Vibration Isolation Optimization

Course No: M02-004

Credit: 2 PDH

Steven Liescheidt, P.E., CCS, CCPR

Continuing Education and Development, Inc.
22 Stonewall Court
Woodcliff Lake, NJ 07677

P: (877) 322-5800
info@cedengineering.com

SAND96-1169
Unlimited Release
Printed May 17, 1996

Discrete Optimization of Isolator Locations for Vibration Isolation Systems: an Analytical and Experimental Investigation

E. R. Ponslet and M. S. Eldred
Structural Dynamics Department
Sandia National Laboratories
Albuquerque, New Mexico 87185-0439

Abstract

An analytical and experimental study is conducted to investigate the effect of isolator locations on the effectiveness of vibration isolation systems. The study uses isolators with fixed properties and evaluates potential improvements to the isolation system that can be achieved by optimizing isolator locations.

Because the available locations for the isolators are discrete in this application, a Genetic Algorithm (GA) is used as the optimization method. The system is modeled in MATLAB™ and coupled with the GA available in the DAKOTA optimization toolkit under development at Sandia National Laboratories. Design constraints dictated by hardware and experimental limitations are implemented through penalty function techniques. A series of GA runs reveal difficulties in the search on this heavily constrained, multimodal, discrete problem. However, the GA runs provide a variety of optimized designs with predicted performance from 30 to 70 times better than a baseline configuration. An alternate approach is also tested on this problem: it uses continuous optimization, followed by "rounding" of the solution to neighboring discrete configurations. Results show that this approach leads to either infeasible or poor designs.

Finally, a number of "optimized" designs obtained from the GA searches are tested in the laboratory and compared to the baseline design. These experimental results show a 7 to 46 times improvement in vibration isolation from the baseline configuration.

Acknowledgments

This work was funded by DOE through Sandia National Laboratories under contract number AP-5115 with AMPARO Corporation, Santa Fe, NM.

The authors wish to thank David Martinez for initiating support for this research, Clay Fulcher for his advice and assistance with many practical problems, Bill Hart for his assistance with the genetic algorithm in DAKOTA, and Tom Rice for his invaluable help in preparing the experimental setup.

Content

List of Figures

List of Tables

1. Introduction

Vibration isolation systems are used in a wide variety of applications to reduce transmission of mechanical vibrations generated by noisy components or carried by the environment to a sensitive device. Examples include isolation of a laser table from floor-borne seismic disturbances, isolation of a car or airplane body from engine vibrations, and suspension systems of vehicles.

Isolation is achieved by inserting soft mechanical links ("isolators") between the subsystem containing the source of the disturbances and the subsystem to be isolated. Based on the relative sizes of these subsystems, two classes of isolation systems can be distinguished (Fig. 1).

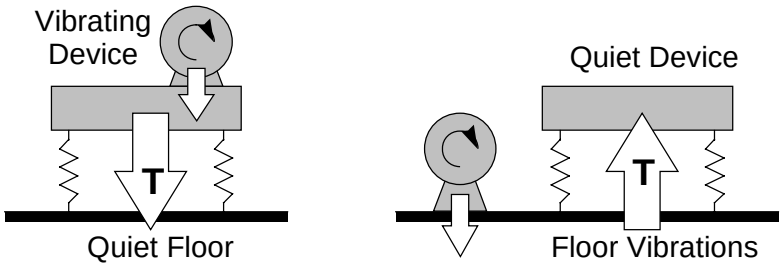

Figure 1 Vibration isolation systems.

In the first situation (left side of Fig. 1), the environment is isolated from vibrations created by a piece of machinery. A typical example is the isolation of a car body from vibrations caused by the engine. In the other, a sensitive device is protected from disturbances carried by its supporting structure. Isolation of a laser table from floor borne vibrations is a common example. In both cases, the effectiveness of the isolation system can be examined in terms of transmissibility functions, *T*, in the frequency domain. In the first class of systems, these transmissibilities can be expressed as ratios of excitation forces to forces transmitted to the floor. In the second class, they are expressed as ratios of component of floor motion to component of device motion. Note that mixed formulations are also possible. Whatever exact definition is used for *T*, its magnitude typically resembles the curve shown in Fig. 2.

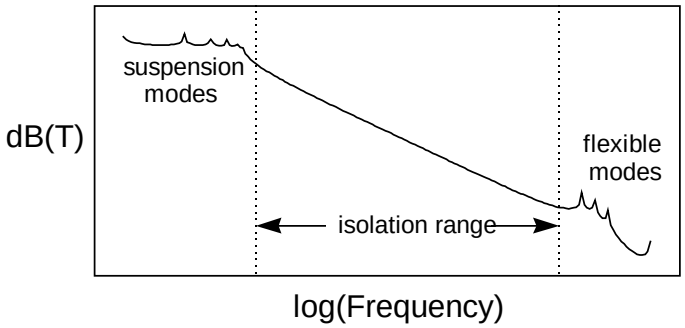

Figure 2 Typical transmission characteristics of isolation system.

Three regions can be distinguished in the figure. At “low” frequencies, resonant peaks corresponding to the suspension modes are observed. There is no isolation in this band (in fact there is amplification). Damping is usually designed into the isolators to limit the amplitudes of these peaks and avoid large responses to transients. At the “high” end of the spectrum, flexible modes of the device and/or supporting structure themselves produce other resonant peaks. In a properly designed isolation system, those two frequency ranges are separated by a wide “isolation band” where transmission decays rapidly with frequency (about -40dB/decade for a lightly damped, single stage system). A properly designed system will “see” most of the disturbance energy occur in this band.

Because of the sharp decay in the isolation band, the traditional design methodology focuses primarily on minimizing the stiffnesses of the individual isolators. This lowers the frequencies of the suspension modes and results in a corresponding drop in transmission in the isolation band (Fig. 3). The configuration of the complete system, in particular the number, locations, and orientations of the isolators, receives little attention in this approach. Actually, the system configuration is often designed to ensure decoupling between translations and rotations in the suspension modes and simplify the analysis[1].

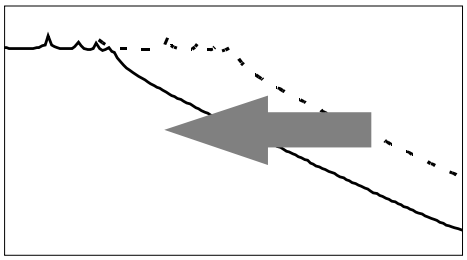

Figure 3 Improving isolation by softening the isolators.

This approach is justified for “generic” isolation problems. Namely, when the location, direction, amplitude, or frequency content of the disturbances are not well known and/or when no particular point on the isolated device more critically requires isolation than others. An example of a generic problem is that of isolating a laser table from floor vibrations: data will often not be available to accurately characterize the disturbance and the designer of the isolation system has no knowledge of what experiment will be performed on the table. In such cases, a generic isolation system (with 4 isolators, one in each corner for example) is appropriate.

However, in some applications, the disturbance(s) are very well known (rotating machinery, for example) and residual motion is critical at one or a few specific points/directions on the isolated device. As an example, consider an isolation system designed to prevent excessive transmission of vibrations from a cryocooler to a telescope mounted on a satellite structure (Fig. 4). Clearly, the source of the disturbance is well known in direction, amplitude, and frequency content. Also, to minimize jitter, residual tilting vibrations of the telescope must be minimized. Vibrations at other points/directions in the system are less critical. In other applications, the critical point/direction might be the location of a vibration-sensitive component or points/directions of strong dynamic

coupling with an elastic sub-system. In all such cases, it is expected that locations and orientations of the isolators will have a substantial effect on the isolation effectiveness[2].

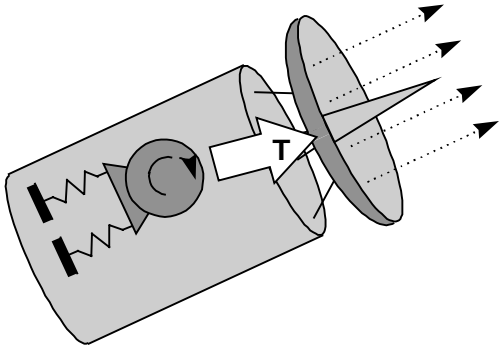

Figure 4 Non-generic isolation problem.

The present study examines this question in more detail. The authors consider the design of a 3-isolator system to minimize transmission of well characterized floor-borne perturbations to a specific point and direction on an optical table. The number and type of isolators used is fixed and their locations under the table are optimized. The optimized designs are compared to a baseline, generic configuration. These designs are then tested in the lab to validate the approach.

Because our optical table provides only a discrete grid of mounting holes for the isolators, the design variables of the optimization problem are discrete and the optimization is performed with SGOPT's[3] Genetic Algorithm (GA) available in the DAKOTA[4] optimization toolkit under development at Sandia National Laboratories.

The goal of the study is twofold: first, investigate the potential of optimization in improving the performance of vibration isolation systems and second, by exercising the GA with a real-life problem, hopefully identify critical directions for future GA research and development efforts.

2. System Description

The experimental vibration isolation setup (Fig. 5) consists of a stainless steel honeycomb sandwich optical table (Newport[5], model RS4000-36-8) measuring approximately 48 by 36 by 8.5 inches and weighing approximately 815 lbs. The table is resting on 3 steel coil spring isolators (custom designed, using springs from Associated Spring-Raymond[6], model CV2000-2500-365) whose locations under the table are the focus of the optimization. This system is in turn resting on a large, solid aluminum seismic mass (custom-made, approximately 70 by 48 by 12 inches, 4085 lb.), itself isolated from the lab floor by four air bags (Firestone Airmount® isolators[7], model 224C). Note that the seismic mass will actually be considered the "floor" in this problem. The air bag suspension is there to eliminate any unknown disturbances from the experiment but will not be part of the optimization problem. The suspension frequencies of the system on these airbags range from about 1.0 to 2.5 Hz.

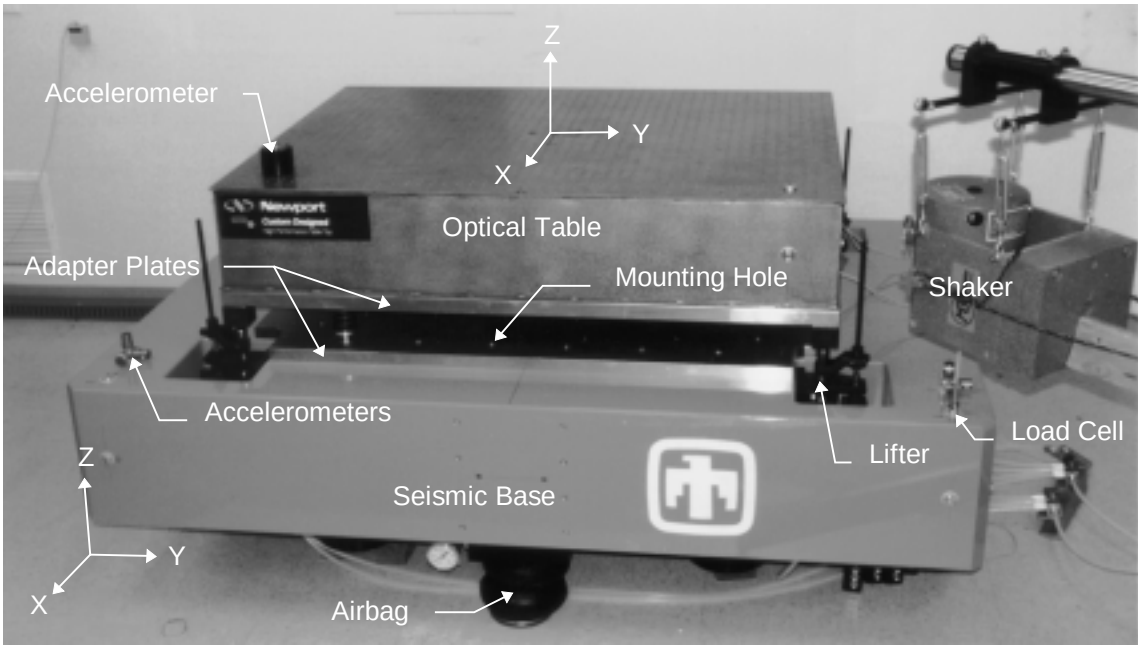

Figure 5 Experimental vibration isolation setup.

The bottom of the optical table and the top of the seismic base are fitted with identical aluminum adapter plates featuring matching arrays of 6 by 8 threaded holes (on a 6 in. grid). The spring isolators are attached to those holes with threaded rods and aluminum end-plates as shown in Fig. 6. Note that because the springs are simply resting in the end-plates, the isolators can only take compressive forces.

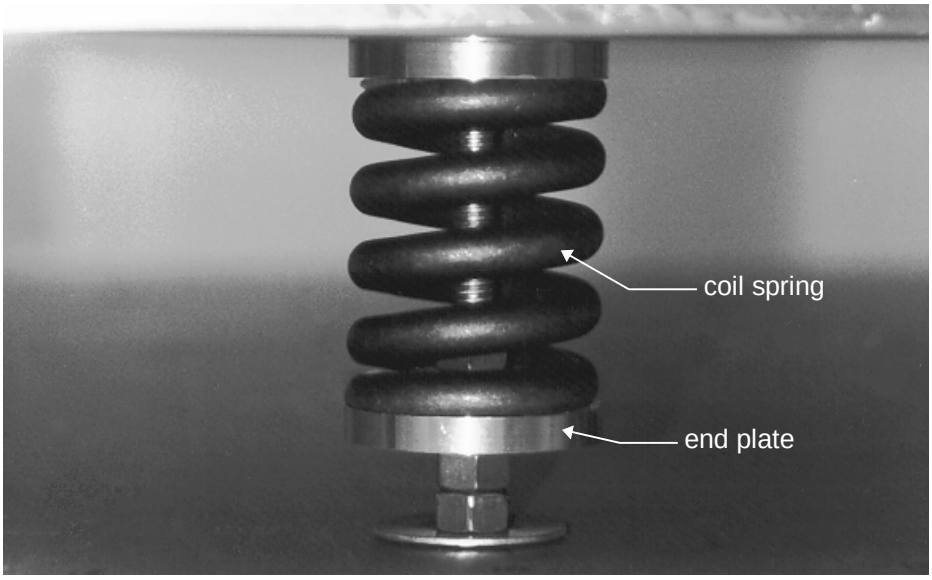

Figure 6 Steel spring isolator.

Four hydro-pneumatic lifters are attached to the seismic block, allowing convenient access to the isolators. The four corners of the aluminum adapter plate attached to the seismic block are machined to provide clearance for the lifters. This eliminates 4 of the 48 possible isolator mounting locations.

Both the seismic base and the optical table are fully instrumented for 6 d.o.f. motion pickup for system identification and model validation purposes (see Section 4.3). Four sets of 3 accelerometers (Endevco[8], model 7751-500mV/g) are mounted near the four corners of the seismic base as seen in Fig. 5. The optical table is equipped with 5 triaxial accelerometers (Endevco[8], model 63-500mV/g), embedded in the aluminum adapter plate near the four corners and the center. All accelerometers are powered by 12-channel signal conditioners (PCB[9], model 483A10). In addition, a high sensitivity seismic accelerometer (Wilcoxon Research[10], model 731, 10,000 mV/in/sec. in velocity mode) measures residual vibration at the critical point, near the front left corner of the optical table.

The "floor" (seismic base) is excited near its front right corner (Figure 4) by an electromagnetic shaker (MB Dynamics[11], model Modal 50A). The excitation force is measured by a piezoelectric load cell (PCB[9], model 208A03, 10mV/lb.).

3. Optimization Problem

The vibration isolation problem is shown schematically in Fig. 7. As mentioned previously, the goal is to find optimal locations for 3 steel spring isolators between the optical table and the seismic base. The number of isolators was set to 3 to minimize uncertainties in the experimental setup that would occur due to static indeterminacy with 4 or more isolators. By design, only 6x8=48 discrete locations are available for these isolators. Four of those locations (at the corners) are not available (see Section 2).

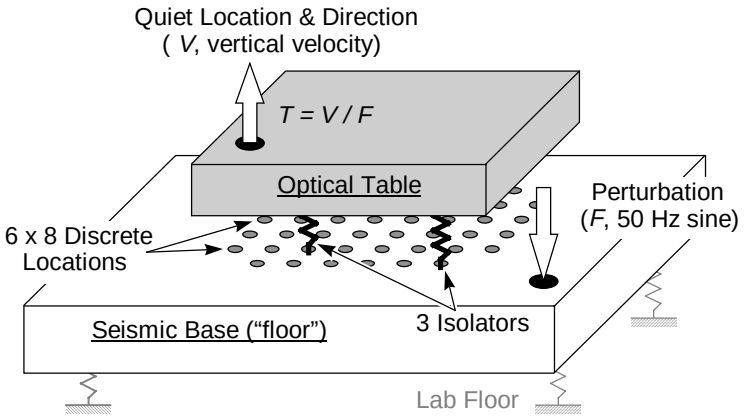

Figure 7 Vibration isolation design problem.

For simplicity, and to reduce computational expenses in the simulations, the disturbance is a pure sinusoidal vertical force *F* at a frequency of 50 Hz. This frequency was chosen to be within the isolation band, i.e. much higher than that of the suspension modes of the optical table but well below the frequency of the lowest elastic modes (150 to 200 Hz). It will be shown that the use of a single target frequency instead of a wide band does not limit the improvements to that single frequency. Instead, the optimized designs actually have improved performance across the isolation range.

The objective is to minimize the residual vertical velocity *V* at a target location on the table (Figure 7), in response to a given amplitude of perturbation force *F*. This amounts to minimizing the transfer function $T = V/F$ (μin/sec/lb) from the excitation force to the residual velocity. This function *T* will be referred to as *transmissibility* in the remainder of this report. The locations of the excitation and the critical point were chosen to create asymmetry in the problem. This is expected to lead to non-intuitive, asymmetric optimal locations for the isolators, in sharp contrast to a baseline configuration.

Practical limitations dictate a number of design constraints on this problem (the implementation of those constraints is explained in more detail in the next section):

- Depending on how the isolator locations are coded into design variables, it may be necessary to enforce the condition that the 4 locations at the corners of the 6x8 grid are not used.
- There cannot be more than one isolator at any given grid location.
- The isolators cannot be aligned on a straight line because the table would then be unstable. Note that, with 3 isolators, this condition also takes care of the previous one: if any two isolators are at the same location they are also on a straight line with the third one.
- Also, since softer isolation generally implies better performance (see Section 1) there may be a tendency for the optimizer to generate designs with very low natural frequencies (by placing the isolators very close to each other for example). This is undesirable in practice because such designs would have very large transient responses to impact and handling loads. To prevent this, a lower limit of 4.0 Hz is enforced on the first natural frequency of the optical table on its isolators. Note that this value also ensures decoupling with the suspension modes of the seismic base on its airbags (1 to 2.5 Hz).
- Because the springs are not attached to their end-plates, the static gravitational load on each isolator must be compressive.
- The static deflections of the springs cannot exceed an upper limit beyond which the springs might not be linear or might be compressed to their solid length. This limit was set to 0.5 in.

The discrete nature of the design variables (isolator locations) calls for the use of specialized optimization techniques. An approach based on a genetic algorithm will be examined in this study and compared to other techniques. It will be shown that the use of classical continuous optimization techniques followed by rounding of the solution is not appropriate.

4. Modeling

4.1 Suspended Rigid Body Modeling

The lowest “flexible” modes in the system occur at frequencies of about 150 to 200 Hz. They correspond to resonances of tuned vibration absorbers which are embedded in the optical table. At frequencies well below these flexible modes (say from 0 to 100Hz), the system can be approximated as a set of 2 rigid bodies (seismic base and optical table) connected by 3-dimensional, linear springs with viscous damping (airbags

and spring isolators). Each rigid body is given 6 degrees of freedom and defined by its mass and inertia tensor. The springs are defined by stiffnesses (k_x, k_y, k_z) and damping coefficients (c_x, c_y, c_z) in 3 mutually orthogonal directions, parallel to the global reference directions X, Y, and Z of Fig. 5. For the system at hand, we have $k_x=k_y=k_{shear}$, $k_z=k_{axial}$, $c_x=c_y=c_{shear}$, and $c_z=c_{axial}$ because of axisymmetry of both the airbags and the steel springs. Bending and torsional stiffnesses of the springs were neglected.

4.2 Mass Properties

Mass properties for most components (seismic base, adapter plates, airbag adapter blocks, etc.) were obtained from Pro-Engineer[12] models. The lifters were weighed and their inertia tensors were approximated based on uniform density assumptions in a Pro-Engineer™ geometric model. The optical table posed special problems: because of the presence of embedded tuned vibration absorbers of unknown properties (accurate data could not be obtained from the manufacturer), the mass properties could not be determined analytically and were instead measured by the Mass Property Laboratory at Sandia National Laboratories. The center of mass of the table was found to be offset by 1.72 in. in the negative *Y* direction from its geometric center (due to uneven distribution of tuned dampers).

4.3 Stiffness and Damping

4.3.1 Airbags

The stiffness of an airbag is almost exactly proportional to the static axial load (the natural frequency of a 1 d.o.f. system made of an airbag and a mass is approximately independent of the magnitude of the mass). Initial data for axial stiffness versus axial load was obtained from the manufacturer's catalog[7] and can be approximated as

$$k_{axial} = 66.51 + 0.4047\ P,$$

where P is the axial static load in lb and k_{axial} is in lb/in. Shear stiffnesses were not available. Modal tests were then performed on the seismic mass alone, supported by the airbags (the optical table and springs were removed for this test). Natural frequencies and modal damping ratios were extracted form those measurements and are listed in Table 1.

Mode #	Frequency [Hz]	Damping [%]	
1	0.961	2.95	shear along X (+ rocking around Y)
2	1.145	3.23	shear along Y (+ rocking around X)
3	1.519	3.36	twist around Z
4	2.161	0.82	symmetric up/down along Z
5	2.464	1.61	rocking around Y
6	2.467	1.28	rocking around X

Table 1 Experimental modes of seismic base on airbags

Mode 4 (pure up and down motion) was used to adjust the axial stiffness and damping (catalog values for stiffness was scaled by 1.0154) and mode 3 (pure twisting motion around Z, straining the airbags in pure shear) provided the ratio of shear to axial stiffness (k_{shear}/k_{axial} = 0.4611) and the shear damping. With those values, the model predictions compare reasonably well with the experiment (Table 2).

	Frequency [Hz]		Damping ratio [%]	
	exp.	anal.	exp.	anal.
shear along X (+ rocking around Y)	0.961	1.192	2.95	1.70
shear along Y (+ rocking around X)	1.145	1.333	3.23	2.39
twist around Z	1.519	1.519	3.36	3.36
symmetric up/down along Z	2.161	2.161	0.82	0.82
rocking around Y	2.464	2.489	1.61	2.90
rocking around X	2.467	2.479	1.28	1.97

Table 2 Seismic base on air springs; analytical and experimental modes.

4.3.2 Steel Springs

With the assumption of linearity, the dynamic axial and shear stiffnesses of a coil spring are independent of static loads. An initial value k_{axial} = 3816 lb/in was obtained from the manufacturer[6]. The shear stiffness was not available. Modal tests were performed on the optical table resting on 4 spring isolators, symmetrically arranged around its geometric center. The airbags supporting the seismic base were replaced with stiff support blocks for those tests. Measured natural frequencies and modal damping ratios are listed in Table 3.

Mode #	Frequency [Hz]	Damping [%]	
1	7.199	0.16	shear along X left side
2	8.007	0.41	shear along X right side
3	8.456	0.22	shear along Y (+ rocking X)
4	12.121	0.15	rocking around Y
5	12.885	0.16	up/down left side
6	14.551	0.21	up/down right side

Table 3 Experimental modes of optical table on 4 spring isolators.

	Frequency [Hz]		Damping ratio [%]	
	exp.	anal.	exp.	anal.
shear along X left side	7.199	7.375	0.16	0.22
shear along X right side	8.007	8.020	0.41	0.26
shear along Y (+ rocking X)	8.456	8.279	0.22	0.27
rocking around Y	12.121	12.203	0.15	0.25
up/down left side	12.885	12.897	0.16	0.17
up/down right side	14.551	14.370	0.21	0.16

Table 4 Optical table on 4 spring isolators; analytical and experimental modes.

The simple analytical technique used to identify airbag stiffnesses and dampings cannot be used here because of the absence of pure up/down or shear modes. Instead, a parameter identification optimization problem was formulated that minimizes a weighted sum of squares of errors on natural frequencies and damping ratios. Four parameters were adjusted to minimize this error: axial stiffness k_{axial} and damping c_{axial}, and shear stiffness k_{shear} and damping c_{shear}. The minimization was performed using OPT++[13] conjugate

gradient optimizer in DAKOTA. The results lead to a correction factor of 1.04 on the catalog value for k_{axial}, a ratio $k_{shear}/ k_{axial} = 0.41$, and damping coefficients $c_{axial} = 0.13$ and $c_{shear} = 0.18$. These values achieve good analytical-experimental match (Table 4).

4.4 Analysis Code

The rigid body equations of motion were coded into a set of M-files in MATLAB[14]. For the optimization, interfacing with DAKOTA† was done directly (without the use of input and output filters[4]) since the MATLAB code could be designed to exchange information (design variables and objective function) in the DAKOTA compatible format.

4.4.1 Objective Function Calculation

The transmissibility T at a single frequency of 50 Hz (see Section 2) is calculated by solving the (12x12) set of linear dynamic equations for that frequency.

4.4.2 Constraint Evaluation

- corner constraints: simple checks are performed for each isolator location. Three Boolean constraints g_1, g_2, g_3 are defined as

$$g_i = < \text{is isolator \# } i \text{ at corner? } >, \quad i=1,\ldots,3, \quad \text{boolean constraints.}$$

- alignment constraint: since natural frequencies are needed for the 4Hz limit in the stability constraint, alignment was checked by monitoring the value of the first natural frequency f_1 (out of 12) of the system. A value of zero indicates alignment (or more than 1 spring at any location). To account for numerical roundoffs, a threshold value of 0.01 Hz was used. This defines the 4th Boolean constraint:

$$g_4 = < f_1 < 0.01 \text{ Hz ? } >, \quad \text{boolean constraint.}$$

Note that if this constraint is violated, the system's stiffness matrix is nearly singular and static equilibrium cannot be computed. Also, there is a switch in the order of the natural frequencies because the first natural frequency is now associated with the upper system instead of being a seismic base suspension mode (which is also why the alignment constraint is treated separately from the stability constraint). For these reasons, all following constraints $g_5 \ldots g_{11}$ are evaluated only if g_4 is *false*.

- stability constraint: As mentioned earlier, the upper system natural frequencies are required to be above 4 Hz, i.e.

$$g_5 = 1 - f_7/4.0 \text{ Hz} < 0, \quad \text{real constraint,}$$

where f_7 is the first natural frequency of the upper half of the system (because the first 6 frequencies correspond to the lower half, i.e. the seismic mass on its airbags).

- compression constraint: with 3 isolators, the upper table is statically determinate so that reaction loads can be readily computed by solving 3 equilibrium equations. Three constraints are formulated to guarantee that the loads are compressive:

$$g_{5+i} = P_i < 0, \quad i=1,\ldots,3, \quad \text{real constraints,}$$

where P_i is the static load on spring #i (positive in traction).

† DAKOTA is an object-oriented C++ toolkit for interfacing broad libraries of optimization methods (e.g. NLP, GA's, coordinate pattern search) with engineering applications in a variety of disciplines.

- static deflection constraint: three constraints enforce that the static deflections δ_i of the springs remain below a limit δ_{max} = 0.5 in.:

$$g_{8+i} = \delta_i / \delta_{max} - 1 < 0, \quad i=1,\ldots,3 , \qquad \textit{real constraints.}$$

5. Baseline Design

Before optimizing the isolation system, we first define a generic, baseline configuration. It will be used as a point of comparison to evaluate improvements achieved by optimization. This baseline configuration is generated following the generic design approach described in Section 1: the three isolators are placed symmetrically around the center of mass of the isolated body (the optical table). Because of the coarse discrete grid of mounting holes for the isolators, perfect symmetry cannot be achieved. The selected locations are shown in Fig. 8.

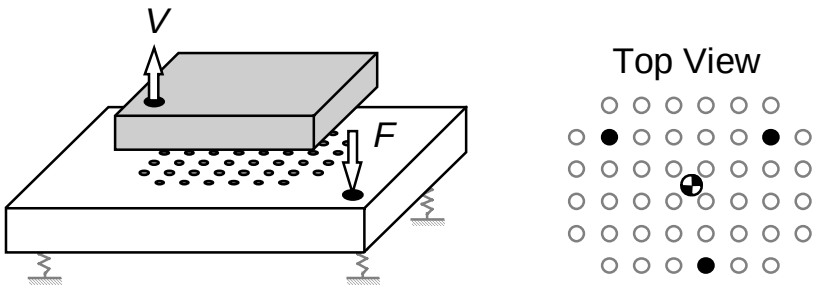

Figure 8 Locations of isolators in baseline design.

Note that, without prior analysis, a configuration similar to that of Fig. 8 would probably be used in practice. This study will show that other configurations can be found that lead to much superior performance.

The transmissibility predicted by the MATLAB model for this design is plotted in Figure 9. A number of resonant peaks corresponding to suspension modes can be seen at frequencies up to about 15 Hz. At higher frequencies, the transmissibility decays rapidly and reaches T = 21.22 μin/sec/lb at the 50 Hz target frequency.

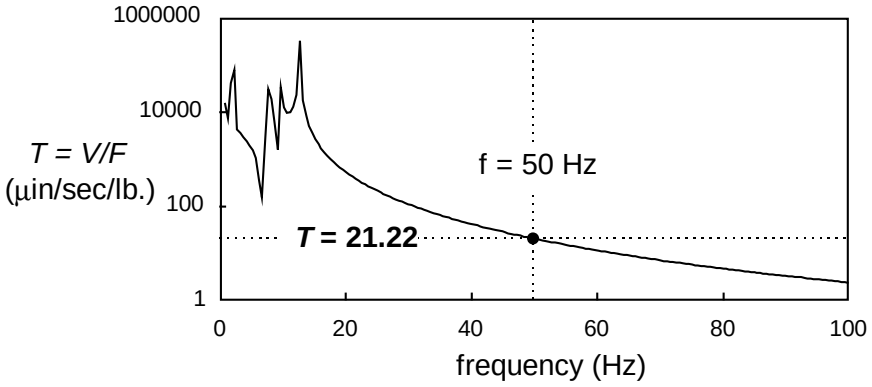

Figure 9 Transmissibility of baseline design.

Flexible modes are absent from the figure because first, they are not represented in the rigid body/spring model and second, they start at frequencies of about 150 Hz.

It should be noted that the performance of this design is very representative of this particular isolation system: Monte Carlo simulations with 1000 random configurations show that the average transmissibility of *feasible* designs is 21.1 μin/sec/lb, almost exactly equal to that of the baseline design.

6. Optimization Techniques

Three optimization techniques are applied to this discrete problem: random search, genetic algorithm (GA), and continuous optimization followed by rounding of the solution. These techniques and their implementation for solving the problem at hand are presented in the following sections. Particular attention is given to the GA solution.

6.1 Random Search

The random search technique (Figure 10) is used as a point of comparison to evaluate the efficiency of the GA search. It consists of generating a given number (*n*) of random configurations of 3 isolators, eliminating those that violate one or more constraint(s) and selecting the best remaining design. This process is extremely simple and general but obviously inefficient.

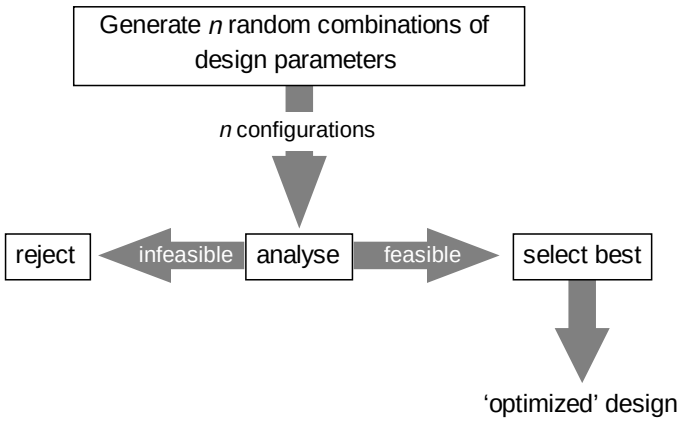

Figure 10 Random search technique.

6.2 Genetic Algorithm

6.2.1 Description

A genetic algorithm (GA) is a random search technique that mimics some mechanisms of natural evolution. The algorithm works on a *population* of designs (*individuals*) which is the counterpart of a population of biological creatures. Following principles of the Darwinian theory, the population e*volves* from *generation* to generation, gradually improving its adaptation to the environment: through natural *selection*, fitter individuals have better chances of transmitting their characteristics to later generations (survival of the fittest).

In the algorithm (Figure 11), the selection of the natural environment is replaced by artificial selection based on a computed *fitness* for each design. This fitness is essentially the objective function of the optimization problem (possibly augmented with constraint penalties). The *chromosomes* that define characteristics of biological beings are replaced by strings of numerical values representing the design variables. When couples of selected *individuals* (designs) reproduce, they combine portions of their genetic material to create an *offspring* that shares traits from each *parent*. In the GA, this recombination of the parents' chromosomes is performed by two *genetic operators* which are the simplified versions of their natural counterparts: *crossover* and *mutation* (several other operators have been introduced but crossover and mutation are almost always present). The crossover combines existing features of both parents to *exploit* the genetic heritage of the population while the mutation introduces new features to *explore* new areas of the design space. In tuning the algorithm, a delicate balance (which unfortunately is problem-dependent) must be achieved between *exploitation* and *exploration*: too little mutation and the GA will "converge" prematurely, possibly to a local optimum, destroying the global character of the search; too much and the search will be exceedingly disrupted, preventing efficient exploitation of existing design features.

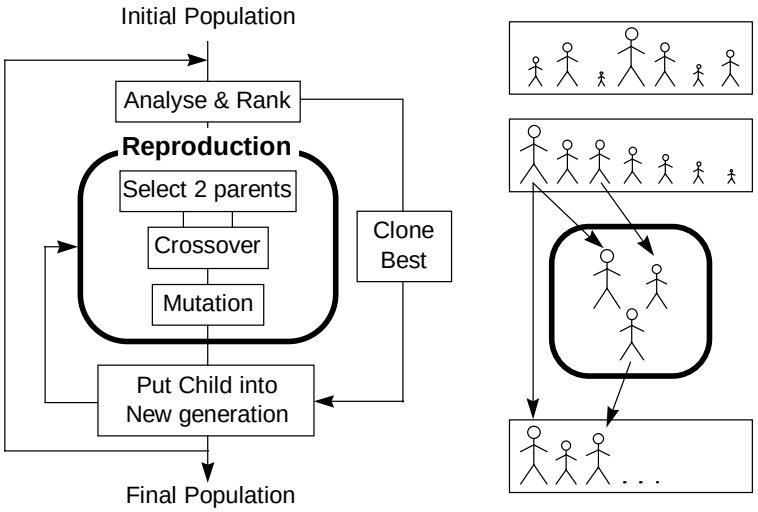

Figure 11 Classical genetic algorithm.

While innumerable variations of genetic algorithms are possible, the following subsections describe the coding, operators, and constraint enforcement strategies specific to SGOPT[3] in DAKOTA[4] and this application.

6.2.2 Defining the Chromosome (Coding)

The first and most important step in preparing an optimization problem for a GA solution is that of defining a particular coding of the design variables and their arrangement into a string of numerical values to be used as the chromosome by the GA.

The choice of a particular coding has large ramifications on the efficiency of the search. Although historically GA's were developed to operate on strings of binary numbers (design variables would be converted to their binary representations and concatenated into a long binary chromosome), studies[15,16] have shown that other

representations (using trinary, decimal, or real alphabets) can be used with similar or better performance.

For the problem at hand, a few coding options are illustrated in Fig. 12. The first three options (a.1 to a.3) are based on a continuous numbering of all available locations from 1 to 44. The last two (b.1 and b.2) use array indices 1 to 6 and 1 to 8 in the *X* and *Y* directions. The first numbering scheme (a.) presents the disadvantage of not embodying the physics of the problem: choosing the location of isolators in a plane is a 2-dimensional problem, calling for a 2-dimensional coding. This numbering also creates artificial discontinuities in that a small change in code value does not always lead to a small change in isolator location. For example, changing an isolator location from #6 to #7 (Fig. 12a) moves it all the way across the adapter plate.

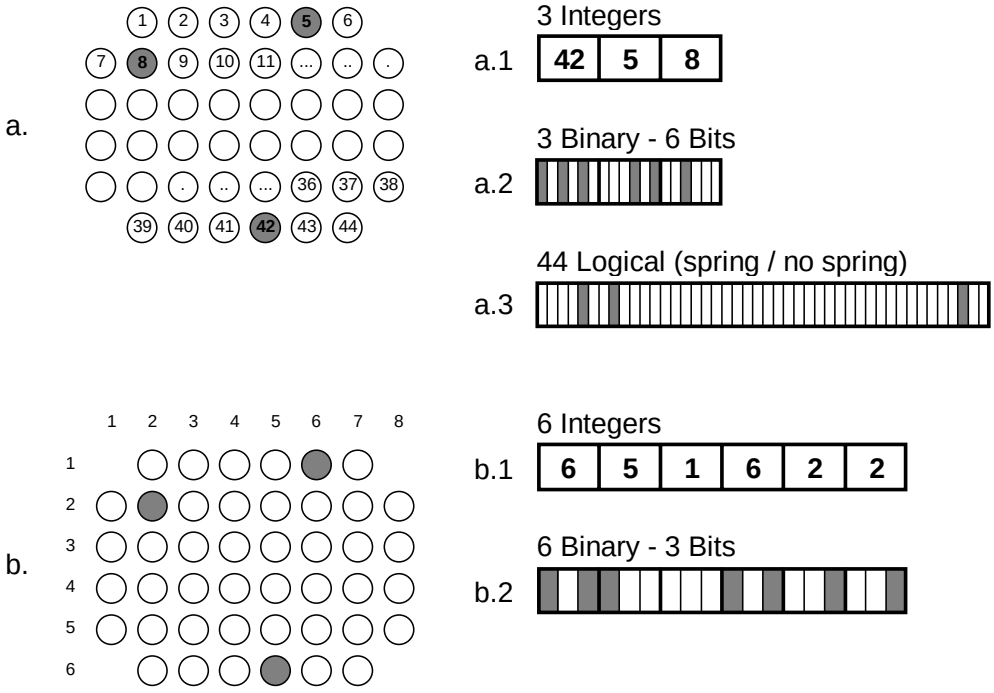

Figure 12 Various ways of coding isolator locations into a "chromosome" for a genetic algorithm.

Coding a.3 is particularly inefficient: crossover operations between two such chromosomes have very little chance of generating an offspring with exactly 3 isolators. The algorithm would then spend most of its time generating and analyzing designs that do not have the required number of isolators. This is a manifestation of a more general idea: a coding should include as many constraints as possible to reduce the probability of generating infeasible designs. Another example of this is the corner constraint: the first numbering scheme (a.) implicitly guarantees satisfaction of that constraint.

Despite this last observation, the designs were coded using a string of 6 integer genes for the X and Y grid indices of the 3 isolator locations, i.e. [x_1, y_1, x_2, y_2, x_3, y_3] (coding b.1 in Fig. 12). This coding closely represents the physics of the problem and does

away with the need for back and forth conversions between integer and binary representations of the design variables. The alphabet (range) is 1..6 for *x* genes and 1..8 for *y* genes.

6.2.3 Selection Operator

The selection operator is in charge of picking individuals for reproduction. It uses a biased roulette wheel where fitter individuals get larger portions of the wheel and have therefore better chances of reproducing and transmitting their characteristics.

A *ranking* technique is used to assign portions on the roulette wheel: the probability of selection of an individual is a function of its *rank* in the population - not its fitness. This avoids the classical problem of the *super-individual*: if, early in the search, a single design is - by chance - vastly superior to all others, a fitness-based selection would almost always pick that super-individual and create a population of clones, leading to complete loss of genetic diversity. This does not happen with a rank-based selection rule.

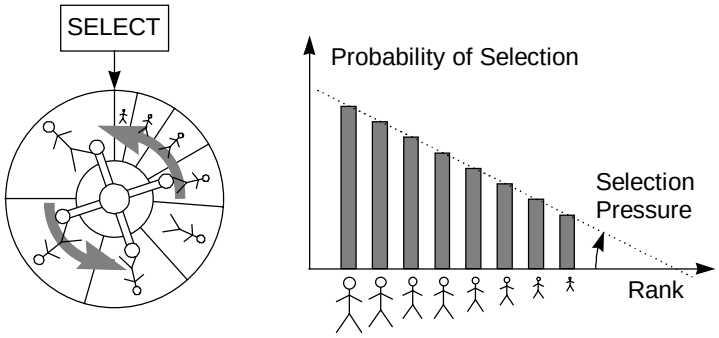

Figure 13 Ranking selection operator.

Figure 13 illustrates the particular rule used in this application. The *selection pressure* is defined as the ratio of the probability of selection of the best individual in the population to that of the worst. High selection pressures push the search to faster improvement but also gives it less time to explore the design space. Again, a compromise must be found. The selection pressure was set to 2 for this application.

6.2.4 Crossover Operator

Once two parents have been selected, their chromosomes undergo a crossover operation that generates an offspring chromosome. This GA uses a 2-point, non-averaging crossover illustrated in Fig. 14.

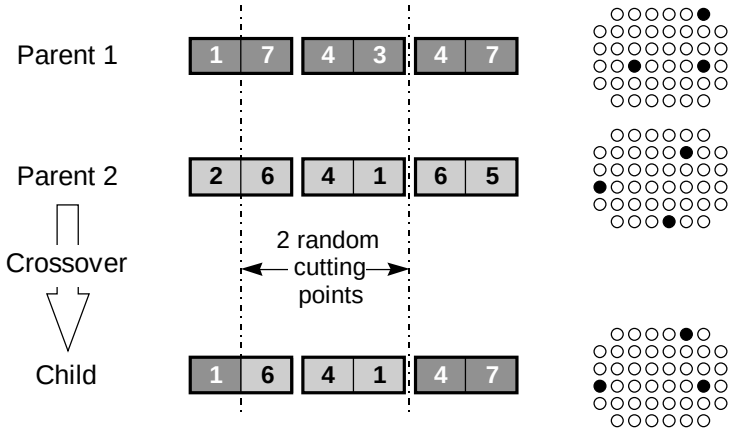

Figure 14 Non-averaging 2-point crossover operator.

The operator selects two random cutting points and creates a child chromosome by assembling the inner and outer substrings from either parent. This operation is applied with a given probability P_c (typically large, 80 to 100%). When crossover is not applied, one of the 2 parent chromosomes (chosen at random) is simply cloned.

Notice that the child typically has some features from each parent but also some new characteristics: the (1,6) location in the child's chromosome in Fig. 14 for example results from the combination of the *x* index from one parent and the *y* index from the other.

This crossover is called *non-averaging* because the cutting points are only allowed to fall *between* genes. In contrast, an averaging crossover (see for example [16]) generates cutting points that can fall anywhere *within* a gene. When this happens, the child gene is computed as a weighted average of the parent genes. The averaging crossover was introduced to improve the GA's behavior for problems with large alphabets: when a gene can take a large number of values, the initial population is unlikely to contain every possible value for that gene. Without averaging crossover, the generation of new values is left exclusively to the mutation. Artificially large and disrupting mutation rates are then needed to maintain diversity. The averaging crossover is also naturally indicated for problems with real-valued genes (which have an infinite alphabet).

For the current application, the alphabet is limited enough (1 to 6, or 1 to 8) that a non-averaging crossover may be sufficient, although further testing would clarify this point.

6.2.5 Mutation Operator

The mutation introduces random changes in the offspring chromosome resulting from the crossover operation. The role of the mutation is to prevent loss of genetic diversity by introducing design features that may have never been present in the population or may have been lost over time.

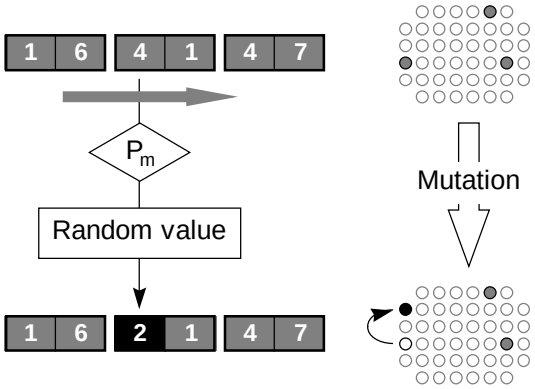

Figure 15 Mutation operator.

Mutation is applied with a small probability P_m to each gene in the chromosome. Figure 15 illustrates this process. When mutation occurs, the current value of the gene is replaced by a random value, uniformly distributed in the range of that particular design variable.

6.2.6 Constraint Enforcement

It is interesting to note that although most realistic engineering design problems involve numerous constraints, little work has been done to investigate constraint enforcement strategies for genetic algorithms. One of four approaches is typically used: data structuring, direct elimination, repair operators, or penalty functions.

Data structuring consists of designing the coding in such a way that constraints are automatically satisfied because infeasible designs can simply not be represented. In our problem for example, the requirement to have exactly 3 isolators is automatically satisfied when using any coding scheme in Fig. 12, except a.3. Satisfaction of the corner constraint on the other hand is implicit with codings a.1 to a.3 but not with b.1 or b.2. Although data structuring is always the most efficient technique, it is only applicable for particular types of constraints and is very problem-specific.

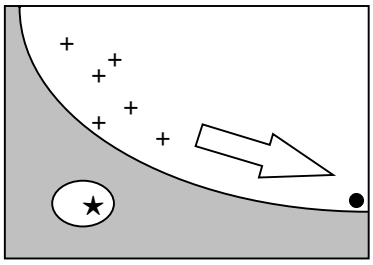

Figure 16 Design problem with disjoint feasible space.

In the *direct elimination* technique, each design resulting from selection, crossover, and mutation is examined for constraint satisfaction before it is included in the new generation. If any constraint is violated, the offspring is eliminated and a replacement is created through new selection, crossover and mutation operations. This process is repeated until a feasible design is found. This creates entirely feasible populations at every generation, constraining the search exclusively within the feasible regions of the design space. In cases where the design space contains non-convex and/or disjoint feasible regions, this makes the search inefficient and unreliable. The reason is best

explained on a simple example. Figure 16 represents the design space of a 2-dimensional design problem. The shaded areas represent infeasible regions and the arrow gives the general direction of improvement of the objective function within the large feasible area at the upper right of the figure. An initial feasible population for this problem is likely to reside entirely in the large feasible region of the design space. If direct elimination is used, the GA will then converge to the local minimum within that area (● in Fig. 16), missing the global optimum (★). It is clear that convergence to the global optimum is likely only if the GA population is allowed to migrate through the infeasible "barrier" to reach the small feasible inclusion. Note that the feasible space does not have to be disjoint for this problem to occur: if the feasible space is non-convex, the GA may have to migrate around a constraint barrier instead of short-cutting through it.

Another approach uses *repair operators* to "fix" infeasible designs before incorporating them in the new generation. Repair operators use some knowledge about the problem to try and eliminate constraint violations through "small" modifications of the design. This approach is inherently problem-specific and is used mostly in research algorithms or GAs designed specifically for a particular class of problems. A simple example is shown in Fig. 17 for the corner constraint: an isolator located at a corner could be moved to one of the neighboring locations (one of three at random for example).

In less trivial cases however, it may be difficult to define design changes that eliminate given constraint violations. Also, in highly constrained problems, one cannot guarantee that 'fixing' a design for one constraint will not cause violation of another. Finally and most importantly, repair operators are problem-specific so they cannot be used in a general purpose optimization package like DAKOTA.

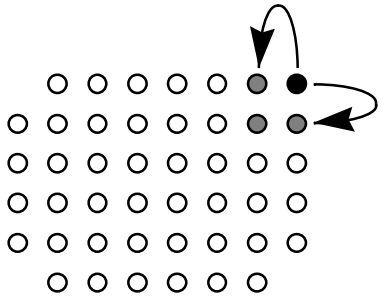

Figure 17 A simple repair operator for the corner constraint.

The fourth constraint enforcement technique uses *penalty functions*: in a minimization problem, each constraint violation produces an increase in the objective function. Because penalty functions were originally introduced to enforce constraints in the context of gradient-based optimization, classical definitions produce a smooth, differentiable transition from feasible to infeasible regions. This 'blurs' the boundaries of the feasible design space. A continuous optimization then converges to either slightly infeasible designs (when using exterior penalty functions[17]) or slightly conservative designs (with interior penalty functions). To avoid this and since there is no need to achieve continuity in the penalty functions with a zero order method like a GA, a combination of step and gradual penalties[18] (as shown in Fig. 18) will be used.

The steps prevent convergence to slightly infeasible designs while gradual penalties maintain a logical hierarchy between designs with more or less severe constraint violations. Gradual penalties are of course applied only to the quantifiable constraints (stability, compressive spring loads, and static spring deflections); Boolean constraints (corner constraints and isolator alignment) receive only a step.

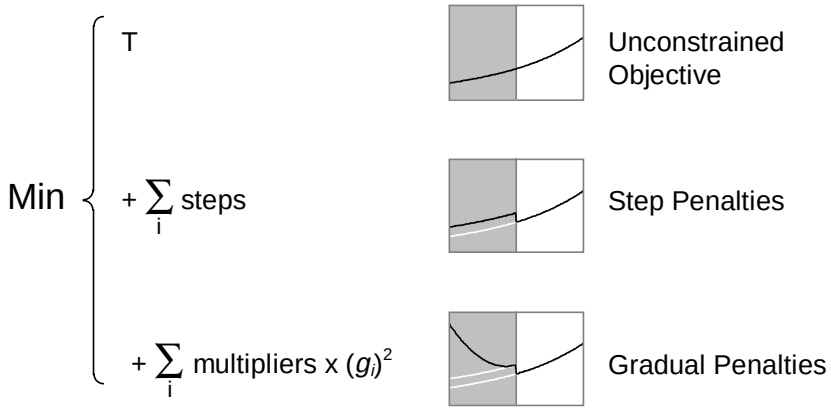

Figure 18 Penalty function approach used in this application; the shaded area is infeasible.

Just like in gradient-based optimization, adjusting penalty multipliers (and steps) is tricky. Too little penalty leads to infeasible designs, while too much makes the search inefficient by restricting it to feasible regions. Unfortunately, because GAs are random searches and necessitate large numbers of function evaluations, trial and error adjustments are impractical. A study by Richardson *et al.*[19] indicates that penalties should be as small as possible, but large enough to prevent frequent convergence to infeasible solutions and that using harsh penalties leads to poor convergence and/or premature convergence to a super-individual. However, that study uses proportional selection (probability of selection based on objective function value); its conclusions do not hold when using a ranking selection rule. In fact, it appears from *limited* experimentation with this problem that the reliability of the search is best with harsh penalties associated with a weak selection pressure (the selection pressure was set to 2, see Section 6.2.3). Note again that because of the random character of GA's and the interaction between multiple parameters (probabilities of mutation and crossover, population size and number of generations per search, penalty multipliers and steps, etc.), statistically significant conclusions can be reached only through extensive (and expensive) experimentation.

6.3 Rounding of Continuous Optimum

In this approach (Fig. 19), all design variables are temporarily viewed as continuous and classical gradient-based techniques are used to solve the optimization problem. The resulting optimal design is infeasible and its parameters must be "rounded" to discrete values.

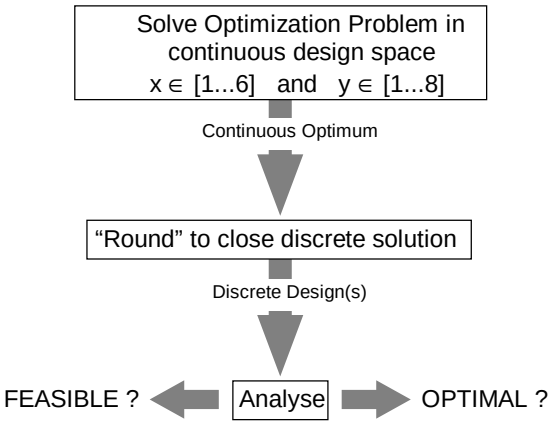

Figure 19 Using continuous approximation followed by rounding to nearby discrete solution.

The appeal of this technique is the much smaller number of function evaluations typically needed to achieve convergence with a gradient based technique than with a random search method. However, the rounding operation can make the design infeasible or suboptimal. Also, many different designs can be defined by rounding up or down the various design variables. To increase the chances of optimality, all these "direct neighbors" must be considered and analyzed. When the number of discrete design variables is large, this may require a substantial number of additional function evaluations (2^n if n is the number of design variables) so that the computational advantage may be lost. These points are further discussed in Section 7.2.

7. Optimization Results

7.1 Genetic Algorithm and Random Search

The GA in DAKOTA provides a number of options and adjustments. The following choices were made for this application:

- population of 10 individuals (designs). The initial population is random. With 10 individuals, the probability of representation of any gene value (1 to 6, or 1 to 8) at any gene location is close to 1, which should ensure good performance with a non-averaging crossover
- probability of crossover: 0.80
- probability of mutation: 0.10. This gives a 60 to 80% probability for any offspring to be affected by mutation.
- elitist strategy always clones the best individual of the current generation into the next generation. This guarantees that the best found design is never lost in future generations.
- number of generations : 15 (experiments with smaller numbers of generations did not achieve sufficient reliability)
- moderate selection pressure: the probability of selection of the best individual is twice that of the worst (i.e. selection pressure=2).

Penalty multipliers and steps were adjusted somewhat arbitrarily. *Limited* trial and error experimentation was performed and lead to the following values:

- corner constraints g_1, g_2, g_3: step = 5.0.
- alignment constraint g_4: step = 20.
- stability constraint g_5: step = 2, multiplier = 20.
- compression constraints g_6, g_7, g_8: step = 5, multiplier = 0.05.
- static deflection constraints g_9, g_{10}, g_{11}: step = 5, multiplier = 200.

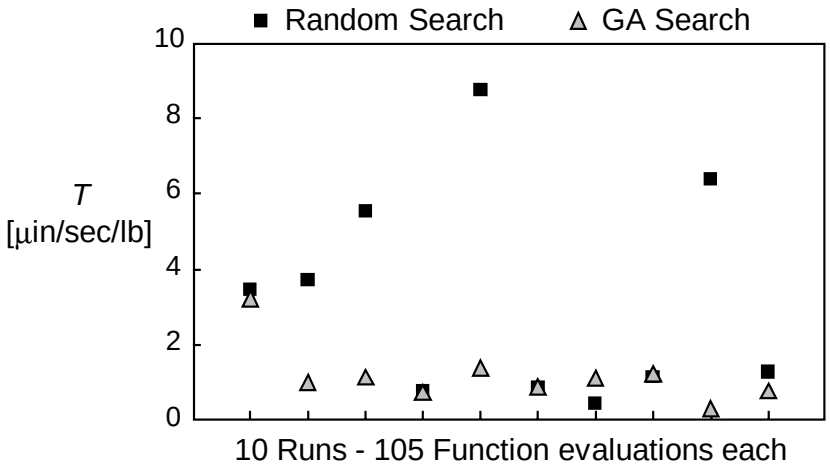

Figure 20 Comparison of genetic algorithm and random search results using the same number of function evaluations (105). Designs shown are the best found for each run.

The design space for this problem is relatively small: 6×8 locations for each of 3 isolators gives $48^3 \cong 110{,}000$. With 15 generations of 10 individuals, each search evaluates up to 150 designs, or 0.14% of the design space. Note that because this GA keeps track of previously analyzed designs, the actual number of function evaluations averages around 105 per search (0.1%). To get an idea of the reliability of the search, a series of 10 runs were performed and the results are compared to a series of 10 random searches. The number of designs in the random search is set to 105 so the computational expense is the same as in the GA. Typical results are shown in Fig. 20. The figure shows only the best design found in each run of the GA or the random search. The random searches generate some good designs and many mediocre ones. In contrast, all 10 designs obtained from the GA represent significant improvements from the baseline case. However, the GA occasionally "converges" to a relatively poor design (T=3.23 in Fig. 20). This implies that more reliable results can be obtained by running a small number of short searches: if the probability that the best found design is "poor" is 0.1 for a one run, then it is only 0.01 for the best of 2 runs, 0.001 for the best of 3, etc. Because GA's are most efficient in the initial phases of the search and further "convergence" is usually slow, this approach is often preferable to running a single longer search[18].

It is interesting to note that the classical argument that a GA provides a choice between several good designs in the final population does not hold in this application. Instead, final populations typically contain only one or two feasible design(s). In fact, all generations in the search are composed mostly of infeasible designs. This shows that the

search is taking place primarily in infeasible regions of the design space. Under these conditions, it is particularly crucial to allow the search to migrate through infeasible regions. This may also explain why a weak selection pressure provides better results with this problem.

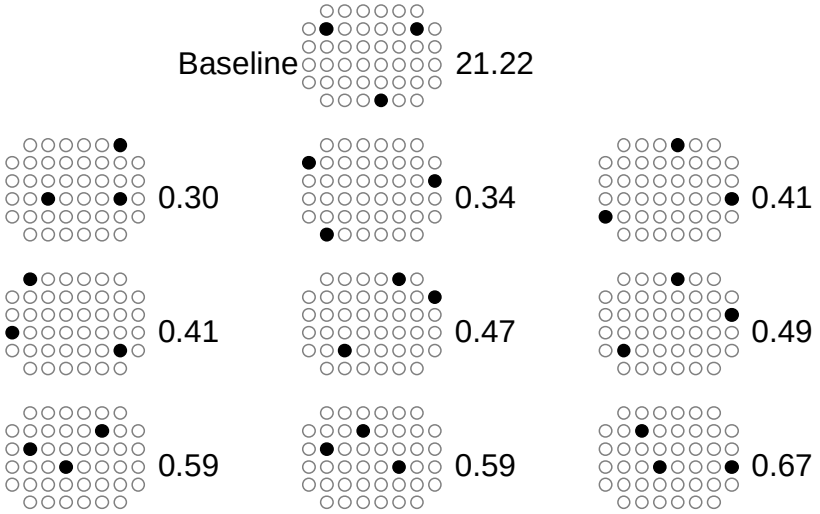

Figure 21 GA optimized designs and their transmissibilities at 50 Hz (in μin/sec/lb), compared to baseline configuration.

Several sets of 10 runs each were performed in the course of this study. Nine of the best designs obtained form those runs are shown in Fig. 21 and compared to the baseline design. The transmissibilities at 50 Hz are also listed in the figure. Clearly, all 9 optimized designs represent very significant improvements from the baseline case: transmissibilities are reduced by factors 32 to 70 compared to baseline.

Note also that the 9 optimized designs do not have any apparent similarities although they provide very similar performances. This indicates the existence of multiple local optima for this problem.

Figure 22 shows frequency response functions (FRF's) for all designs of Fig. 21. They show that the GA is seeking out an anti-resonance condition in the vicinity of 50 Hz. The fact that the anti-resonances "miss" the 50 Hz target is due to the discrete isolator locations. In fact, the continuous optimum of the next section places the antiresonance almost exactly at 50Hz, achieving a transmissibility of 0.2 μin/sec/lb. Another important observation is that there is significant broad-band improvement in the transmissibilities of the optimized designs compared to the baseline design. That is, the improvement is not confined only to the 50 Hz target frequency. This is an especially important observation since it indicates that performance is not seriously degraded for off-nominal excitation inputs and more sophisticated objective function formulations minimizing broad band transmissibility are probably unnecessary.

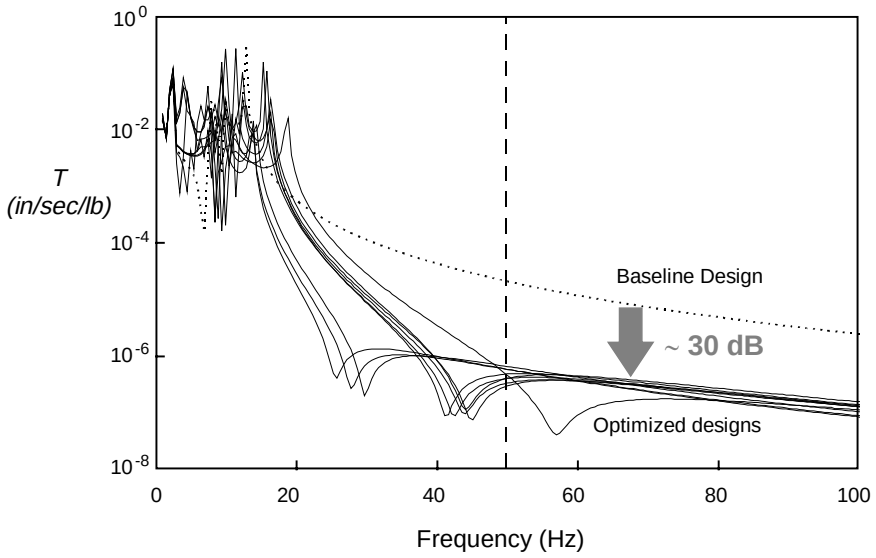

Figure 22 Transmissibility curves of optimized designs compared to baseline configuration (model predictions).

7.2 Rounding of Continuous Optimum

DOT's[20] modified method of feasible directions (accessible through DAKOTA) was used to solve the constrained, non-linear continuous problem. A continuous solution (★ in Fig. 23) was found at (2.22, 1.56, 2.49, 4.94, 4.29, 3.79) with a transmissibility T=0.20 μin/sec/lb at 50 Hz. Rounding to the closest discrete solution (● in Fig. 23) leads to (2,2,2,5,4,4), which is infeasible (violates the 4Hz stability limit). If we consider all immediate neighbors of the continuous solution (all combinations of ● and ● in Fig. 23), we find that out of the 64 designs, only 12 are feasible and the best of these (2,1,3,5,5,3) gives T=3.67 μin/sec-lb. This transmissibility is 22 times higher than that of the best GA solution (T=0.30) and only 6 times better than the baseline configuration (T=21.22).

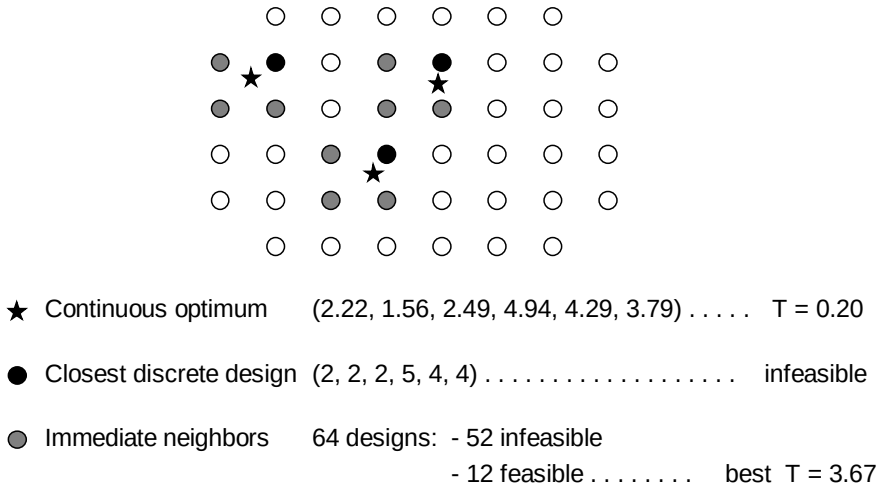

Figure 23 Results from continuous optimization and rounding.

Continuous approximation leads - for this problem - to many infeasible designs and a few sub-optimal solutions. Although this particular application is particularly difficult for continuous approximation (because of the low density of feasible designs and the very coarse discrete grid), two general observations can be made:

- continuous solutions for constrained problems tend to make one or more constraint(s) active. Rounding those solutions is likely to produce violations of the active constraint(s).
- optimal regions in the continuous problem may not contain any discrete solution. The discrete optimum may then be very different from the continuous one. This is particularly true when the discrete grid is coarse compared to the shortest wavelengths in the objective function response surface.

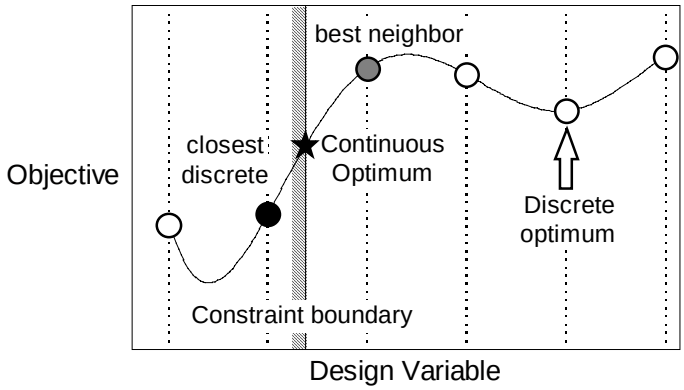

Figure 24 Rounding a continuous optimum to neighboring discrete solutions.

Figure 24 illustrates these points for a one-dimensional constrained problem. The curve represents the variation of the objective function in a continuous design space. Dotted vertical lines show the discrete grid of the actual problem and the solid line gives the constraint boundary. If appropriate starting points are used for the continuous optimization, the global continuous optimum will be found at ★ (the global optimum in the continuous sense). This design makes the constraint active. The closest discrete solution (●) violates the constraint while the other direct neighbor solution is far from optimal. The discrete optimum (shown by the arrow in the figure) does not "resemble" its continuous counterpart.

7.3 A Look at the Design Space

To understand the behavior of the optimizers, it is interesting to take a look at the topography of the design space. In particular, the relative sizes of feasible and infeasible regions and the rates of variation of the objective function are critical factors that influence the efficiency of optimization algorithms.

To answer the first question, a Monte Carlo simulation was performed in which 1000 random combinations of isolator locations were selected at random and their objective function and constraints were calculated. The results show that only 12.7% of the designs are feasible, indicating a heavily constrained problem. The mean objective function value for the feasible designs is equal to T=21.1 μin/sec/lb. As mentioned

earlier, this value is almost exactly equal to the transmissibility of the baseline configuration which can therefore be considered representative (in other words, the baseline design is not exceptionally poor). Also, the distribution of isolator locations among the feasible designs does not depart significantly from uniform (except for the 4 corners which are never used). This indicates that "good" designs do not tend to use particular locations on the grid. It is only the combination of 3 locations that determines a design's feasibility.

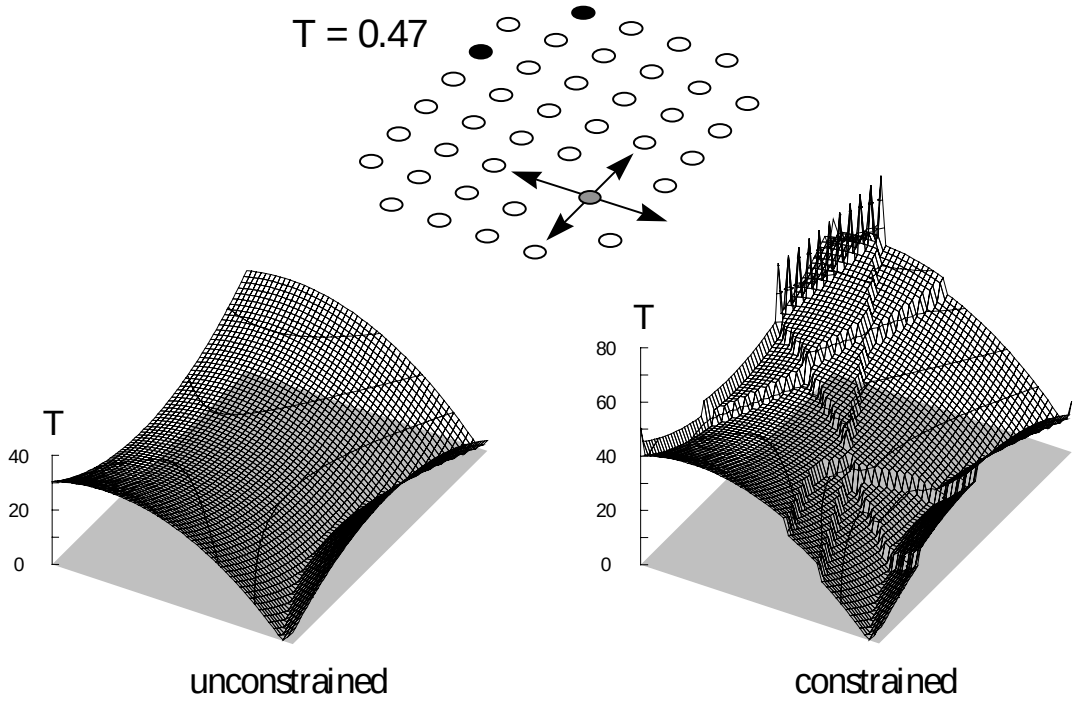

Figure 25 Two-dimensional cut through the design space obtained by moving one isolator in both directions; for clarity, the constrained objective plot only shows step penalties.

The 6-dimensional design space of this problem cannot be visualized easily. However, 2-dimensional cuts can be obtained by fixing the locations of 2 of the isolators and moving the third one in the *X* and *Y* directions. The GA design (1,6,2,8,5,3) with *T*=0.47 from Fig. 21 is used as a starting point. The locations of the two isolators near the top right corner (1,6) and (2,8) are fixed while the third one is moved across the plane, ignoring the grid. This generates the 2-dimensional cut (1,6,2,8,*x*,*y* with $x,y \in \Re$) in the design space. Figure 25 shows both unconstrained and constrained (steps penalties only) response surfaces. Note that —in the continuous sense— there is not a unique optimum but rather a infinity of locations (the dotted curve in Fig. 26) for the third isolator that achieve a small transmissibility (0.2) by designing an anti-resonance at the exact location of the pickup point.

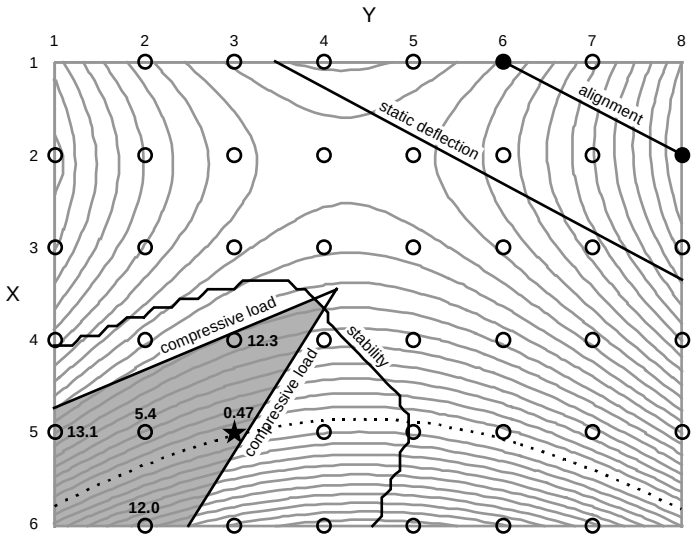

Figure 26 Design space cross section showing unconstrained objective, discrete grid, contour lines, constraint boundaries, feasible domain (shaded), and feasible discrete designs with their transmissibilities; the ★ shows the discrete optimum.

The same cut is represented in Fig. 26: here, contour lines of the unconstrained objective function are plotted together with constraint boundaries. Discrete isolator locations are shown with small circles. Note the small size of the feasible domain (shaded). Only 5 discrete designs are contained in that area; their transmissibilities *T* at 50 Hz are listed in the figure.

Note also that the location (5,3) found by the GA for the 3rd isolator is the discrete optimum (★) for the given locations of the other 2 isolators. It was found that this is the case of almost all locations used in the designs of Fig. 21, which indicates that they correspond to various local optima and confirms the strong multi-modality of this problem.

The small size of the feasible regions (12.7% of design space) and the coarse discrete grid create small feasible "pockets", each containing few discrete solutions. This explains the difficulties encountered in the GA searches. The search has to take place almost entirely in the infeasible design space because the number of designs in each "pocket" is too small to allow efficient exploitation by the GA. This also leads to mostly infeasible populations so that each run provides only one or two feasible design(s).

Also, there is very strong coupling between design variables: it is only specific combinations of 3 locations that enable the small transmissibilities of Fig. 21. In Fig. 26 for example, the locations of the 2 fixed isolators selected by the GA are such that there is a feasible discrete location almost exactly on the anti-resonance line (dotted curve in the figure).

8. Experimental Results

The baseline configuration and all 9 optimized designs of Fig. 21 were tested in the laboratory. Because of the limited load capability of the shaker, the very large inertia of the seismic base, and the effectiveness of the optimized isolation systems, residual

velocities proved difficult to measure because of marginal signal to noise ratio and the presence of acoustic disturbances. To “clean up” the velocity signal, about 200 samples triggered on the 50Hz sinusoidal excitation signal were averaged in the time domain. The results are shown in Fig. 27 and compared to analytical predictions. The figure also shows predicted ranges of transmissibilities with ±5% scatter in the spring stiffnesses. Those ranges were obtained from Monte Carlo simulations using uniform distributions of spring stiffnesses with ±5% variations; the stiffnesses of the 3 spring isolators were varied independently from each other.

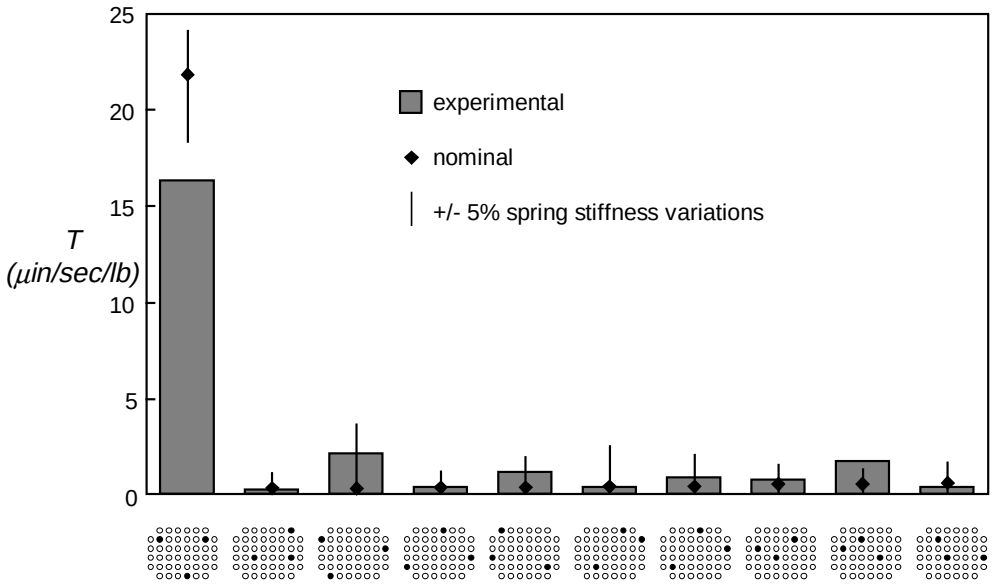

Figure 27 Experimental transmissibilities at 50 Hz of baseline and optimized designs, compared to analytical predictions.

The analytical-experimental agreement is qualitatively excellent: the dramatic improvement in performance achieved through optimization as predicted by the analysis is confirmed by the experiment. All 9 optimized designs perform between 7 and 46 times better than the baseline configuration (predicted ratios were between 32 and 70).

9. Conclusions

9.1 Optimum Isolation System Design

Our results confirm that in cases where vibration isolation must be provided at specific points/directions on a device and sufficient information is available about the disturbances, very significant improvements in performance can be achieved by explicitly optimizing the locations of the isolators. In the particular application, the performance of the optimized designs was predicted between 32 to 70 times better than that of a baseline configuration. Those improvements were also observed in the laboratory, with performance ratios between 7 and 46. It was also shown that, even though the optimization was formulated to “target” transmissibility at a single frequency, significant broad-band improvement was obtained in the optimized designs compared to the baseline configuration.

9.2 Discrete Optimization

- Applying constraints through penalty functions in a GA problem is a delicate operation. A balance must be achieved between the desire to obtain a feasible final design and the need to allow the search to cross infeasible regions of the design space. Surprisingly little research has been devoted to this aspect. One reason is that, in research GA's, problem-specific repair operators are often introduced to enforce constraints. This approach is more efficient but is highly application-specific and cannot be included in general purpose codes like DAKOTA.
- The combination of multi-modality, large number of constraints, and limited design options (coarse discrete grid in this case) makes the problem difficult to handle, even for a zero-order random search technique like the GA.
- The classical argument that a GA provides multiple design alternatives in its final population does not hold in heavily constrained discrete problems with small design spaces. Instead, each run provides only one or two acceptable designs.
- Multiple design options and improved reliability of the search can be obtained by running a few short searches, rather than a single long search.
- Continuous optimization followed by rounding to neighboring discrete solutions does not generally lead to an optimal design. For problems with coarse discrete grids, heavily constrained design space, and rapidly varying objective function, this approach leads to few, relatively poor feasible designs.

These observations have uncovered the need for further research if discrete optimization is to become a practical, easy to implement technique for use in real life design problems. In particular, techniques for efficient implementation of multiple constraints in genetic algorithm optimization need further exploration and development.

10. References

1. Crede, C. E., *Vibration and Shock Isolation*, J. Wiley & Sons, ed., New York, NY, 1951.

2. Ashrafiuon, H., "Design Optimization of Aircraft Engine-Mount Systems," *Journal of Vibration and Acoustics*, ASME, Vol. 115, October 1993, pp.463-467.

3. Hart, W. E., *Evolutionary Pattern Search Algorithms*, Sandia Report SAND95-2293, September 1995.

4. Eldred, M. S., Outka, D. E., Bohnhoff, W. J., Witkowski, W. R., Romero, V. J., Ponslet, E. R., and Chen, K. S., "Optimization of Complex Mechanics Simulations with Object-Oriented Software Design," *Computer Modeling and Simulation in Engineering*, to appear.

5. Newport Corp., Irvine, CA.

6. Associated Springs-Raymond (BARNES Group Inc.), Cerritos, CA 90703.

7. Firestone Industrial Products Company, Noblesville, IN.

8. Endevco Corp., San Juan Capistrano, CA.

9. PCB Piezotronics Inc., Depew, NY.

10. Wilcoxon Research Inc., Gaithersburg, MD.

11. MB Dynamics Inc., Cleveland, OH.

12. Pro-Engineer, *Users Manual*, Parametric Technology Group, Waltham, MA.

13. Meza, J. C., *OPT++: An Object-Oriented Class Library for Nonlinear Optimization*, Sandia Report SAND94-8225, March 1994.

14. *MATLAB Users Manual*, The MathWorks, Inc., Natick, MA.

15. Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold Pub., 1991.

16. Ponslet, E., Haftka, R. T., and Cudney, H. H., "Optimal Placement of Tuning Masses on Truss Structures by Genetic Algorithms," Proceedings of the 34th AIAA/ ASME/ ASCE/ AHS /ASC Structures, Structural Dynamics, and Materials Conference, April 19-22, 1993, La Jolla, CA, part 4, pp. 2448-2457 (AIAA paper 93-1586).

17. Haftka, R. T., Gurdal, Z., and Kamat, M. P., *Elements of Structural Optimization*, second revised edition, Kluwer Academic Pub., Dordrecht, The Netherlands, 1990.

18. Leriche, R., *Composite Structures Optimization by Genetic Algorithms*, Ph.D. Dissertation, Virginia Tech, Blacksburg, VA, 1994.

19. Richardson, J. T., Palmer, M. R., Liepins, G., and Hilliard, M., "Some Guidelines for Genetic Algorithms with Penalty Functions," Proceedings of the Third International Conference on Genetic Algorithms, George Mason Univ., Morgan Kaufmann Pub., June 4-7, 1989, pp. 191-197.

20. *DOT Users Manual*, Vanderplaats Research and Development, Inc., Version 4.20, Colorado Springs, CO, 1995.