COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 43 (1984) 195–219
NORTH-HOLLAND

DESIGN SENSITIVITY ANALYSIS AND OPTIMIZATION OF DYNAMIC RESPONSE*

C.C. HSIEH and J.S. ARORA

Design Optimization Laboratory, College of Engineering, The University of Iowa, Iowa City, IA 52242, U.S.A.

Received 29 June 1983
Revised manuscript received 9 November 1983

The paper presents methods of design sensitivity analysis and optimization of dynamic response of mechanical and structural systems. A key feature of the paper is the development of procedures to handle point-wise state variable constraints. Difficulties with a previous treatment where such constraints were transformed to equivalent integral constraints are noted and explained from theoretical as well as engineering standpoints. An alternate treatment of such constraints is proposed, developed and evaluated. In this treatment each point-wise state variable constraint is replaced by several constraints that are imposed at all the local max-points for the original constraint function. The differential equations of motion are formulated in the first-order form so as to handle more general problems. The direct differentiation and adjoint variable methods of design sensitivity analysis to deal with the point-wise constraints are presented. With the adjoint variable methods, there are two ways of calculating design sensitivity coefficients. The first approach uses an impulse load and the second approach uses a step load for the corresponding adjoint equation. Since the adjoint variable methods are better for a large class of problems, an efficient computational algorithm with these methods is presented in detail. Optimum results for several problems are obtained and compared with those available in the literature. The new formulation works extremely well as precise optimum designs are obtained.

1. Introduction

Point-wise state variable constraints appear naturally in dynamic response optimization problems. In a previous treatment [1, 2], all the point-wise constraints were transformed to equivalent functional constraints by integrating them over the time interval. This idea was adopted from optimal control theory [3]. Only then it was possible to calculate the design sensitivity coefficients. Such treatment of constraints leads to theoretical difficulties in numerical optimization methods [4]. Therefore, we need alternative methods to treat point-wise state variable constraints in dynamic response optimization problems.

The purpose of this paper is as follows:

(1) To present an alternative treatment of point-wise state variable constraints in dynamic response optimization problems.

(2) To explain for the first time difficulties in the previous treatment of point-wise state variable constraints as equivalent integral constraints.

*This research is supported by the National Science Foundation Grant No. CEE 82-13851.

(3) To develop and evaluate numerical methods for design sensitivity analysis and optimization with point-wise state variable constraints.

(4) To integrate the design sensitivity analysis procedures into a recently developed new design optimization method based on bounding optimum value of the cost function.

(5) To solve some known dynamic response optimization problems to completely evaluate the numerical methods presented herein and to compare results with the ones available in the literature.

Hsieh and Arora [5] present new methods to compute design sensitivity coefficients for the point-wise constraints for the dynamic response optimization problems. There, differential equations of motion are considered in the second-order form as for a linear structural dynamics problem. That presentation is valid only for linear problems. This paper extends the design sensitivity analysis methods of [5] by incorporating the differential equations of motion that are written in the first-order form. Advantages are that we can treat nonlinear problems, consider the constraints for velocity and acceleration without any effort, and obtain more accurate response due to the availability of good variable stepsize methods for first-order equations, such as DE [6] and GERK [7]. Design sensitivity analysis with the direct differentiation and two adjoint variable methods are presented. Comparison of the three methods is made to claim that the adjoint variable methods are better for a general purpose code. An efficient computational algorithm for optimal design of dynamic response problems is presented by incorporating the design sensitivity analysis procedures into the recently developed design optimization algorithm based on bounding the optimum cost function value. Finally, three examples—a single degree of freedom nonlinear impact absorber, a linear two degree of freedom vibration isolator and a five degree of freedom vehicle suspension system—are optimized using these methods. Results are compared with those available in the literature.

2. Optimal design problem formulation

Let $b \in R^k$ be a vector of design variables and $z \in R^n$ be a vector of generalized velocities and displacements. A general optimization problem for dynamic response of mechanical and structural systems is defined as follows: Find b to

$$\min \psi_0(b)\,, \tag{2.1}$$

subject to the state equation

$$P(b)\dot{z} = f(b, z, t)\,, \quad 0 \leq t \leq T\,, \tag{2.2}$$

$$z(0) = z^0\,, \tag{2.3}$$

and the constraints

$$\psi_i(b, z, t) \leq 0\,, \quad \text{for } 0 \leq t \leq T,\ i = 1, \ldots, m\,, \tag{2.4}$$

$$\phi_j \equiv \int_0^T h_j(b, z, t)\mathrm{d}t \leq 0\,, \quad j = 1, \ldots, p\,, \tag{2.5}$$

$$g_i(b) \leq 0\,, \quad i = 1, \ldots, l\,. \tag{2.6}$$

In the above formulation, the cost function is taken to be dependent only on design variables. This is no real restriction as state dependent cost functions can be handled with the use of an artificial design variable. The min–max problem can also be handled with the use of an artificial variable. This is explained in [1].

The differential equations of motion (2.2) are written in a general first-order form, where $\dot{z}$ denotes the time derivative of the state variable, $P(b)$ is an $n \times n$ matrix whose elements depend upon design variables, $f(b, z, t)$ is an n vector of forcing functions and support reactions, z^0 is a vector of given initial conditions, and T is the time interval of interest. It should be noted that even though $\dot{z}$ appears linearly, these equations can be nonlinear. The constraints of (2.6) are explicit functions of design variables. Such constraints can be routinely handled in any nonlinear programming algorithm. The constraints of (2.5) are the integral-type state variable constraints and they have been treated in [1]. The constraints of (2.4) are the point-wise state variable constraints that present certain difficulties. The present paper concentrates on the treatment of such constraints.

In a previous treatment, the constraints of (2.4) were transformed to an equivalent functional constraint of the form

$$\int_0^T \langle \psi_i(b, z, t) \rangle \mathrm{d}t \leq 0\,, \tag{2.7}$$

where

$$\langle \psi_i(b, z, t) \rangle = \begin{cases} 0 & \text{if } \psi_i < 0\,, \\ \psi_i & \text{if } \psi_i \geq 0\,, \end{cases} \tag{2.8}$$

at any time t. The constraint of (2.7) is equivalent to the constraint of (2.4) only in the sense that if (2.7) is satisfied then (2.4) is also satisfied, and vice versa. However, from the optimization theory point of view, the two constraints are very different. The optimality conditions with the two constraint formulations are different. This problem has been studied in [4] for static response structural optimization problems. The essence of that study is that when constraints of (2.4) are transformed to the equivalent functional constraints of (2.7), some information about the original constraint (2.4) is lost. The lost information concerns the Lagrange multipliers for individual constraints in the static response problem and the Lagrange function for the constraint of (2.4). This is explained as follows:

According to the optimization theory, there exists a Lagrange function $\mu(t)$, $0 \leq t \leq T$, for each of the constraints of (2.4). However, for the equivalent constraint functional of (2.7) there is only one Lagrange multiplier. To see this, we consider the following problem with only one constraint and write necessary conditions for it:

$$\min \psi_0(b)\,,$$

subject to the constraint

$$\psi(b, z, t) \leq 0\,.$$

We assume the optimal point $\bar{b}$ to be a regular point of the feasible region, and $\psi_0(b)$ and $\psi(b, z, t)$ to be differentiable functions. For $\bar{b}$ to be a relative minimum, there exists a multiplier $\mu(t)$, $0 \leq t \leq T$, such that

$$\mathrm{d}L(\bar{b}, z)/\mathrm{d}b = 0\,,$$
$$\mu(t)\psi(b, z, t) = 0\,, \quad 0 \leq t \leq T\,,$$

where

$$L(b, z) = \psi_0(b) + \int_0^T \mu(t)\psi(b, z, t)\mathrm{d}t\,.$$

On the other hand, the necessary conditions for the problem with the equivalent functional constraint of (2.7) are: there exists a multiplier $\bar{\mu}$ (scalar) ≥ 0 such that

$$\mathrm{d}\bar{L}(\bar{b}, z)/\mathrm{d}b = 0\,,$$
$$\bar{\mu}\int_0^T \langle\psi(b, z, t)\rangle\mathrm{d}t = 0\,,$$

where

$$\bar{L}(b, z) = \psi_0(b) + \bar{\mu}\int_0^T \langle\psi(b, z, t)\rangle\mathrm{d}t\,.$$

Clearly, the two sets of necessary conditions are not the same unless $\bar{\mu}$ is also taken as a function of t. Thus, equivalent functional treatment of point-wise state variable constraints makes an implicit assumption that the Lagrange function for the constraint is constant throughout the interval $0 \leq t \leq T$. The Lagrange multiplier also represents the force of the constraint. Therefore, in engineering terms, equivalent functional treatment of (2.7) implies a constant force of constraint for the original constraint of (2.4). This is not true because the force of the constraint corresponding to the constraint of (2.4) must be a function of time.

In addition to the above mentioned difficulty, there is a deeper mathematical flaw in the equivalent functional formulation of (2.7). Fig. 1 represents the constraint of (2.7). The constraint is violated over three time sub-domains and the amount of the violation is the sum of the areas A, B and C. In such a situation, the constraint functional of (2.7) is not differentiable in the ordinary sense. Therefore, in general, the equivalent functional formulation of the state variable point-wise constraints can lead to numerical difficulties. In fact such

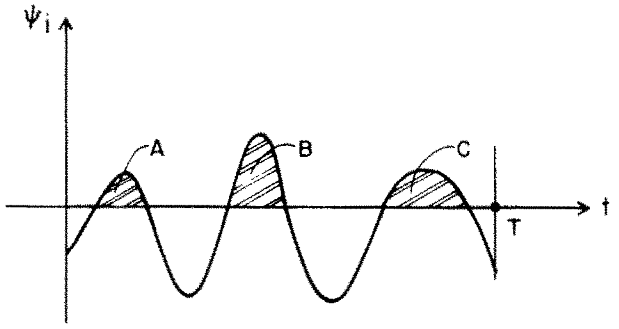


Fig. 1. Point-wise state variable constraint $\psi_i \leq 0$, $0 \leq t \leq T$ violated over three sub-domains.

difficulties were encountered in solving many of the dynamic response optimization problems presented in [2] and in [1, Chapter 5]. For some problems the optimality conditions could not be satisfied. Some of the problems from these references are solved later in the paper with the new treatment of constraints presented in the following paragraphs. More discussion about these points will be presented there.

It is proposed to treat the point-wise state variable constraint in the sense of 'worst-case design formulation' presented in [1, Chapter 3]. There the worst-case design problem for static response optimization is considered. Therefore, in this paper that formulation is extended to dynamic response problem. In that setting the time parameter t is treated an 'environmental parameter' [1]. As shown there, the continuum constraint of (2.4) can be replaced by the constraint

$$\max_t \psi_i(b, z, t) \leq 0 . \tag{2.9}$$

The constraint of (2.9) is equivalent to the constraint of (2.4) in the sense that if one is satisfied the other is also satisfied. In a sense, (2.9) represents a constraint at the 'worst' response time point and hence the name 'worst-case design formulation'. In numerical computation, it is necessary to impose the constraint at all points of local maximum for the function ψ_i. Otherwise, oscillations in active constraints can occur causing convergence difficulties in numerical optimization. Thus, the ith constraint of (2.4) is replaced by the following constraints:

$$\psi_i(b, z, t)|_{t=t_j} \leq 0 , \quad j = 1, \ldots, m(i) , \tag{2.10}$$

where t_j is a point of local maximum for the function ψ_i, and $m(i)$ is the total number of max-points for the ith constraint. The numerical procedure then is to locate all the points t_j at a given design for the constraint and impose the constraint there. The max-points can be located by simply searching through a table of values as explained later in the paper. It is obvious that the max-points t_j depend on the system design variable vector b at any iteration. That is, for the changed design in an iterative optimization algorithm, the points t_j will change. However, it can be shown that such changes have no effect on the first derivatives of the function $\psi_i(b, z, t)$ at the point t_j with respect to the design variables. In other words, once points t_j have been located for a design point, the state variable constraints can be imposed only at those points treating t_j's as fixed during an iteration of the optimization algorithm. To see this, the constraint function $\psi_i(b, z, t)|_{t=t_j}$ is written as

$$\psi_i(b, z, t)|_{t_j} = \psi_i(b, z(t_j, b), t_j(b)) . \tag{2.11}$$

Then the total derivative of ψ_i with respect to the design variable is

$$\frac{\mathrm{d}\psi_i}{\mathrm{d}b} = \frac{\partial\psi_i}{\partial b} + \frac{\partial\psi_i}{\partial z}\bigg|_{t=t_j} \left[\frac{\mathrm{d}z}{\mathrm{d}b}\bigg|_{t=t_j} + \dot{z}\bigg|_{t=t_j} \frac{\mathrm{d}t_j}{\mathrm{d}b}\right] + \frac{\partial\psi_i}{\partial t}\bigg|_{t=t_j} \frac{\mathrm{d}t_j}{\mathrm{d}b} . \tag{2.12}$$

Rewriting (2.12),

$$\frac{\mathrm{d}\psi_i}{\mathrm{d}b} = \frac{\partial\psi_i}{\partial b} + \left[\frac{\partial\psi_i}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}b}\right]_{t=t_j} + \left[\frac{\partial\psi_i}{\partial z}\dot{z} + \frac{\partial\psi_i}{\partial t}\right]_{t=t_j} \frac{\mathrm{d}t_j}{\mathrm{d}b} . \tag{2.13}$$

Since t_j is a point of local maximum for $\psi_i(b, z(t), t)$, the total derivative of ψ_i with respect to $t(\mathrm{d}\psi_i/\mathrm{d}t)$ is zero at t_j. That is,

$$\left[\frac{\partial\psi_i}{\partial z}\dot{z} + \frac{\partial\psi_i}{\partial t}\right]_{t=t_j} = 0\,. \tag{2.14}$$

Therefore, the last term of (2.13) drops out from the design derivative of ψ_i at t_j. Clearly, then, change in t_j as design is changed has no effect (to first order) on $\mathrm{d}\psi_i/\mathrm{d}b$. This shows that once t_j is calculated in the current iteration, it can be treated as a constraint; i.e., independent of design variables.

Thus, to treat constraints of (2.10) in an optimization process, we need to develop procedures for calculating derivatives of a state variable inequality constraint when it is imposed at a particular time point. Such procedures are developed in the following sections.

3. Design sensitivity analysis

In this section, we investigate methods for calculating design gradients of a point-wise state variable constraint imposed at a particular time t_j. Let a constraint of (2.10) be represented as

$$\psi = Q(b, z, t)|_{t_j}\,. \tag{3.1}$$

Now introducing the Dirac delta function, as in [5, 8] (3.1) is written as

$$\psi = \int_0^T Q(z, b, t)\delta(t - t_j)\mathrm{d}t\,. \tag{3.2}$$

A first variation of (3.2) gives

$$\delta\psi = \int_0^T \left[\frac{\partial Q(z, b, t)}{\partial b}\delta b + \frac{\partial Q(z, b, t)}{\partial z}\delta z\right]\delta(t - t_j)\mathrm{d}t\,. \tag{3.3}$$

That is,

$$\delta\psi = \left.\frac{\partial Q(z, b, t)}{\partial b}\right|_{t=t_j}\delta b + \left.\frac{\partial Q(z, b, t)}{\partial z}\delta z\right|_{t=t_j} \equiv \frac{\mathrm{d}\psi}{\mathrm{d}b}\delta b\,. \tag{3.4}$$

The problem is to find $\mathrm{d}\psi/\mathrm{d}b$ with the condition that an explicit form for $z(b)$ is not known. There are two approaches of calculating derivatives: direct differentiation and adjoint variable methods [5, 9, 10].

3.1. Direct differentiation method

Taking the first variation of (2.2),

$$P\,\delta\dot{z} - \frac{\partial f}{\partial z}\delta z = \frac{\partial f}{\partial b}\delta b - \frac{\partial(P\bar{z})}{\partial b}\delta b\,, \tag{3.5}$$

where ~ indicates that this variable is kept fixed during the partial differentiation. Now, let us define a matrix q as

$$\delta z = \frac{\mathrm{d}z}{\mathrm{d}b}\,\delta b \equiv q\,\delta b\,. \tag{3.6}$$

Differentiating the above equation with respect to t,

$$\delta \dot{z} = \frac{\mathrm{d}\dot{z}}{\mathrm{d}b}\,\delta b \equiv \dot{q}\,\delta b\,. \tag{3.7}$$

Substituting (3.6) and (3.7) into (3.5),

$$P\dot{q} - \frac{\partial f}{\partial z} q = \frac{\partial f}{\partial b} - \frac{\partial (P\tilde{\dot{z}})}{\partial b}\,. \tag{3.8}$$

The right-hand side of (3.8) can be calculated explicitly since dependence of f and P on the design variable b is known. Also, since initial conditions of (2.2) are design independent, a variation with respect to the design variables gives the initial condition for (3.8) as

$$q(0) = 0\,. \tag{3.9}$$

Obviously, (3.8) and (3.9) are in the same form as (2.2) and (2.3). Therefore, any method used to solve for state variable z can be employed to solve for q. The design sensitivity for the constraint function ψ is then obtained by inserting (3.6) into (3.4) as

$$\frac{\mathrm{d}\psi}{\mathrm{d}b} = \left.\frac{\partial Q(z,b,t)}{\partial b}\right|_{t=t_j} + \left.\frac{\partial Q(z,b,t)}{\partial z} q\right|_{t=t_j}\,. \tag{3.10}$$

3.2. Adjoint variable methods

Premultiplying (3.5) by the transpose of an adjoint variable vector $\lambda(t)$ and then integrating over the time interval 0 to T,

$$\int_0^T \lambda^{\mathrm{t}}\left(P\,\delta\dot{z} - \frac{\partial f}{\partial z}\,\delta z\right)\mathrm{d}t = \int_0^T \lambda^{\mathrm{t}}\left[\frac{\partial f}{\partial b}\,\delta b - \frac{\partial (P\tilde{\dot{z}})}{\partial b}\,\delta b\right]\mathrm{d}t\,. \tag{3.11}$$

Integrating the first term in (3.11) by parts, one has the identity

$$\lambda^{\mathrm{t}} P\,\delta z|_0^T - \int_0^T \left[\dot{\lambda}^{\mathrm{t}} P + \lambda^{\mathrm{t}}\frac{\partial f}{\partial z}\right]\delta z\,\mathrm{d}t = \int_0^T \lambda^{\mathrm{t}}\left[\frac{\partial f}{\partial b} - \frac{\partial (P\tilde{\dot{z}})}{\partial b}\right]\delta b\,\mathrm{d}t\,. \tag{3.12}$$

Now, $\delta z(0)$ vanishes due to the initial conditions of (2.2) and we require the boundary at T to be zero in (3.12). That is,

$$\lambda^{\mathrm{t}} P\,\delta z|^T = 0\,. \tag{3.13}$$

Since $\delta z(T)$ is arbitrary, (3.13) is satisfied by setting

$$\lambda(T) = 0\,. \tag{3.14}$$

This eliminates the boundary term in (3.12).

Let the adjoint variable $\lambda(t)$ be defined as a solution of

$$P^{\mathrm{t}}\dot{\lambda}(t) + \frac{\partial f^{\mathrm{t}}}{\partial z}\lambda(t) = \left[\frac{\partial Q(z, b, t)}{\partial z}\right]^{\mathrm{t}} \delta(t - t_j)\,, \tag{3.15}$$

subject to the terminal condition of (3.14). Substituting (3.15) into (3.12) and then substituting the result into (3.3), we have the sensitivity coefficients for the constraint function ψ as

$$\frac{\mathrm{d}\psi}{\mathrm{d}b} = \left.\frac{\partial Q(z, b, t)}{\partial b}\right|_{t=t_j} - \int_0^{t_j} \lambda^{\mathrm{t}}\left[\frac{\partial f}{\partial b} - \frac{\partial (P\dot{z})}{\partial b}\right]\mathrm{d}t\,. \tag{3.16}$$

Note from (3.14) and (3.15) that $\lambda = 0$ for $t_j < t \leq T$. Therefore, the upper limit on integration in (3.16) has been changed to t_j.

We can transform the terminal value problem of (3.14) and (3.15) to an initial value problem. We will set $\lambda = 0$ for $t_j < t \leq T$. Defining $t = t_j - \tau$, $y(\tau) = \lambda(t)$, $\partial \bar{f}(z, b, \tau)/\partial z = \partial f(z, b, t)/\partial z$ and $\bar{Q}(z, b, \tau) = Q(z, b, t)$, (3.14) and (3.15) become

$$-P^{\mathrm{t}}\dot{y}(\tau) + \frac{\partial \bar{f}^{\mathrm{t}}}{\partial z}y(\tau) = \left[\frac{\partial \bar{Q}(z, b, \tau)}{\partial z}\right]^{\mathrm{t}}\delta(\tau)\,, \tag{3.17}$$

$$y(0) = 0\,. \tag{3.18}$$

Note that (3.14) and (3.15) or (3.17) and (3.18) can be put in the same form as (2.2) and (2.3). However, (3.15) and (3.17) are linear in the variables $\lambda(t)$ and $y(\tau)$. In addition, they have constant coefficients when f is linear in z. Advantage of this property will be taken later in numerical calculations.

As in [5], there is another form of the adjoint equation and the design sensitivity expression for the constraint function $Q(b, z, t)|_{t=t_j}$ of (3.1). Let us first integrate the constraint function Q over the time domain 0 to t_j as

$$\bar{\psi} = \int_0^{t_j} Q(z, b, t)\mathrm{d}t\,. \tag{3.19}$$

The procedure is to derive design sensitivity expressions for $\bar{\psi}$ and then use Leibnitz rule to recover corresponding sensitivity expressions for ψ. It should be noted that constraint (3.19) for $\bar{\psi}$ is similar to the constraint function ψ of (3.2) except there is no $\delta(t - t_j)$. Therefore, the sensitivity expression for $\mathrm{d}\bar{\psi}/\mathrm{d}b$ is obtained from (3.16), as

$$\frac{\mathrm{d}\bar{\psi}}{\mathrm{d}b} = \int_0^{t_j}\left(\frac{\partial Q(z, b, t)}{\partial b} - \bar{\lambda}^{\mathrm{t}}(t)\left[\frac{\partial f}{\partial b} - \frac{\partial (P\dot{z})}{\partial b}\right]\right)\mathrm{d}t\,, \tag{3.20}$$

where the new adjoint vector $\bar{\lambda}$ is defined to be a solution of an adjoint equation similar to (3.15), as

$$P^{\text{t}}\dot{\bar{\lambda}}(t)+\frac{\partial f^{\text{t}}}{\partial z}\bar{\lambda}(t)=\left[\frac{\partial Q(z,b,t)}{\partial z}\right]^{\text{t}}, \tag{3.21}$$

with the terminal condition

$$\bar{\lambda}(t_j)=0\,. \tag{3.22}$$

Again, these equations can be transformed to an initial value problem,

$$-P^{\text{t}}\dot{\bar{y}}(\tau)+\frac{\partial \bar{f}^{\text{t}}}{\partial z}\bar{y}(\tau)=\left[\frac{\partial \bar{Q}(z,b,\tau)}{\partial z}\right]^{\text{t}} \tag{3.23}$$

and

$$\bar{y}(0)=0\,, \tag{3.24}$$

where $\bar{y}(\tau)=\bar{\lambda}(t)$ and τ, $\partial\bar{f}^{\text{t}}/\partial z$ and $\partial\bar{Q}^{\text{t}}/\partial z$ are the same as before. Note again that either (3.21) and (3.22) or (3.23) and (3.24) can be put in the same form as (2.2) and (2.3). Equations (3.21) and (3.23) have the same property as (3.15) and (3.17). These equations are linear in $\lambda(t)$ and $y(\tau)$. Also, they have constant coefficients when f is linear in z.

Next, we need to recover the sensitivity expression for the function ψ from (3.20). The relationship is given as

$$\frac{\text{d}\psi}{\text{d}b}(t_j)=\frac{\text{d}}{\text{d}t_j}\frac{\text{d}\bar{\psi}}{\text{d}b}\,. \tag{3.25}$$

Using $\bar{\lambda}(t)=\bar{y}(t_j-t)$ and Leibnitz rule of differentiation, (3.25) yields

$$\begin{aligned}\frac{\text{d}\psi}{\text{d}b}&=\left.\frac{\partial Q(z,b,t)}{\partial b}\right|_{t=t_j}-\int_0^{t_j}\frac{\partial\bar{y}^{\text{t}}(t_j-t)}{\partial t_j}\left[\frac{\partial f}{\partial b}-\frac{\partial(P\dot{z})}{\partial b}\right]\text{d}t+\bar{\lambda}^{\text{t}}(t_j)\left.\left[\frac{\partial f}{\partial b}-\frac{\partial(P\dot{z})}{\partial b}\right]\right|_{t=t_j}\\&=\left.\frac{\partial Q(z,b,t)}{\partial b}\right|_{t=t_j}+\int_0^{t_j}\dot{\bar{\lambda}}^{\text{t}}(t)\left[\frac{\partial f}{\partial b}-\frac{\partial(P\dot{z})}{\partial b}\right]\text{d}t\,,\end{aligned} \tag{3.26}$$

since $\bar{\lambda}(t_j)=0$ and $\partial\bar{y}(\tau)/\partial\tau=-\partial\bar{\lambda}(t)/\partial t$.

3.3. Comparison of methods

In determining which of the three methods presented above is to be employed, one needs to compare the number of equations to be solved by each method. First of all, it should be observed that even if the system equations of motion are nonlinear in the state variable z, the adjoint equations (3.8), (3.15) and (3.21) are linear in the terms of q or λ. In addition, they have constant coefficients when f is linear in z. Advantage of this fact can be taken in calculations. If the number of time points at which the ith point-wise constraint function is active is $n(i)$, then there are $n(i)$ of adjoint equations with impulse loading and step function loading in (3.15) and (3.21), respectively. However, one just needs to solve the adjoint equation with impulse loading only once, since it is a linear system and the loading vectors are proportionally applied at different time points t_j. There is a large class of systems for which $f(b, z, t)$ is linear in z. Thus, one may solve the adjoint equation using any good first-order method considering the impulse loading applied at the time t_j. Then the adjoint response for other impulse loading applied at different time points is recovered by appropriately scaling the calculated response and shifting the starting point for the solution. Some other characteristics for the adjoint variable method with impulse loading may be observed as well. If one loading vector is a linear combination of others, then the adjoint equation associated with the loading need not be solved again. One obtains the required response simply as a linear combination of the known responses. In any case, the maximum number of adjoint equations for an optimization problem that are needed to be solved is equal to the number of state variables.

The adjoint equation (3.21) has the same properties as the adjoint equation (3.15) for a large class of problems. When f is linear in z and the loading in (3.21) is proportionately applied, the principle of linear superposition can be used to advantage in reducing the number of adjoint vectors to be solved.

Let k be the number of design variables, NL the number of loading cases, NC the number of active constraint functions and $n(i)$ the number of time points for the ith point-wise constraint function at which the constraint is active. The number of equations needed to be solved for all three methods are: $K \times \mathrm{NL}$ for the direct differentiation method, $\sum_{i=1}^{\mathrm{NC}} n(i)$ for the adjoint variable method with step function loading and for the adjoint variable method with impulse loading when $f(b, z, t)$ is not linear in z, and at most NC for the adjoint variable method with impulse loading when f is linear in z. When f is linear in z, the adjoint variable method with impulse loading is superior to the same method with step function loading when superposition of solution cannot be used with the latter method. Note, however, that superposition of solutions for the adjoint variable method with step function loading is actually applicable to a large class of problems. The adjoint variable methods are also better than the direct differentiation method because NC is smaller than $k \times \mathrm{NL}$ in many optimization problems. Thus, one should use the adjoint variable methods for efficient calculation of the design sensitivity coefficients.

4. Optimization algorithm for dynamic response problems

Based on the above derivations and discussions, an efficient algorithm for the optimal design of dynamic response of systems is stated as follows:

Step 1. Read initial design data.

Step 2. Solve state equations for z and $\dot{z}$ for the entire time interval of interest.

Step 3. Loop on each point-wise state variable constraint.

Step 4. Compute the constraint function ψ_i and its time derivative $\dot{\psi}_i$.

Step 5. Locate all the points of local maxima for the function ψ_i. Since time history for ψ_i is available, these points are located by simply searching through a table of values.

Step 6. Check if constraints at the local max points are active for the ith point-wise state variable constraint. If not, go to Step 9.

Step 7. If $f(b, z, t)$ is linear in z, solve the adjoint equation (3.15) for λ and $\dot{\lambda}$ up to the maximum point t_j in the time domain. If $f(b, z, t)$ is not linear in z, solve the adjoint equation (3.15) for each of the $n(i)$ loading cases and then go to Step 8(b).

Step 8. Compute the sensitivity coefficients for each constraint corresponding to the ith point-wise constraint:

(a) Compute the true value of the adjoint variable λ and $\dot{\lambda}$ using the loading factor and the time shift from t_j.

(b) If time grids for the state variable z and the adjoint variable λ do not match, interpolate for the values of λ such that it can be calculated at the points where the state variable is known. Hermite cubic interpolation [11] formula is used here.

(c) Carry out the integration of (3.16) to calculate the sensitivity coefficients for each of the constraints.

Step 9. End of loop for point-wise state variable constraint.

Step 10. Compute the design sensitivities for the integral-type and time independent constraints, and the cost function.

Step 11. Compute the design change δb using some optimization method [12]. Put $b^{\nu+1} = b^{\nu} + \delta b$, where ν is the iteration number.

Step 12. If the convergence criteria are satisfied or limit on iterations exceeded, then stop; otherwise, go to Step 2.

The calculation of $\dot{z}$ is for evaluating $\dot{\psi}$ that is needed for locating local max points. The time derivative of the adjoint variable λ is needed if Hermite cubic interpolation is used in Step 8(b). Note that the above algorithm incorporates the adjoint variable method with impulse loading. A similar algorithm can be stated when the step loading method is applicable.

There are certain numerical considerations in implementing the algorithm that are discussed here. Numerical experience with the method is discussed later in the paper. In general, the point-wise state variable constraint is written as $\psi_i(b, z, t) = |\theta_i(b, z, t)| - \theta_{ia} \leq 0$. It is replaced by constraints $\psi_i(b, z, t_j) \leq 0$, $j = 1, \ldots, m(i)$, where t_j is a point of local maximum for the function ψ_i. However, it may not be necessary to consider all of the local maxima. The treatment is to select t_j for the constraint function ψ_i which belongs to the M^i set, where the M^i set is defined as

$$M^i = \{t_j \mid t_j \text{ is a point of local maximum of } |\theta_i| \text{ and } |\theta_i(b, z, t)| \geq \alpha\theta_{i\,\max}, \text{ where } \theta_{i\,\max} \text{ is the global maximum of } |\theta_i| \text{ and } \alpha \text{ is a scalar, } 0 \leq \alpha \leq 1\}\,.$$

The number of elements of the M^i set is dependent on the parameter α. If $\alpha = 0$, then all the local maximum points are considered. If $\alpha = 1$, then only the global maximum point is considered. For an intermediate value of α, only some severely violated local maximum points are included. To explain this graphically, consider a function $|\theta_i|$ that is plotted in Fig. 2. There are five local maximum points, and point B has the global maximum value as $\theta_{i\,\max}$. Points A, B and D will be in the M^i set if the value α is set as shown in Fig. 2. It should be noted that constraints are imposed only at the points in the set M^i. Only violated constraints are retained in the active set. Points C and E are neglected even if the constraint is violated at these points.

The local maximum points can appear at the ends of the interval and points having zero slope for the function $|\theta_i|$. For the former case, it does not take any effort to locate maximum points. However, in the latter case, some root-finding algorithm must be used to locate the

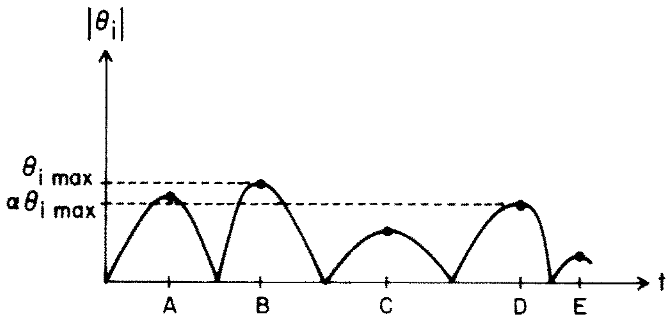


Fig. 2. Graphical description of the M' set.

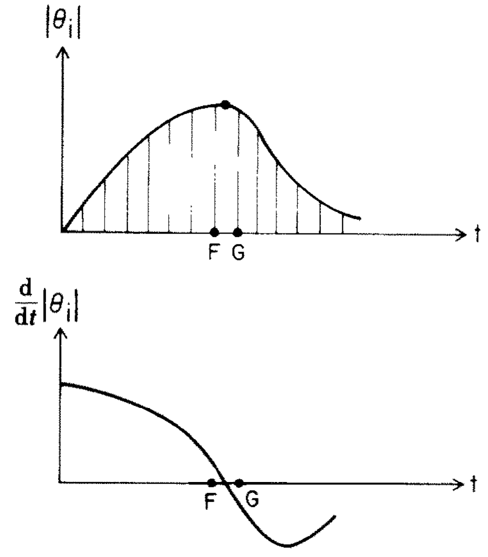


Fig. 3. Graphical representation of a local maximum point.

zero slope points. Since the constraint function values are determined only at some grid points along the time domain, the zero slope points may exist, in general, between two grid points. One may multiply the slope value at two adjacent grid points to check if there is a local maximum between these two grid points. The local maximum exists when the multiplication yields a negative value. Thus, in Fig. 3, there may be a local maximum between points F and G. Once the sub-domain of local maximum has been found, we can start to locate the point of zero slope. Any root-finding algorithm can be employed, such as the bisection and secant methods. For obtaining the slope values at intermediate points during the root-finding, state equations are resolved by starting from the first grid point of that region. For example, we start solving state equations from point F for determining a root between points F and G in Fig. 3. This process has the advantage of less error for calculating the information at intermediate points during the root-finding stage.

In Step 11 of the algorithm, any optimization algorithm can be used to compute the new design variable vector. However, some algorithms are better than others for engineering design problems. Many engineering design optimization problems, such as dynamic response problems, have functions that are implicitly dependent on design variables. For example, in the dynamic response problems, an explicit functional form for $z(t)$ in terms of design variables is not known. However, once the design variable vector b is specified, $z(t)$ can be obtained by integrating the state equation (2.2). That is, whenever functions of the optimization problem are needed, we have to integrate the equations of motion. This is an important point relative to an optimization algorithm. Many optimization algorithms require one-dimensional search to calculate δb in Step 11 of the above algorithm. One-dimensional search may require function evaluations several times. Thus algorithms requiring one-dimensional search are likely to be less efficient for engineering design problems.

With the above point in mind, a new optimization algorithm has been recently developed that does not require one-dimensional search [12]. The method is based on recursive quadratic programming and optimum solution bounding techniques. In the present paper, the new algorithm is used for the first time on dynamic response optimization problems. Performance of the method relative to another method that uses one-dimensional search is evaluated in the next section.

5. Sample problems

To evaluate effectiveness of the methods presented in the foregoing, three design examples from the literature [1] are solved and the results compared with the available results. These examples are solved for the optimum solutions by the optimization program IDESIGN that is based on recursive quadratic programming and cost function bounding concepts [12, 13]. IDESIGN is a general purpose design optimization program. Its structure is shown in Fig. 4. The user needs to supply the subroutine USERC which calculates the cost function, constraints, cost function gradient and gradients of active constraints. During each iteration of the optimization algorithm, the functions are evaluated only once. The first example is also solved by the optimization program LINRM [14] to compare solutions with two programs. LINRM uses the linearization method of Pshenichny [15] and has a structure similar to IDESIGN. However, the algorithm of LINRM requires one-dimensional search and, therefore, function evaluation several times during an iteration. It is noted that for two examples $f(b, z, t)$ is linear

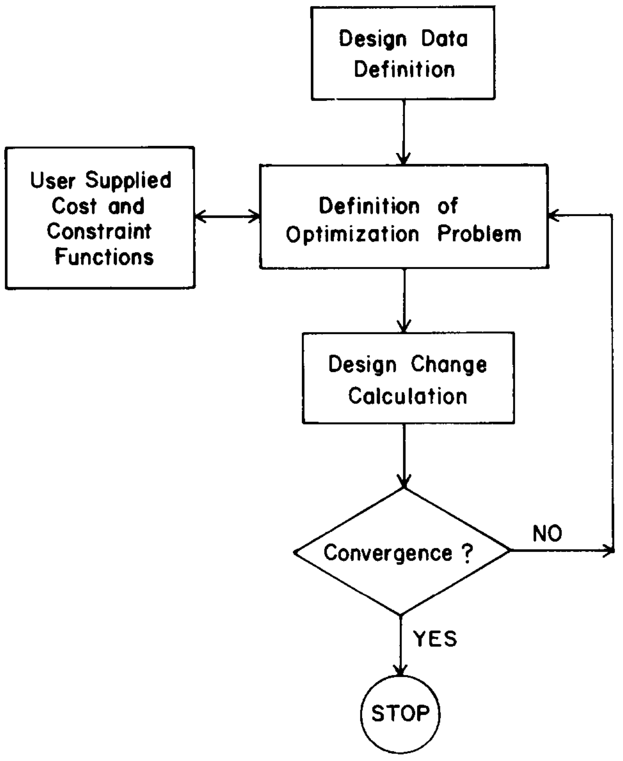

Design Data Definition
User Supplied Cost and Constraint Functions
Definition of Optimization Problem
Design Change Calculation
Convergence ?
NO
YES
STOP

Fig. 4. Structure of general purpose optimization program IDESIGN.

in z. Therefore the linear adjoint equation for these examples has constant coefficients. Advantage of this property is realized in design sensitivity analysis, as explained before.

EXAMPLE 5.1 (A single degree of freedom nonlinear impact absorber). The nonlinear single degree of freedom system of Fig. 5 has a fixed mass M and two design variables b_1 and b_2 that represent spring and damper coefficients, respectively. The exponents ν and ω are held fixed for each design. The problem is solved in [1].

The system impacts a fixed barrier at time $t = 0$ with initial velocity v. Defining $z_1 = x$ and $z_2 = \dot{x}$ one obtains the equations of motion in the first-order form as

$$\begin{bmatrix} 1 & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ -b_2|z_2|^{\omega}\operatorname{sgn}(z_2) - b_1|z_1|^{\nu}\operatorname{sgn}(z_1) \end{bmatrix}, \tag{5.1}$$

where $z_1(0) = 0$ and $z_2(0) = v$. It is desired to minimize the maximum acceleration of the mass. Therefore, it is a min–max problem which is converted to the standard nonlinear programming problem by defining an artificial variable b_3 as

$$\frac{1}{M}|\{b_2|z_2|^{\omega}\operatorname{sgn}(z_2) + b_1|z_1|^{\nu}\operatorname{sgn}(z_1)\}| - b_3 \leq 0\,, \quad 0 \leq t \leq T\,. \tag{5.2}$$

The problem then is to minimize b_3. There are three design variables for the problem. A point-wise constraint on the displacement is imposed as

$$|z_1(t)| - z_{1\,\max} \leq 0\,, \quad 0 \leq t \leq T\,. \tag{5.3}$$

To compare results with [1], numerical data is taken as $M = 1$, $v = 1$, $z_{1\,\max} = 1$, $\nu = 2$ and $\omega = 1$, 2, 3 and 4 giving four cases to be solved. The starting points are [0.50, 0.50, 0.53], [0.50, 0.50, 0.80], [0.50, 0.50, 0.65] and [0.50, 0.50, 0.47] for $\omega = 1$, 2, 3 and 4, respectively. The total time interval is considered as 12 sec. and the number of time grid points is set to 1200. The parameter α is set to 0.95. Since the constraint functions have a few peaks in the time interval of interest, no root-finding algorithm is used for locating t_j's. The program DE with local error of 10^{-7} and global error of 3×10^{-7} is employed to solve the first-order equations of motion and all the adjoint equations.

The state equation and the direct differentiation method equation are solved by the forward integration, and the adjoint equation either with impulse or with step function loading is

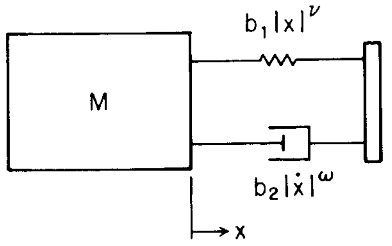


Fig. 5. Nonlinear impact absorber.

solved by backward integration. The impulse function $\delta(t - t_j)$ is simulated as a uniform load of value $1/\Delta t$ over a small region between $t_j - \Delta t$ and t_j, where Δt is the size of the time grid. Simpson's rule is selected to carry out the integration for design sensitivity coefficients.

All three methods of sensitivity analysis are employed separately to optimize above four cases for the problem. The optimization programs IDESIGN and LINRM are used for obtaining the optimum solutions. The final designs and the number of iterations taken by IDESIGN and LINRM with 10^{-3} tolerance for the convergence parameter and the constraint violation at the optimum are listed in Tables 1 and 2, respectively. Tables 3 and 4 show the cost function histories with the two programs. Let us designate ψ_1 as the displacement constraint of (5.3) and ψ_2 as the artificial acceleration constraint of (5.2). The active constraints at optimum for $\omega = 1$, 2, 3 and 4 are: ψ_1 at 2.01 sec and ψ_2 at 0.0 and 1.29 sec, ψ_1 at 1.93 sec and ψ_2 at 0.0 and 1.93 sec, and ψ_1 at 1.89 sec and ψ_2 at 0.0 and 1.89 sec, and ψ_1 at 1.84 sec and ψ_2 at 0.0 and 1.84 sec, respectively.

The number of function evaluations is greater than the number of gradient evaluations in LINRM, since it performs one-dimensional search after determining the direction of design change. Conversely, there is no line search in IDESIGN. Therefore, there is only one call to function and gradient evaluation subroutine at each iteration. From the results in Tables 1 and 2, it is concluded that all three methods of design sensitivity analysis work pretty well. Both IDESIGN and LINRM are reliable, but IDESIGN is more efficient than LINRM. Therefore, for subsequent design examples only IDESIGN is used to optimize problems.

Table 5 shows results obtained in [1]. Optimal solutions reported here are the same as given there. It is noted that the solutions of [1] were obtained by slightly modifying the formulation of the problem. The constraint function ψ_2 of (5.2) was active at two separate regions. With the

Table 1
Optimal solutions for nonlinear impact absorber with IDESIGN

Method	b	$\omega = 1$	$\omega = 2$	$\omega = 3$	$\omega = 4$
Direct differentiation method	b_1	0.441	0.597	0.683	0.754
	b_2	0.526	0.597	0.683	0.754
	Cost fun. b_3	0.526	0.597	0.683	0.754
	No. of iterations	8	3	4	8
	CPU time (sec)[a]	70.4	27.3	41.0	84.6
Adjoint variable method (impulse loading)	b_1	0.441	0.597	0.683	0.754
	b_2	0.527	0.597	0.683	0.754
	Cost fun. b_3	0.526	0.597	0.683	0.754
	No. of iterations	5	3	4	5
	CPU time (sec)[a]	31.3	20.9	30.9	38.1
Adjoint variable method (step function loading)	b_1	0.441	0.597	0.683	0.753
	b_2	0.527	0.598	0.683	0.754
	Cost fun. b_3	0.526	0.598	0.683	0.754
	No. of iterations	5	5	6	8
	CPU time (sec)[a]	29.5	32.6	43.0	58.2

[a]PRIME 750.

Table 2
Optimal solutions for nonlinear impact absorber with LINRM

Method	b	$\omega = 1$	$\omega = 2$	$\omega = 3$	$\omega = 4$
Direct differentiation method	b_1	0.441	0.597	0.683	0.754
	b_2	0.526	0.597	0.683	0.754
	Cost fun. b_3	0.526	0.597	0.683	0.754
	No. of fun. eval.	36	24	19	31
	No. of gra. eval.	8	7	6	8
Adjoint variable method (impulse loading)	b_1	0.441	0.597	0.683	0.754
	b_2	0.526	0.597	0.683	0.754
	Cost fun. b_3	0.526	0.597	0.683	0.754
	No. of fun. eval.	36	24	19	31
	No. of gra. eval.	8	7	6	8
Adjoint variable method (step function loading)	b_1	0.441	0.597	0.683	0.754
	b_2	0.527	0.598	0.683	0.754
	Cost fun. b_3	0.526	0.598	0.683	0.754
	No. of fun. eval.	23	31	11	24
	No. of gra. eval.	6	8	5	8

Table 3
Cost function histories for nonlinear impact absorber with IDESIGN

Method	Iteration No.	1	2	3	4	5	6	7	8
Direct differentiation method	$\omega = 1$	0.500	0.544	0.544	0.527	0.526	0.513	0.526	0.526
	2	0.500	0.584	0.597					
	3	0.500	0.642	0.681	0.683				
	4	0.500	0.687	0.755	0.755	0.754	0.720	0.752	0.754
Adjoint variable method (impulse loading)	$\omega = 1$	0.500	0.544	0.544	0.527	0.526			
	2	0.500	0.584	0.597					
	3	0.500	0.643	0.682	0.683				
	4	0.500	0.687	0.755	0.755	0.754			
Adjoint variable method (step function loading)	$\omega = 1$	0.500	0.545	0.545	0.527	0.526			
	2	0.500	0.577	0.600	0.600	0.598			
	3	0.500	0.642	0.690	0.690	0.684	0.683		
	4	0.500	0.713	0.713	0.713	0.764	0.764	0.757	0.754

equivalent constraint functional formulation of (2.7) for ψ_2, the optimization algorithm of [1] could not converge. It was observed that the acceleration would be high at the initial impact. Therefore, acceleration at the initial impact was treated as a separate constraint. Then there was only one additional region at which the constraint ψ_2 was active. With this modification the optimization method of [1] converged without any difficulty. Therefore, this simple example clearly demonstrates the theoretical difficulties with the equivalent functional for-

Table 4
Cost function histories for nonlinear impact absorber with LINRM

Method	Iteration No.	1	2	3	4	5	6	7	8
Direct differentiation method	$\omega = 1$	0.500	0.476	0.451	0.435	0.430	0.513	0.512	0.526
	2	0.500	0.469	0.461	0.573	0.568	0.596	0.597	
	3	0.500	0.465	0.628	0.623	0.679	0.683		
	4	0.500	0.462	0.447	0.443	0.651	0.640	0.741	0.754
Adjoint variable method (impulse loading)	$\omega = 1$	0.500	0.476	0.450	0.435	0.429	0.513	0.512	0.526
	2	0.500	0.469	0.461	0.573	0.568	0.596	0.597	
	3	0.500	0.465	0.629	0.624	0.680	0.683		
	4	0.500	0.460	0.446	0.442	0.650	0.639	0.741	0.754
Adjoint variable method (step function loading)	$\omega = 1$	0.500	0.479	0.461	0.521	0.519	0.526		
	2	0.500	0.480	0.484	0.480	0.565	0.559	0.594	0.598
	3	0.500	0.498	0.637	0.681	0.684			
	4	0.500	0.489	0.473	0.455	0.665	0.747	0.754	0.754

Table 5
Optimal solutions for nonlinear impact absorber from [1]

b	$\omega = 1$	$\omega = 2$	$\omega = 3$	$\omega = 4$
b_1	0.453	0.597	0.682	0.752
b_2	0.531	0.597	0.682	0.752
b_3	0.531	0.597	0.682	0.752
Iterations	6–10			

mulation for the point-wise state variable constraint. That is, the equivalent constraint functional treatment of point-wise state variable constraint is not correct. The max-value treatment of such constraints is much more reasonable, theoretically correct and works well numerically.

EXAMPLE 5.2 (Optimal design of a linear two degree of freedom vibration isolator). The linear two degree of freedom vibration isolator is shown in Fig. 6. The objective here is to find the damping and spring constants that minimize the peak transient dynamic response (displacement) of the main mass over a finite number of excitation frequencies for the forcing function, subject to constraints on transient and steady state responses, and explicit bounds on design variables. Two design problems are solved. The first problem has just one excitation frequency and the second problem has five excitation frequencies; [1] may be consulted for detailed formulation of these problems.

The total time interval is set to 2.0 sec and the number of time grid points is set to 1000. The scalar α is defined as 0.95. This means that we consider points with constraint values more than 0.95 times the maximum constraint value as active constraints for each constraint

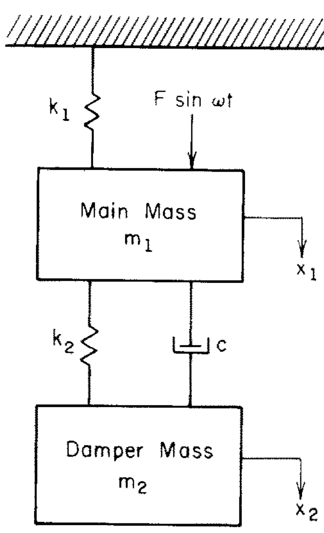


Fig. 6. Two degree of freedom dynamic absorber.

function. The method does not converge if we use $\alpha = 1$. The bisection method is selected to locate the points t_j with an accuracy of 10^{-4} sec. The program DE with local error of 10^{-7} and global error of 3×10^{-7} is employed for solving the first-order differential equations. The Hermite cubic interpolation is used to calculate the values of adjoint variable at those points that are needed in integrating for design sensitivities. The simulation of $\delta(t - t_j)$ and integration method are the same as in Example 5.1. The initial designs for the problems are [1.60, 0.02, 3.189] and [1.00, 0.02, 9.844], respectively.

The efficient program IDESIGN with 10^{-3} tolerance for convergence parameter and constraint violation is used to solve for the optimal solutions. Optimal solutions for the Design

Table 6
Optimal solutions for two degree of freedom dynamic absorber with IDESIGN ($10^{-4} \leq b_1 \leq 2.0$, $10^{-6} \leq b_2 \leq 0.16785$)

	Design Problem 1			Design Problem 2	
b	Direct differentiation method	Adjoint variable method (impulse loading)	[1]	Adjoint variable method (impulse loading)	[1]
b_1	1.3277	1.3277	1.3380	0.9213	0.9213
b_2	0.03047	0.03054	0.02121	0.1545	0.1548
Cost fun. b_3	2.3562	2.3566	2.3670	4.296	4.291
No. of iterations	11	11	21	11	22
CPU time (sec)	415.4[a]	260.8[a]	109.3[b]	1507.2[a]	421.7[b]

[a]PRIME 750.
[b]IBM 360/65.

Table 7
Cost function histories with IDESIGN for two degree of freedom dynamic absorber

	Design Problem 1		Design Problem 2
Iteration No.	Direct differentiation method	Adjoint variable method (impulse loading)	Adjoint variable method (impulse loading)
1	3.189	3.189	9.844
2	3.189	3.189	9.844
3	1.967	1.794	9.844
4	1.967	2.118	9.844
5	1.967	2.118	5.422
6	2.298	2.326	5.422
7	2.334	2.372	5.422
8	2.359	2.367	5.422
9	2.356	2.362	4.034
10	2.359	2.362	4.280
11	2.356	2.357	4.296

Problem 1 are obtained by the direct differentiation method and the adjoint variable method with the impulse loading. The other adjoint variable method is not used. Since for the Design Problem 2, the equations needed to be solved by the direct differentiation method are much larger than the adjoint variable method with impulse loading, only the latter method is used. In other words, the latter method is efficient compared to the direct differentiation method. All of the final designs are listed in Table 6. The cost function histories are given in Table 7. Active constraints at the optimum for Design Problem 1 are: the steady state relative displacement between two masses, the transient response for the main mass at $t = 0.13768$, and 0.51795. The active constraints at the optimum for Design Problem 2 are: the transient response of the main mass at $t = 0.8531$ sec for excitation frequency $\zeta = 0.8$, at $t = 0.2504$ sec for $\zeta = 1.0$ and at $t = 0.9629$ sec for $\zeta = 1.1$. The results by the new methods, shown in Table 6, are similar to [1]. However, IDESIGN obtains these results in approximately half the number of iterations as compared to the method in [1]. It clearly shows superiority of the new method.

EXAMPLE 5.3 (Design of a 5 degree of freedom vehicle suspension system). Fig. 7 shows a 5 degree of freedom vehicle suspension system. The vehicle suspension system is to be designed to minimize the extreme acceleration of the driver's seat for a variety of vehicle speeds and road conditions. Spring constants k_1, k_2, and k_3 and damping coefficients c_1, c_2, and c_3 of the system are chosen as design parameters. The motion of the vehicle is constrained so that the relative displacements between the chassis and the driver's seat, and the chassis and the front and rear axles are within given limits. Further, it is desirable to constrain seat acceleration due to an additional set of extreme road conditions. The design variables are also constrained; [1] may be consulted for detailed derivation of various equations. Two design problems are considered here that are exactly the same as Design Problems 1 and 3 of [1]. In the first design problem only one road condition is considered, i.e., Profile No. 1 in Fig. 8. The second design problem has two road conditions as shown in Fig. 8.

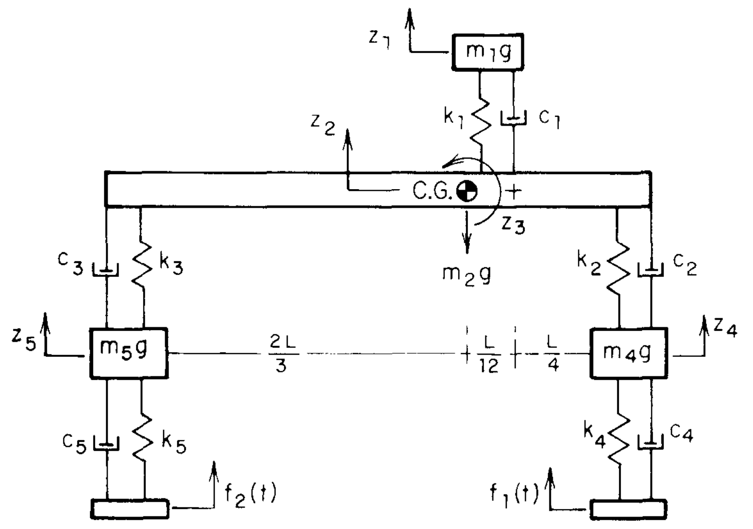
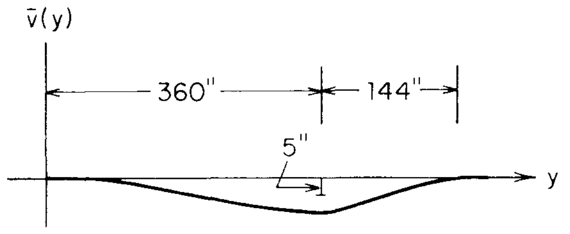


Fig. 7. Five degree of freedom vehicle model.

Most of the numerical considerations for this example such as the number of time grids, the value of α, Hermite cubic interpolation, first-order method and Simpson's rule are the same as for the previous problem. The total time interval is considered as 5.0 sec. Bisection method is employed with an accuracy of 10^{-9} sec. The impulse function $\delta(t - t_j)$ is simulated as a uniform load of value $5/\Delta t$ over a region of $t_j - 0.2\,\Delta t$ to t_j, where Δt is size of the time grid. The initial designs for two problems are [100.0, 300.0, 300.0, 10.0, 25.0, 25.0, 332.6] and [100.0, 300.0, 300.0, 10.0, 25.0, 25.0, 198.6], respectively. Lower and upper bounds for the design variables are [50, 200, 200, 2, 5, 5, 1] and [500, 1000, 1000, 50, 80, 80, 500], respectively.

The efficient adjoint variable method with impulse loading is employed for solving the two



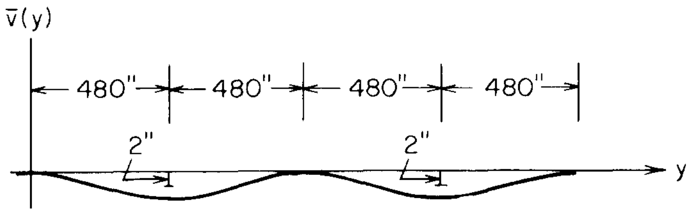


Fig. 8. Road surface profiles; (a) Profile No. 1; (b) Profile No. 2.

Table 8
Optimal solutions for vehicle suspension system with IDESIGN

	Design Problem 1 [1]		Design Problem 3 [1]	
b	Adjoint variable method (impulse loading)	[1]	Adjoint variable method (impulse loading)	[1]
b_1	50.00	50.00	50.00	50.00
b_2	204.1	200.0	200.0	200.0
b_3	293.9	241.9	200.0	200.0
b_4	30.87	12.89	31.88	46.58
b_5	76.94	77.52	77.40	78.44
b_6	80.00	80.00	80.00	26.21
Cost fun. b_7	255.8	257.4	100.2	103.5
No. of iterations	23	55	30	44
CPU time (sec)	3755.8[a]	861.8[b]	6063.7[a]	1535.2[b]

[a]PRIME 750.
[b]IBM 360/65.

design problems of this example. Again, IDESIGN is used as the optimizer. The tolerances for convergence parameter and constraint violation in IDESIGN are set to 10^{-3} for both design problems. The optimal solutions obtained with these tolerances are shown in Table 8. The cost function history for the first design problem is as follows: 332.6, 332.6, 332.6, 332.6, 332.6, 199.6, 199.6, 199.6, 199.6, 199.6, 246.2, 246.2, 246.2, 246.2, 260.8, 260.8, 260.8, 253.5, 253.5, 253.5, 253.5, 255.8, 255.8. The active constraints for this design problem are the artificial acceleration constraint at 0.9774 sec, the relative displacement constraint between the chassis and the front axle at 1.056 sec, the lower bound for design variable b_1 and the upper bound for design variable b_6. The cost function history for the second design problem is as follows: 198.6, 198.6, 198.6, 198.6, 198.6, 119.2, 119.2, 119.2, 119.2, 72.72, 72.72, 72.72, 72.72, 72.72, 95.01, 95.01, 95.01, 95.01, 95.01, 95.01, 100.2, 100.2, 97.59, 99.43, 99.43, 99.43, 99.43, 99.70, 99.33, 100.2. The active constraints for this design problem are the artificial acceleration constraint due to road profile No. 2 at 1.165 sec, the lower bound for design variables b_1, b_2 and b_3, and the upper bound for design variable b_6. Results of two design problems in Table 8 show not only better solutions, but also the reduced number of iterations as compared to those in [1]. Thus, we can conclude that the new method is more reliable and IDESIGN is more efficient.

6. Numerical experience

Some numerical experience is worthy of discussion for the accuracy of design sensitivity coefficients and the optimal solution. First of all, the time interval for the integration of equations of motion should be selected as large as possible to include all worst cases in the optimal design process. If the time interval is small, the optimal solution may converge to a different point. In addition, there may be some violations of constraints outside the specified time interval.

The number of time grids does not affect accuracy of solution of the first-order equations of motion, if a variable-stepsize integration method is used. However, it will affect the accuracy of the interpolation values of adjoint variable and the accuracy of integration for sensitivity coefficients. Thus, one needs to carefully select the number of time grid points. The program DE uses the Adams family of formulas, but modifies them to deal with a variable stepsize. It has very good error control, including the choice of the order and the stepsize. It has been so well designed from both the viewpoint of error control and user convenience that it is a well-recommended program for solving first-order differential equations [11].

The simulation of impulse load is based on a rectangular load over the region $0 \leq t \leq \Delta t$ such that the multiplication of load value by Δt is equal to one. Solution with unit impulse is first obtained. It is then properly scaled to obtain the real solution. For better simulation of impulsive load, Δt should be as small as possible.

The Hermite cubic polynomial formula is good for interpolating the values of adjoint variable since the time derivative of the adjoint variable is easy to obtain.

Selection of the parameter α $(0 \leq \alpha \leq 1)$ is an art. Too big a value results in failure to converge to the optimal solution, and a smaller value results in more computational time. However, a smaller value of α is recommended for stable behavior of the optimization process. The method cannot converge for all the previous examples except the Design Problem 2 of Example 5.2 and two design problems of Example 5.3 if α is set to 1. Use of $\alpha = 0.95$ has worked pretty well for all examples.

The bisection and secant methods of root-finding algorithms are tried to locate local maximum points. Although the secant method is more efficient, it diverges in some situations. It is therefore necessary to adopt the bisection method. One may combine both the methods to use a hybrid approach. In such an approach the secant method would be used first. Then the process would be switched to the bisection method if divergence occurred. It should be noted that the position of the local maximum point is very sensitive for calculations of the constraint function values and design sensitivity coefficients for some problems, such as Example 5.3. In those cases, the root-finding process must continue until a very tight convergence criterion is satisfied. Finally, fourth-order integration rule (Simpson's rule) is accurate enough to carry out all integrations for design sensitivity analysis, if the number of time grids is large.

7. Discussion and conclusions

The problem of treating point-wise state variable constraints in dynamic response optimization of systems is addressed in the paper. In a previous treatment, such constraints were transformed to equivalent functional constraints by integrating over the time interval of interest. Theoretical as well as numerical difficulties with such transformations of point-wise constraints are studied. These difficulties are interpreted from an engineering point of view.

An alternative treatment for point-wise state variable constraints is proposed. In this treatment the point-wise state variable constraint is replaced by a finite number of constraints imposed only at the points of local maxima for the original constraint function. The treatment is interpreted in terms of 'worst-case design formulation' that was developed for static problems in [1]. In the 'worst-case formulation', the constraint is imposed only at points of local maxima. Therefore, in this sense the present paper extends the 'worst-case design formulation' of static response problems to the dynamic response problem.

The points of local maxima (worst cases) for a function depend on the design for the system, i.e., the location of worst cases changes as the design is changed in the iterative optimization process. However, it is shown that the design dependence of location of worst cases has no effect on first-order design sensitivity coefficients. In other words, the location of worst cases can be kept fixed (not an assumption) during an iteration of the design optimization process. This greatly simplifies numerical calculations for design gradients of constraints.

Three methods of design sensitivity analysis for the state variable constraint imposed at a particular point in time—the direct differentiation method, and two adjoint variable methods, one with impulse loading and the other with step function loading—are then presented. A comparison is made to claim that the adjoint variable methods are efficient. In many applications, these methods call for solution of the adjoint equation with only one right-hand side for calculating the design sensitivity coefficients at all the local maximum points of each constraint function.

Numerical results show the new formulation of the optimal design problem to be very reliable. It does not have any of the convergence difficulties that were encountered with the equivalent functional treatment of the point-wise state variable constraint of [1]. It was possible to satisfy very strict convergence criteria for optimum design in the computer program IDESIGN. The program IDESIGN and its algorithm worked extremely well. It is a well-designed program as it does not require specification of any optimization related parameters by the user. Although all of the examples presented belong to mechanical system design, the formulation can be easily applied to design structural systems.

Some parallel literature [16–18] on dynamic response optimization and the associated design sensitivity analysis has been brought to our attention (by a reviewer). Whereas the general problem area addressed in these references and the present paper is the same, there are several key differences in the material, methods and associated analyses. The literature, however, is important and relevant. Therefore, we will briefly review the literature and relate it to the present work.

In the present paper, difficulties with the previous treatment of point-wise state variable constraints are studied. The difficulties are explained from mathematical, engineering and numerical view points. It is extremely important to understand these difficulties so that they can be avoided in the future. Such fundamental analyses and discussions are not presented in the above references.

The dynamic response constraints are imposed at the terminal time in [16], or some extreme response point in [17]. These time points are assumed to be design dependent (as in the present paper) and are determined by certain additional conditions of the form

$$\Omega_j(t_j, z, b) = 0\,. \tag{7.1}$$

The above equation is also used to determine the derivative of t_j with respect to the design variables b. Differentiating (7.1) with respect to b, we obtain [17, 18]

$$\frac{\partial \Omega_j}{\partial b} + \frac{\partial \Omega_j}{\partial z}\frac{\partial z(t_j)}{\partial b} + \left[\frac{\partial \Omega_j}{\partial t_j} + \frac{\partial \Omega_j}{\partial z}\dot{z}(t_j)\right]\frac{\mathrm{d}t_j}{\mathrm{d}b} = 0\,. \tag{7.2}$$

Since (7.1) is to determine t_j, the coefficient of $\mathrm{d}t_j/\mathrm{d}b$ in (7.2) cannot be zero [17, 18]. Thus, $\mathrm{d}t_j/\mathrm{d}b$ can be solved from (7.2) and substituted into (2.13). The design sensitivity analysis procedure gives rise to some additional terms in definition of the adjoint variable problem as well as the final design derivative expressions [17, 18].

There is nothing wrong with the above procedure. However, a key point has been missed in [17, 18]. The above procedure is unnecessary when the constraints are imposed at the max-points for the point-wise state variable function. As shown in Section 2 of this paper, the coefficient of $\mathrm{d}t_j/\mathrm{d}b$ in (2.13) is zero (see (2.14)) when the constraint is imposed at the max-points for the function. Thus, the numerical procedure of design sensitivity analysis advocated in [17, 18] needs unnecessary additional calculations. There is, however, a positive aspect to the above analysis: When a point-wise state variable constraint is imposed at a point t_j which is not the max-point, and the point t_j is design dependent, then the coefficient of $\mathrm{d}t_j/\mathrm{d}b$ in (2.13) does not vanish. Therefore, the above procedure must be utilized to calculate correct design sensitivity coefficients. These types of applications are not considered in the present paper. However, design sensitivity analysis procedures presented in Section 3 can be routinely extended by keeping the terms associated with $\mathrm{d}t_j/\mathrm{d}b$ in (2.13) and (3.3), and other expressions. The definition of the associated adjoint variable problems will be somewhat modified.

There is another flaw in the numerical procedure of [17, 18]. It suggests imposing the constraint at one extreme point for each point-wise state variable function; see [17, (43)] and [18; (34), (35), (51) and the explanation below (51)]. In general, this procedure causes numerical difficulties, as oscillations in the location of global max-point occur in the optimization process. This is confirmed in the numerical examples presented in the paper. Therefore, constraints must be imposed at all the local max-points, as proposed in the present paper.

Numerical details for methods used to integrate differential equations, locate extreme points, and various integrations needed to calculate sensitivity coefficients are missing in [17, 18]. In the present paper all the numerical details are presented. In addition, the observation of linear adjoint equations in design sensitivity analysis procedure is made. This property is fully exploited in developing the most efficient procedure for design sensitivity analysis. Unlike [16–18], three methods of design sensitivity analysis are presented and evaluated from theoretical as well as numerical standpoints.

In [16–18], the Dirac delta measure is not introduced to calculate design sensitivity coefficients for the constraint of (3.1). With that approach, one need not calculate adjoint response due to an implusive load. The adjoint equation is homogeneous but the initial conditions are nonhomogeneous. This formulation of the present paper is equivalent to the one presented in [16–18]. The impulsive response adjoint problem of the present paper can be transformed to an equivalent problem with homogeneous differential equation and nonhomogeneous initial conditions. From numerical standpoint this is perhaps a better approach. In any case, the present paper presents two other methods that do not use the Dirac delta measure.

In [17, 18], the program LINRM is used to solve optimization problems. In the present paper, we have used and evaluated algorithms of both the LINRM and the IDESIGN programs. It is concluded that IDESIGN is more efficient compared to LINRM.

Acknowledgment

Suggestions given by the reviewers to improve presentation of the material of the paper are gratefully acknowledged.

References

[1] E.J. Haug and J.S. Arora, Applied Optimal Design (Wiley, New York, 1979).
[2] T.T. Feng, J.S. Arora and E.J. Haug, Optimal structural design under dynamic loads, Internat. J. Numer. Meths. Engrg. 11 (1977) 39–52.
[3] A.E. Bryson, Jr. and Y.C. Ho, Applied Optimal Control (Ginn, Waltham, MA, 1969).
[4] J.S. Arora, Efficient constraint treatment in structural optimization, Paper 80-501, Proc. ASCE Annual National Convention, Hollywood, FL, 1980.
[5] C.C. Hsieh and J.S. Arora, Structural design sensitivity analysis with general boundary conditions: Dynamic problem, Internat. J. Numer. Meths. Engrg. (1984) to appear.
[6] L. Shampine and M. Gordon, Computer Solution of Ordinary Differential Equations (Freeman, San Francisco, CA, 1975).
[7] L. Shampine and H. Watts, Global error estimation for ordinary differential equations, ACM Trans. Math. Software 2 (1976) 172–186.
[8] R.T. Haftka, Techniques for thermal sensitivity analysis, Internat. J. Numer. Meths. in Engrg. 17 (1981) 71–80.
[9] J.S. Arora and E.J. Haug, Methods of design sensitivity analysis in structural optimization, AIAA J. 17 (9) (1979) 970–973.
[10] C.C. Hsieh and J.S. Arora, Structural design sensitivity analysis with general boundary conditions: Static problem, Internat. J. Numer. Meths. Engrg. (1984) to appear.
[11] K.E. Atkinson, An Introduction to Numerical Analysis (Wiley, New York, 1978).
[12] J.S. Arora, An algorithm for optimum structural design without line search, in: R.H. Gallagher et al., eds., New Directions in Structural Optimization (Wiley, New York, 1984).
[13] J.S. Arora and W.B. Liu, User's manual for program IDESIGN, Tech. Rept., Division of Materials Engineering, University of Iowa, Iowa City, IA, 1983.
[14] A.D. Belegundu and J.S. Arora, A new recursive quadratic programming method with active set strategy for optimal design, Internat. J. Numer. Meths. Engrg. (1984) to appear.
[15] B.N. Pshenichny and Y.M. Danilin, Numerical Methods in Extremal Problems (MIR, Moscow, 1978).
[16] E.J. Haug and P.E. Ehle, Second order design sensitivity analysis of mechanical system dynamics, Internat. J. Numer. Meths Engrg. 18 (1982) 1699–1717.
[17] K.K. Choi, E.J. Haug, J.W. Hou and V.N. Sohoni, Pshenichny's linearization method for mechanical system optimization, ASME J. Mechanisms, Transmissions, and Automation in Design 105 (1983) 97–103.
[18] E.J. Haug, V.N. Sohoni, S.S. Kim and H.-G. Seong, Vehicle suspension dynamic optimization, Proc. International Conference on Modern Vehicle Design Analysis, London, 1983.