AIAA JOURNAL
Vol. 36, No. 11, November 1998

Optimal Location and Gains of Feedback Controllers at Discrete Locations

Makola M. Abdullah*
Florida A&M University, Tallahassee, Florida 32310

The dynamic responses of elastic structures due to vibrations are very important and can produce conditions that can vary from uncomfortable to unsafe. For this reason, optimally designed and placed control devices are very important for the reduction of these vibrations. Many structures have limitations on where these control devices can be placed. However, many existing algorithms are not able to provide solutions to these nonlinear/discrete location problems. A new method is presented for optimal design of the placement and gains of actuators and sensors at discrete locations in output feedback control systems. The method extends the method of Xu et al. (Xu, K., Warnitchai, P., and Ingusa, T., "Optimum Locations and Gains of Sensors and Actuators for Feedback Control," AIAA Paper 93-1660, 1993) by solving for the optimal placement at certain discrete locations. A new algorithm is introduced that will allow constrained optimization problems with solutions possible only at discrete locations to be modeled as unconstrained optimization problems. Numerical studies were performed on a two-dimensional structure. Convergence to optimal discrete solutions was efficient.

Nomenclature

A_{2N}	= integer transformation function constant
a, b	= dimensions of the membrane structure
$\boldsymbol{b}(\boldsymbol{x}_a)$	= $n \times m$ actuator placement matrix
$\boldsymbol{C}_d$	= $n \times n$ modal damping matrix
$\boldsymbol{c}(\boldsymbol{x}_s)$	= $r \times n$ sensor placement matrix
c_0	= damping per unit area
$\boldsymbol{F}$	= $m \times r$ time-invariant output feedback gain matrix
$\boldsymbol{F}(\boldsymbol{x}, t)$	= applied force
$g(x)$	= sample function
$\boldsymbol{H}_i$	= approximate inverse of the Hessian matrix
$h(y)$	= integer transformation function
$\boldsymbol{I}$	= $n \times n$ identity matrix
J	= expected value of the control cost function; cost function
$\hat{J}$	= control cost function
$\boldsymbol{K}$	= matrix integral
L	= linear differential operator
$\boldsymbol{L}$	= matrix integral
m	= number of actuators
m_0	= mass per unit area
N	= integer for the approximation of the step function
n	= number of modes in the structural model
$\boldsymbol{Q}$	= weight matrix for structural response
$\boldsymbol{q}(t)$	= generalized coordinates
$\boldsymbol{R}$	= weight matrix for control forces
r	= number of sensors
$\boldsymbol{S}_i$	= search direction vector
t	= time
$\boldsymbol{u}(t)$	= applied control force matrix
$w(\boldsymbol{x}, t)$	= displacements
$\boldsymbol{X}_a$	= boundary for actuator placement
$\boldsymbol{X}_i$	= vector of optimization variables
$\boldsymbol{X}_s$	= boundary for sensor placement
$\boldsymbol{x}$	= spatial coordinate
$\boldsymbol{x}_a$	= actuator placement coordinate
$\boldsymbol{x}_s$	= sensor placement coordinate
$\boldsymbol{Y}_i$	= integer transformed vector of optimization variables
$\boldsymbol{y}_a$	= integer transformed actuator placement coordinate
$\boldsymbol{y}_s$	= integer transformed sensor placement coordinate
$\boldsymbol{y}(t)$	= r vector of sensor outputs
α_j, β_j	= parameters for open-loop modes
δ_{ij}	= Kronecker delta function
ε	= convergence control parameters
η, ξ	= unconstrained coordinates
Λ	= $n \times n$ modal stiffness matrix
λ	= elastic constant
$\phi(x)$	= n vector of eigenfunctions
ω_i	= ith natural frequency of the open-loop system
$*$	= optimal values

Introduction

IN the design of feedback control systems for flexible structures, there are two sets of design parameters, the control gains and the placement of sensors and actuators. There are well-established methods for determining optimal gains; however, methods for optimal placement are relatively new. Early optimal placement methods minimized control energy, whereas the design of the feedback gains was considered separately.[1–3] Schultz and Heimbold[4] developed a method of concurrent design of both placement and gains. The method is optimal in that it maximizes energy dissipation due to control action. The solution to the optimization problem was obtained by a gradient-based nonlinear programming technique.

In this paper, the method of Xu et al.[5] is extended. The original method consisted of calculating the gradients of the performance function not only with respect to the feedback gains but also with respect to sensor and actuator placement. Only the dimensions of the example structure limited the placement of the collocated sensors and actuators. However, in many cases, there is a discrete and finite set of possible locations within the structure where the sensors and actuators can be placed. In solving problems of this nature, it is necessary to use integer-programming techniques. In most gradient-based optimal control problems, each of the design variables are permitted to take any real value. For problems where all or some of the design variables must be from a discrete set, the most feasible way to find a solution is using integer-programming methods. An integer-programming method is used in this paper to find optimal placement of controllers and actuators at discrete locations.

It is not always possible to place control sensors and actuators at arbitrary positions. It is possible that for many structures the optimal locations for sensors and actuators to minimize vibrations are not feasible. Simply rounding solutions can give very unpredictable results. It is possible to round the location of an actuator to a position where a mode would be uncontrollable. In this case, the desired and actual control effects can differ drastically. Also, it is not always

Received Aug. 19, 1997; revision received July 20, 1998; accepted for publication July 31, 1998.

*Assistant Professor, Department of Civil Engineering, 2525 Pottsdamer Street.

possible to search every combination of locations, especially with the concurrent design of placement and gains. Using the analytical expressions for the performance function defined by Xu et al.,[5] a transformation will be used to find the optimal placements from a set of discrete locations.

Relatively very little work has been done in the area of nonlinear and mixed integer (problems with both integer and continuous design variables) programming. In many cases, linear integer-programming techniques have been used to solve some nonlinear problems. It is known that these nonlinear integer-programming methods are not as robust as their continuous counterparts.[6] Several integer-programming techniques that have been used for nonlinear optimization include the zero–one[7] and branch and bound[8] and interior penalty function methods[8] for integer and mixed integer problems.

The zero–one method has been used successfully for certain types of nonlinear problems. In this method, each possible solution for each design variable is transformed into a zero–one variable. If a solution is chosen to be optimal it is given the value of one, if it is not chosen it is given the value of zero. The advantage of this method is that it can handle a large number of integer values effectively for convex problems. However, in using the standard control cost functions, it is rare for more than one mode of the problem to be convex. Also, this method is not efficient for mixed integer problems and requires significant changes in the structure of the optimization algorithm to change to the binary zero–one system. The branch and bound method is very efficient for an integer problem, but it is only useful for a small number of variables, less than 10 or 20. The interior penalty function method for integer and mixed integer problems is capable of handling mixed integer problems effectively; however, it was felt that the addition of a penalty function would affect the generic qualities of the control algorithm. Many other methods have been developed for very discrete problems: the knapsack problem or the traveling salesman problem. It was not clear, however, whether these methods could be used for the quadratic control cost functions that are popular for the control of flexible structures. Note that, because there has been little study of nonlinear integer and mixed integer programming, these methods are not yet reasonably guaranteed to converge. It was important to develop a gradient-based method that handled the mixed integer problem effectively and was easy to program. This method must also have a satisfactory rate of convergence.

A new method is proposed here. For well-defined functions, it has been shown that a constrained nonlinear problem can be converted into an unconstrained problem using a transformation of variables.[5] This technique is very easy to program. In the proposed method, the use of transformations is generalized where a change of the dependent variables will be used to transform the constrained integer problem to an unconstrained continuous one. Because the performance function is well defined, it is demonstrated that this change of variables is an effective means of finding integer optimal values. One of the major advantages to this method is its ease of programming. It is shown to be quite effective with the very robust Davidion–Fletcher–Powell (DFP) algorithm used in the numerical optimization.

Discrete Location Programming Technique

Consider the optimization problem

$$\min J(\boldsymbol{x}_a, \boldsymbol{x}_s, \boldsymbol{F}) \rightarrow \boldsymbol{x}_a^*, \boldsymbol{x}_s^*, \boldsymbol{F}^* \tag{1}$$

where J is the cost function defined later, $\boldsymbol{x}_s$ and $\boldsymbol{x}_a$ are the placement of sensors and actuators, respectively, and $\boldsymbol{F}$ is a feedback gain matrix, whereas $\boldsymbol{x}_s^*$, $\boldsymbol{x}_a^*$, and $\boldsymbol{F}^*$ are the values for the design variables that minimize J. The variables $\boldsymbol{x}_s$ and $\boldsymbol{x}_a$ in a continuous case are subject to constraints

$$\boldsymbol{x}_a \in \boldsymbol{X}_a, \qquad \boldsymbol{x}_s \in \boldsymbol{X}_s \tag{2}$$

where $\boldsymbol{X}_a$ and $\boldsymbol{X}_s$ are subsets of the domain of the structure, which in most cases is only limited to the dimensions of the structure. If $\boldsymbol{x}_a$ and $\boldsymbol{x}_s$ are restricted to integers, or a set of equally spaced numbers $\boldsymbol{X}_a$ and $\boldsymbol{X}_s$, this problem cannot be solved by standard techniques. This problem is now (or can be represented as) a mixed integer problem, where a partial set of variables (actuator and sensor placements) are integers and the remaining variables (gain assignments) are nonintegers. It would be useful to find a transformation to make all of the variables noninteger so that the problem becomes a noninteger problem. This objective would be accomplished if a function $\boldsymbol{x} = h(\boldsymbol{y})$ could be found so that

$$\min J[\boldsymbol{h}(\boldsymbol{y}_a), \boldsymbol{h}(\boldsymbol{y}_s), \boldsymbol{F}] \rightarrow \boldsymbol{y}_a^*, \boldsymbol{y}_s^*, \boldsymbol{F}^* \tag{3}$$

subject to the constraints

$$\boldsymbol{y}_a \in \boldsymbol{X}_a, \qquad \boldsymbol{y}_s \in \boldsymbol{X}_s \tag{4}$$

where $\boldsymbol{y}$ is a continuous variable that produces an integer value for $\boldsymbol{x}$, and $\boldsymbol{X}_a$ and $\boldsymbol{X}_s$ still are the subsets of the domain of the structure.

The function h in theory should be a round-off step function, which would round any number to the closest integer. However, this step function is discontinuous and, therefore, its derivative is undefined at each step. This renders gradient-based nonlinear-programming methods unusable because the gradients of $J[h(\boldsymbol{y}), \boldsymbol{F})$ would now also be discontinuous. Thus, it is important to find a function that looks like this step function and is yet continuous.

It can be shown that the sine function raised to an even power, as the even power becomes large, begins to look like a periodic impulse function.[9] Reference 9 should be consulted for more details. If this function is multiplied by a constant to make the area under each impulse equal to one, the following integral becomes an approximation for the step function:

$$x_i = h(y_i) \equiv A_{2N} \int_0^{y_i} [\sin(\pi u)]^{2N} \, \mathrm{d}u \tag{5}$$

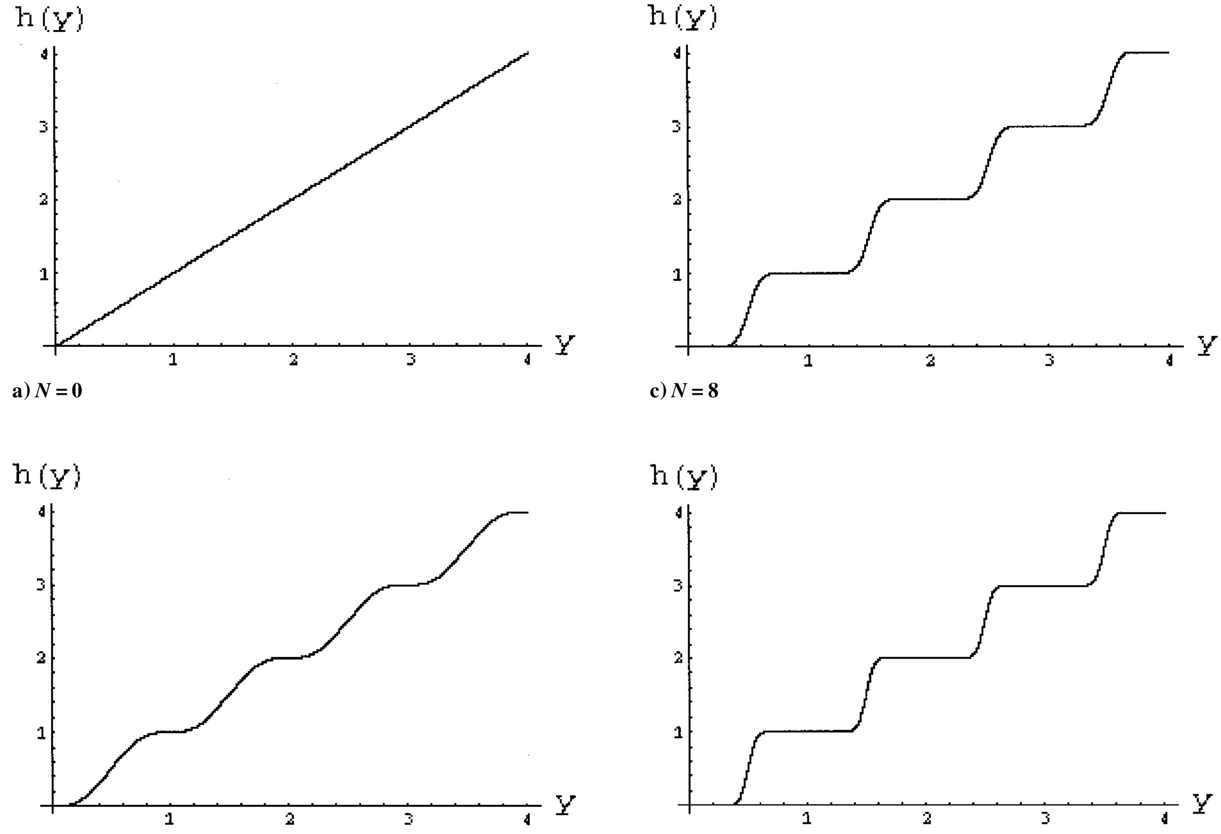
where

$$A_{2N} = \frac{2^{2N}(N!)^2}{(2N)!} \tag{6}$$

and N is an integer for the approximation of the step function. The constant values for A_{2N} were found analytically. As N increases, the approximation to the step function improves. As a point of reference, if $N = 0$, $h(y) = y$. Then to render certain variables noninteger, all that is necessary is to set n equal to zero for those variables. Figure 1 shows a plot of the function $x = h(y)$ for increasing values of N ($N = 0$, 1, 8, and 16). The approximation gets quite accurate for reasonable values of N. In all examples, the value of N did not exceed 61.

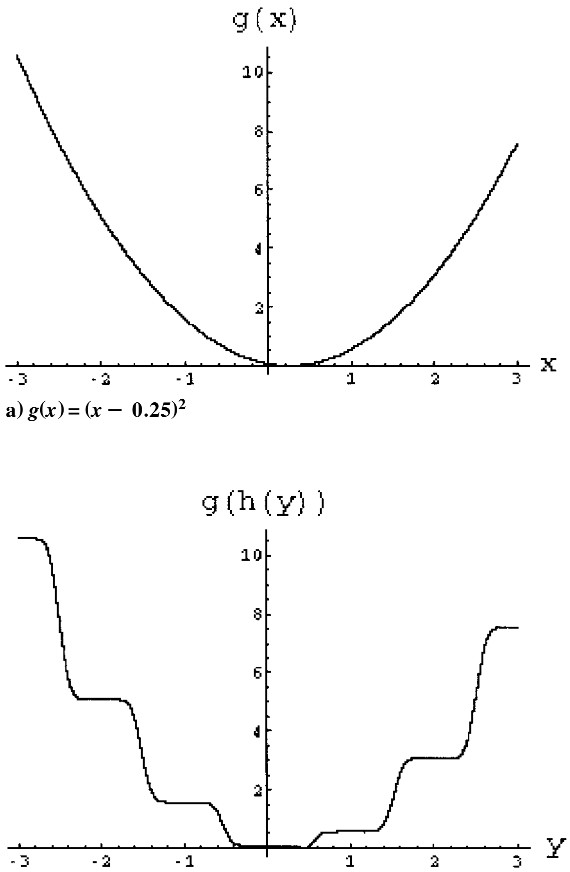
The transformation just described makes integer values more attractive in the optimization process. As N gets large, it becomes increasingly difficult for nonlinear-programming methods to find noninteger values. There are two reasons: First, the gradients at integer values are much smaller that those at noninteger values. This allows a program to perform as if it has found a minimum or maximum. More maxima or minima are not created but plateaus are that are similar to saddle points. Saddle points are neither local maximum nor local minimum points; rather, they are points where the gradients are equal to zero. Second, the noninteger values begin to occupy a smaller range. For instance, for $N = 27$ all values of x between 1.00 and 2.00 correspond to values of y values between 1.4 and 1.6, this can also be seen in Fig. 1.

To illustrate the preceding concepts, consider the simple curve $g(x) = (x - 0.25)^2$. This is a curve where the minimum occurs at $x = 0.25$, which is not an integer value. The graphs of $g(x)$ and $g[h(y)]$ are shown in Fig. 2. From these two curves, it is easy to see how the integer values become more attractive. This transformation using $h(y)$ was done for $N = 5$, a relatively low number. The plateau that exists at all integers for values of y will make it very difficult to find $x = 0.25$. For $N = 10$, the point $x = 0.25$ would be even harder to find. However, if N is too large, integer values become too attractive; an iterative method might refuse to move from the first integer value it finds because of the small value of the gradient, and the search vector (which is proportional to the gradient) might be zero or close to zero. This would yield integer convergence, but not the optimal integer convergence. Optimal integer convergence is much more likely if $N = 0$ is chosen first and then increased at each nonconvergent iteration. In other words, N should be a function of the



a) $N = 0$ b) $N = 1$ c) $N = 8$ d) $N = 16$

Fig. 1 Transformation of variables.



a) $g(x) = (x - 0.25)^2$

b) $g[h(y)] = [h(y) - 0.25]^2$

Fig. 2 Graphs where $N = 5$.

iteration step i, where $N = f(i)$. This gives the algorithm direction before integer solutions become too attractive, which allows a minimum region to be found early, but not a noninteger solution. Another helpful addition to the algorithm to increase optimal convergence is to start increasing the value of N after a minimum region has been found. In other words, let

$$N = f(i - i_{\mathrm{mr}}) \tag{7}$$

where i_{mr} is the first iteration step where the performance function satisfies $J < J_{\mathrm{mr}}$ and J_{mr} is a chosen value of the cost function that defines the minimum region. This implies the assumption that the minimum integer solution will occur in the region near a local minimum. The two-dimensional examples shown later will show that this is true for most cases. Generally, the value for J_{mr} is chosen after the values of the noninteger local minimums have been found.

The optimal design problem defined by Eqs. (3) and (4) is a constrained nonlinear-programming problem. However, as long as the subspaces $\boldsymbol{X}_a$ and $\boldsymbol{X}_s$ are polygons, or other well-defined geometries, the constrained problem can be mathematically transformed to an unconstrained problem.[5] Therefore, the optimization problem presented here can be treated as an unconstrained minimization problem. It is important now to define the cost function J as a function of sensor/actuator placement.

Formulation of the Optimization Problem for Control of Linear Elastic Structures

The vibration response of an elastic structure subjected to an applied force $\boldsymbol{F}(\boldsymbol{x}, t)$ is given by

$$\left[m_0 \frac{\mathrm{d}^2}{\mathrm{d}t^2} + c_0 \frac{\mathrm{d}}{\mathrm{d}t} + \lambda L\right] w(\boldsymbol{x}, t) = \boldsymbol{F}(\boldsymbol{x}, t) \tag{8}$$

The first two terms are the inertial and viscous damping forces. The third term is the elastic restoring force, expressed in terms of a linear differential operator L. If Eq. (8) can be solved by separation of variables, then $w(\boldsymbol{x}, t)$ is of the form

$$w(\boldsymbol{x}, t) = \phi^T(\boldsymbol{x})\boldsymbol{q}(t) \tag{9}$$

where $\phi^T(\boldsymbol{x})$ are the mode shapes of structure and $\boldsymbol{q}(t)$ are the generalized coordinates given by the solution of

$$\boldsymbol{I}\ddot{\boldsymbol{q}}(t) + \boldsymbol{C}_d\dot{\boldsymbol{q}}(t) + \Lambda\boldsymbol{q}(t) = \boldsymbol{b}\boldsymbol{u}(t) \tag{10}$$

The applied control force is expressed as a product of the $n \times m$ matrix $\boldsymbol{b}$ and an m vector $\boldsymbol{u}(t)$, whereas $\boldsymbol{I}$ is the identity matrix, and Λ and $\boldsymbol{C}_d$ are the decoupled modal stiffness and damping matrix, whose diagonal terms are defined as follows:

$$\Lambda_i = \frac{\int_0^L \phi_i L\phi_j \,\mathrm{d}x}{\int_0^L m\phi_i^2 \,\mathrm{d}x} = \delta_{ij}\omega_j^2 \tag{11}$$

and

$$C_{di} = \frac{\int_0^L \phi_i c_0\phi_j \,\mathrm{d}x}{\int_0^L m\phi_i^2 \,\mathrm{d}x} = \delta_{ij}2\zeta\omega_j \tag{12}$$

where ω_j are the natural frequencies, ζ is the damping ratio, and δ_{ij} is the Kronecker delta.

Consider m point-force actuators located at $\boldsymbol{x}_{aj}$, where $j = 1, \ldots, m$, and r velocity sensors located at $\boldsymbol{x}_{sp}$, where $p = 1, \ldots, r$. Then, $\boldsymbol{u}(t)$ are the vectors of the actuator forces, and

$$\boldsymbol{b}(\boldsymbol{x}_a) = [\phi(x_{a1}) \quad \cdots \quad \phi(x_{am})] \tag{13}$$

For direct output velocity feedback control used by Balas,[10] the r-dimensional vector $\boldsymbol{y}(t)$ is given by

$$\boldsymbol{y}(t) = \boldsymbol{c}(\boldsymbol{x}_s)\dot{\boldsymbol{q}}(t) \tag{14}$$

where $\boldsymbol{c}(\boldsymbol{x}_s)$ is the placement matrix

$$\boldsymbol{c}(\boldsymbol{x}_s) = \begin{bmatrix} \phi^T(x_{s1}) \\ \vdots \\ \phi^T(x_{sm}) \end{bmatrix} \tag{15}$$

The control force is proportional to the output measurement

$$\boldsymbol{u}(t) = -\boldsymbol{F}\boldsymbol{y}(t) \tag{16}$$

where $\boldsymbol{F}$ is an $m \times r$ time invariant gain matrix.

It is convenient to express Eqs. (10) and (13–16) as a first-order state equation

$$\dot{\mathbf{x}}(t) = \boldsymbol{A}_0\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \qquad \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \tag{17}$$

where

$$\boldsymbol{A}_0 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\Lambda & -\boldsymbol{C}_d \end{bmatrix}, \qquad \boldsymbol{x}(t) = \begin{bmatrix} \boldsymbol{q}(t) \\ \dot{\boldsymbol{q}}(t) \end{bmatrix}$$
$$\boldsymbol{B} = \begin{bmatrix} 0 \\ \boldsymbol{b} \end{bmatrix}, \qquad \boldsymbol{C} = [0 \quad \boldsymbol{c}] \tag{18}$$

The free vibration response of the closed-loop system for any initial condition $\boldsymbol{x}(0)$ is[11,12]

$$\boldsymbol{x}(t) = e^{At}\boldsymbol{x}(0) \tag{19}$$

where e^{At} is the fundamental transition matrix and

$$\boldsymbol{A} = \boldsymbol{A}_0 - \mathbf{BFC} \tag{20}$$

Gradient-Based Optimization Technique

In the following, an optimal design procedure is developed for the actuator placement $\boldsymbol{x}_{aj}$, the sensor placement $\boldsymbol{x}_{sp}$, and the feedback gains $\boldsymbol{F}$. First, a performance function is chosen that includes both the structural response and the control effort. The standard performance function is

$$\hat{J} = \frac{1}{2}\int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\,\mathrm{d}t \tag{21}$$

which can be expanded in terms of the fundamental transition matrix by substituting for $\boldsymbol{u}$ as a function of $\boldsymbol{x}$:

$$\hat{J} = \boldsymbol{x}^T(0)\left[\frac{1}{2}\int_0^\infty e^{A^Tt}(\boldsymbol{Q} + \boldsymbol{C}^T\boldsymbol{F}^T\boldsymbol{RFC})\,e^{At}\,\mathrm{d}t\right]\boldsymbol{x}(0) \tag{22}$$

A design procedure that uses this performance function will require discrete values for the initial state $\boldsymbol{x}(0)$ (Ref. 4). This dependence on $\boldsymbol{x}(0)$ is eliminated here by using a performance function proposed by Levine and Athans.[12] The initial state is modeled as a random vector uniformly distributed on the surface of the $2n$-dimensional unit sphere. It has been shown that the average, or expected, value of $\hat{J}$ scaled by $2n$ is

$$J = \frac{1}{2}\int_0^\infty \mathrm{tr}\left[e^{A^Tt}(\boldsymbol{Q} + \boldsymbol{C}^T\boldsymbol{F}^T\boldsymbol{RFC})\,e^{At}\right]\mathrm{d}t \tag{23}$$

This performance function, which is expressed explicitly in terms of actuator and sensor placement vectors $\boldsymbol{x}_{aj}$ and $\boldsymbol{x}_{sp}$ (which are imbedded in the matrices $\boldsymbol{C}$ and $\boldsymbol{B}$), and the feedback gain matrix $\boldsymbol{F}$ are used in the following.

The optimization problem is

$$\min J(\boldsymbol{x}_a, \boldsymbol{x}_s, \boldsymbol{F}) \rightarrow \boldsymbol{x}_a^*, \boldsymbol{x}_s, \boldsymbol{F}^* \tag{24}$$

subject to the constraints

$$\boldsymbol{x}_a \in \boldsymbol{X}_a, \qquad \boldsymbol{x}_s \in \boldsymbol{X}_s \tag{25}$$

where $\boldsymbol{X}_a$ and $\boldsymbol{X}_s$ are subsets of the domain of the structure. For simplicity, the subscripts j and p are dropped. If $\boldsymbol{X}_a$ and $\boldsymbol{X}_s$ are noninteger values the optimization problem can be solved with standard nonlinear-programming techniques. Because the performance function gradients can be found analytically, descent methods can be used to solve this problem. The derivations of the gradients to be shown were previously shown by Xu et al.[5] and Levine and Anthans.[12]

To determine the gradients of the performance function, the following two matrix integrals are needed:

$$\boldsymbol{K} = \int_0^\infty e^{A^Tt}(\boldsymbol{Q} + \boldsymbol{C}^T\boldsymbol{F}^T\boldsymbol{RFC})\,e^{At}\,\mathrm{d}t \tag{26}$$

and

$$\boldsymbol{L} = \int_0^\infty e^{At}e^{A^Tt}\,\mathrm{d}t \tag{27}$$

The matrix $\boldsymbol{K}$ is related to the performance function by

$$J = \tfrac{1}{2}\mathrm{tr}[\boldsymbol{K}] \tag{28}$$

The matrices can be solved, without numerical integration, by solving the associated Lyapunov equations[13]

$$\boldsymbol{KA} + \boldsymbol{A}^T\boldsymbol{K} + \boldsymbol{Q} + \boldsymbol{C}^T\boldsymbol{F}^T\boldsymbol{RFC} = 0 \tag{29}$$

$$\boldsymbol{LA} + \boldsymbol{A}^T\boldsymbol{L} + \boldsymbol{I} = 0 \tag{30}$$

The Lyapunov equations are efficiently solved by the method of Bartels and Stewart.[14] The gradients of the matrices $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{F}$ are as follows:

$$\frac{\partial J}{\partial \boldsymbol{B}} = -\boldsymbol{KLC}^T\boldsymbol{F}^T \tag{31}$$

$$\frac{\partial J}{\partial \boldsymbol{C}} = -\boldsymbol{F}^T\boldsymbol{B}^T\boldsymbol{KL} + \boldsymbol{F}^T\boldsymbol{RFCL} \tag{32}$$

$$\frac{\partial J}{\partial \boldsymbol{F}} = -\boldsymbol{RFCLC}^T + \boldsymbol{B}^T\boldsymbol{KLC}^T \tag{33}$$

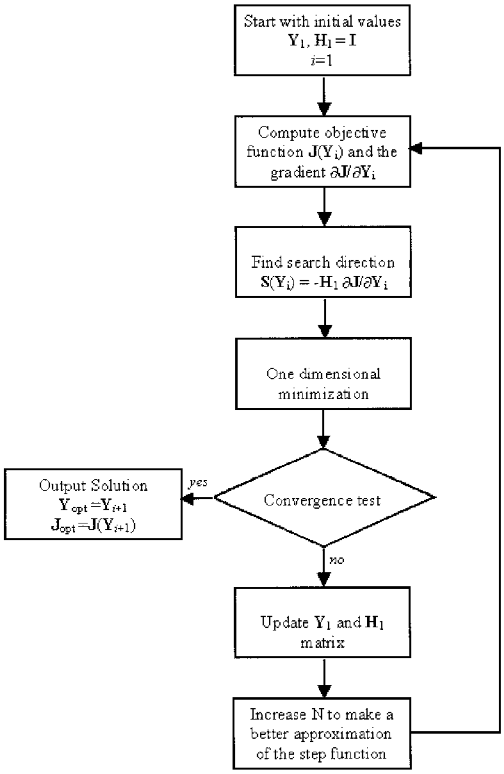


Fig. 3 Flowchart for optimization technique.

The control systems used for the present study are now limited to single-loop collocated sensors and actuators with direct velocity feedback gains. This eliminates the effect of residual modes causing instabilities in the system.[15] Because the system is collocated, $\boldsymbol{C} = \boldsymbol{B}^T$, and it is necessary to sum the preceding two gradients to yield

$$\frac{\partial J}{\partial \boldsymbol{B}} = -\boldsymbol{KLBF}^T + \boldsymbol{LBF}^T\boldsymbol{RF} + \boldsymbol{KLBF}^T \tag{34}$$

The gradients with respect to the coordinate vectors $\boldsymbol{x}_a$ and the elements of the gain matrix $\boldsymbol{F}$ will be shown later.

One of the most robust gradient-based unconstrained optimization techniques is the DFP algorithm,[6,16] which is summarized by the flowchart in Fig. 3. The basic parameters for each iteration i are: the vector of optimization variables $\boldsymbol{X}_i$, an approximate inverse of the Hessian matrix $\boldsymbol{H}_i$, and a search direction vector $\boldsymbol{S}_i$. After the search direction vector is computed, the one-dimensional minimization proceeds as follows. An accelerated step size algorithm is used to determine an interval in the search direction that contains at least one local minimum. Then, within this interval, cubic interpolation is used to find a minimum. However, if there is more than one local minima in the search direction, convergence may be slow; therefore, a combination of cubic interpolation and direct root methods is used. If convergence does not occur after several iterations, then the direct root method is used to reduce the domain so that it includes only one local minimum. As the iterations increase and the step function approximation improves, additional saddle points will become prevalent. Saddle points are neither maximum nor minimum conditions and have zero gradients. These points will exist at integer solutions and will increase the complexity of the problem. In the last step of the algorithm, the following convergence tests are used:

$$\frac{J[h(Y_{i+1})] - J[h(Y_i)]}{J[h(Y_i)]} \le \varepsilon_1 \tag{35}$$

$$\frac{\partial J}{\partial Y_j} \le \varepsilon_2 \tag{36}$$

$$\mathrm{round}(Y_j) - Y_j \le \varepsilon_3 \tag{37}$$

Here, ε_1, ε_2, and ε_3 are convergence control parameters and Y_j is the jth component of the vector $\boldsymbol{Y}_i$. Round(Y) is a function that rounds the vector component to an integer value, and $\partial J/\partial y$ can be found by using the chain rule and multiplying $\partial J/\partial x$ by $\partial x/\partial y$, which is the derivative of the function h as shown in Eq. (5).

Example Analysis

The example structure is a rectangular membrane, with dimensions $a = 1.00$ and $b = 1.03$, as shown in Fig. 4. The derivation of the equations of motion can be found in standard texts.[17] The open-loop modal damping ratios are 0.005 for all modes; a total of 11 modes are considered in the following numerical analysis. The mode shapes of the open-loop system are

$$\phi_j(x, y) = \sin[\alpha_j(x/a)]\sin[\beta_j(y/b)] \tag{38}$$

The modal parameters α_j and β_j and the natural frequencies are given in Table 1.

The output feedback control system consists of m point force actuators and collocated sensors located at $\{x_j, y_j\}$, where $j = 1, \ldots, m$. The matrix $\boldsymbol{b}$ defined in Eq. (11) is

$$\boldsymbol{b} = \begin{bmatrix} \phi_1(x_1, y_1) & \cdots & \phi_1(x_m, y_m) \\ \vdots & \ddots & \vdots \\ \phi_n(x_1, y_1) & \cdots & \phi_n(x_m, y_m) \end{bmatrix} \tag{39}$$

For the performance function used in Eq. (21), the following weighting matrices are used:

$$\boldsymbol{Q} = Q\begin{bmatrix} \Lambda & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \qquad \boldsymbol{R} = R\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{40}$$

When $\boldsymbol{Q}$ is chosen as in Eq. (40), the first term in the original cost function in Eq. (21) represents the total energy of the function. R is chosen to be 10 for the example analysis.

The constraints for actuator placement are the boundaries of the membrane, i.e.,

$$0 \le x_i \le a, \qquad 0 \le y_i \le b, \qquad i = 1, \ldots, m \tag{41}$$

Table 1 Modal properties of the open-loop system

Mode number	Frequencies	α_j	β_j
1	1.394	π	π
2	2.184	π	2π
3	2.223	2π	π
4	2.788	2π	2π
5	3.080	π	3π
6	3.153	3π	π
7	3.533	2π	3π
8	3.574	3π	2π
9	4.010	π	4π
10	4.116	4π	π
11	4.181	3π	3π

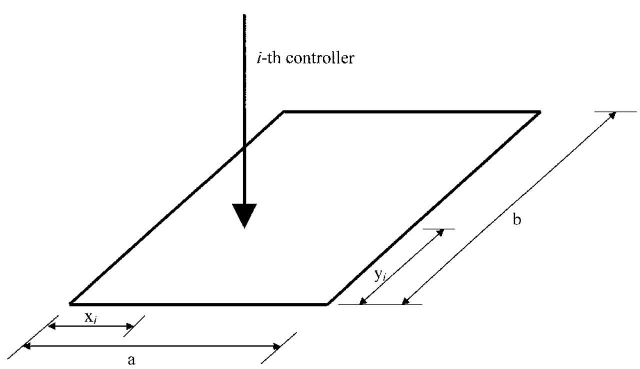


Fig. 4 Elastic membrane.

To eliminate these mathematical constraints, the constrained coordinates x_j and y_j are transformed to unconstrained coordinates η_j and ξ_j by

$$x_i = a\sin^2(\eta_i), \qquad y_i = b\sin^2(\xi_i) \qquad i = 1, \ldots, m \tag{42}$$

Although there are an infinite number of values for η_j and ξ_j for each pair of values for x_j and y_j, the algorithm is well behaved for any initial values for the optimization variables. For the purpose of restricting the values of x_j and y_j to equally spaced intervals, we will introduce the step function approximation from Eq. (5),

$$x_i = ah\left[\sin^2(\eta_i)\right], \qquad y_i = bh\left[\sin^2(\xi_i)\right] \qquad i = 1, \ldots, m \tag{43}$$

The gradients for these placement variables can be obtained from the chain rule:

$$\frac{\partial J}{\partial \eta_i} = \frac{\partial J}{\partial B_{kl}} \frac{\partial B_{kl}}{\partial b_{mn}} \frac{\partial b_{mn}}{\partial \phi_p(x_s)} \frac{\partial \phi_p(x_s)}{\partial x_p} \frac{\partial x_s}{\partial \eta_i} \tag{44}$$

where

$$\frac{\partial B_{kl}}{\partial b_{mn}} = \begin{cases} 0, & \text{if} \quad 1 \le k \le n\,\text{mode} \\ \delta_{(k-n\,\text{mode})m}\delta_{\ln}, & \text{if} \quad n\,\text{mode} + 1 \le k \le 2n\,\text{mode} \end{cases} \tag{45}$$

$$\frac{\partial b_{mn}}{\partial \phi_p(x_s)} = \delta_{mr}\delta_{ns} \tag{46}$$

$$\frac{\partial \phi_p(x_s)}{\partial x_p} = \delta_{sp}\frac{\partial \phi_p(x_s)}{\partial x_s}\{\text{no sum on } s\}$$

$$= \delta_{sp}\frac{\alpha_r}{a}\cos\left(\alpha_r\frac{x_s}{a}\right)\sin\left(\beta_r\frac{y_s}{b}\right) \tag{47}$$

$$\frac{\partial x_p}{\partial \eta_i} = \delta_{pi}2ah'\left[\sin^2(\eta_i)\right]\sin(\eta_i)\cos(\eta_i) \tag{48}$$

After substitution and manipulation, it is found that Eq. (44) can be simplified to

$$\frac{\partial J}{\partial \eta_i} = \sum_{k=1}^{n\,\text{mode}} 2\frac{\partial J}{\partial B_{(k+n\,\text{mode})i}}\alpha_k\cos\left(\alpha_k\frac{x_i}{a}\right)\sin\left(\beta_k\frac{y_i}{b}\right) \times h'\left[\sin^2(\eta_i)\right]\sin(\eta_i)\cos(\eta_i) \tag{49}$$

The gradients for the gain matrix $\boldsymbol{F}$ are much simpler. If we assume that $\boldsymbol{F}$ is a diagonal matrix and

$$F_{ii} = f_i^2 \tag{50}$$

then the gradient with respect to f_i would be

$$\frac{\partial J}{\partial f_i} = \frac{\partial J}{\partial F_{ii}}2f_i \tag{51}$$

During the study, 20 numerical examples of an elastic membrane were completed. The elastic membrane is shown in Fig. 4 Three arrangements were considered: the placement of 1 actuator with 1 mode, the placement of 1 actuator with 11 modes, and the placement of 11 actuators using 11 modes. The discrete location one-actuator example was done for five equally spaced divisions on both the x and y axes. This produces 36 discrete locations, 16 on the interior and 20 on the boundary. The discrete location 11-actuator example was done for 11 equally spaced divisions on both the x and y axes. This produces 144 discrete locations, 100 on the interior and 44 on the boundary. Each case was performed with and without the integer-programming technique to compare with the results of Xu et al.[5]

Each of the examples were run for 20 random initial starting values. Because of the complexity of the cost function, it is possible for the control algorithm to settle on many possible local minima. The minima that yielded the lowest value of the cost function were deemed the optimal solutions. The percentage of occurrence of the optimal solutions is also found for each of the examples. This is defined as the percentage of optimal solutions found from the 20 initial points. Each initial value set was allowed 100 iterations. All of the continuous examples converged in much less than the allowed 100 iterations for 100% convergence. These examples were computed for comparison to the discrete cases. A discrete location solution had to be correct to three decimal places. The results are shown in Tables 2–4. Figure 5 shows the cost function with respect to placement coordinates. Figure 6 shows the actuator/sensor placement for one mode. Figure 7 shows optimal local minima, and Figs. 8–10 show discrete location solution results.

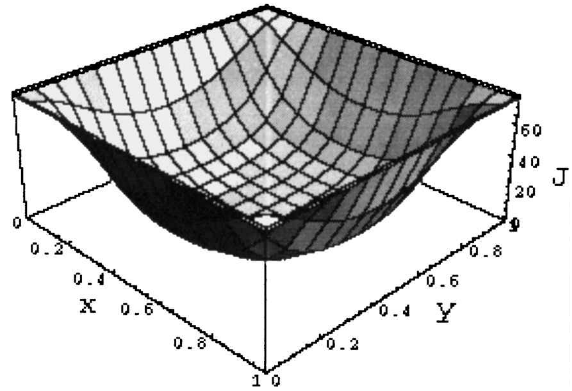


Fig. 5 Cost function J with respect to single actuator/sensor placement, 1 mode, gain = 0.160.

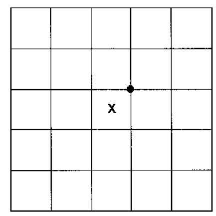


Fig. 6 Actuator/sensor placement for one mode: ●, $J(0.6, 0.6) = 3.564$ and gain = 0.106, and ×, $J(0.5, 0.5) = 3.244$ and gain = 0.125.

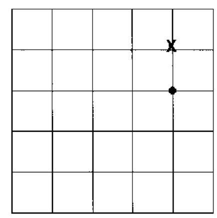


Fig. 7 Actuator/sensor placement for 11 modes: ●, $J(0.800, 0.600) = 45.775$ and gain = 0.279, and ×, $J(0.790, 0.825) = 45.468$ and gain = 0.264.

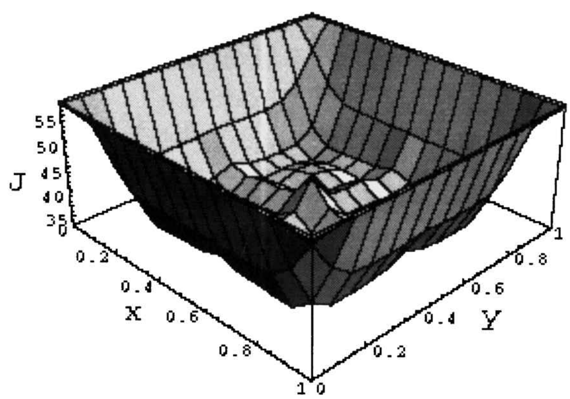


Fig. 8 Cost function J with respect to single actuator/sensor placement, 11 modes, gain = 0.160.

Discussion

Single Actuator/Sensor for One Mode

The simplest optimal design problem is a single actuator and collocated sensor with one mode. There are only three optimization

Table 2 One actuator/sensor, one-mode case

Design type	Convergence, %	Optimal convergence, %	Optimal placement (x, y)	Optimal gain	Cost function J
Continuous	100	100	(0.50, 0.50)	0.425	3.244
Discrete location	85	55	(0.60, 0.60) or (3/5, 3/5)	0.470	3.564
Percent change					9.8%

Table 3 One actuator/sensor, 11-mode case

Design type	Convergence, %	Optimal convergence, %	Optimal placement (x, y)	Optimal gain	Cost function J
Continuous	100	40	(0.79, 0.82)	0.264	45.468
Discrete location	75	45	(0.80, 0.60) or (4/5, 3/5)	0.279	45.775
Percent change					0.6%

Table 4 Case of 11 actuators/sensors, 11 modes

Design type	Convergence, %	Optimal convergence, %	Optimal placement (x, y)	Optimal gain	Cost function J
Continuous	100	10	(0.20, 0.20)	0.190	5.873
			(0.20, 0.79)	0.192	
			(0.79, 0.20)	0.192	
			(0.81, 0.80)	0.192	
			(0.16, 0.50)	0.199	
			(0.84, 0.50)	0.195	
			(0.50, 0.84)	0.195	
			(0.50, 0.16)	0.199	
			(0.48, 0.58)	0.192	
			(0.53, 0.53)	0.208	
			(0.58, 0.48)	0.192	
Discrete location	40	5	(0.09, 0.18) or (1/11, 2/11)	0.281	7.614
			(0.18, 0.82) or (2/11, 9/11)	0.178	
			(0.82, 0.73) or (9/11, 8/11)	0.168	
			(0.82, 0.18) or (9/11, 2/11)	0.221	
			(0.46, 0.18) or (5/11, 1/11)	0.275	
			(0.36, 0.82) or (4/11, 9/11)	0.221	
			(0.82, 0.36) or (9/11, 4/11)	0.246	
			(0.18, 0.46) or (1/11, 5/11)	0.215	
			(0.55, 0.46) or (6/11, 5/11)	0.206	
			(0.82, 0.55) or (8/11, 6/11)	0.194	
			(0.36, 0.27) or (4/11, 3/11)	0.227	
Percent change					29.6%

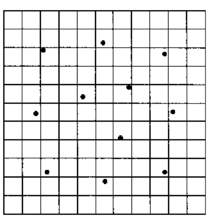


Fig. 9 Optimal placement for 11 actuators/sensors for 11 modes where $J = 5.873$ and gains $= \{0.190, 0.192, 0.192, 0.192, 0.192, 0.192, 0.195, 0.195, 0.199, 0.199, 0.208\}$.

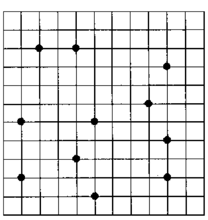


Fig. 10 Optimal discrete placement for 11 actuators/sensors where $J = 7.614$ and gains $= \{0.168, 0.178, 0.194, 0.206, 0.215, 0.221, 0.221, 0.227, 0.246, 0.275, 0.281\}$.

variables: the velocity gain and the x and y coordinates of the controller. The control penalty is $R = 10$, which yields control with relatively high authority. With only one mode, there is only one local minima, which is at the center of the membrane. It is expected that the best solution will be the one nearest the local minima. For the plot of the cost function J with respect to the placement coordinates, shown in Fig. 5, the gain of the controller was held constant. Figure 5 shows that the minimum value of J is at the crossing of the grid lines nearest to the minimum, as shown in Fig. 6, as expected. Because this membrane is symmetric about the center, this solution represents all four solutions surrounding the center.

For this example, the algorithm converged to the optimal position in Fig. 6 for 11 of the 20 times, as shown in Table 2. This example was chosen for discussion because of its simplicity. Even though it is possible in this case to simply round the value of the location to the nearest grid point, an optimization program would still have to be run to find the value of the optimal gain. The algorithm correctly rounded to the nearest coordinate position, while simultaneously finding the optimal gain. It is, in general, possible to round the values only when the function has only one absolute minimum.

Single Actuator/Sensor for 11 Modes

The second example is a single actuator with 11 modes. The optimal location for the local minima is shown in Fig. 7, which is consistent with results of Xu et al.[5] Note that the optimal discrete location of the grid is not the one closest to that minimum; in fact, it is closer to another local minimum. Figure 8 is a plot of the cost function for a constant gain. This plot is not as simple as the plot for one mode. It was necessary to search all of the local minima to find the optimal grid-line solution, and it does not necessarily exist around the absolute minimum. For this example, for the 20 random initial values used, the problem converged to the grid lines 17 of 20 times, as shown in Table 3, with 9 converging to the optimal discrete location. This problem shows the difficulty of rounding solutions effectively. By rounding in this case (and then attempting to optimize the gain), an incorrect solution would be found. The integer algorithm converged consistently to only three solutions, and all of the convergent values had performance function values below 47, with gains less than 0.27. This allows a choice of control placements with very little difference in control cost, which shows the effectiveness of finding minimal discrete locations.

11 Actuators/Sensors for 11 Modes

In this example, the number of actuators is equal to the number of modes in the structural model. This problem was chosen because it is the case with the most variables to optimize. There are 33 optimization variables, 22 of which are integer variables. As in the preceding example, the control penalty is $R = 10$. The nonlinear-programming algorithm is executed with 20 computer-generated random initial values for the optimization variables.

The noninteger example shown in Fig. 9 is again consistent with Xu et al.[5] For this example, the algorithm converged to the grid pattern 9 of 20 times. Xu et al.[5] found five different results for this problem without discrete locations. The increase in the number of actuators and modes results in many possible solutions. Because of the increasing number of combinations and 100 possible interior points to choose from, the optimal discrete location, as shown in Fig. 10, was found only once. In fact, each of the nine solutions were different. In only one case was the value of the cost function higher that 10. What is also very interesting is that, whereas the gains for the minimum solution found by Xu et al.[5] were all near 0.2, the gains for the minimum grid-line solution were in the range 0.178 –0.281. Because of the restrictions on actuator/sensor placement, the range of gains was increased. In general, it is advantageous for controllers not to fall on mode lines where a mode would be uncontrollable and unobservable; this greatly effected the placement and gains of the present controllers. It can be shown that several mode lines come very close to the grid lines. The convergence rates can be found in Table 4.

Though the convergence value was lower for this example (as expected), it is shown that this algorithm can work with larger problems. This example shows the advantage of integer programming in control because it would be next to impossible to try all 100 possible locations for 11 actuators/sensors and optimize gains.

Conclusions

A method is presented for the optimal design of placement and gains of actuators and sensors in output feedback control systems at discrete locations. This method used an integer transformation to extend the method of Xu et al.[5] It has been shown that the present method is efficient and can find optimal solutions on an evenly spaced grid pattern. The method can handle a relatively large number of mixed integer variables.

The iterations for the discrete case were approximately one and a half as long than for the nondiscrete case. This was observed even though the number of iterations to find convergent solutions was maintained or decreased. The added time for nonconvergent solutions increased the overall running time by about a factor of three.

References

[1]Arbel, A., “Controllability Measures and Actuator Placement in Oscillatory Systems,” *International Journal of Control*, Vol. 33, No. 3, 1981, pp. 565–574.

[2]Baruh, H., and Meirovitch, L., “On the Placement of Actuators in the Control of Distributed Parameter Systems Applicable to Large Flexible Spacecraft Design,” AIAA Paper 81-0638, 1981.

[3]Sepulveda, A. E., and Schmit, L. A., Jr., “Optimal Placement of Actuators and Sensors in Control Augmented Structural Optimization,” *International Journal for Numerical Methods in Engineering*, Vol. 32, 1991, pp. 1165–1187.

[4]Schulz, G., and Heimbold, G., “Dislocated Actuator/Sensor Positioning and Feedback Design for Flexible Structures,” *Journal of Guidance, Control, and Dynamics*, Vol. 6, No. 5, 1983, pp. 361–367.

[5]Xu, K., Warnitchai, P., and Igusa, T., “Optimal Locations and Gains of Sensors and Actuators for Feedback Control,” AIAA Paper 93-1660, 1993.

[6]Rao, S. S., *Optimization: Theory and Application*, Wiley Eastern, Ltd., Daryaganj, India, 1979, Chap. 10.

[7]Hansen, P., Jaumard, B., and Mathon, V., “Constrained Nonlinear 0-1 Programming,” *Operations Research Society of America Journal on Computing*, Vol. 5, No. 2, 1993, pp. 97–118.

[8]Nemhauser, G. L., and Wolsey, L. A., *Integer and Combinatorial Optimization*, Wiley, New York, 1988, Chap. 2.

[9]Abdullah, M., “Active Control of Tall Buildings,” Ph.D. Thesis, Civil Engineering, Northwestern Univ., Evanston, IL, June 1994.

[10]Balas, M. J., “Feedback Control of Flexible Systems,” *IEEE Transactions on Automatic Control*, Vol. AC-23, No. 4, 1978, pp. 673–679.

[11]Meirovitch, L., *Dynamics and Control of Structures*, Wiley, New York, 1990, Chap. 3.

[12]Levine, W. S., and Athans, M., “On the Determination of the Optimal Constant Output Feedback Gains for Linear Multivariable Systems,” *IEEE Transactions on Automatic Control*, Vol. AC-15, No. 1, 1970, pp. 44–48.

[13]Bellman, R., *Introduction to Matrix Analysis*, 2nd ed., McGraw–Hill, New York, 1970, p. 174.

[14]Bartels, R. H., and Stewart, G. W., “Solution of the Matrix Equation $AX + XB = C$,” *Communications of the ACM*, Vol. 15, No. 9, 1972, pp. 820–826.

[15]Balas, M. J., “Trends in Large Space Structure Control Theory: Fondest Hopes, Wildest Dreams,” *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 3, 1982, pp. 522–535.

[16]Fletcher, R., and Powell, M. J. D., “A Rapidly Convergent Descent Method for Minimization,” *Computer Journal*, Vol. 6, No. 2, 1963, pp. 163–168.

[17]Timoshenko, S., Young, D. H., and Weaver, W., Jr., *Vibration Problems in Engineering*, Wiley, New York, 1974, Chap. 3.

G. M. Faeth
Editor-in-Chief