AIAA JOURNAL
Vol. 43, No. 10, October 2005

Alternative Formulations for Transient Dynamic Response Optimization

Qian Wang* and Jasbir S. Arora†
University of Iowa, Iowa City, Iowa 52242

Three alternative formulations for transient dynamic response optimization of mechanical systems are presented, analyzed, and evaluated. The basic idea of these formulations is to treat various state variables as independent variables in the optimization process (in addition to the real design variables), that is, generalized displacements, velocities, and accelerations. Because all functions of the optimization problem become explicit in terms of the variables in these formulations, their gradients can be calculated easily as compared to the conventional approach where special design sensitivity analysis methods need to be used. Also, the equations of motion are not integrated explicitly; they are imposed as equality constraints. For the alternative formulations, the optimization problem is quite large in terms of the numbers of variables and constraints. However, the problem functions are quite sparse, which is exploited in the optimization process. Advantages and disadvantages of the formulations are discussed. Several cases of two example problems are solved to study the performance of the formulations. Based on the extensive numerical experiments, it is concluded that the proposed formulations work well for the example problems and have potential for further development for practical applications.

I. Introduction

TRANSIENT dynamic response optimization problems are difficult to solve because they involve integration of linear or nonlinear differential-algebraic equations or just differential equations (DEs). The most common approach for optimization of such problems has been the one where only the design variables are treated as optimization variables.[1–8] All other response quantities, such as displacements, velocities, and accelerations, are treated as implicit functions of the design variables. Therefore, in the optimization process, a system of DEs is integrated to obtain the response (state) variables and to calculate values of various functions of the optimization problem. Then an optimization algorithm is used to update the design. This nested process of solution of DEs and design update, also called the conventional approach, is repeated until a stopping criterion is satisfied. This optimization process, however, is difficult to use in practice. The main difficulty is that the response quantities are implicit functions of the design variables, which require special methods for their gradient evaluation, such as the direct differentiation method or the adjoint variable method.[4] These methods require integration of additional DEs. Unless a finite difference method is used to calculate the gradients, it is difficult to optimize systems with any existing simulation software, especially for multidisciplinary problems requiring use of different discipline-specific analysis software.

Another interesting approach for transient dynamic optimization is the equivalent static load method,[9,10] where the problem is transferred to a quasi-static problem. The idea is to find a static load set that can generate the same displacement field as that with the dynamic load at certain times. Therefore, multiple equivalent static load sets obtained at all of the time intervals can represent various states of the structure under the dynamic load. However, with this approach, DEs must still be integrated a number of times and design sensitivity analysis must also be performed for the resulting static problems.

To alleviate the difficulties mentioned, a fundamental shift in the direction of research on optimization of dynamic systems is needed. It will be useful to develop alternative formulations that do not require explicit solution of DEs at each iteration and where there is no need for special design sensitivity analysis procedures. By the formulation of the optimization problem in a mixed space of design and state variables, these two objectives can be met. Thus, the purpose of this paper is to expand on this idea by proposing alternative formulations for optimization of transient dynamic systems and to evaluate these using extensive numerical experiments. Various state variables, such as the displacements, velocities, and accelerations, are treated as independent variables in the formulations. The equations of motion become equality constraints. All constraints of the problem in the proposed formulations are expressed explicitly in terms of the optimization variables. Therefore, their gradient evaluations become quite simple. Although the resulting optimization problem is large, it is quite sparse and can be solved using sparse nonlinear programming (NLP) algorithms. Two numerical examples are used to study the alternative approaches and compare solutions with the conventional approach. Advantages and disadvantages of the formulations are also discussed.

This idea of using state and design variables simultaneously in the optimization process has been used in the literature for the static response structural optimization problems. This is known as simultaneous analysis and design approach.[11–14] A similar approach has also been used to solve optimal control problems,[15–21] which is called the direct collocation-transcription method. The main applications of the approach are for aerospace trajectory optimization[15,18] and chemical process engineering.[17] The basic idea is to discretize the system of first-order DEs and define finite-dimensional approximations for the state and control variables. The discretized state equations are treated as equality constraints in the optimization process. The collocation conditions are determined using an implicit formula, such as the implicit Runge–Kutta method, or some polynomial interpolations. Bounds on the state variables are usually enforced at the time grid points, or at all of the collocation points in the time domain. The optimization variables are the state variables at time-grid points, the first derivatives of the state variables, and the control and the algebraic variables at all of the collocation

Received 6 July 2004; presented as Paper 2005-2277 at the AIAA/ASME/ASCE/AHS/ASC 46th Structures, Structural Dynamics, and Materials Conference and AIAA 1st Multidisciplinary Design Optimization Specialist Conference, Austin, TX, 18–21 April 2005; revision received 12 May 2005; accepted for publication 20 May 2005.

*Research Assistant, Optimum Design Laboratory/Center for Computer-Aided Design, College of Engineering; qwang2@engineering.uiowa.edu. Student Member AIAA.

†F. Wendell Miller Distinguished Professor of Engineering, Optimum Design Laboratory/Center for Computer-Aided Design, College of Engineering; Jasbir-Arora@uiowa.edu. Senior Member AIAA.

points. The resulting NLP is usually solved by a sequential quadratic programming (SQP) method[17,18] or an interior point method,[19] which exploit the sparse and block features of the problem structure. Some detailed formulations for optimal control problems may be found in Refs. 20 and 21.

The present work on simultaneous analysis and design for dynamic systems differs from the foregoing work in the following ways: 1) Three alternative formulations are proposed and evaluated for optimal design of transient dynamic mechanical systems. 2) Second-order forms of the equations of motion are also treated directly in the formulations. 3) The state variables are discretized using standard finite difference methods, making the numerical implementation quite easy and straightforward. 4) A more recent and powerful optimization algorithm and an associated software are used that take full advantage of the sparsity structure of the alternative formulations. 5) Conventional and alternative formulations are compared and evaluated. 6) Advantages and disadvantages of the two approaches are delineated.

II. Dynamic Response Optimization Problem

The basic dynamic response optimization problem is to determine design parameters related to stiffness and damping properties of the dynamic system, to achieve certain goals, for example, minimization of a cost function, such as the maximum displacement or acceleration, while satisfying all of the performance requirements. In this section, a general problem for optimization of dynamic systems is presented. In the remaining sections, the conventional and alternative formulations and numerical results for two example problems are presented and discussed.

A. Simulation Model

Let $\boldsymbol{x}$ be an m-dimensional vector to represent the design variables for the problem, which may include the sizing variables of a structure, or directly mass, stiffness, and damping parameters of the dynamic system. Here, z is a d-dimensional vector that represents the state variables for the problem. The equation of motion for a linear system to determine the state variables can be written as follows. Nonlinear problems can be treated similarly as seen later in an example problem. Thus,

$$\boldsymbol{M}(\boldsymbol{x})\ddot{z}(t) + \boldsymbol{C}(\boldsymbol{x})\dot{z}(t) + \boldsymbol{K}(\boldsymbol{x})z(t) = \boldsymbol{F}(\boldsymbol{x}, t) \tag{1}$$

with the initial conditions $z(0) = z_0$ and $\dot{z}(0) = \dot{z}_0$. $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{K}$ are $(d \times d)$ the generalized mass, damping, and stiffness matrices, and $\boldsymbol{F}(\boldsymbol{x}, t)$ is a $d \times 1$ generalized force vector. Note that Eq. (1) cannot be solved in a closed form to obtain explicit functional forms for $\ddot{z}$, $\dot{z}$, and z in terms of the design variables $\boldsymbol{x}$, which causes difficulty in the optimization process. The equation of motion can be solved directly in second-order form[22] as is done in the structural engineering literature where rotations of various parts of the structure are quite small. The equations can also be transformed to first-order form,[23] as is done in the controls and mechanical systems literature, by defining new variables,

$$y_j = z_j, \qquad y_{d+j} = \dot{z}_j, \qquad j = 1, d \tag{2}$$

Then Eq. (1) is transformed into a system of first-order differential equations, the so-called the state space representation of Eq. (1), as

$$\bar{\boldsymbol{C}}(\boldsymbol{x})\dot{\boldsymbol{y}}(t) + \bar{\boldsymbol{K}}(\boldsymbol{x})\boldsymbol{y}(t) = \bar{\boldsymbol{F}}(\boldsymbol{x}, t) \tag{3}$$

where $\bar{\boldsymbol{C}}$ and $\bar{\boldsymbol{K}}$ are system matrices $(2d \times 2d)$ and the dimension of $\bar{\boldsymbol{F}}(\boldsymbol{x}, t)$ is $2d \times 1$. Both the first-order and the second-order forms are examined for optimization formulations.

B. Cost Function and Constraints

In general, a cost functional includes the state and design variables, as

$$f = c_0(\boldsymbol{x}, T) + \int_0^T h_0(\boldsymbol{x}, z, \dot{z}, \ddot{z}, t)\,\mathrm{d}t \tag{4}$$

where T is the total time interval considered. The objective f may be the cost of the system, performance measures, or any other function of the state variables. A time-dependent functional, such as maximum acceleration or displacement, can also be treated, as will be seen later in the example problems. Design requirements are imposed mostly as inequality constraints. (Equality constraints can also be treated.) One type of constraint that involves integration over time as

$$g_i = c_i(\boldsymbol{x}, T) + \int_0^T h_i(\boldsymbol{x}, z, \dot{z}, \ddot{z}, t)\,\mathrm{d}t \le 0 \tag{5}$$

The other type of constraint is the so-called pointwise constraint, which needs to be satisfied at each point of the entire time interval $t \in [0, T]$,

$$g_i(\boldsymbol{x}, z, \dot{z}, \ddot{z}, t) \le 0 \tag{6}$$

These constraints may include the following displacement, velocity, and acceleration constraints and the time-independent constraints on the design variables:

$$z^L \le z \le z^U, \qquad \dot{z}^L \le \dot{z} \le \dot{z}^U, \qquad \ddot{z}^L \le \ddot{z} \le \ddot{z}^U \tag{7}$$

$$\boldsymbol{x}^L \le \boldsymbol{x} \le \boldsymbol{x}^U \tag{8}$$

where $z^L, \dot{z}^L, \ddot{z}^L$, and $\boldsymbol{x}^L$ and $z^U, \dot{z}^U, \ddot{z}^U$, and $\boldsymbol{x}^U$ are the lower and upper bounds for the generalized displacements, velocities, accelerations, and design variables, respectively. Any other design requirements may also be included in Eqs. (5) and (6).

III. Optimal Design Formulations

In this section, the conventional design formulation and three alternative formulations are presented and discussed.

A. Conventional Formulation: Only Design Variables as Optimization Variables

1. Formulation

In the conventional formulation, optimization is carried out only in the space of design variables. This is the most common way to formulate the transient dynamic response optimization problems,[4] and it includes the minimum number of optimization variables. The total time interval $[0, T]$ is divided into N intervals ($N + 1$ time grid points), and the equations of motion are integrated to determine the system response and, thus, calculate various functions of the optimization problem. The optimization problem is to find $\boldsymbol{x}$ to minimize the cost functional of Eq. (4) subject to the constraints of Eqs. (5), (6), and (8).

Note that five treatments of the pointwise constraints in Eq. (6) have been presented and evaluated in the literature.[3,4] These include the equivalent integral and critical point methods, as well as the conventional approach where the constraints are imposed at each time grid point. It has been concluded that the imposition of constraints at each time grid point is quite effective. The reason is that, although the number of constraints becomes very large with this approach, the number of active constraints is quite small (during the iterative process and at the optimum). Therefore, the approach works quite well and is quite easy to implement for numerical calculations. The equivalent integral and critical point constraint methods appear to be quite attractive as well because the number of constraints remains quite small. However, there are some theoretical difficulties with them causing the optimization process to fail sometimes.[3–5] These two methods were implemented for some of the examples presented later. As expected, the performance of the conventional optimization formulation was not good as compared to the constraint imposed at each time point in terms of the computational effort as well as convergence to the optimum point. Therefore, in this study, results are presented with only the approach where the constraints are imposed at each time grid point.

2. Gradient Evaluation

Because the constraint functions in this formulation are implicit functions of the design variables, implicit differentiation procedures need to be used to evaluate the gradients. Certainly, the finite difference methods can be used to evaluate the gradients because they are quite easy to implement. However, the accuracy of the gradients is questionable, which may affect convergence to the optimal solution. Analytically, the direct differentiation or the adjoint variable method can be used.[4,5] These procedures are difficult to implement with an existing simulation code because the code needs to be recalled to solve for displacement, velocity, and acceleration gradients, or the adjoint vectors. Then, the gradients of the response functionals need to be assembled using the adjoint vectors, or the displacement, velocity, and acceleration gradients. This is one of the main drawbacks of the conventional optimization formulation.

In the present work, both the finite difference and direct differentiation approaches were used to solve the example problems. Both of the methods worked well and converged to the same solutions. However, only the results requiring least amount of computational effort are reported.

B. Alternate Formulation 1: Design Variables and Displacements as Optimization Variables

In this formulation, the displacements $z(t)$ are also treated as optimization variables. Thus, the problem is to determine $\boldsymbol{x}$ and $z(t)$ to minimize the cost functional of Eq. (4) subject to the constraints of Eqs. (1), (5), (6), and (8). In the numerical solution process, Eqs. (1) and (6) are discretized into $N+1$ equations in the entire time interval [0, T], and they represent in fact the constraints that need to be satisfied at each time grid point, as

$$\boldsymbol{M}(\boldsymbol{x})\ddot{z}_i + \boldsymbol{C}(\boldsymbol{x})\dot{z}_i + \boldsymbol{K}(\boldsymbol{x})z_i = \boldsymbol{F}(\boldsymbol{x}, t_i), \qquad i = 0, N \tag{9}$$

$$\boldsymbol{g}_i(\boldsymbol{x}, z_i, \dot{z}_i, \ddot{z}_i, t_i) \le \boldsymbol{0}, \qquad i = 0, N \tag{10}$$

The optimization variables now become $\boldsymbol{x}$ and z_i, $i = -1, N+1$. Note that Eqs. (9) are the equality constraints between the variables representing the equations of motion for the system. It is obvious that to make Eqs. (9) and (10) explicit with respect to the optimization variables $\boldsymbol{x}$ and z the velocity and acceleration vectors $\dot{z}$ and $\ddot{z}$ need to be written explicitly in terms of z. One simple way is to use the finite difference approximations for them as

$$\dot{z}_i = \dot{z}_i(t_i) = \frac{z_{i+1} - z_{i-1}}{2\Delta t}, \qquad i = 0, N \tag{11}$$

$$\ddot{z}_i = \ddot{z}_i(t_i) = \frac{z_{i+1} - 2z_i + z_{i-1}}{\Delta t^2}, \qquad i = 0, N \tag{12}$$

where Δt is the time interval defined as $\Delta t = T/N$. Thus, in terms of the variables $\boldsymbol{x}$ and z_i, all of the problem functions have an explicit form. Therefore, special design sensitivity analysis procedures are not needed to evaluate the gradients. Even the equations of motion are not integrated, but are imposed as equality constraints in the optimization process. Note that these equations may not be satisfied at each optimization iteration. They are, however, satisfied at the final solution. Note also that Eqs. (9) are linear in z_i if the original DEs are linear and nonlinear if they are nonlinear.

Note that ideas similar to direct collocation can also be used here[21]; however, they usually involve more complicated relationships among the state variables as equality constraints. That will make the gradient evaluation as well as computer implementation more complex.

C. Alternate Formulation 2: Design Variables, Displacements, and Velocities as Optimization Variables

In alternate formulation 1, all functions needed to be expressed in terms of the design variables and displacements for numerical calculations. However, if the velocities or accelerations are also treated as variables, it will give choice of expressing some constraints, for example, the equations of motion, in terms of velocities or accelerations, to simplify their expressions. This may lead to simpler gradient evaluation and computer implementation. To evaluate this idea, velocities and accelerations are also treated as variables in this and the next subsection.

If displacements and velocities are treated as optimization variables, accelerations can be expressed in terms of the velocities. Therefore, another explicit form can be obtained for the dynamic optimization problem. The problem is to determine $\boldsymbol{x}$, z, and $\dot{z}$ to minimize the cost function of Eq. (4), subject to Eqs. (1), (5), (6), and (8). The optimization variables in this formulation are $\boldsymbol{x}$, z_i, and $\dot{z}_i$, $i = -1, N+1$, and this is denoted as form 1 of the formulation. In fact, this form is similar to the state-space representation. If the state-space variables $\boldsymbol{y}$ in Eq. (3), that is, displacements z and velocities $\dot{z}$, are treated as variables in the optimization formulation, the problem functions become explicit in terms of these variables. The problem is to determine $\boldsymbol{x}$ and $\boldsymbol{y}$ to minimize the cost function of Eq. (4), subject to the first-order DEs in Eq. (3) and the pointwise constraints

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, t) \le \boldsymbol{0} \tag{13}$$

Note that Eqs. (3) and (13) are discretized into $N+1$ equations in the entire time interval [0, T] and they in fact represent the system of constraints as follows:

$$\bar{\boldsymbol{C}}(\boldsymbol{x})\dot{\boldsymbol{y}}_i + \bar{\boldsymbol{K}}(\boldsymbol{x})\boldsymbol{y}_i = \bar{\boldsymbol{F}}_i(\boldsymbol{x}, t_i), \qquad i = 0, N \tag{14}$$

$$\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{y}_i, t_i) \le \boldsymbol{0}, \qquad i = 0, N \tag{15}$$

The optimization variables for this formulation are $\boldsymbol{x}$ and $\boldsymbol{y}_i$, $i = -1, N+1$. To make Eqs. (14) and (15) explicit with respect to the optimization variables $\boldsymbol{x}$ and $\boldsymbol{y}$, vector $\dot{\boldsymbol{y}}$ needs to be written explicitly in terms of $\boldsymbol{x}$ and $\boldsymbol{y}$. By using the finite difference approximations, we get

$$\dot{\boldsymbol{y}}_i = \dot{\boldsymbol{y}}_i(t_i) = \frac{\boldsymbol{y}_{i+1} - \boldsymbol{y}_{i-1}}{2\Delta t}, \qquad i = 0, N \tag{16}$$

Note that Eq. (16) is not as accurate as Eq. (12) because the error involved here is $\mathcal{O}[(2\Delta t)^2]$ (Ref. 23), which is larger than that for Eq. (12). To obtain an approximation of the accelerations with higher accuracy, the velocities at the middle of each grid points, $\dot{z}_{(2i-1)/2}$, $i = 0, N+1$, are introduced as additional variables, and this is called form 2 of the formulation. The accelerations are, therefore, expressed as

$$\ddot{z}_i = \frac{\dot{z}_{(2i-1)/2} - \dot{z}_{(2i-1)/2}}{\Delta t}, \qquad i = 0, N \tag{17}$$

The optimization variables now are $\boldsymbol{x}$; z_i, $i = -1, N+1$; $\dot{z}_i$, $i = 0, N$; and $\dot{z}_{(2i-1)/2}$, $i = 0, N+1$. More equality constraints are needed due to the introduction of additional variables. From the finite difference approximation, the velocities are expressed as follows:

$$\dot{z}_i = \frac{z_{i+1} - z_{i-1}}{2\Delta t}, \qquad i = 0, N \tag{18}$$

$$\dot{z}_{(2i-1)/2} = \frac{z_i - z_{i-1}}{\Delta t}, \qquad i = 0, N+1 \tag{19}$$

D. Alternate Formulation 3: Design Variables, Displacements, Velocities, and Accelerations as Optimization Variables

From Eq. (1), it can be seen that other alternative formulations are possible. If the displacements z, velocities $\dot{z}$, and accelerations $\ddot{z}$ are treated as variables simultaneously, another explicit formulation can be obtained. Because the state variables are related to each other, more equality constraints need to be imposed in the formulation, such as Eqs. (11) and (12).

The problem is to determine $\boldsymbol{x}$, z, $\dot{z}$, and $\ddot{z}$ to minimize the cost function of Eq. (4), subject to Eqs. (1), (5), (6), and (8). After discretization, the optimization variables in this formulation are $\boldsymbol{x}$, z_i ($i = -1, N+1$), $\dot{z}_i$ and $\ddot{z}_i$, $i = 0, N$.

E. Discussion of Formulations

Table 1 shows the sizes of all of the formulations. In Table 1, m is the number of elements in the design variable vector $\boldsymbol{x}$, N is the number of time intervals (where number of grid points $= N+1$), e is the number of constraints in $\boldsymbol{g}$, and d is the number of DEs in Eq. (1), or the dimension of vector z. Note that some of the inequality constraints g_i in alternative formulations 1–3 may become simple bounds on variables, as seen later in the examples.

Note that because the pointwise constraint is imposed only at the grid points, it is possible that the constraint could be violated between the grid points. This is the usual difficulty whenever a

Table 1 Number of variables and constraints for different formulations

Item	Conventional	Alternate 1	Alternate 2	Alternate 3
Number of variables	m	$m+d(N+3)$	$m+d(2N+6)$ or $m+d(3N+6)$	$m+d(3N+5)$
Number of equality constraints	0	$d(N+1)$	$d(2N+2)$ or $d(3N+4)$	$d(3N+3)$
Number of inequality constraints	$e(N+1)$	$e(N+1)$	$e(N+1)$	$e(N+1)$

continuum problem is discretized for numerical calculations. In the present case, if the system response is relatively smoother and slowly varying, the constraint violation, if any, between the grid points will be quite small. When the response is varying rapidly, a smaller grid size can reduce the possibility of violation between the grid points at the expense of additional calculations.

In the alternative formulations, because the objective and constraint functions are all explicit in terms of the optimization variables, the gradients of functions can be obtained easily by direct differentiation. Note that the alternative formulations do not require the equations of motion in Eq. (1) to be satisfied exactly at each optimization iteration. They need to be satisfied only at the final solution point. This has an advantage for the alternative formulations because they avoid possible instabilities or failure of the integration process for the equations of motion. Also, unnecessary simulations of the system are avoided at intermediate designs, where it might be difficult to obtain a solution. However, it can be seen that the number of variables and constraints becomes very large in the alternative formulations and the size depends on the number of grid points. Large-scale NLP solution algorithms with sparse matrix capabilities are required to solve the alternative formulations efficiently. Some aspects of the alternative formulations will be discussed in details in Sec. IV, when particular design problems are presented.

IV. Numerical Examples

For numerical evaluation, the conventional and three alternative formulations presented in Sec. III are applied to solve two examples problems. The alternative formulations are solved using the sparse SQP algorithm in SNOPT,[24] whereas the conventional formulation is solved using the dense SQP solver in SNOPT because the problem is dense. SNOPT has an option to compute derivatives using a combined forward and central difference method. This option is also used to solve the example problems with the conventional formulation. Results of the examples are listed and compared. Advantages and disadvantages of the formulations are discussed.

For the conventional formulation, two subroutines in the International Mathematical and Statistical Library (IMSL) mathematical library[25] are used to integrate the first-order DEs in all of the examples. These are DIVPRK (Runge–Kutta–Verner method) and DIVPAG (Adams–Moulton method or backward differentiation method). These are very good methods that can also solve (DIVPAG) stiff DEs, which are encountered in some mechanical system applications. All of the foregoing three methods are tried for the example problems presented here, and only the most efficient one is selected for optimization. Note that these methods automatically adjust the step size internally to maintain stability and accuracy of the solution process. Therefore, the computational effort to integrate the equations of motion and sensitivity equations can be substantial. In contrast, the alternative formulations use a fixed time grid in the solution process. To have a fairer comparison of the performance of different formulations, a fixed time step option was tried for integration of the DEs. In some cases, the integration procedure failed to converge, causing the optimization process to terminate prematurely. Therefore, it is concluded that a reliable and robust DE integrator is crucial for success of the conventional formulation. In addition, the governing equations are sometimes stiff and include algebraic equations as side constraints. Therefore, it is better to use more general approaches for integration of equations of motion. For all of the results with the conventional formulation reported later, the time step size was allowed to be adjusted automatically within the integration routine.

A personal computer with 2.53-GHz processor and 512-MB RAM is used for running the programs and recording the CPU times.

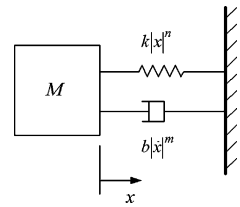

Fig. 1 Configuration of 1-DOF impact absorber.

Each solution case of the example problems was run several times with different starting points, and the shortest time was recorded. Also differences in the solution points, if any, were noted to observe global optimality of the solution.

A. Example 1: One-Degree-of-Freedom Impact Absorber

The problem is to design a nonlinear one-degree-of-freedom (1-DOF) shock absorber with nth-order stiffness and mth-order damping[1] (Fig. 1). The objective is to minimize the maximum acceleration of the attached mass during the transient response. The equation of motion is given as

$$M\ddot{x} + b|\dot{x}|^m \mathrm{sgn}\dot{x} + k|x|^n \mathrm{sgn} x = 0 \tag{20}$$

with initial conditions as $x(0) = 0$ and $\dot{x}(0) = V$ and $\mathrm{sgn} y = y/|y|$. Equation (20) is normalized by considering transformation of the displacement and the time parameter as $X = x/L$ and $\tau = Vt/L$, where L is the maximum displacement of the mass during the resulting transient motion. Therefore, the normalized equation of motion is

$$\ddot{X} + B|\dot{X}|^m \mathrm{sgn}\dot{X} + K|X|^n \mathrm{sgn} X = 0 \tag{21}$$

where the overdots now represent differentiation with respect to τ, the initial conditions become$X(0) = 0$ and $\dot{X}(0) = 1$, and the two new parameters are defined as

$$B = bLV^{m-2}/M, \qquad K = kL^{n+1}/MV^2 \tag{22}$$

The optimal design problem is to find B and K, representing the mass, stiffness, and damping properties of the system to

$$\underset{B,K}{\text{minimize}}\ |\ddot{X}|_{\max} \text{ subject to } X_{\max} \leq 1 \tag{23}$$

When an artificial variable E is introduced and the normalized time interval T of 2 ($0 \leq \tau \leq 2$) is discretized into N steps ($\Delta t = T/N$), the conventional formulation for the optimal design problem is to find three variables E, B, and K to minimize E subject to

$$|\ddot{X}_i| \leq E, \qquad i = 0, N \tag{24}$$

$$X_i \leq 1, \qquad i = 0, N \tag{25}$$

The numbers of optimization variables, constraints (excluding simple bound constraints), and nonzero elements in the constraint Jacobian are listed in Table 2. The initial values of the variables B, K, and E are taken as unity for the conventional formulation and all three alternative formulations. No special techniques are used to find the starting values of other variables for the three alternative formulations. The starting values of displacements, velocities and accelerations are all taken as unity. Note also that the inequality constraints in Eq. (25) become simple bound constraints in the alternative formulations that can be treated efficiently in the optimization algorithms.

The optimum solutions obtained by all of the formulations are quite similar to those obtained in the literature[1] and are plotted in Fig. 2. Figures 2a and 2b show the optimum values for nondimensional damping and stiffness parameters B and K for various values of m and n between zero and four. Figure 2c shows the maximum

Table 2 Size of design example 1

Item	Conventional	Alternate 1	Alternate 2	Alternate 3
Number of variables	3	$N+5$	$2N+7$	$3N+7$
Number of constraints	$2N+2$	$3N+3$	$4N+4$	$5N+4$
Number of nonzero elements in Jacobian	$6N+6$	$13N+7$	$15N+18$	$16N+13$
Total number of elements in Jacobian	$6N+6$	$3N^2+18N+15$	$8N^2+36N+28$	$15N^2+47N+28$

Table 3 Optimum designs for examples

1 DOF ($N=50$)				5 DOF ($N=300$)				
Variable	Conventional	Alternates 1 and 3	Alternate 2	Variable	Conventional (Ref. 2)	Conventional	Alternate 1	Alternates 2 and 3
E	0.59725	0.59752	0.59735	E	257.40	254.56	254.69	254.38
B	0.59725	0.59752	0.59735	k_1	50.00	50.00	50.00	50.00
K	0.59725	0.59752	0.59735	k_2	200.00	200.00	200.00	200.00
				k_3	241.90	200.00	200.00	200.00
				c_1	12.89	45.45	19.97	45.23
				c_2	77.52	77.35	76.99	77.39
				c_3	80.00	80.00	80.00	80.00

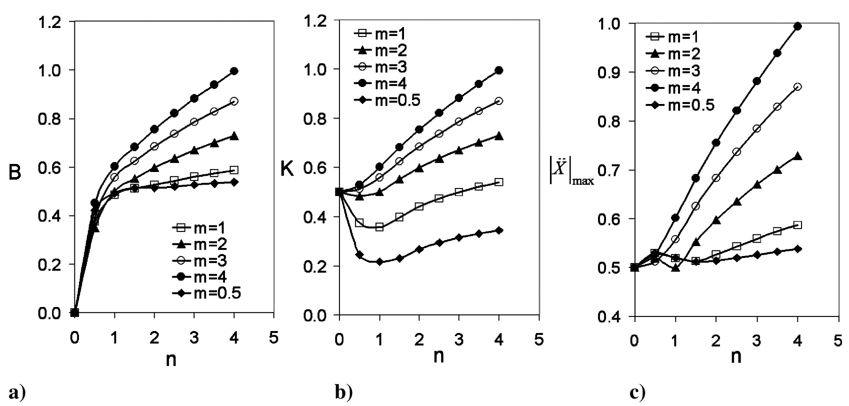

Fig. 2 Example 1: optimum values for a) nondimensional damping B and b) stiffness K and c) maximum acceleration magnitude $|\ddot{X}|_{max}$ by using optimum B and K.

acceleration magnitude $|\ddot{X}|_{max}$ for the optimum values of B and K. The optimum solutions corresponding to the nonlinear system with m and n taken as 2 and the number of grid points as 50 are listed in Table 3.

B. Example 2: 5-DOF Vehicle Suspension System

A 5-DOF vehicle suspension system[2] is shown in Fig. 3 (The state variables z_i are defined there.) The objective is to minimize the extreme acceleration of the driver's seat (mass m_1) for a variety of vehicle speeds and road conditions defined by the functions $f_1(t)$ and $f_2(t)$. The design variables are the spring constants k_1, k_2, and k_3 and the damping constants c_1, c_2, and c_3. The motion of the vehicle is also constrained so that the relative displacements between the chassis, and the driver's seat, the chassis, and the front and rear axles are within given limits. This is the design problem 1 of example 5.3 on pages 348–354 in Ref. 2. The equations of motion for the system are given there.

The optimum design problem is to find k_1, k_2, k_3, c_1, c_2, and c_3 that minimize $\max_{t\in[0,T]} |\ddot{z}_1(t)|$ for the given road profile 1 (Ref. 2). By the introduction of an artificial variable E, the reformulated problem is to minimize E subject to the state equations and the following inequality constraints in the time interval $[0, T]$:

$$|\ddot{z}_1(t)| \le E \tag{26}$$

$$|z_2(t) + (L/12)z_3(t) - z_1(t)| \le 2, \qquad 0 \le t \le T \tag{27}$$

$$|z_4(t) - z_2(t) - (L/3)z_3(t)| \le 5, \qquad 0 \le t \le T \tag{28}$$

$$|z_5(t) - z_2(t) + (2L/3)z_3(t)| \le 5, \qquad 0 \le t \le T \tag{29}$$

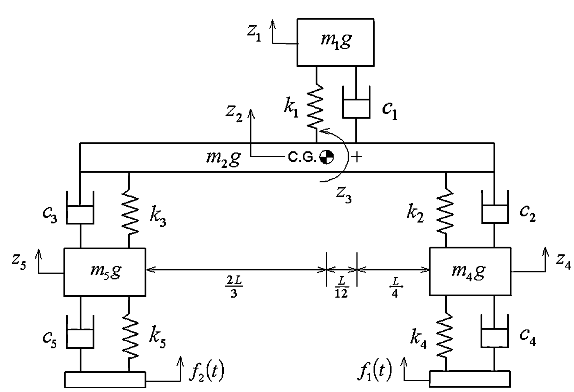

Fig. 3 Vehicle suspension system with 5 DOF.

$$|z_4(t) - f_1(t)| \le 2, \qquad 0 \le t \le T \tag{30}$$

$$|z_5(t) - f_2(t)| \le 2, \qquad 0 \le t \le T \tag{31}$$

and simple bounds on the design variables. According to the literature,[2] the initial design, lower and upper bounds for the design variable $[k_1, k_2, k_3, c_1, c_2, c_3, E]$ are taken as [100, 300, 300, 10, 25, 25, 332.6], [50, 200, 200, 2, 5, 5, 1], and [500, 1000, 1000, 50, 80, 80, 500], respectively. The total time interval is considered as 2.5 s, and the number of time steps is set to 300. No special techniques are used to find an initial point for the alternative formulations. The starting values for the displacements, velocities, and accelerations

Table 4 Size of design example 2

Item	Conventional	Alternate 1	Alternate 2	Alternate 3
Number of variables	7	$5N+22$	$15N+37$	$15N+32$
Number of constraints	$6N+6$	$10N+15$	$20N+25$	$20N+20$
Number of nonzero elements in Jacobian	$42N+42$	$92N+102$	$111N+126$	$109N+109$
Total number of elements in Jacobian	$42N+42$	$50N^2+295N+330$	$300N^2+1115N+925$	$300N^2+940N+640$

Table 5 Final objective results of design examples

Number of time intervals, N	Conventional	Alternate 1	Alternate 2	Alternate 3
		1 DOF		
50	0.59725	0.59752	0.59735	0.59752
300	0.59725	0.59726	0.59726	0.59726
500	0.59725	0.59725	0.59725	0.59725
		5 DOF		
100	252.81	251.24	252.57	251.24
300	254.56	254.69	254.38	254.38
500	254.71	254.74	254.96	254.64

Table 6 Numbers of iterations for design examples

Number of time intervals, N	Conventional	Alternate 1	Alternate 2	Alternate 3
		1 DOF		
50	4	5	16	6
300	4	8	15	3
500	4	9	15	4
		5 DOF		
100	23	23	10	24
300	29	27	23	20
500	22	18	23	38

are taken as zero. Note that in all of the alternative formulations Eq. (26) is treated as a pair of linear inequalities, Eqs. (27–29) become linear constraints, and Eqs. (30) and (31) become simple bounds on the variables. Table 4 lists the sizes of the problems for different formulations.

Table 3 gives the optimum solutions with different formulations. It is seen that alternative formulations 2 and 3 find the same optimum design, whereas alternate formulation 1 and the conventional formulation converge to a slightly different solution. The optimum solution obtained here is slightly better than that available in the literature.

C. Discussion of Results

1. *Number of Time Steps*

It is obvious that the number of time steps used in the numerical solution process can affect the final solution and performance of the formulations. If the step size is too large, the time-dependent constraints may have larger violation between the grid points and the optimal solution will not be accurate. If the step size is too small, the sizes of the alternative formulations become very large, which requires additional calculations and computer storage. To evaluate the performance of various formulations, a few different grid sizes are tried for the two examples and various results are summarized in Tables 5–7. Table 5 lists the final objective function values, Table 6 gives the numbers of iterations, and Table 7 gives the CPU efforts for different grid sizes. Table 5 shows that all of the formulations converged essentially to the same optimum solution for example 1 with $N=50$, 300, and 500. For example 2, all of the formulations converged to a slightly lower objective function value for $N=100$ compared to the solution with $N=300$, and 500. This indicates that there was some violation of the constraints between the grid points when $N=100$ was used. Table 8 shows that as the number of grid points is increased the computational effort with all of the formulations also increases, the increase being more dramatic for

Table 7 Computing efforts for different formulations, seconds

Number of time intervals, N	Conventional	Alternate 1	Alternate 2	Alternate 3
		1 DOF		
50	0.04	0.03	0.06	0.05
300	0.25	0.64	1.35	0.75
500	0.45	1.62	2.11	2.00
		5 DOF		
100	13.62	4.30	4.80	10.99
300	23.19	12.00	45.99	49.31
500	17.77	74.55	171.26	117.72

the alternative formulations. It is also observed that, for example 2, alternate 1 is about three times faster with $N=100$ and about two times faster with $N=300$ compared to the conventional formulation. With $N=500$, the conclusion is reversed. Alternates 2 and 3 are more efficient than the conventional formulation only for $N=100$. In any case, all of the alternative formulations converged to an optimum solution, with alternate 1 requiring the least computational effort.

2. *Scaling of Variables*

Note that the alternative formulations include different types of variables, which have different orders of magnitudes. Therefore, scaling of some of the variables is necessary to reduce numerical difficulties. In alternates 2 and 3, velocity and acceleration variables are normalized by positive numbers. These normalizers are comparable to the velocity and acceleration limits. If the generalized displacements have large differences in their orders of magnitude, for example, rotations and translations, some variables, such as the rotations, may also need to be scaled. This is true for the rotational DOF in example 2. The current scaling option in SNOPT works well only if a good starting point is provided or the problem is not too nonlinear. Efficient automatic scaling procedures need to be developed and incorporated into the alternative formulations.

3. *Global Solution*

The 5-DOF example problem has been solved by starting from several different points to determine all of the local minimum points. All of the starting points converged to one of the two objective function values (254.69 and 254.38). These objective function values are almost the same but have slightly different design variable values, especially the damping parameter c_1 for the driver's seat. Apparently the optimum cost function is insensitive with respect to this parameter. The two solutions can be considered essentially as global optima for the problem.

4. *Advantages and Disadvantages of Formulations*

The advantages and disadvantages of the conventional and alternative formulations are listed in Table 8, and the differences between the alternative formulations are given in Table 9. Note that it is not necessary to include velocities and accelerations as variables to obtain explicit expressions for constraints and Jacobians. Their inclusion gives more optimization variables and constraints and requires more data storage; however, their inclusion simplifies the constraint expressions and computer implementations. Some constraints become linear or simple bounds on the variables, such as the constraints on the velocities and accelerations. The gradients of the linear constraints in the alternative formulations can be programmed

Table 8 Advantages and disadvantages of two formulations

Advantages	Disadvantages
Conventional	
Small optimization problems.	Equations of motion must be integrated at each iteration; a good DE integrator is needed.
Equations of motion are satisfied at each iteration; intermediate solutions are usable.	Constraints are implicit functions of the variables; design sensitivity analysis must be performed.
Error in the solution of DEs can be controlled.	Implementation is more tedious.
	The optimization problem is always dense.
Alternative	
Equations of motion are not integrated at each iteration; no DE integrator is needed.	Intermediate solutions are not usable.
Formulations are explicit in terms of variables; design sensitivity analysis is not needed.	Error in the state variables cannot be controlled.
Many constraints become linear or simple bounds on variables.	Optimization algorithms for large-scale problems must be used.
Jacobians and Hessians are sparse.	For efficiency, advantage of sparsity of the Jacobians and Hessians must be utilized.
Implementation is easier.	Scaling of optimization variables is needed.

Table 9 Advantages and disadvantages of alternative formulations

Advantages	Disadvantages
Alternate 1	
Fewer optimization variables and constraints.	Derivative calculation and implementation slightly more tedious.
	Denser Jacobians and Hessians.
Alternates 2 *and* 3	
Very sparse Jacobians and Hessians.	Larger numbers of variables and constraints.
Implementation very straightforward.	Larger number of nonzero elements in Jacobians.
Velocity or acceleration constraints become linear or simple bounds.	

independently and called only once in the solution process, resulting in fewer calculations.

V. Conclusions

Three alternate formulations for optimization of transient dynamic mechanical system were proposed and evaluated. Different state variables were treated as optimization variables in the formulations, that is, generalized displacements, velocities, and accelerations. Therefore, the analysis equations (DEs) could be treated as equality constraints in the optimization process. When more variables were introduced into the formulations, the forms of the constraints and their derivatives were changed. All functions of the formulations became explicit in terms of the optimization variables. Derivatives of the functions with respect to the variables were computed explicitly. The formulations were implemented with a sparse NLP code for evaluation. Different time steps were tested. The alternative formulations had more variables and constraints, although the constraints had a simpler form compared to the conventional formulation. Therefore, an optimization algorithm for large numbers of variables and constraints was used to solve the problem. Two example problems were solved extensively to study and evaluate the formulations. The solutions for the sample problems were also compared with those available in the literature.

Based on the current research, the following conclusions are drawn:

1) All of the proposed alternative formulations using simple finite difference approximations for the state variables worked well and obtained optimum solutions for the example problems.

2) The proposed alternate formulation 1 was more efficient than formulations 2 and 3.

3) The proposed alternate formulation 1 was competitive and more efficient than the conventional formulation for reasonable number of time grid points.

4) The alternative formulations have potential for further development to optimize practical dynamic systems.

The alternative formulations represent a fundamental shift in the way analysis and design optimization of dynamic systems are currently treated. This shift in paradigm needs to be further investigated by studying and evaluating 1) different forms for approximation of velocities and acceleration in terms of the displacements, 2) use of existing simulation software, and 3) other optimization algorithms for large-scale optimization problems.

Acknowledgment

Support for this research provided by the Virtual Soldier Research program at the University of Iowa is gratefully acknowledged.

References

[1]Afimiwala, K. A., and Mayne, R. W., "Optimum Design of an Impact Absorber," *Journal of Engineering for Industry*, Vol. 96, No. 1, 1974, pp. 124–130.

[2]Haug, E. J., and Arora, J. S., *Applied Optimum Design, Mechanical and Structural Systems*, Wiley, New York, 1979, pp. 348–354.

[3]Tseng, C. H., and Arora, J. S., "Optimum Design of Systems for Dynamics and Controls Using Sequential Quadratic Programming," *AIAA Journal*, Vol. 27, No. 12, 1989, pp. 1793–1800.

[4]Arora, J. S., "Optimization of Structures Subjected to Dynamic Loads," *Structural Dynamic Systems Computational Techniques and Optimization: Optimization Techniques*, edited by C. T. Leondes, Gordon and Breach Science, New York, 1999, pp. 1–72.

[5]Hsieh, C. C., and Arora, J. S., "Design Sensitivity Analysis and Optimization of Dynamic Response," *Applied Mathematics and Optimization*, Vol. 43, No. 2, 1984, pp. 195–219.

[6]Lim, O. K., and Arora, J. S., "Dynamic Response Optimization Using an Active Set RQP Algorithm," *International Journal for Numerical Methods in Engineering*, Vol. 24, No. 10, 1987, pp. 1827–1840.

[7]Kim, M.-S., and Choi, D.-H., "Min–Max Dynamic Response Optimization of Mechanical Systems Using Approximate Augmented Lagrangian," *International Journal for Numerical Methods in Engineering*, Vol. 43, No. 3, 1998, pp. 549–564.

[8]Grandhi, R. V., Haftka, R. T., and Watson, L. T., "Design-Oriented Identification of Critical Times in Transient Response," *AIAA Journal*, Vol. 24, No. 4, 1986, pp. 649–656.

[9]Kang, B. S., Choi, W. S., and Park, G. J., "Structural Optimization Under Equivalent Static Loads Transformed from Dynamic Loads Based on Displacement," *Computers and Structures*, Vol. 79, No. 2, 2001, pp. 145–154.

[10]Choi, W. S., and Park, G. J., "Structural Optimization Using Equivalent Static Loads at All Time Intervals," *Computational Methods in Applied Mechanics and Engineering*, Vol. 191, No. 19–20, 2002, pp. 2077–2094.

[11]Haftka, R. T., "Simultaneous Analysis and Design," *AIAA Journal*, Vol. 23, No. 7, 1985, pp. 1099–1103.

[12]Kirsch, U., and Rozvany, G. I. N., "Alternative Formulations of Structural Optimization," *Structural Optimization*, Vol. 7, No. 1–2, 1994, pp. 32–41.

[13]Orozco, C. E., and Ghattas, O. N., "Sparse Approach to Simultaneous Analysis and Design of Geometrically Nonlinear Structures," *AIAA Journal*, Vol. 30, No. 7, 1992, pp. 1877–1885.

[14]Arora, J. S., and Wang, Q., "Review of Formulations for Structural and Mechanical System Optimization," *Structural and Multidisciplinary Optimization* (to be published).

[15]Betts, J. T., and Huffman, W. P., "Application of Sparse Nonlinear Programming to Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, 1993, pp. 198–206.

[16]von Stryk, O., and Bulirsch, R., "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, Vol. 37, No. 1–4, 1992, pp. 357–373.

[17]Cervantes, A., and Biegler, L. T., "Large-Scale DAE Optimization Using a Simultaneous NLP Formulation," *AIChE Journal*, Vol. 44, No. 5, 1998, pp. 1038–1050.

[18]Betts, J. T., "Very Low-Thrust Trajectory Optimization Using a Direct SQP Method," *Journal of Computational and Applied Mathematics*, Vol. 120, No. 1, 2000, pp. 27–40.

[19]Cervantes, A. M., Wächter, A., Tütüncü, R. H., and Biegler, L. T., "A Reduced Space Interior Point Strategy for Optimization of Differential Algebraic Systems," *Computers and Chemical Engineering*, Vol. 24, No. 1, 2000, pp. 39–51.

[20]Barclay, A., Gill, P. E., and Rosen, J. B., "SQP Methods and Their Application to Numerical Optimal Control," Dept. of Mathematics, Rept NA 97-3, Univ. of California, San Diego, CA, 1997.

[21]Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[22]Bathe, K.-J., *Finite Element Procedures in Engineering Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1982.

[23]Atkinson, K. E., *An Introduction to Numerical Analysis*, Wiley, New York, 1988.

[24]Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[25]IMSL® Math/Library 3.0 Manual, Visual Numerical, Inc., Houston, TX, 1997.

A. Messac
Associate Editor