An Evaluation of Some Alternative Formulations for Transient Dynamic Response Optimization

Qian Wang [*] and Jasbir S. Arora[†]
Optimal Design Lab/CCAD, College of Engineering, The University of Iowa, Iowa City, IA 52242

Some alternative formulations for transient dynamic response optimization are described and analyzed. A critical review and comparative study of the formulations are presented. The key feature of the formulations is that the state variables, in addition to the real design variables, are treated as independent variables in the optimization process. The state variables include different combinations of generalized displacements, velocities and accelerations. Formulations based on different discretization techniques for first-order and second-order differential equations are presented. Finite difference, Newmark's method and some methods based on collocation are all discussed. Similar to the simultaneous analysis and design (SAND) approach used for optimization of static structures and the direct collocation/transcription method for optimal control, the state governing equations are treated as equality constraints. A major advantage of these formulations is that special design sensitivity analysis methods are no longer needed. However, the formulations have larger numbers of variables and constraints. Therefore sparsity of the problem functions must be exploited in all the calculations. Advantages and disadvantages of the formulations are discussed. The results of a numerical example and performance features of the formulations are compared.

I. Introduction

Transient dynamic response optimization problems involve integration of linear or nonlinear differential equations (DEs), which is usually time-consuming. The most common approach for optimization of such problems has been the one where only the design variables are treated as optimization variables[1-4]. All other response quantities, such as displacements, velocities, and accelerations, are implicit functions of the design variables. This optimization process, however, is difficult to use in practice. The main difficulty is that since the response quantities are implicit functions of the design variables, they require special methods for gradient evaluation[5,6]. The entire process is difficult to use with existing simulation software. Another interesting approach for transient dynamic optimization is the so-called equivalent static load method, where the problem is transferred to a quasi-static problem[7,8]. 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 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 above, alternative formulations have been developed[9]. By formulating the optimization problem in a mixed space of design and state variables, explicit solution of DEs at each iteration is no longer required and there is no need for design sensitivity analysis. This is similar to the simultaneous analysis and design (SAND)[10-12] method developed for static structural optimization. The simultaneous approach has also been used to solve optimal control problems, which is called the direct collocation or transcription method[13-19]. Detailed formulations for the optimal control problems can be found in Refs. 20 and 21.

One major application field for direct collocation method is the optimal trajectory design in aerospace engineering[13-17]. The procedure of simultaneous solution of DAEs and the optimization problems has also been applied to chemical process engineering[18,19]. Starting form a system of first order differential equations, polynomial approximations in the time domain for the state, control and algebraic variables is obtained. Bounds on the state

[*] Research Assistant; Student Member AIAA (qwang2@engineering.uiowa.edu).
[†] F. Wendell Miller Distinguished Professor of Engineering; Corresponding author (Jasbir-Arora@uiowa.edu); Senior Member AIAA.

variables are enforced at the time grid points, or at all collocation points in the time domain. The optimization variables are the state variables at the time-grid points, first derivatives of the state variables, and the control and the algebraic variables at all collocation points. The resulting nonlinear programming (NLP) are usually solved by a sequential quadratic programming (SQP) method[21-22], or an interior point method[19].

In this research, several simultaneous formulations are reviewed and compared for optimization of transient dynamics problems. Various state variables, such as the displacements, velocities and accelerations are treated as independent variables in the optimization process. The equations of motion are treated as equality constraints. All constraints in these formulations are expressed explicitly in terms of the optimization variables. Therefore their gradient evaluations become quite simple. Although the resulting optimization problem is large but it is quite sparse which can be solved using sparse NLP algorithms. One numerical example is optimized and its solutions are compared with those available in the literature. Advantages and disadvantages of the formulations are discussed.

The present work describes and evaluates some simultaneous formulations for transient dynamic mechanical systems. The major contributions of the paper are as follows: (1) simultaneous formulations based on the SAND and direct collocation concepts are described, evaluated and compared, (2) some formulations based on the direct discretization of the second order form of the equations of motion (Newmark's method, Hermite interpolation) are proposed, which will facilitate use of the existing simulation software in the optimization process, and (3) a modern and powerful optimization algorithm and associated software are used that take full advantage of the sparsity structure of the simultaneous formulations.

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 (e.g., minimization of a cost function, such as the maximum displacement or acceleration) while satisfying all the performance requirements.

A. Governing Equations

Let $\mathbf{x}$ be an *m* dimensional vector to represent the design variables for the problem, which may directly include the mass, stiffness and damping parameters of the dynamic system, and $\mathbf{z}$ be a *d* dimensional vector that represents the state variables for the problem. The equation of motion for a linear system are written as follows (nonlinear problem can be treated similarly):

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

with the initial conditions $\mathbf{z}(0)=\mathbf{z}_0$, and $\dot{\mathbf{z}}(0)=\dot{\mathbf{z}}_0$. Note that $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ represent system matrices $(d\times d)$. In mechanical systems, these are the generalized mass, stiffness and damping matrices. $\mathbf{F}(\mathbf{x},t)$ is the generalized force vector of dimension $d\times 1$. The equations of motion can be solved directly in the second order form[23,24] in Eq. (1), or by converting it to a first order form, the so-called the *state space representation* of Eq. (1), as:

$$\dot{\mathbf{y}}(t)=\mathbf{f}(\mathbf{y}(t),\mathbf{x},t)=\overline{\mathbf{K}}(\mathbf{x})\mathbf{y}(t)+\overline{\mathbf{F}}(\mathbf{x},t) \tag{2}$$

where $\mathbf{y}^T=[\mathbf{z} \quad \dot{\mathbf{z}}]$, $\overline{\mathbf{K}}$ is the system matrix $(2d\times 2d)$, and $\overline{\mathbf{F}}(\mathbf{x},t)$ is $2d\times 1$ vector given as:

$$\overline{\mathbf{K}}(\mathbf{x})=\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{M}^{-1}(\mathbf{x})\mathbf{K}(\mathbf{x}) & -\mathbf{M}^{-1}(\mathbf{x})\mathbf{C}(\mathbf{x})\end{bmatrix};\ \overline{\mathbf{F}}(\mathbf{x},t)=\begin{bmatrix}\mathbf{0}\\ \mathbf{M}^{-1}(\mathbf{x})\mathbf{F}(\mathbf{x},t)\end{bmatrix} \tag{3}$$

in which $\mathbf{I}$ is a $d\times d$ identity matrix. It is worthwhile to examine both the second and the first order forms of the dynamics equations.

B. Cost Function and Constraints

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

$$J=c_0(\mathbf{x},T)+\int_0^T h_0(\mathbf{x},\mathbf{z},\dot{\mathbf{z}},\ddot{\mathbf{z}},t)\,dt \tag{4}$$

where *T* is the total time interval considered. The objective *J* may be the cost of the system, a performance measure, or any other function of the state variables. A time-dependent functional, such as maximum acceleration or displacement, can also be treated as an objective.

Design requirements are imposed as inequality constraints. One type of constraints is not dependent on time, which is treated easily in the optimization process. The other type of constraints is the so-called *point-wise constraint*, which needs to be satisfied at each point of the entire time interval $t\in[0,T]$:

$$\mathbf{g}(\mathbf{x},\mathbf{z},\dot{\mathbf{z}},\ddot{\mathbf{z}},t)\le\mathbf{0} \tag{5}$$

Five treatments of the point-wise constraints have been discussed in the literature[5]. In this study, the conventional treatment is used where the time interval is divided into N subintervals (N+1 time grid points) and the constraints are imposed at each time grid point:

$$\mathbf{g}_i(\mathbf{x}, \mathbf{z}_i, \dot{\mathbf{z}}_i, \ddot{\mathbf{z}}, t_i) \le \mathbf{0}, \qquad i = 0, N \tag{6}$$

The length of the subinterval is Δt, and is defined as $\Delta t = T / N$. In dynamic systems, the time-dependent inequality constraints in Eq. (5) may include the displacement, velocity and acceleration constraints. Some time-independent constraints include the lower and upper bounds for the design variables.

If the first-order form of the DEs is used, state space variables $\mathbf{y}$ in Eq. (2), i.e., displacements $\mathbf{z}$ and velocities $\dot{\mathbf{z}}$ are used in the constraint expressions. The problem is to determine $\mathbf{x}$ to minimize the cost function in Eq. (4), subject to the first order DEs in Eq. (2), and

$$\mathbf{g}(\mathbf{x}, \mathbf{y}, \dot{\mathbf{y}}, t) \le \mathbf{0} \tag{7}$$

Note that Eqs. (2) and (7) are discretized into N+1 equations in the entire time interval $[0, T]$, and they represent a system of constraints as follows:

$$\mathbf{g}_i(\mathbf{x}, \mathbf{y}_i, \dot{\mathbf{y}}_i, t_i) \le \mathbf{0}, \qquad i = 0, N \tag{8}$$

The optimization problem is to find **x** to minimize the cost functional in Eq. (4) subject to the constraints of Eqs. (1) and (5), or (2) and (6), and other time-independent constraints. If the problem is solved based on the conventional formulation, where the optimization is carried out only in the space of design variables, the equations of motion in (1) and (2) need to be integrated to determine the system response and thus calculate various functions of the optimization problem. Since the constraint functions in this formulation are implicit functions of the design variables, implicit differentiation procedures need to be used to evaluate the gradients. Analytically, the direct differentiation or the adjoint variable method can be used[5,6]. 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 gradients.

III. Simultaneous Formulations Based on First Order DEs

In this section, simultaneous formulations based on the direct collocation/transcription method of optimal control are presented and discussed. In these formulations, the system of first order DEs is directly discretized and imbedded into the optimization formulation. Starting form a system of first order differential equations, approximations in the time domain for the state variables are set up and collocation is enforced at certain time points. Direct collocation/transcription methods are used often in trajectory optimization, chemical process engineering, and robotic motion planning.

These formulations are based on different discretization and integration techniques. The first order state differential equations in Eq. (2) are approximated within each segment using an integration formula. The approximate integration formulation is then transformed to a set of algebra equations, the so-called defect equations, which need to be set to zero to enforce the DEs. In this section, the following notations are used:

$$\mathbf{y}_{c,i} = \mathbf{y}(t_{c,i}); \qquad \mathbf{f}_{c,i} = \mathbf{f}(\mathbf{y}_{c,i}, \mathbf{x}, t_{c,i}) \tag{9}$$

where $t_{c,i} = (t_{i-1} + t_i)/2$, ($i$ = 1, N). $t_{c,i}$, $\mathbf{y}_{c,i}$ and $\mathbf{f}_{c,i}$ are the center of the time interval and the corresponding state variables and derivatives.

A. Simultaneous Formulation Based on Trapezoidal Discretization (TR)

This formulation is based on the trapezoidal rule of numerical integration[25]. The general form of integration from time t_{i-1} to t_i is given as

$$\mathbf{y}_i = \mathbf{y}_{i-1} + \int_{t_{i-1}}^{t_i} \dot{\mathbf{y}} dt = \mathbf{y}_{i-1} + \int_{t_{i-1}}^{t_i} \mathbf{f} dt \tag{10}$$

If the trapezoidal approximation is considered, the defect equations can be set up[16] as

$$\boldsymbol{\zeta}_i = \mathbf{y}_i - \mathbf{y}_{i-1} - \frac{\Delta t}{2}(\mathbf{f}_i + \mathbf{f}_{i-1}), \qquad i = 1, N \tag{11}$$

where

$$\mathbf{f}_i = \mathbf{f}(\mathbf{y}(t_i), \mathbf{x}, t_i) \tag{12}$$

In this formulation, the state variables $\mathbf{y}_i$ (i = 0, N) are chosen as the optimization variables. The defect equation in Eq. (12) needs to be satisfied to enforce the DEs; therefore, this formulation is to minimize the objective function in Eq. (4), subject to the equality constraints as

$$\mathbf{y}_i - \mathbf{y}_{i-1} - \frac{\Delta t}{2}\left(\mathbf{f}_i + \mathbf{f}_{i-1}\right) = \mathbf{0}\,, \qquad i = 1, N \tag{13}$$

and the discretized form of inequality constraints in Eq. (8). Note that time-dependent constraints on the state variable $\mathbf{y}$ can be imposed on the discrete time grid points, i.e., $\mathbf{y}_i$ (i = 0, N). This formulation is based on a linear approximation of the state gradient $\mathbf{f}$, and therefore a quadratic approximation of the state variables $\mathbf{y}$ in each time subinterval. It can also be derived from the forward difference approximation of the average velocities and accelerations in each time subinterval.

B. Simultaneous Formulation Based on Compressed Hermite-Simpson Discretization (CHS)

If the Simpson rule is used in the numerical integration in Eq. (10), the defect equations[13,17] are

$$\zeta_i = \mathbf{y}_i - \mathbf{y}_{i-1} - \frac{\Delta t}{6}\left(\mathbf{f}_i + 4\mathbf{f}_{c,i} + \mathbf{f}_{i-1}\right), \qquad i = 1, N \tag{14}$$

where $\mathbf{f}_{c,i}$ and $\mathbf{f}_i$ are defined in Eqs. (9) and (12), respectively. And

$$\mathbf{y}_{c,i} = \frac{1}{2}\left(\mathbf{y}_{i-1} + \mathbf{y}_i\right) + \frac{\Delta t}{8}\left(\mathbf{f}_{i-1} - \mathbf{f}_i\right) \tag{15}$$

The state variables $\mathbf{y}_i$ (i = 0, N) are chosen as the optimization variables. This formulation is to minimize the objective function in Eq. (4), subject to the inequality constraints in Eq. (8), and the equality constraints (approximated state equation) as

$$\mathbf{y}_i - \mathbf{y}_{i-1} - \frac{\Delta t}{6}\left(\mathbf{f}_i + 4\mathbf{f}_{c,i} + \mathbf{f}_{i-1}\right) = \mathbf{0}\,, \qquad i = 1, N \tag{16}$$

Note that time-dependent constraints on the state variables can be imposed at the discrete time grid points, or both the discrete time grid points and the center of each time subintervals. In this formulation, the state variables $\mathbf{y}$ are chosen as continuous differentiable functions and defined piecewise as cubic polynomials between $\mathbf{y}_{i-1}$ and $\mathbf{y}_i$. This formulation can also be derived by collocating the first derivatives as equality constraints at the center point of each subinterval[13,17].

C. Simultaneous Formulation Based on Separated Hermite-Simpson Discretization (SHS)

An alternate form can be obtained, if both the state variables $\mathbf{y}_i$ (*i = 0, N*) and $\mathbf{y}_{c,i}$ (*i = 0, N-1*) are treated as optimization variables[26]. In this case, the equality constraints are in Eq. (16) and

$$\mathbf{y}_{c,i} - \frac{1}{2}\left(\mathbf{y}_{i-1} + \mathbf{y}_i\right) - \frac{\Delta t}{8}\left(\mathbf{f}_{i-1} - \mathbf{f}_i\right) = \mathbf{0}\,, \qquad i = 1, N \tag{17}$$

where $\mathbf{f}_{c,i}$ and $\mathbf{f}_i$ are defined in Eqs. (9) and (12), respectively. Equation (17) contains additional constraints that need to be satisfied. The time-dependent constraints on the state variables can be imposed at the discrete time grid points, or both the discrete time grid points and the center of each time subintervals.

IV. Simultaneous Formulations Based on Second Order DEs

Although a large number of accurate, high-order or multi-step numerical methods are available to solve the system of first order DEs[25], the analysis of practical structural and mechanical dynamics also heavily rely on those single-step direct time integration methods that solve the second order DEs directly[23,24]. These methods are usually more efficient, and they are widely used in many commercial codes. Therefore, the development of simultaneous formulations combined with the second order form of the DEs will bring benefit to use these available codes directly. In the next section, some of these formulations are presented, and they are based on Newmark's method, central difference method and Hermite interpolation.

A. Simultaneous Formulations Based on Newmark's Method (Newmark)

1. *The Newmark's method*

The Newmark's method is one of the most popular single step direct integration methods for second order DEs. Based on the assumption that the acceleration is linear within each time step, the following equations can be set up:

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \Delta t\dot{\mathbf{z}}_i + \left(\tfrac{1}{2} - \beta\right)\Delta t^2\ddot{\mathbf{z}}_i + \beta\Delta t^2\ddot{\mathbf{z}}_{i+1} \tag{18}$$

$$\dot{\mathbf{z}}_{i+1} = \dot{\mathbf{z}}_i + (1-\gamma)\Delta t \ddot{\mathbf{z}}_i + \gamma \Delta t \ddot{\mathbf{z}}_{i+1} \tag{19}$$

Two widely used cases of Eqs. (18) and (19) are as follows:

$\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$, called the average acceleration method.

$\gamma = \frac{1}{2}$ and $\beta = \frac{1}{6}$, called the linear acceleration method.

Note that when $\gamma = \frac{1}{2}$ and $\beta = 0$, Eqs. (18) and (19) can be reduced to very simple form, and this is in fact the central difference method, also the constant acceleration method. For general cases when $\beta \neq 0$, it is not easy to use the displacements to express the velocities and accelerations. When $\beta = 0$, it is possible to express the velocities and accelerations with respect to the displacements in a simple form. The central difference method will be discussed in next section. Two possible simultaneous formulations based on Newmark's method in Eqs. (18) and (19) are discussed in this section.

2. *Design variables, displacements and velocities as optimization variables (Newmark-1)*

If displacements and velocities are treated as optimization variables, accelerations can be expressed in terms of the displacements and velocities. Therefore, an explicit form can be obtained for the dynamic optimization problem. The problem is to determine $\mathbf{x}$, $\mathbf{z}$ and $\dot{\mathbf{z}}$ (*i = -1, N+1*) to minimize the cost function of Eq. (4), subject to the state equations (1), and constraint in Eq. (6). More equality constraints are needed due to the introduction of additional variables. Rearranging the Newmark's equations in Eqs. (18) and (19), the accelerations are expressed as follows:

$$\ddot{\mathbf{z}}_i = \frac{1}{\varphi}\left[-\gamma \mathbf{z}_i + \gamma \mathbf{z}_{i+1} - (\gamma - \beta)\Delta t \dot{\mathbf{z}}_i - \beta \Delta t \dot{\mathbf{z}}_{i+1}\right] \tag{20}$$

$$\ddot{\mathbf{z}}_{i+1} = \frac{1}{\varphi}\left[(1-\gamma)\mathbf{z}_i - (1-\gamma)\mathbf{z}_{i+1} + \left(\tfrac{1}{2} - \gamma + \beta\right)\Delta t \dot{\mathbf{z}}_i + \left(\tfrac{1}{2} - \beta\right)\Delta t \dot{\mathbf{z}}_{i+1}\right] \tag{21}$$

where $\varphi = \left(\frac{1}{2}\gamma - \beta\right)\Delta t^2$. The equality constraints among the variables $\mathbf{z}_i$, and $\dot{\mathbf{z}}_i$ can be constructed by substituting Eqs. (20) or (21) into (19) or (20), as

$$(1-\gamma)\mathbf{z}_i + (2\gamma - 1)\mathbf{z}_{i+1} - \gamma \mathbf{z}_{i+2} + \left(\tfrac{1}{2} - \gamma + \beta\right)\Delta t \dot{\mathbf{z}}_i + \left(\tfrac{1}{2} - 2\beta + \gamma\right)\Delta t \dot{\mathbf{z}}_{i+1} + \beta \Delta t \dot{\mathbf{z}}_{i+2} = \mathbf{0} \tag{22}$$

Note that in Eqs. (20) and (21), $\frac{1}{2}\gamma - \beta \neq 0$. This means that the average acceleration method where $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$, is not applicable here. The basic idea of the formulations based on the direct discretization of second order DEs is that the system of DEs is discretized as equality constraints. However, it is not possible to express $\ddot{\mathbf{z}}_i$ in terms of $\mathbf{z}_i$ and $\dot{\mathbf{z}}_i$, from Newmark's equations in Eqs. (18) and (19). Therefore, it is not easy to express and evaluate the equality equations in Eq. (1).

3. *Design variables, displacements, velocities and accelerations as optimization variables (Newmark-2)*

From Eq. (1), it is seen that other simultaneous formulations are possible. If the displacements $\mathbf{z}$, velocities $\dot{\mathbf{z}}$ and accelerations $\ddot{\mathbf{z}}$ are treated as variables simultaneously, another explicit form is obtained. The problem is to determine $\mathbf{x}$, $\mathbf{z}$ (*i = -1, N+1*), $\dot{\mathbf{z}}$ and $\ddot{\mathbf{z}}$ (*i = 0, N*) to minimize the cost function of Eq. (4), subject to the equations of motion (1), and discretized constraints in Eq. (6). It is obvious that the equality constraints in Eq. (1) are explicit with respect to the optimization variables $\mathbf{x}$, $\mathbf{z}$, $\dot{\mathbf{z}}$ and $\ddot{\mathbf{z}}$. However, more equality constraints are needed, since the state variables are related to each other by the Newmark's equations in Eqs. (18) and (19). These need to be imposed in the formulation.

B. Simultaneous Formulations Based on Central Difference Method (CD)

Three simultaneous formulations based on the central difference approximation have been presented and evaluated in Ref. 9. Details of these three formulations are not presented here. However, these formulations include displacement only (CD-1), displacement and velocity (CD-2), and displacement, velocity and acceleration (CD-3) as additional optimization variables, respectively. Finite difference relationships are used as equality constraints in the formulations, which makes the implementation of the algorithm quite simple, as

$$\dot{\mathbf{z}}_i = \dot{\mathbf{z}}_i(t_i) = \frac{\mathbf{z}_{i+1} - \mathbf{z}_{i-1}}{2\Delta t}, \qquad i = 0, N \tag{23}$$

$$\ddot{\mathbf{z}}_i = \ddot{\mathbf{z}}_i(t_i) = \frac{\mathbf{z}_{i+1} - 2\mathbf{z}_i + \mathbf{z}_{i-1}}{\Delta t^2}, \qquad i = 0, N \tag{24}$$

C. Simultaneous Formulation Based on Piecewise Hermite interpolation (Hermite)

The basic discretization scheme is as follows: the state variables $\mathbf{z}$ are chosen as continuous differentiable functions and defined as piecewise cubic polynomials between $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$, with the state equations (1) satisfied at t_i and t_{i+1}. For $t \in [t_i, t_{i+1}]$, using a parameter $u \in [0,1]$, such that $t = t_i + u\Delta t$, the approximation of state variables $\mathbf{z}$ is

$$\mathbf{z}_a(t) = \sum_{k=0}^{3} \mathbf{c}_k^i u^k \tag{25}$$

where

$$\begin{aligned} \mathbf{c}_0^i &= \mathbf{z}_i; & \mathbf{c}_1^i &= \Delta t \dot{\mathbf{z}}_i \\ \mathbf{c}_2^i &= -3\mathbf{z}_i - 2\Delta t \dot{\mathbf{z}}_i + 3\mathbf{z}_{i+1} - \Delta t \dot{\mathbf{z}}_{i+1}; & \mathbf{c}_3^i &= 2\mathbf{z}_i + \Delta t \dot{\mathbf{z}}_i - 2\mathbf{z}_{i+1} + \Delta t \dot{\mathbf{z}}_{i+1} \end{aligned} \tag{26}$$

where $\Delta t = t_{i+1} - t_i$. The approximation function (25) of the state variables must satisfy the DEs (1) at the grid point t_i (*i = 0, N*). The time dependent constraints are satisfied at the grid points, and at the centers $t_{c,i} = (t_i + t_{i+1})/2$ (*i = 0, N-1*).

The optimization problem is to determine $\mathbf{x}$, $\mathbf{z}$, and $\dot{\mathbf{z}}$ to minimize the cost function of Eq. (4), subject to the equality constraints in Eq. (1) at the discrete time grid points, and the discretized inequality constraints in Eq. (6) at the time grid point and the middle of each point. From Eq. (25),

$$\dot{\mathbf{z}}_a(t) = \sum_{k=1}^{3} \frac{k}{\Delta t} \mathbf{c}_k^i u^{k-1}; \qquad \ddot{\mathbf{z}}_a(t) = \sum_{k=2}^{3} \frac{k(k-1)}{\Delta t^2} \mathbf{c}_k^i u^{k-2} \tag{27}$$

at the grid point $t = t_i$,

$$\ddot{\mathbf{z}}_i = \ddot{\mathbf{z}}_a(t_i) = \frac{2}{\Delta t^2}\left(-3\mathbf{z}_i + 3\mathbf{z}_{i+1} - 2\Delta t \dot{\mathbf{z}}_i - \Delta t \dot{\mathbf{z}}_{i+1}\right) \tag{28}$$

A system of equality constraints is obtained at each time grid point, as

$$\mathbf{M}(\mathbf{x})\left[\frac{2}{\Delta t^2}\left(-3\mathbf{z}_i + 3\mathbf{z}_{i+1} - 2\Delta t \dot{\mathbf{z}}_i - \Delta t \dot{\mathbf{z}}_{i+1}\right)\right] + \mathbf{C}(\mathbf{x})\dot{\mathbf{z}}_i + \mathbf{K}(\mathbf{x})\mathbf{z}_i = \mathbf{F}_i(\mathbf{x}, t_i), \qquad i = 0, N \tag{29}$$

The relationships between the displacement and velocity variables are obtained from Eq. (28), as

$$3\mathbf{z}_i - 3\mathbf{z}_{i+2} + \Delta t\left(\dot{\mathbf{z}}_i + 4\dot{\mathbf{z}}_{i+1} + \dot{\mathbf{z}}_{i+2}\right) = \mathbf{0} \tag{30}$$

D. Simultaneous Formulation Based on Cubic B-Spline Interpolation (BSpline)

If the final state variable history needs to be very smooth, cubic B-spline interpolation (Mortenson 1985) can be used. B-splines have many important properties such as continuity, differentiability, and local control. There are a number of ways to define the B-spline basis functions, and here the uniform B-spline is used. Let $T = \{t_0, t_1, .., t_m\}$ be a non-decreasing sequence of real numbers, i.e., $t_i \le t_{i+1},\ i = 0, .., m\text{-}1$. The t_i are called *knots*, and they are evenly spaced for a uniform B-spline. A cubic B-spline is defined as

$$z(t) = \sum_{j=0}^{n} N_{j,4}(t) P_j; \qquad 0 \le t \le T \tag{31}$$

where the $\{P_j\},\ j = 0, .., n$ are the $(n+1)$ *control points*, and the $\{N_{j,4}(t)\}$ are the cubic B-spline basis functions defined on the non-periodic knot vector ($(m+1)$ knots). Using a parameter $u \in [0,1]$ defined as such that $t = t_i + u\Delta t$, the basis functions of a cubic B-spline are as follows,

$$\begin{aligned} N_{j,4}(u) &= \tfrac{1}{6}\left(-u^3 + 3u^2 - 3u + 1\right); & N_{j+1,4}(u) &= \tfrac{1}{6}\left(3u^3 - 6u^2 + 4\right) \\ N_{j+2,4}(u) &= \tfrac{1}{6}\left(-3u^3 + 3u^2 + 3u + 1\right); & N_{j+3,4}(u) &= \tfrac{1}{6}\left(u^3\right) \end{aligned} \tag{32}$$

Since each segment of the curve is defined by four control points, for $t_i \le t_{i+1}$, Eq. (31) can be simplified as,

$$z(u) = N_{i,4}(u)P_i + N_{i+1,4}(u)P_{i+1} + N_{i+2,4}(u)P_{i+2} + N_{i+3,4}(u)P_{i+3} \tag{33}$$

Since the first and second order derivatives of the displacement are needed in the dynamic optimization, the *k*th derivatives of a cubic B-spine curve can be easily obtained from Eq. (32), since only the basis functions are functions of time.

In this formulation, the control vector $\mathbf{P}$ for each DOF are chosen as the optimization variables. This formulation is to minimize the objective function in Eq. (4), subject to the inequality constraints in Eq. (5), as

$$\mathbf{g}(\mathbf{x}, \mathbf{P}, t) \le \mathbf{0} \tag{34}$$

Note that the equilibrium equations in Eq. (1) can be expressed by $\mathbf{P}$ similarly and treated as equality constraints in the formulation. The time-dependent constraints on the state variables can be imposed on the discrete time grid points.

V. Evaluation of Formulations

Table 1 shows the sizes of all the formulations. The following symbols are used in the table: m = number of elements in the design variable vector $\mathbf{x}$; N = number of time intervals (number of grid points = N + 1); n = number of control points in a cubic B-spline; d = number of DEs in Eq. (1), or the dimension of vector $\mathbf{z}$; e = number of constraints in $\mathbf{g}$; however, some of the inequality constraints in some alternative formulations become simple bounds on the variables.

Table 1. Number of variables and constraints for different formulations

Formulations		Variables	No. of Variables	No. of Equality Constraints	No. of Inequality Constraints
Conventional		$\mathbf{x}$	*m*	*0*	*e(N+1)*
1st order DEs	**TR**	$\mathbf{x}, \mathbf{y}$	*m +2dN+2d*	*2dN*	*e(N+1)*
	CHS	$\mathbf{x}, \mathbf{y}, \mathbf{y}_c$	*m +2dN+2d*	*2dN*	*e(2N+1)*
	SHS	$\mathbf{x}, \mathbf{z}, \dot{\mathbf{z}}$	*m +4dN+2d*	*4dN*	*e(2N+1)*
2nd order DEs	**Newmark-1**	$\mathbf{x}, \mathbf{z}, \dot{\mathbf{z}}$	*m+2dN+6d*	*2dN+2d*	*e(N+1)*
	Newmark-2	$\mathbf{x}, \mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}$	*m+ 3dN+9d*	*3dN+5d*	*e(N+1)*
	CD-1	$\mathbf{x}, \mathbf{z}$	*m+dN+3d*	*dN+d*	*e(N+1)*
	CD-2	$\mathbf{x}, \mathbf{z}, \dot{\mathbf{z}}$	*m+2dN+6d*	*2dN+2d*	*e(N+1)*
	CD-3	$\mathbf{x}, \mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}$	*m+3dN+5d*	*3dN+3d*	*e(N+1)*
	Hermite	$\mathbf{x}, \mathbf{z}, \dot{\mathbf{z}}$	*m+2dN+4d*	*2dN+d*	*e(2N+1)*
	BSpline	$\mathbf{x}, \mathbf{P}$	*m+nd*	*dN+d*	*e(N+1)*

Table 2. Advantages and disadvantages of different formulations

Formulations		Advantages	Disadvantages
Conventional		• Smaller NLP problems. • DEs are satisfied at each iteration; intermediate solutions are usable. • Error in the solution of DEs can be controlled.	• Design sensitivity analysis must be performed. • Dense Jacobian and Hessian matrices. • DEs must be integrated.
1st order DEs	**TR**	• Implementation is straightforward. • Velocity constraints become linear or simple bounds.	• Larger number of variables and constraints. • Acceleration constraints become complex.
	HS	• Good smoothness • Smaller NLP problems • Velocity constraints become linear or simple bounds.	• Implementation is not straightforward. • Acceleration constraints become complex.
2nd order DEs	**CD**	• Very sparse Jacobians and Hessian. • Implementation is very straightforward. • Velocity or acceleration constraints become linear or simple bounds.	• Larger number of variables and constraints. • Larger number of non-zero elements in Jacobians.
	Newmark	• Very sparse Jacobians and Hessian. • Implementation is very straightforward. • Velocity or acceleration constraints become linear or simple bounds. • Very general	• Larger number of variables and constraints. • Larger number of non-zero elements in Jacobians.
	Hermite	• Good smoothness. • Smaller NLP problems • Velocity or acceleration constraints become linear or simple bounds.	• Implementation is not straightforward.
	BSpline	• Good smoothness. • Smaller NLP problems • Displacement, velocity or acceleration constraints are linear.	• Implementation is not straightforward. • Displacement, velocity or acceleration constraints are not simple bounds.

Since the objective and constraint functions are all explicit in terms of the optimization variables in the alternate formulations, the gradients of functions can be obtained easily by direct differentiation. Note that the alternate formulations do not require the equations of motion in Eq. (1) to be satisfied exactly at each iteration of the optimization process. It needs to be satisfied only at the final solution point. This has advantage if instabilities occur or no solution exists for certain designs in the design space. Also, unnecessary simulations of the system are avoided at intermediate designs, where it might be difficult to obtain a solution.

The differences between the simultaneous formulations are discussed in Table 2. For the simultaneous formulations based on the first order DEs, the number of time grid size is usually not very large; therefore, the resulting NLP may not need to be too large. Since most of the methods are based on polynomial interpolations between grid points, only a reasonable number of grid points is needed. Moreover, these methods can provide more accurate solution, provided that multi-step or higher order method are used. As can be seen that the defect equations (i.e., Eqs. (13) and (16)) involve repeated substitutions of the equations of motion (2), if multi-step methods of discretization or higher order polynomial interpolations are used. Thus implementation becomes tedious. Sparsity patterns of those formulations are hard to identify and define. For the constraints involving accelerations, the expressions for some cases become complex, since the equations of motion are embedded in the expressions for accelerations.

With the second order DEs, since the equations are directly discretized and treated as equality constraints, the complexity of the constraint equations is well defined. The implementation and sparsity pattern are very straightforward. For the finite difference-based methods, e.g., central difference or Newmark's methods, since a small time step is used to guarantee convergence and stability, the number of grid points N is usually very large, resulting in large numbers of variables and constraints in the formulations. Large-scale NLP solution algorithms with sparse matrix capabilities are required to solve the simultaneous formulations efficiently. These methods are more suitable for dynamic response that may not necessarily be smooth. The methods treat the acceleration constraints more efficiently. Some aspects of the simultaneous formulations are discussed in details in Section VI, when a design example is solved.

VI. Numerical Example

All the simultaneous formulations developed in Sections III and IV are applied to a dynamic mechanical example for evaluation. All the simultaneous formulations are solved using the sparse SQP algorithm in SNOPT[27], while the conventional formulation is solved using the dense SQP solver in SNOPT. A Dell PC with 2.53 GHz P4 processor and 1.0 GB RAM is used for running the programs and recording the relative CPU times. Each solution case of the example problem was run several times with different starting point and the shortest time was recorded. Results of the examples are listed and compared. Advantages and disadvantages of the formulations are discussed.

A. A 5-DOF Vehicle Suspension System

A 5-DOF vehicle suspension system[1] is shown in Figure 1, which is the design problem 1 of Example 5.3 on pages 348-354 in Ref. 1. 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 total time interval is considered as 2.5 seconds. 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.

The optimal 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[1]. By introducing 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 $[0,T]$:

$$|\ddot{z}_1(t)| \le E \tag{35}$$

$$\left|z_2(t)+\frac{L}{12}z_3(t)-z_1(t)\right| \le 2, \quad 0 \le t \le T \tag{36}$$

$$\left|z_4(t)-z_2(t)-\frac{L}{3}z_3(t)\right| \le 5, \quad 0 \le t \le T \tag{37}$$

$$\left|z_5(t)-z_2(t)+\frac{2L}{3}z_3(t)\right| \le 5, \quad 0 \le t \le T \tag{38}$$

$$\left| z_4(t) - f_1(t) \right| \le 2, \quad 0 \le t \le T \tag{39}$$

$$\left| z_5(t) - f_2(t) \right| \le 2, \quad 0 \le t \le T \tag{40}$$

and simple bounds on the design variables. The initial design, lower and upper bounds for the design variable vector $[k_1, k_2, k_3, c_1, c_2, c_3, E]$ 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[1]. No special techniques are used to find an initial point for the simultaneous formulations. The starting values for the displacements, velocities, and accelerations are taken as zero. Note that in all the simultaneous formulations, Eq. (35) is treated as a pair of linear inequalities, Eqs. (36) - (38) become linear constraints, and Eqs. (39) and (40) become simple bounds on the variables.

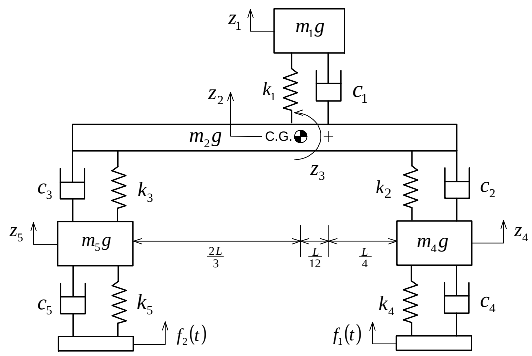

Figure 1. A 5-DOF vehicle suspension system

Table 3. Number of variables and constraints for different formulations

Formulations		No. of Variables	No. of Constraints	No. of Non-zero Elements in Jacobian (w. Sparsity)	No. of Non-zero Elements in Jacobian (w/o Sparsity)
Conventional		*7*	*6N+6*	*42N+42*	*42N+42*
1st order DEs	**TR**	*10N+17*	*15N+5*	*141N+27*	*150N²+305N+85*
	CHS	*10N+17*	*22N+5*	*349N+27*	*220N²+424N+85*
	SHS	*20N+17*	*30N+5*	*335N+27*	*600N²+610N+85*
2nd order DEs	**Newmark-1**	*10N+37*	*15N+15*	*115N+115*	*150N²+705N+555*
	Newmark-2	*15N+52*	*20N+30*	*119N+164*	*300N²+1490N+1560*
	CD-1	*5N+22*	*10N+15*	*92N+102*	*50N²+295N+330*
	CD-2	*15N+37*	*20N+25*	*111N+126*	*300N²+1115N+925*
	CD-3	*15N+32*	*20N+20*	*109N+109*	*300N²+940N+640*
	Hermite	*10N+27*	*22N+10*	*160N+85*	*220N²+694N+270*
	BSpline	*5N+22*	*12N+22*	*148N+178*	*60N²+374N+484*

B. Discussion of Results

1. Sizes of optimization problem

Table 3 lists the sizes of the problems for different formulations. For the conventional formulation, the size of the design problem is well defined. The number of variables is 7 in the design example. This is a quite small problem in terms of numerical optimization. In the alternative formulations, there are large numbers of variables, depending on the number of time grid points chosen. If the number of grid points is large, the resulting optimization problem may include hundreds and even thousands of variables, which may make the problem ill-conditioned sometimes. This is the major disadvantage of these alternative formulations. Even though the alternative formulations are quite sparse, the storage requirement is still higher than that for the conventional one, as can be seen in Tables 3.

2. 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 (N = 100, 300, and 500) are tried for the example and the final optimization results, the numbers of iterations and CPU efforts for different grid sizes are summarized in Tables 4 and 5.

Table 4 gives the final optimum solutions with different formulations for N=300. The final objective values of the design example are listed in Table 5. It is seen that various simultaneous formulations work well and optimal solutions are obtained. The optimum solutions obtained by the conventional and simultaneous formulations in this study are slightly better than those available in the literature. Most 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 4. Final results of design example (N=300)

	Formulations	E	k_1	k_2	k_3	c_1	c_2	c_3
	Conventional [1]	257.40	50.00	200.00	241.90	12.89	77.52	80.00
	Conventional	254.56	50.00	200.00	200.00	45.45	77.35	80.00
1st order DEs	**TR**	254.91	50.00	200.00	200.00	20.10	76.95	80.00
	CHS	255.82	50.00	200.00	200.00	50.00	77.39	80.00
	SHS	255.74	50.00	200.00	200.00	31.42	77.19	80.00
2nd order DEs	**CD-1**	254.69	50.00	200.00	200.00	19.97	76.99	80.00
	CD-2	254.38	50.00	200.00	200.00	45.23	77.39	80.00
	CD-3	254.38	50.00	200.00	200.00	45.23	77.39	80.00
	Newmark-1 (Const. ace.)	254.69	50.00	200.00	200.00	19.97	76.99	80.00
	Newmark-1 (Linear ace.)	254.84	50.00	200.00	200.00	20.06	76.96	80.00
	Newmark-2 (Const. ace.)	254.69	50.00	200.00	200.00	19.97	76.99	80.00
	Newmark-2 (Linear ace.)	254.56	50.00	200.00	200.00	45.61	77.36	80.00
	Newmark-2 (Ave. ace.)	254.91	50.00	200.00	200.00	20.10	76.95	80.00
	Hermite	254.55	50.00	200.00	200.00	28.44	77.15	80.00
	BSpline	254.65	50.00	200.00	200.00	50.00	77.41	80.00

Since different approximations of state variables are introduced in the simultaneous formulations, the quality of the final solutions is different. Note that the quality of the final solution depends on the approximation made between the state variables. In this work, the finite difference based methods, such as CDs and Newmarks provide similar performance in terms of the final optimal solutions, numbers of iterations and computing efforts. In order to have good results, the number of grid points usually needs to be large. The second order Hermite and Bspline formulations do not provide better accuracy with small a number of time grid points, because the equations of motion constraints are only imposed at the discrete time grid points, and no intermediate points for constraints are considered.

For simultaneous formulations based on first order DEs, such as CHS and SHS, the optimal solutions with N = 100 are already very good. This can be explained by the piecewise cubic polynomial approximations for both the generalized displacements and velocities, and the imposing of constraints at the center of each time step. These provide accurate approximations for the state variables; therefore, for CHS and SHS to reach similar accuracy of optimal solutions as other alternative formulations, a smaller number of time steps N is needed. Although SHS is easier to implement than CHS, due to the inclusion of more variables, the overall performances of CHS and SHS are similar. TR usually performs better than CHS or SHS with respect to the computational efforts for various N, because of simpler expressions of constraints and the exclusion of constraints or variables at the centers of time steps. However since the approximation is not as good as those by CHS or SHS, the optimal solution obtained by TR for N = 100 is not as good, either.

Table 5 contains comparison of the numbers of iterations and CPU efforts for different grid sizes. It shows that as the number of grid points is increased the computational effort with all the alternative formulations increases; the increase being more dramatic for larger N. Although the numbers of iterations needed for the alternative formulations are mostly in the range of 20~40, the CPU/iteration is quite large, and becomes larger as the size of the optimization problem increases. However, the CPU effort for the conventional formulation keeps relatively stable for various N. It is seen that the alternative formulations are more efficient and require less CPU effort than the conventional formulation when the number of time grid points is small (N = 100), except for CHS and SHS. For N = 300, only CD-1 is more efficient than the conventional formulation, and with N = 500, the conventional formulation is the most efficient one among all the formulations. This can be explained by the sizes of the formulations. When the number of time grid points becomes large, there are much more optimization variables and constraints in the alternative formulations which makes the convergence slow. Apparently, the QP solver for calculation of the search direction becomes less efficient as the numbers of variables and constraints increase. Therefore, better QP solvers for large sparse problems need to be developed.

3. Advantages and disadvantages of formulations

The main advantages of the simultaneous formulations for dynamic systems are as follows. (i) The equations of motion for the system need not be integrated explicitly. They need to be satisfied only at the final solution point. This has advantage if instabilities occur or no solution exists for certain designs in the design space. Also, unnecessary simulations of the system are avoided at intermediate designs, where it might be difficult to obtain a solution. (ii) Design sensitivity analysis of the systems (which is quite tedious and difficult to implement) is not needed since all the problem functions are explicit in terms of the variables. (iii) The inclusion of more state variables in the formulations simplifies the constraint expressions and computer implementations. Some constraints may become linear or simple bounds on the variables, such as the constraints on the displacements, velocities and accelerations. The gradients of the linear constraints in the simultaneous formulations can be programmed independently and calculated only once in the solution process. A major disadvantage of the simultaneous formulations is that the optimization problem is very large and large-scale sparse NLP methods are needed. Also, since the simultaneous formulations include different types of variables, which have different orders of magnitudes, scaling of some of the variables is necessary to reduce numerical difficulties.

Table 5. Final objective values, numbers of iterations and computing efforts for different formulations

	Formulations	Final objective values			Numbers of iterations			CPU time (Sec.)		
		***N*=100**	***N*=300**	***N*=500**	***N*=100**	***N*=300**	***N*=500**	***N*=100**	***N*=300**	***N*=500**
	Conventional	252.81	254.56	254.71	23	29	22	13.6	23.2	17.8
1st order DEs	**TR**	253.09	254.91	254.73	21	32	43	4.8	51.3	117.5
	CHS	255.00	254.81	254.78	34	190	18	24.6	323.9	465.3
	SHS	254.31	254.74	254.85	20	27	38	32.4	288.0	539.5
2nd order DEs	**CD-1**	251.24	254.69	254.74	23	27	18	4.3	12.1	74.4
	CD-2	252.57	254.38	254.96	10	23	26	4.9	46.4	172.8
	CD-3	251.24	254.38	254.64	24	20	38	10.9	49.1	116.8
	Newmark-1 (Const. ace.)	251.24	254.69	254.64	22	30	35	5.9	38.4	180.3
	Newmark-1 (Linear ace.)	252.45	254.84	255.01	26	42	31	7.3	51.8	227.8
	Newmark-2 (Const. ace.)	251.24	254.69	254.96	48	22	12	8.4	54.8	64.5
	Newmark-2 (Linear ace.)	252.45	254.56	254.80	30	37	21	7.8	84.5	142.4
	Newmark-2 (Ave. ace.)	253.09	254.91	254.73	20	21	29	9.0	58.0	171.7
	Hermite	252.51	254.55	255.02	37	28	17	8.3	211.5	406.3
	BSpline	251.95	254.65	254.95	30	26	24	6.1	72.0	209.3

4. Scaling of variables

In most alternative formulations, 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 difference in their orders of magnitude, e.g., rotations and translations, some variables, such as the rotations may also need to be scaled. This is true for the rotational degrees of freedoms in the example. The current scaling option in SNOPT works fine 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.

5. Other formulations

It is clear that other simultaneous formulations are possible. These are based on different discretization techniques of the first or second-order DEs, such as the Runge-Kutta formula for numerical solution of first order DEs[14,20,26], and piecewise higher degree polynomial approximations of state variables for first order DEs[16]. However, these multi-step methods or higher degree of polynomials may result in significant complexity of numerical implementation for the simultaneous formulations, which is not desired. The application of these methods for transient dynamic optimization needs further evaluation.

VII. Concluding Remarks

Simultaneous formulations for optimization of transient dynamic mechanical system were described and evaluated. Different state variables or their parametric approximations were treated as optimization variables in the formulations, i.e., generalized displacements, velocities and accelerations. Therefore the discretized state equations (DEs), either in the first or second order forms could be treated as equality constraints in the optimization process. By introducing more variables into the formulations, the forms of the constraints and their derivatives were changed. The formulations were implemented with a sparse NLP code for evaluation. The simultaneous formulations had more variables and constraints, although the constraints had simpler form compared to the conventional formulation. Therefore an optimization algorithm for large numbers of variables and constraints was used to solve the problem.

The solutions for one sample problem were obtained and compared. In terms of CPU times, most SAND formulations outperform the conventional formulation for small number of time grid points. When the problem size is too large, the sparse QP subproblem solver becomes slow to converge, resulting in much more computational effort than the conventional formulation.

The simultaneous 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. The exploitation of the matrix sparsity, decomposition schemes, efficient solution methods of sparse QP subproblems, and parallel algorithms for the alternative SAND formulations need to be further studied, developed and combined for much broader practical applications of these formulations.

Acknowledgement

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

References

[1]Haug, E.J. and Arora, J.S., *Applied Optimum Design, Mechanical and Structural Systems*, John Wiley & Sons, Inc., New York, 1979.

[2]Grandhi, R.V., Haftka, R.T. and Watson, L.T., “Design-oriented identification of critical times in transient response,” *AIAA J.*, Vol. 24, 1986, pp. 649-656.

[3]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(10), 1987, pp. 1827-1840.

[4]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, 1998, pp. 549-564.

[5]Arora, J.S., “Optimization of structures subjected to dynamic loads,” in *Structural Dynamic Systems Computational Techniques and Optimization: Optimization Techniques*, edited by Leondes, C.T., Gordon and Breach Science Publishers, 1999, pp. 1-72.

[6]Hsieh, C.C. and Arora, J.S., “Design sensitivity analysis and optimization of dynamic response,” *Applied Mathematics and Optimization*, Vol. 43(2), 1984, pp. 195-219.

[7]Choi, W.S. and Park, G.J., “Structural optimization using equivalent static loads at all time intervals,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 191(19-20), 2002, pp. 2077-2094.

[8]Kang, B.S., Park, G.J. and Arora, J.S., “Optimization of flexible multibody dynamic systems using the equivalent static load method,” *AIAA J.*, Vol. 43(4), 2005, pp. 846-852.

[9]Wang, Q. and Arora, J.S. “Alternative formulations for transient dynamic response optimization,” *AIAA J.* **43**(10), 2188-2195.

[10]Haftka, R.T., “Simultaneous analysis and design,” *AIAA Journal*, Vol. 23(7), 1985, pp. 1099-1103.

[11]Orozco, C.E. and Ghattas, O.N., “Sparse approach to simultaneous analysis and design of geometrically nonlinear structures,” *AIAA Journal*, Vol. 30, 1992, pp. 1877-1885.

[12]Arora, J.S. and Wang, Q., “Review of formulations for structural and mechanical system optimization,” *Structural and Multidisciplinary Optimization*, to appear, 2005.

[13]Hargraves, C.R. and Paris, S.W., “Direct trajectory optimization using nonlinear programming and collocation,” *J. Guidance, Cont. Dyn.* Vol. 10(4), 1987, pp. 338-342.

[14]Enright, P.J. and Conway, B.A., “Discrete approximation to optimal trajectories using direct transcription and nonlinear programming,” *J. Guidance, Cont. Dyn.* Vol. 15(4), 1992, pp. 994-1002.

[15]Betts, J.T. and Huffman, W.P., “Application of sparse nonlinear programming to trajectory optimization,” *J. Guidance, Cont. Dyn.* Vol. 15(1), 1993, pp 198-206.

[16]Herman, A.L. and Conway, B.A., “Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules,” *J. Guidance, Cont. Dyn.* Vol. 19(3), 1995, pp. 592-599.

[17]von Stryk, O. and Bulirsch, R., “Direct and indirect methods for trajectory optimization,” *Annals of Operations Research*, Vol. 37, 1992, pp. 357-373.

[18]Cervantes, A. and Biegler, L.T., “Large-scale DAE optimization using a simultaneous NLP formulation,” *AIChE Journal*, Vol. 44(5), 1998, pp. 1038-1050.

[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(1), 2000, pp. 39-51.

[20]Betts, J.T., “Survey of numerical methods for trajectory optimization,” *J. Guidance, Cont. Dyn.* Vol. 21(2), 1998, pp 193-207.

[21]Barclay, A., Gill, P.E. and Rosen, J.B., *SQP methods and their application to numerical optimal control*, Report NA 97-3, Department of Mathematics, University of California, San Diego, 1997.

[22]Betts, J.T., “Very low-thrust trajectory optimization using a direct SQP method,” *Journal of Computational and Applied Mathematics*, Vol. 120(1), 2000, pp. 27-40.

[23]Bathe, K.-J., *Finite Element Procedures in Engineering Analysis*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1982.

[24]Chopra, A., *Dynamics of structures: Theory and applications to earthquake engineering*, Prentice-Hall, Upper Saddle River, N.J., 2001.
[25]Atkinson, K.E., *An Introduction to Numerical Analysis*, John Wiley and Sons, Inc., New York, 1988.
[26]Betts, J.T. and Huffman, W.P., "Exploiting sparsity in the direct transcription method for optimal control," *Comp. Optim. & Appl.* Vol. 14(2), 1999, pp. 179-201.
[27]Gill, P.E., Murray, W. and Saunders, M.A., "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM J. Optim.* Vol. 12, 2002, pp. 979-1006.