations with a parameter which belongs to an interval and which enters linearly into the system description. The application of this algorithm to three specific examples shows that the estimates for the solution bounds are quite good.

References

[1] E. P. Oppenheimer and A. N. Michel, "Application of interval analysis techniques to linear systems—Part II: The interval matrix exponential function," *IEEE Circuits Syst.*, vol. 35, pp. 000–000, this issue.

[2] R. K. Miller and A. N. Michel, *Ordinary Differential Equations*. New York: Academic, 1982.

[3] E. P. Oppenheimer and A. M. Michel, "Application of interval analysis techniques to linear systems—Part I: Fundamental results," *IEEE Circuits Syst.*, vol. 35, pp. 000–000, Sept. 1988.

[4] E. P. Oppenheimer, "Application of interval analysis to problems of linear control systems," Ph.D. dissertation, Iowa State Univ. Library, Ames, IA.

[5] Carol A. Clark, "Implementation of the Fortran precompiler, CLUDGE, for the IBM 360/67," Computing Center Library, Washington State Univ., Seattle, 1971.

[6] ___, "Interval arithmetic package for the IBM 360/67," computing Center Library, Washington State Univ., Seattle, 1971.

[7] ___, "Implementation of best possible floating point arithmetic for the IBM 360/67," Computing Center Library, Washington State Univ., Seattle, 1971.

[8] "IBM high-accuracy arithmetic subroutine library (ACRITH)," IBM Systems Reference Library, File S370/4300-82, GC33-6163-02, IBM Product Pub., Kingston, NY, Apr. 1986.

[9] L. B. Rall, "Introduction to the scientific computing language Pascal-SC," presented at the *Trans. Army Conf. on Appl. Math. and Computing*, Washington, DC, May 22–25, 1984.

[10] J. J. D'Azzo and C. H. Houpis, *Linear Control Systems*. New York: McGraw-Hill, 1981.

[11] B. A. Chubb, *Modern Analytical Design of Instrument Servomechanisms*. Reading, MA: Addison-Wesley, 1967.

[12] P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations*. New York: Wiley, 1962.

[13] V. V. Ciric, "Design of minimum sensitivity control systems," Ph.D. dissertation, Rice Univ., Houston, TX.