AIAA JOURNAL
Vol. 40, No. 7, July 2002

Structural Damage Detection by Genetic Algorithms

Kaazem Moslem* and Ramin Nafaspour†
Ferdowsi University, 91775-1111 Mashad, Iran

A two-stage procedure utilizing incomplete measurements to assess the location and extent of structural damage is presented. In the first stage, candidate damaged elements are identified using the residual force method. Based on a priori knowledge from the first stage, the damage extent is determined from candidate elements using a proposed optimization scheme based on the method of simulated evolution.

Introduction

HEALTH monitoring of structures, especially large space structures intended for long-term operation, is essential in developing short-term or long-term repair/implementation plans. Because the expense of maintenance, replacement, and time out of service is costly, it is advantageous to develop methods that can detect, locate, and estimate the extent of the damage with the least disruption of the operation process.

System identification techniques have been extensively employed by many researchers for the assessment of structural integrity.[1–8] The theoretical background of system identification methods is that the differences in structural dynamic characteristics between the undamaged and damaged structures are translated into the changes of property matrices (stiffness/mass) that reveal the structural damage.

The problem of incomplete measurement is a common issue in system identification techniques used to address structural damage detection. It is impossible to measure experimentally the structural response at all degrees of freedom (DOF) and in all modes of vibration. Another difficulty is that the measured modes used for the damage detection are corrupted by the measurement noise. This factor makes it hard to compare responses of damaged structures with those of undamaged structures, especially when the differences between them are not considerable.

The difficulties cited have been the subject of some current works[6–10] and created the motivation for the present study. The output error approach of system identification is employed in this work to determine changes in the undamaged structure necessary to minimize differences between the measured and predicted responses. One important advantage of this approach is that the complete set of modes or displacements is not needed because the objective function involves only the difference between components of these vectors. This approach was used in Ref. 7, in which nonlinear mathematical programming methods were employed to determine a solution to the unconstrained optimization problem proposed by them.

A major drawback of nonlinear mathematical programming algorithms is that they are susceptible to convergence to local optima due to nonconvexities of the design space.[7] In addition, these algorithms depend on the existence of derivatives of the objective functions, thus, in a complex or a discontinuous search space it will be very difficult or sometimes impossible to obtain an optimum solution.

In this paper, genetic algorithms have been utilized for minimization of objective functions. Because the size of search space is also a critical issue when dealing with genetic algorithms, a few strategies are introduced to reduce the size of the search space. An attempt is first made to improve the convergence of genetic algorithms. To overcome the difficulties of incomplete measurements in large structures, a systematic procedure based on the work of Kim and Bartkowicz[8] is presented next to locate possible areas of damage. The result is a two-stage damage detection method that constitutes the basis of this study.

This study includes only numerical simulations and assumes that the initial analytical model is an accurate representation of the undamaged structure.

Theoretical Background

The free vibration eigenvalue problem for an N-DOF undamped dynamic system is given by

$$\left[\boldsymbol{K} - \omega_i^2 \cdot \boldsymbol{M}\right] \cdot \{\phi_i\} = 0, \qquad i = 1, 2, \ldots, N \tag{1}$$

where $\boldsymbol{M}$ and $\boldsymbol{K}$ are the $N \times N$ mass and stiffness matrices, respectively; ω_i is the ith eigenvalue; and $\{\phi_i\}$ is the corresponding ith mode shape of the size $N \times 1$.

This equation indicates that a change in the property matrices results in a change in the response. However, from a damage assessment point of view, it is important to relate these differences to changes in specific elements of the property matrices. Assuming the mass matrix is constant because internal structural damage usually does not result in a loss of materials, Hajela and Soeiro[7] used the stiffness matrix in a functional form as follows:

$$\boldsymbol{K}_i = f(A_i, I_i, J_i, t_i, L_i, E_i, G_i) \tag{2}$$

in which A_i, I_i, and J_i are cross-sectional area, bending, and polar moments of the inertia of the ith element; t_i and L_i are the element dimensions; and E_i and G_i are the extensional and shear moduli, respectively. The net changes in these quantities due to damage were lumped into a single coefficient for each element that constituted the design variables of the optimization problem.

In the present work, the functional form of Eq. (2) is considered for the dependency of the stiffness matrix to the properties of various elements. Furthermore, it is assumed that any change in the properties of each element due to the damage is equivalent to a decrease in its modulus of elasticity. The net change in element properties is, thus, lumped into a single coefficient d_i for each element in the form of a fraction of modulus of elasticity E_i. These coefficients (named stiffness coefficients) are multiplied by the stiffness matrices of their respective elements and constitute the vector of design variables d. The optimization problem is formulated as determining the vector of design variables d that minimizes a natural cost function representing the mismatch between the undamaged and damaged eigendata as defined in Eq. (3). Upper and lower bounds of 1 and 0 are assigned to design variables:

$$\sum_{i=1}^{r} a_i \left[1 - \left(\frac{\omega_i^m}{\omega_i^a}\right)\right]^2 + \sum_{i=1}^{r}\sum_{j=1}^{s} b_{ij}\left(\phi_{ij}^m - \phi_{ij}^a\right) \tag{3}$$

The undamaged eigenvalue for the ith mode is ω_i^a and ϕ_{ij}^a is the jth element of the ith eigenvector from the undamaged modal matrix ϕ. Superscript m indicates a measured quantity obtained from damaged

Received 30 December 2000; revision received 15 December 2001; accepted for publication 4 January 2002.

*Assistant Professor, Department of Civil Engineering.
†Graduate Student, Department of Civil Engineering.

structure. The positive coefficients a_i and b_{ij} allow for individual weightings in the objective function. These weightings are set to unity, representing the case where all measurements are assumed at the same level of uncertainty. The summation upper limits r and s represent the number of eigenvalues/eigenvectors and elements of the eigenvectors, respectively, from the measured data.[3]

Genetic Algorithms

Genetic algorithms (GAs), which take advantage of random selections in a directed optimization process, were established as valid search algorithms by Holland.[11] Because of their flexibility, globality, parallelism, simplicity, and good problem-solving capability, GAs have received increasing applications in a wide variety of fields with promising results.

When GAs are used in an optimization process, first an initial population is generated at random, or chosen heuristically, as a set of candidate solutions. Design variables are coded onto strings that are called "chromosomes." Each string represents a solution point in the search space, and is composed of characters that are analogous to genes. The initial population is evaluated to obtain the population statistics such as minimum, maximum, and average fitness values, as well as the sum of the population fitness for further utilization. A new population is reproduced on the basis of these evaluations through genetic operators. Genetic operators contain selection, crossover, mutation, inversion, dominance, etc. Therefore, a newly generated population replaces members of the old population and enters the evaluation stage again as the cycle of evaluation and reproduction continues. The new population usually has higher fitness values, which means that the population improves from generation to generation. A search concludes when it reaches the prescribed maximum generation, or when it satisfies other stopping criteria.

Though GAs have selection, crossover, and mutation operators in common, their capabilities to simulate natural processes are different. In the present work, we take advantage of steady-state genetic algorithms (SSGA), which have better performance with respect to custom GAs.[12] The following sections provide details of such algorithms as implemented in this work.

Representation Scheme

Among the different kinds of representation, the binary digit string representation is chosen in the present study. The design variables are coded onto fixed-length binary digit string representation, which is constructed over the binary alphabet $\{0, 1\}$ and concatenated head to tail to form a long string, the aforementioned chromosome. If each variable is represented by λ bit string, then the string will contain 2^{λ} discrete values. The total length of the chromosome is obtained by summing the lengths of strings λ_i.

Design Variables

Because the design variables considered in this study are continuous, it is necessary to put the discrete values to a continuous form. If the required accuracy assigned to a variable is denoted by ε, then the string length is calculated from the following equation:

$$\lambda \geq \log_2[(y^u - y^l)/\varepsilon] \tag{4}$$

where y^u and y^l are the upper and lower bounds of the variable y, respectively. For example, if $\varepsilon = 0.01$, $y^u = 1$, and $y^l = 0$, then $\lambda \geq 7$. Thus, the binary string must first be decoded to an unsigned decimal integer. The physical value of the variable y is then calculated from the following equation:

$$y = y^l + I(y^u - y^l)/(2^{\lambda} - 1) \tag{5}$$

where I is an unsigned decimal integer.

Fitness Function

Fitness is used to allocate reproductive trails and, thus, is some measure of worthiness to be maximized. This means that strings with higher fitness value will have higher probability of being selected as parents. Therefore, we have to transform the objective function minimization to the fitness function maximization problem. To accomplish this, we subtract the strings' fitness in each population from the highest fitness value within the population to obtain the new population fitness.

Selection

Selection, which is sometimes called reproduction, is equivalent to the survival of the fittest. In other words, the probability that the fittest strings (individuals) are selected for the next generations is higher than that of other strings.

Selection operator samples from current population and copies strings, with regard to their fitness, to generate the next population. Here, the well-known weighted roulette wheel method is used for selection. In this method, each string of the population has a place on the wheel proportional to its fitness. When the wheel is twisted each time, the random stopping point will be over one of individuals. This is done to the number of required copies.

In custom GAs, the number of copies is equal to the number of population individuals, that is, in each generation the whole population is replaced. This leads to large number of fitness function evaluations. Besides, it may corrupt the structure of the population, prohibiting the transfer of good schemata to the next generation and, consequently, the convergence of the algorithm. If a small generation gap is used, the aforementioned difficulties will be removed.[12] It is, thus, sufficient to select only two strings in each generation as parents. The generated offspring will then be replaced with two of the worst strings in the current population to generate the new population. The procedure finds a steady-state convergence by doing this.

A vital issue in the selection stage is the necessity to scale the fitness function values. On one hand, there is a great tendency for extraordinary individuals to dominate in the selection process. On the other hand, during the final steps of selection, when the population has largely converged, competition between individuals is decreased, which causes wandering.[13] In both cases, scaling of fitness values will enhance the results. The power scaling of fitness values[14] has been used in this work according to the following equation:

$$f' = [1 - (f_{\max} - f)/(f_{\max} - f_{\min})]^3 \tag{6}$$

in which $f_{\min}$ and $f_{\max}$ are the minimum and maximum fitness values of the population, respectively, and f and f' represent initial and scaled fitness values.

Crossover

By the use of the crossover operator, parts of parent strings are exchanged to generate offspring strings. There are many kinds of crossovers, such as one-point, two-point, three-point, uniform, etc. Typical crossovers are shown in Fig. 1.

To execute crossover in GAs, a probability must be assigned. With regard to the importance of crossover in the algorithms, this probability must be considered relatively high.

Mutation

The mutation operator arbitrarily alters the gene value according to a predetermined probability. For binary digit string representation, the mutation operator flips the bit from 1 to 0, or vice versa, on a bit-by-bit basis. If the mutation probability is very low, then the

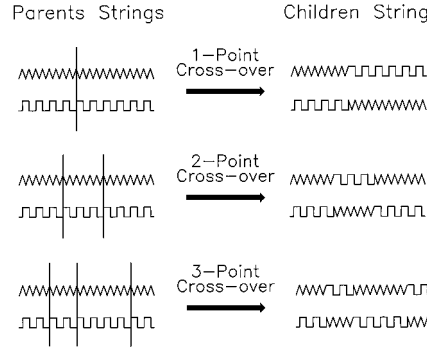

Fig. 1 Types of crossover.

algorithms often get trapped at local optima. However, if the mutation probability is high, the algorithms will degenerate to a random search. Mutation operators introduce diversity and reflect features that are not presented in the current population and, therefore, can prevent premature convergence.

Improved GA

The number of design variables in GAs is a critical factor in obtaining an optimum solution. Let binary string representation be used for the design variables. Concatenating these strings head to tail, a chromosome is formed that represents a point in the search space. With the number of strings (design variables) denoted by n, and the number of genes in each string by m, the search space will consists of $2^{m \times n}$ cases because each gene may take a value of 0 or 1. If one can make the product of $m \times n$ smaller, without significant reduction in the accuracy of the problem, then the search space will become smaller and the probability of finding the optimum solution will be increased accordingly.

An attempt was made here to reduce the number of variables n according to the concept of passive variables.[15] Variables that coincide to their upper or lower bounds in several generations are considered passive and are left out in the next generations. The remaining variables are, thus, active variables. This strategy, in spite of its generality, did not show reliability in our problem where the design variables are members stiffness coefficients d_i. This is because, in some cases, the stiffness coefficients of damaged members are close to their boundary values. Treating these variables as passive will cause the failure of the search.

To improve the performance of GAs, we used another strategy in which, through a multistep processes, the number of genes m were reduced instead. To accomplish this goal, relatively short length strings were first produced using Eq. (4) with an initial upper and lower bounds, and an accuracy assigned to each variable. The GAs were then executed for the first time. This approach will make the search space smaller, which in turn speeds up the search process and increases the possibility of obtaining approximate solutions. In the next step, the length of strings were kept unchanged, but the difference between the upper and lower bounds were taken to be one-half of their differences in step 1. To locate the center of the new bounds, the approximate solutions obtained in step 1 were used. A schematic of the procedure is shown in Fig. 2. In cases where the approximate solutions assume values close to their initial upper or lower bound values, the new upper or lower bound value will be limited to the initial value. With these new values, GAs are performed once again to produce more accurate solutions. The procedure is repeated several times until the desired accuracy is achieved. Typically, a total of three iterations results in good solutions.

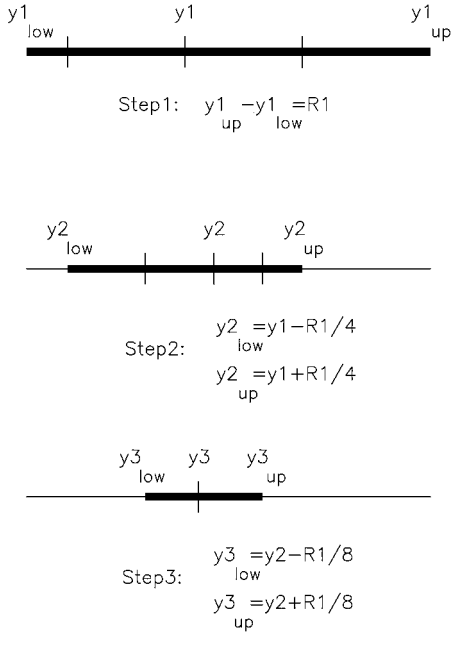

Fig. 2 Multistep reduction of design variable bounds.

Note that two major factors greatly influence the success of this procedure. The first is the required accuracy assigned to the variables. Equation (4) indicates that this parameter affects the string length. Because initial bounds are constant, if ε is chosen, such that the strings lengths become too short, the process will not cover optimum solutions because the search space becomes too small. The second factor (not as important as the first one) is the number of generations considered in the algorithm. This number should be high enough to result in good convergence toward optimum solutions in step 1.

Reducing the Size of Search Space

To locate the possible damage areas, measured mode shapes, natural frequencies of the damaged structure, and those calculated from the analytical model were exploited. The dynamic residual method was used. Consider the characteristic equation of a damaged structural system as follows:

$$\left[\boldsymbol{K}_d - \omega_{d_i}^2 \cdot \boldsymbol{M}_d\right] \cdot \left\{\phi_{d_i}\right\} = 0 \tag{7}$$

where $\boldsymbol{K}_d$ and $\boldsymbol{M}_d$ represent the stiffness and mass matrices of the damaged structure, respectively, and ω_{d_i} and ϕ_{d_i} denote its ith natural frequency and eigenmode, respectively. If the stiffness and the mass of the undamaged model denoted by $\boldsymbol{K}_a$ and $\boldsymbol{M}_a$, respectively, one can write

$$\boldsymbol{K}_d = \boldsymbol{K}_a - \Delta\boldsymbol{K}, \qquad \boldsymbol{M}_d = \boldsymbol{M}_a - \Delta\boldsymbol{M} \tag{8}$$

in which $\Delta\boldsymbol{K}$ and $\Delta\boldsymbol{M}$ are the changes in the stiffness and mass matrices, respectively, due to the damage. Substituting Eq. (8) into Eq. (7) gives

$$\left[\boldsymbol{K}_a - \omega_{d_i}^2 \cdot \boldsymbol{M}_a\right] \cdot \left\{\phi_{d_i}\right\} = \left[\Delta\boldsymbol{K} - \omega_{d_i}^2 \cdot \Delta\boldsymbol{M}\right] \cdot \left\{\phi_{d_i}\right\} = \boldsymbol{R}_i \tag{9}$$

The right-hand side of Eq. (9) is defined as dynamic residual. Assuming the mass matrix remains unchanged, that is, $\Delta\boldsymbol{M} = 0$, one obtains

$$\left[\boldsymbol{K}_a - \omega_{d_i}^2 \cdot \boldsymbol{M}_a\right] \cdot \left\{\phi_{d_i}\right\} = \Delta\boldsymbol{K} \cdot \left\{\phi_{d_i}\right\} = \boldsymbol{R}_i \tag{10}$$

or

$$\boldsymbol{Z}_{d_i} \cdot \left\{\phi_{d_i}\right\} = \boldsymbol{R}_i \tag{11}$$

where

$$\boldsymbol{Z}_{d_i} = \boldsymbol{K}_a - \omega_{d_i}^2 \cdot \boldsymbol{M}_a$$

If the stiffness matrix does not experience any changes, the right-hand side of Eq. (10) is zero. However, if some degrees of freedom are affected by damage, the corresponding elements in the residual force vector $\boldsymbol{R}_i$ will be nonzero or a large value.

Because of incomplete measurement, the order of the undamaged model and measured responses are not the same. Thus, it is necessary to use model reduction/expansion techniques to make them compatible. This introduces some errors in the calculations that will make the interpretation of the residual force vector $\boldsymbol{R}_i$ more difficult.

To circumvent the problem just discussed, we took advantage of the definition of damage vector.[5] Equation (11) can be rewritten in the following form:

$$R_i^j = Z_{d_i}^j \cdot \left\{\phi_{d_i}\right\} = \left\|Z_{di}^j\right\| \cdot \left\|\phi_{d_i}\right\| \cdot \cos\left(\theta_i^j\right) \tag{12}$$

where R_i^j represents the jth DOF of the damage vector in mode i, Z_{di}^j is the jth row of the Z_{d_i} matrix, and θ_i^j denotes the angle between Z_{di}^j and $\{\phi_{d_i}\}$ vectors. $R_i^j = 0$ corresponds to $\theta_i^j = 90$ deg, provided that measurements are free of noise. Errors in modal measurements cause θ_i^j to deviate slightly from 90 deg; However, this deviation will be more pronounced at the DOF affected by the damage. From deviation of these angles from 90 deg, one can locate the possible areas of damage. Another advantage of this method is that for a particular DOF the angles of force vectors calculated for different modes can be added together to increase the reliability of damage detection in that DOF.

Two-Stage Damage Detection Method

This procedure systematically locates possible damage areas in the first stage. In the second stage, the candidate damaged elements are further examined to determine their extent of damage.

Stage 1: Locating Possible Areas of Damage

In this stage, the order of the undamaged model and the number of DOF at which measurements are taken are made compatible. This is accomplished by considering a set of intermediate DOF. The order of the undamaged model, that is, mass and stiffness matrices, is reduced, and the order of measured eigenmodes (mode shapes) is expanded accordingly to these intermediate DOF.[8] Damage vectors are then calculated from Eq. (12), and areas of possible damage are located.

Stage 2: Determining Extent of Damage in Members

In the second stage, stiffness coefficients of the members in possible damaged areas are chosen as design variables, and the objective function defined in Eq. (3) is minimized using SSGAs. The undamaged model is refined such that the difference between the model response and measured response assumes a least value. Final values of design variables will indicate the extent of the damage in the members.

Numerical Studies

To validate the damage detection procedure proposed in this paper, several planar structures were examined. The results of this examination for two truss structures, one small and the other a relatively large truss, are presented here. The responses of damaged structures were simulated by reducing the initial stiffness of one or several members of the trusses in different scenarios. Random noise was added to the measured responses to account for the influence of measurement inaccuracy.

Three types of measurement noise, α, β, and γ, were introduced in the simulations: frequency random scale noise of $\alpha\%$, mode shape random scale noise of $\beta\%$, and mode shape random bias noise of $\gamma\%$ (Ref. 8). Mode shape random bias noise was added to account for mode shape errors near nodal points.

The proposed two-stage damage detection method was only applied to the larger truss. For the small truss, due to limited number of design variables, the location and the extent of the damage was identified in one stage only. A number of FORTRAN programs, developed by the authors, were used to implement the procedures outlined in the preceding sections. Model reduction/expansion was performed using the algorithms developed by Guyan[16] and Alvin.[4] More exact model reduction techniques such as IRS[17] and SEREP[18] were also used, but they led to worse results compared to Guyan method.[16]

In each scenario, the objective function was minimized using SSGAs. Two different cases were considered. In case 1, the commonly used SSGAs were employed, whereas in case 2, the proposed improved GAs (multistep procedure) were utilized. In Tables 1–7, GA1 and GA2 refer to the results obtained in case 1 and case 2, respectively.

To better demonstrate the effectiveness of the SSGAs proposed in this paper for the assessment of structural damage, Broydon–Fletcher–Goldfarb–Shanno variable metric method (see Ref. 19)

Table 1 Estimation of members' stiffness coefficients in TR15-1 damage scenario

	First 5 modes, $\alpha=3\%, \beta=3\%, \gamma=1\%$			First 6 modes, $\alpha=3\%, \beta=3\%, \gamma=1\%$			12 modes, $\alpha=3\%, \beta=3\%, \gamma=1\%$			Exact
Member no.	NPM	GA1	GA2	NPM	GA1	GA2	NPM	GA1	GA2	value
1	0.99	1.00	1.00	1.00	1.00	1.00	0.96	1.00	1.00	1.00
2	1.00	0.88	1.00	1.00	0.97	1.00	0.97	1.00	0.99	1.00
3	0.90	1.00	0.98	1.00	0.97	1.00	1.00	1.00	1.00	1.00
4	0.91	1.00	1.00	1.00	0.97	1.00	1.00	0.97	1.00	1.00
5	1.00	0.97	0.98	1.00	1.00	0.98	1.00	0.97	1.00	1.00
6	1.00	1.00	1.00	1.00	1.00	0.98	1.00	0.97	0.98	1.00
7	1.00	1.00	1.00	1.00	0.97	0.98	1.00	1.00	0.99	1.00
8	1.00	1.00	0.92	0.98	0.88	0.98	0.96	1.00	0.98	1.00
9	1.00	1.00	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	1.00	1.00	0.99	1.00	1.00	1.00	1.00	0.97	1.00	1.00
11	0.68	0.97	1.00	0.95	0.97	1.00	1.00	0.97	0.98	1.00
12	0.70	1.00	0.98	1.00	0.98	0.98	1.00	0.97	0.98	1.00
13	0.78	0.85	0.73	0.66	0.79	0.75	0.66	0.73	0.73	0.75
14	0.98	0.73	0.88	1.00	0.82	0.84	1.00	0.82	0.87	0.85
15	1.00	1.00	0.98	1.00	0.97	0.98	0.99	0.98	1.00	1.00
Value of objective function	0.22441	0.05632	0.01695	0.05499	0.02716	0.01653	0.12455	0.04191	0.03681	

Table 2 Estimation of members' stiffness coefficients in TR15-2 damage scenario

	6 modes, $\alpha=0\%, \beta=0\%, \gamma=0\%$			6 modes, $\alpha=3\%, \beta=3\%, \gamma=1\%$			6 modes, $\alpha=5\%, \beta=7\%, \gamma=2\%$			Exact
Member no.	NPM	GA1	GA2	NPM	GA1	GA2	NPM	GA1	GA2	value
1	0.26	0.11	0.11	0.23	0.10	0.09	0.22	0.31	0.13	0.10
2	0.41	0.97	0.97	0.40	0.95	0.94	0.54	0.49	1.00	1.00
3	0.38	0.52	0.50	0.40	0.52	0.51	0.47	0.52	0.52	0.50
4	1.00	0.97	1.00	1.00	0.97	0.99	0.80	1.00	1.00	1.00
5	1.00	0.97	1.00	1.00	0.88	0.93	1.00	1.00	1.00	1.00
6	0.73	0.97	1.00	0.75	1.00	1.00	1.00	0.64	0.87	1.00
7	0.46	1.00	0.98	0.48	1.00	1.00	1.00	1.00	1.00	1.00
8	0.90	1.00	0.98	0.93	0.97	0.98	0.72	1.00	0.91	1.00
9	0.73	0.97	1.00	0.73	0.97	0.97	0.56	0.52	0.84	1.00
10	1.00	0.97	0.98	1.00	1.00	1.00	1.00	1.00	0.87	1.00
11	0.75	1.00	0.98	0.76	1.00	1.00	0.49	0.97	0.96	1.00
12	1.00	0.52	0.50	1.00	0.52	0.52	0.46	0.61	0.47	0.50
13	1.00	0.13	0.11	1.00	0.13	0.12	0.97	0.11	0.12	0.10
14	0.64	0.95	0.97	0.67	0.95	0.98	0.78	0.88	0.84	1.00
15	0.99	0.92	0.96	1.00	0.91	0.92	1.00	1.00	0.88	1.00
Value of objective function	0.39572	0.00377	0.00116	0.37420	0.00912	0.00812	0.50266	0.11476	0.02932	

Table 3 Estimation of members' stiffness coefficients in TR15-3 damage scenario

	4 modes, $\alpha=0\%, \beta=0\%, \gamma=0\%$			4 modes, $\alpha=3\%, \beta=3\%, \gamma=1\%$			4 modes, $\alpha=5\%, \beta=7\%, \gamma=2\%$			Exact
Member no.	NPM	GA1	GA2	NPM	GA1	GA2	NPM	GA1	GA2	value
1	1.00	1.00	1.00	1.00	1.00	1.00	0.95	0.97	0.98	1.00
2	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00	0.97	0.98	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	0.99	0.97	1.00	1.00	1.00	1.00	1.00
5	0.99	0.73	0.70	1.00	0.79	0.80	1.00	0.85	0.76	0.70
6	0.99	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00
8	0.96	1.00	1.00	1.00	1.00	1.00	0.92	0.88	0.91	1.00
9	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.98	0.97	1.00
10	1.00	0.97	0.98	1.00	0.88	1.00	1.00	1.00	0.94	1.00
11	0.98	1.00	1.00	0.99	1.00	1.00	0.81	0.88	0.90	1.00
12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	0.47	0.50	0.51	0.44	0.49	0.45	0.42	0.40	0.45	0.50
14	0.99	1.00	0.99	0.99	0.97	1.00	0.99	1.00	1.00	1.00
15	1.00	0.58	0.58	1.00	0.58	0.66	1.00	0.92	0.74	0.60
Value of objective function	0.00586	0.00009	0.00001	0.00895	0.00537	0.00490	0.02111	0.01955	0.01923	

Table 4 TR40-1 damage vector

DOF no.	Damage vector	Possible damage[a]
1	18.62	——
2	36.54	——
3	57.22	×
4	84.16	×
5	77.67	×
6	81.61	×
7	100.00	×
8	38.17	——
9	5.74	——
10	12.76	——
11	18.94	——
12	35.34	——
13	31.78	——
14	37.94	——
15	50.28	×
16	17.88	——

[a] × Indicates possible damage.

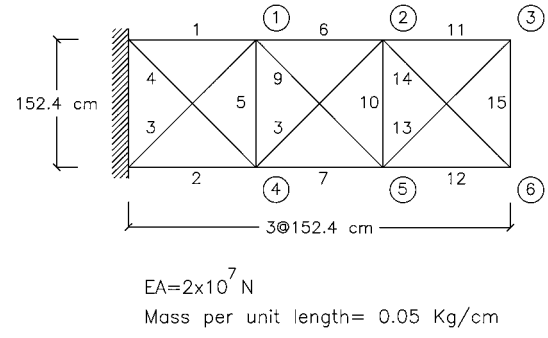

Fig. 3 Plane truss TR-15.

was also employed to determine a solution to the unconstrained optimization problem. This method is a nonlinear mathematical programming method (NPM). Results obtained from this analysis are marked by NPM in Tables 1–7.

The values of genetic parameters used in implementation of GAs are as follows: The number of design variables is equal to the number of selected truss elements, string length is equal to 5 or 6, chromosome length is equal to the number of design variables times string length, population in each generation equals chromosome length, mutation probability is 0.05, crossover probability is 0.9–1.0, and the maximum number of generations in each iteration is 5000.

One-, two-, and three-point crossover, as well as uniform crossover, were examined. Among them, the two-point crossover gave the best results. Mutation had a more pronounced effect than crossover. A crossover probability between 0.9–1.0 did not affect the results significantly, whereas a mutation probability much less than 0.05 caused an early convergence of the algorithm.

Example 1: 15-Bar Planar Truss

The 15-bar truss with 12 DOF is shown in Fig. 3. To account for the problem of incomplete measurement, it was assumed that the measurements are taken only at the six vertical DOF. In this example, the number of design variables was set equal to the number of truss members.

In the first scenario, TR15-1, damage was introduced in elements 13 and 14 by reducing their stiffness to 75 and 85% of the initial values, respectively. Measurement noise was simulated and entered into the analysis. Measured and analytically predicted responses were used to detect the location and the extent of the damage in the structure. The results are shown in Table 1. For the first five eigenmodes, both GAs and NPM identified the damage in member 13. However, the performance of GA2 is more accurate than those of both GA1 and NPM, as can be observed by comparing the values of design variables and the values of the objective functions. When six eigenmodes were utilized, NPM was only able to locate the damage in member 13, whereas both GA1 and GA2 identified the damaged elements, and the extent of the damage was assessed with good accuracy. The values of objective functions show the better performance of GA2. Inadequacy of NPM in detecting the damage is clear from Table 1 because complete measurement at 12 DOF and full set of eigenmodes could not locate the damage in member 14.

In the second scenario, TR15-2, damage was introduced in members 1, 3, 12, and 13 of the 15-bar truss. Damaged members were assumed to have 10, 50, 50, and 10% of their initial stiffness, respectively. The first six modes were used to identify the damage in the structure. Two cases were considered: 1) measurements are free of noise and 2) measurements are corrupted by different levels of noise. The results are shown in Table 2. It is observed that GAs easily identify the location as well as the extent of the damage in both cases, whereas NPM, even in the noise-free case, confronts failure. Values of objective functions show the better performance of GA2. However, some differences are observed between the exact and the estimated values of members stiffness for the case where the noise level is high.

In the third scenario of the 15-bar truss, members 5, 13, and 15 were damaged by reduction of their stiffness's to 70, 50, and 60% of the initial values, respectively. The first four modes were used to identify the damage in the presence of corrupted noise. From the results shown in Table 3, it is observed that GAs detect the location and the extent of the damage with good accuracy. GA2 performs better than GA1, and NPM fails to locate the damage in members 5 and 15.

Example 2: 40-Bar Planar Truss

This structure, which has been used as a test and/or simulation model by many researchers, is examined in this section. Figure 4

Table 5 Estimation of members' stiffness coefficients in TR40-1 damage scenario

	9 modes, $\alpha=0\%, \beta=0\%, \gamma=0\%$			9 modes, $\alpha=3\%, \beta=3\%, \gamma=1\%$			9 modes, $\alpha=5\%, \beta=7\%, \gamma=2\%$			Exact
Member no.	NPM	GA1	GA2	NPM	GA1	GA2	NPM	GA1	GA2	value
4	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
26	0.99	0.75	0.80	1.00	0.94	0.86	1.00	1.00	1.00	0.80
27	0.99	1.00	1.00	0.97	1.00	1.00	0.97	0.97	0.97	1.00
28	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
29	1.00	1.00	1.00	1.00	0.97	0.95	1.00	1.00	1.00	1.00
30	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
31	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
32	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
33	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
34	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
35	1.00	0.94	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00
36	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	1.00	1.00
37	1.00	0.97	1.00	1.00	1.00	0.98	1.00	0.97	1.00	1.00
38	1.00	0.94	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
Value of objective function	0.00063	0.00006	0.00003	0.10851	0.02939	0.02812	0.10851	0.10864	0.10855	

Table 6 TR40-2 damage vector

DOF no.	Damage vector	Possible damage[a]
1	10.26	——
2	29.26	——
3	22.04	——
4	44.25	——
5	28.86	——
6	82	×
7	100	×
8	48.95	×
9	4.09	——
10	10.65	——
11	20.98	——
12	10.57	——
13	11.83	——
14	32.29	——
15	49.41	×
16	28.36	——

[a] × Indicates possible damage.

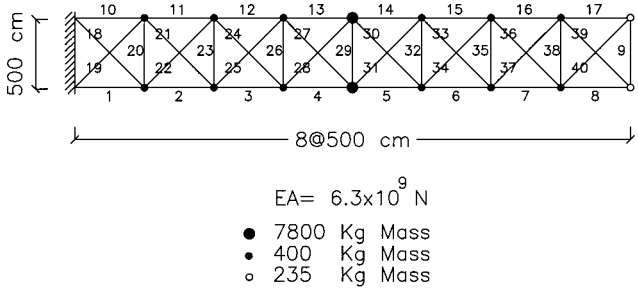

Fig. 4 Plane truss TR-40.

shows the 40-bar truss, which assumes lumped masses at its nodes. Because this structure is considered to be large, the two-stage damage detection method proposed in this paper was used for its analysis.

The truss has 32 translational DOF, from which only 8 DOF were assumed to be instrumented. The intermediate DOF were taken equal to 16. The order of the analytical model was reduced and that of the measured eigenmodes was expanded to 16 accordingly.

Selection of the intermediate DOF and the DOF to be instrumented is of great importance. If the sole objective is to increase the accuracy of the reduced model, use can be made of algorithms that select master DOF. The procedure proposed by Shah and Raymund[20] was employed in this work for the selection of master DOF. The results indicated that by increasing the accuracy of the reduced model, the detection of the damaged areas is not guaranteed.

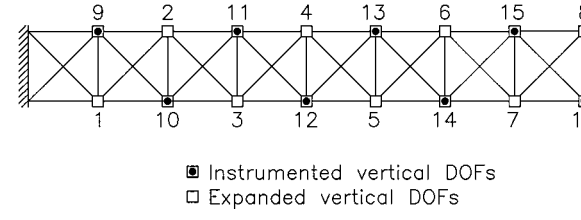

Fig. 5 Intermediate DOF.

It is, thus, of interest to select those DOF as master that not only ensure a good accuracy for the reduced model but also extend over the full length of the structure and divide it into relatively small portions such that any changes in the stiffness characteristics of these portions can be related to the DOF at their two ends. The intermediate and instrumented DOF for the 40-bar truss under consideration are shown in Fig. 5.

In the first scenario, TR40-1, the stiffness of member 26 was reduced to 80% of its initial value. Where the first nine modes are used, the results obtained in the first stage of the two-stage method are shown in Table 4. DOF having values larger than 50% of the maximum component of the damage vector were considered those affected by damage.

From the results shown in Table 4, bays 4–7 are identified as possible damaged bays. Next, the stiffness coefficients of participant members of these bays as design variables were considered, and the extent of the damage was estimated and shown in Table 5. Table 5 indicates that GA2 was able to estimate the extent of the damage in member 26 with good accuracy when the measurements are free of noise, or when the corrupted noise is of relatively low level.

In TR40-2 scenario, the stiffness of members 7, 38, and 40 were reduced to 80, 50, and 80% of their initial values, respectively. The first nine modes were used. Locations of possible damage are shown in Table 6. Bays 7 and 8 are identified as damaged bays. The stiffness coefficients of members in these bays were next considered as design variables to detect the extent of the damage in them. The results are shown in Table 7. It is observed that, when measurements are free of noise, GA2 clearly detects the extent of the damage in the members.

Table 7 Estimation of members' stiffness coefficients in TR40-2 damage scenario

	9 modes, $\alpha = 0\%, \beta = 0\%, \gamma = 0\%$			9 modes, $\alpha = 3\%, \beta = 3\%, \gamma = 1\%$			9 modes, $\alpha = 5\%, \beta = 7\%, \gamma = 2\%$			Exact
Member no.	NPM	GA1	GA2	NPM	GA1	GA2	NPM	GA1	GA2	value
7	0.76	0.83	0.80	0.76	0.83	0.82	0.76	0.82	0.80	0.80
8	1.00	0.85	1.00	1.00	0.91	0.91	1.00	0.94	0.98	1.00
16	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00	0.97	0.98	1.00	0.97	0.95	1.00
35	1.00	1.00	1.00	1.00	0.85	0.93	1.00	1.00	1.00	1.00
36	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
37	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
38	1.00	0.46	0.51	1.00	0.47	0.50	1.00	0.43	0.48	0.50
39	1.00	1.00	0.99	1.00	0.98	0.98	1.00	0.92	0.92	1.00
40	1.00	0.91	0.81	1.00	0.89	0.95	1.00	0.91	0.93	0.80
9	1.00	0.79	0.99	1.00	1.00	0.74	1.00	1.00	1.00	1.00
Value of objective function	0.00460	0.00100	0.00001	0.05515	0.0491	0.04886	0.10986	0.10278	0.10272	

In the presence of noise, however, the extent of the damage is not accurate for member 40, and minor damages are also assessed to other members such as 8, 9, and 39.

Conclusions

A two-stage damage detection method was proposed in this paper to circumvent the problem of incomplete measurements. In the first stage, possible damaged areas are detected. SSGAs are then used as an optimization tool to detect the extent of the damage in members.

SSGAs were modified to act as a multistep process, which, in turn, increased the reliability of the damage detection algorithm considerably. In many cases where nonlinear mathematical programming methods based on gradients of objective functions were not able to detect the damage, the SSGAs performed well. There were, however, some cases (not presented) in which GAs were not able to solve the intrinsic difficulty of the identification techniques as used in damage detection. Where the damaged member assumes a minimum kinetic energy in those measured modes that experience the greatest changes because of the damage, the identification of that scenario will be either very difficult or sometimes impossible.

Though the proposed algorithm has the advantage of being systematic, the inevitable errors such as measurement noise, model expansion, and especially model reduction errors presented difficulties in locating possible damaged areas. More exact model reduction techniques such as IRS[17] and SEREP[18] led to worse results compared to the Guyan method.[16]

References

[1]Chiang, D.-Y., and Lai, W.-Y., "Structural Damage Detection Using the Simulated Evolution Method," *AIAA Journal*, Vol. 37, No. 10, 1999, pp. 1331–1333.

[2]Papadopaulos, L., and Garcia, E., "Structural Damage Identification: A Probabilistic Approach," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 2138–2145.

[3]Cobb, R. G., and Liebst, B. S., "Structural Damage Identification Using Assigned Partial Eigenstructure," *AIAA Journal*, Vol. 35, No. 1, 1997, pp. 152–158.

[4]Alvin, K. F., "Finite Element Model Update via Bayesian Estimation and Minimization of Dynamic Residuals," *AIAA Journal*, Vol. 35, No. 5, 1997, pp. 879–886.

[5]Kaouk, M., and Zimmerman, D. C., "Structural Damage Assessment Using a Generalized Minimum Rank Perturbation Theory," *AIAA Journal*, Vol. 32, No. 4, 1994, pp. 836–842.

[6]Zimmerman, D. C., Smith, S. W., Kim, H. M., and Bartkowicz, T. J., "Experimental Study of Structural Damage Detection Using Incomplete Measurements," *Proceedings of the 35th AIAA Structures, Structural Dynamics, and Materials Conference*, AIAA, Washington, DC, 1994, pp. 307–317.

[7]Hajela, P., and Soeiro, F. J., "Structural Damage Detection Based on Static and Modal Analysis," *AIAA Journal*, Vol. 28, No. 6, 1990, pp. 1110–1115.

[8]Kim, H. M., and Bartkowicz, T. J., "Damage Detection and Monitoring of Large Space Structures," *Journal of Sound and Vibration*, Vol. 27, No. 6, 1993, pp. 513–519.

[9]James, G., Zimmerman, D. C., and Cao, T., "Development of a Coupled Approach for Structural Damage Detection with Incomplete Measurements," *AIAA Journal*, Vol. 36, No. 12, 1998, pp. 2209–2217.

[10]Zimmerman, D. C., Kaouk, M., and Simmermacher, T., "Structural Health Monitoring Using Vibration Measurements and Engineering Insight," *Journal of Vibration and Acoustics*, Vol. 117, No. 6, 1995, pp. 214–221.

[11]Holland, J. H., *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press, Ann Arbor, MI, 1975.

[12]Wu, S. J., and Chow, P. T., "Steady-State Genetic Algorithms for Discrete Optimization of Trusses," *Computers and Structures*, Vol. 56, No. 4, 1995, pp. 695–702.

[13]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley, Reading, MA, 1989.

[14]Wu, S. J., and Chow, P. T., "Integrated Discrete and Configuration Optimization of Trusses Using Genetic Algorithms," *Computers and Structures*, Vol. 55, No. 4, 1995, pp. 695–702.

[15]Chen, T. Y., and Chen, C. J., "Improvement of Simple Genetic Algorithm in Structural Design," *International Journal for Numerical Methods in Engineering*, Vol. 40, 1997, pp. 1323–1334.

[16]Guyan, R. J., "Reduction of Stiffness and Mass Matrices," *AIAA Journal*, Vol. 3, No. 2, 1965, p. 380.

[17]O'Callahan, J., "A Procedure for an Improved Reduced System (IRS) Model," *Proceedings of the 7th International Modal Analysis Conferences*, Union College, Schenectady, NY, 1989, pp. 17–21.

[18]O'Callahan, J., Avitabile, P., and Riemer, R., "System Equivalent Reduction Expansion Process (SEREP)," *Proceedings of the 7th International Modal Analysis Conference*, Union College, Schenectady, NY, 1989, pp. 29–37.

[19]Vanderplaats, G. N., *ADS—A Fortran Program for Automated Design Synthesis*, Ver. 1.10, Univ. of California, Santa Barbara, CA, 1985.

[20]Shah, V. N., and Raymund, M., "Analytical Selection of Masters for the Reduced Eigenvalue Problem," *International Journal for Numerical Methods in Engineering*, Vol. 18, No. 1, 1982, pp. 89–98.

A. Berman
Associate Editor