An efficient and exact solution for Random Vibration analysis using MSC/NASTRAN. Part II: General Random Spectrum

E. de la Fuente and J. San Millán
Instituto Nacional de Tecnica Aeroespacial
Structures and Mechanisms Department
28850 Torrejon de Ardoz. Madrid, Spain.
email: fuentee@inta.es, sanmillan@inta.es

Presented at the 1999 MSC Aerospace Conference. June 7th-10th, 1999. Long Beach (CA)

Abstract

A new method for performing RANDOM vibration analysis within MSC/NASTRAN is presented in this paper. The method is a direct application of a well known result of Linear Systems Theory, and allows computation of RMS values of any number of structural outputs: displacements, stresses, element forces, accelarations, etc. These results can be postprocessed as if they were originated in a conventional static analysis (in colour plots for instance). The method is implemented within MSC/NASTRAN by means of a DMAP Alter, that is carefully described in the paper. The results of this DMAP are compared with those given by standard RANDOM solution of MSC/NASTRAN, in terms of accuracy, CPU time, and calculation capability, showing clear advantages of the presented method. Finally, extensions of the method are outlined.

1 Introduction

RANDOM vibration analysis is a common task in structural analysis. In the space field, for instance, random vibration is usually needed to simulate acoustic loading or loading due to atmosphere turbulence during launch of spacecrafts. In the aeronautical field, random analysis is required to assess the response of the aircraft structure to continuous turbulence (i.e., turbulence defined by its Power Spectral Density). Moreover, the need for including the effects of structural flexibility is becoming more and more usual, leading to the so called aeroelastic analysis.

The way in which the structural analysis packages (MSC/NASTRAN for instance) deal with the problem of random vibration starts with the calculation of the transfer functions of the outputs requested by the user, (i.e. stresses, forces, accelerations, etc.), followed by the computation of the individualized PSD of each output, and finally by performing a numerical integration of the resulting PSD curves. The final result of this process is the steady state RMS value of the corresponding output. This calculation process presents some difficulties very well known by the users:

- The outputs have to be requested one by one (no instruction such as STRESS=ALL is available). This inconvenient is important when dealing with large Finite Element Models.
- A significant computational effort is needed for each individual result, if this is required to high accuracy. This usually results, when dealing with big models, in long CPU times, and problems with the disk space.
- It is not possible to visualize RMS results (stresses for instance) in color or contour plots as it is normally made in static or modal analysis. This happens because RMS results are not stored in the OUTPUT file produced by MSC/NASTRAN. The postprocessing of the RMS results is obviously essential to have a clear idea of what part of the structure that is most loaded.
- The solution obtained is not exact because the results are calculated by means of a numeric integration. The exactness of the results depends on the ability of the user to choose a sufficient number of frequencies adequately distributed, for which MSC/NASTRAN calculates the transfer function of the requested outputs.

The method presented in this paper tries to solve all of those shortcomings. The method is taken from direct application of a known result of the Theory of Linear Systems, but that is however not handled by structural analysts, since its main use is in Linear Systems Control Theory. Its main advantage with respect to standard MSC/NASTRAN sequence (SOL 111), is that *exact* RMS results (stresses, forces, accelerations and so on), are computed using a process by far much cheaper (in terms of computational effort) than the standard one. A much lower number of mathematical operations is required for the obtention of a given result, therefore typical problems of non-enough disk space do not appear. Finally, it is also possible to postprocess the RMS results as if they came from a conventional static analysis (color plots can be obtained showing RMS stresses, displacements and so on). We have used MSC/ARIES to that purpose.

In a previous work ([5]), the analysis was limited to *uniform white noise input spectrum*. This paper follows the same theoretical approach, but generalizing the results to general random spectra. A DMAP ALTER was developed to implement the method in MSC/NASTRAN sequence SOL 11. This DMAP sequence is essentially the same as that provided in [5] but with the possibility of introducing general spectra.

The paper is structured as follows. First, the mathematical background of structural analysis under random loads is reviewed, taking the approach given in [5] as a starting point. Then, a direct application to structure-like systems is shown and the sequence of mathematical operations needed is identified. The different ways of characterizing general random spectra is then described. Afterwards, details of the implementation of this sequence into DMAP form is given, and the obtained results are validated through a representative example. Finally, details on possible extensions of the proposed method to more general problems is anticipated and discussed.

2 Mathematical Background

Ref. [1] gives a fairly complete and rigorous derivation of the theory of response of linear systems to white noise random excitation sources. The application of the theory to structural problems was described in detail in [5], where the excitation was assumed to be white noise. Here the extension to general PSD spectra will be dealt with, but before, a brief summary of the main results obtained in [5] will be presented.

2.1 Output variance matrix for linear systems submitted to white noise

It is well known from Linear Systems Theory how to calculate the RMS response of a linear, time invariant, dynamic system submitted to white noise zero mean excitation. Such a system, may always be written in the form

$$\begin{aligned} \dot{x}_e &= A_{ee}x_e + B_{ep}w \\ z &= C_{ze}x_e \end{aligned} \tag{1}$$

where A_{ee} is a stable matrix (i.e. all its eigenvalues have negative real parts)[1], x_e is the state vector, w is a vector of excitation sources consisting of zero mean white noise with PSD matrix given by the symmetric positive definite matrix W and z is the system output.

It can be demonstrated ([1]) that the *steady state* variance matrix X_{ee} of the state vector $x_e(t)$, defined as

$$X_{ee} = \lim_{t\to\infty} \mathcal{E}\left(x_e(t)x_e(t)^T\right) \tag{2}$$

($\mathcal{E}$ is the mathematical expectation operator), is given by the solution of the linear Lyapunov equation

$$A_{ee}X_{ee} + X_{ee}A_{ee}^T + B_{ep}WB_{ep}^T = 0 \tag{3}$$

The steady state variance matrix, Z, of the output $z(t)$, i.e.

$$Z = \lim_{t\to\infty} \mathcal{E}\left(z(t)z(t)^T\right) \tag{4}$$

can be calculated from X by

$$Z = C_{ze}X_{ee}C_{ze}^T \tag{5}$$

and finally, the steady state mean square values of the response are simply

$$\overline{z^2} = \text{diag}\left(C_{ze}X_{ee}C_{ze}^T\right) \tag{6}$$

2.2 Extension to general spectra

Although white noise can be used to estimate the response of a dynamic system in many practical situations, specially when the PSD does not vary substantially over the frequency range covered by the lowest natural modes, a formal extension to the general PSD spectrum case is straightforward. It suffices to recall that any PSD spectrum can be obtained as the output of some *linear* dynamic system submitted to a white noise input. Furthermore, the actual PSD spectrum can be approached as accurately as desired, the cost being only the complexity (i.e., the number of states) of the model. Hereinafter, this system, which in fact act as a filter, will be referred to as the *auxiliary system*.

The procedure for obtaining the description of the auxiliary system will be discussed later. Here, we will assumed that the linear system has been found, and is described by the state space equations

$$\begin{aligned} \dot{x}_n &= A_{nn}x_n + B_{np}w_{\text{white\_noise}} \\ f_p &= C_{pn}x_n \end{aligned} \tag{7}$$

where x_n is the state vector, A_{nn}, B_{np}, C_{pn} are the system matrices, $w_{\text{white\_noise}}$ is the input to the auxiliary system, consisting of p uncorrelated, unitary white noise inputs, and f_p is the output of the auxiliary system. The PSD of the input is the identity matrix I_{pp}, and the PSD of the output is given by

$$W_{pp} = H_{pp}(i\omega)\, H_{pp}(-i\omega)^T$$

[1] The response of an *unstable* system would be infinite.

where

$$H_{pp}(i\omega) = C_{pn}(i\omega I_{nn} - A_{nn})^{-1} B_{np}$$

is the transfer matrix of the auxiliary system.

When the output f_p of the auxiliary system (7) is injected as input to the main system (1), i.e., w in (1) is taken as f_p in (7) the equations of the complete system are given by

$$\begin{aligned} \dot{x}_s &= A_{ss}x_s + B_{sp}w_{\text{white\_noise}} \\ z &= C_{zs}x_s \end{aligned} \tag{8}$$

where $\dot{x}_s = \begin{bmatrix} x_e & x_n \end{bmatrix}$ is the state vector of the complete system,

$$\begin{aligned} A_{ss} &= \begin{bmatrix} A_{ee} & B_{ep}C_{pn} \\ 0 & A_{nn} \end{bmatrix} \\ B_{sw} &= \begin{bmatrix} 0 \\ B_{np} \end{bmatrix} \\ C_{zs} &= \begin{bmatrix} C_{ze} & 0 \end{bmatrix} \end{aligned}$$

are the system matrices of the complete system. Note that the complete system is now submitted to white noise input, and thus, the theoretical frame described in the former section can be applied. Should the auxiliary system provide exactly the required output PSD, the complete system will in turn provide the *exact* solution to the problem of a system submitted to general spectrum. In any case, since any output PSD can be approached as accurately as desired, the exact solution can also be approached as accurately as needed.

Before going into details on how the procedure can be applied to a general structural problem, we will describe how the auxiliary system can be synthesized to produce the required PSD spectrum.

3 Synthesis of the auxiliary system

3.1 Theoretical background

The problem we must solve can be stated as follows: for a given PSD spectrum $W_{\text{req}}(\omega)$, how can we obtain a linear dynamical system whose output is $W_{\text{req}}(\omega)$ when submitted to unitary white noise, zero mean input.

This problem is well known to both dynamic system control and signal processing engineers, so that a number of solution strategies are available. However, since the structural engineer is not usually acquainted with those procedures, we will give some general details about the solution. We will start with the problem of one single source of excitation, and after the general problem of multiple correlated sources of excitation will be dealt with.

3.1.1 Single source of random excitation

Assume a single input single output linear system described by the state equations (7), but now the B and C matrices are vectors, that is,

$$\begin{aligned} \dot{x} &= A_{nn}x + b_n w_{\text{white\_noise}} \\ z &= c_n^T x \end{aligned}$$

where A is a $n \times n$ matrix and b and c are column vectors of size $n \times 1$. The input $w_{\text{white\_noise}}$ is a scalar white noise signal whose PSD is equal to 1. The frequency response function of the system is given by

$$h(s) = \frac{z(s)}{w_{\text{white\_noise}}(s)} = c^T(sI - A)^{-1}b = \frac{p(s)}{q(s)}$$

with I the identity matrix, and $p(s)$ and $q(s)$ are real polynomials in the Laplace variable s, with real coefficients. The PSD of the output is, in the frequency domain

$$W(\omega) = |h(i\omega)|^2 = h(-i\omega)\,h(i\omega)$$

and our problem is to find A, b and c for a given function $W_{\text{req}}(\omega)$.

The first fact that must be known is that, for a given accuracy, there is an infinite number of such systems. This can be easily understood by the following reasoning. Assume that $h(s)$ is both *stable* and *minimum phase*[2]. It is clear that the modulus of the transfer function is unaltered when multiplied by factors of the type

$$\frac{s+a}{s-a}$$

The only change that this additional factors produces is in the phase, which does not have any influence on the PSD. If $a > 0$, the original system remains minimum phase but becomes unstable, and if $a < 0$, it remains stable but becomes non-minimum phase. We conclude that there is an infinite number of systems having the same modulus of their transfer functions. However, it was demonstrated by Bode that *if we restrict ourselves to stable and minimum phase systems, then the solution is unique.*

The above considerations leads us to an immediate formulation of the problem. Let us call $h_{\text{approx}}(s)$ be an analytical transfer function which gives an approximation to the required PSD law, $W_{\text{req}}(\omega)$. We will assume that this system is both stable and minimum phase, and hence, it can be written in the form

$$h_{\text{approx}}(s) = k\frac{(s+\alpha_3)\prod_i (s^2+\alpha_{1i}s+\alpha_{2i})}{(s+\beta_3)\prod_j (s^2+\beta_{1j}s+\beta_{2j})}$$

with all k, α's, and β's are positive[3]. The factors $(s+\alpha_3)$, $(s+\beta_3)$ may or may not exist, depending on whether the number of zeroes or poles is odd or even. In any case, the *approximate* PSD will be given by

$$W_{\text{approx}} = \left|h_{\text{approx}}(i\omega)\right|^2$$

A "natural" manner of obtaining the solution is as follows. Given a function $W_{\text{req}}(\omega)$, and an assumed order of the numerator and denominator polynomials, find the positive scalars k, α_{1i}, α_{2i}, β_{1i}, β_{2i}, (α_3, β_3) which make the expression

$$\int \left(W_{\text{req}}(\omega) - W_{\text{approx}}(\omega)\right)^2 d\omega$$

to be minimum. The above expression is a measure of the error between the required and the approximate PSD's.

In practice, it happens that the analytical expression of $W_{\text{req}}(\omega)$ is not known, the curve being given defined piecewise by its values at specific frequencies. In this case, the error function to be minimized may be expressed as a sum of error terms calculated at different frequencies. Also, the error may be weighted in the frequency domain by a weighting function $g(\omega)$, which can be used to penalize the error in some frequency ranges. For this case, the function to be minimized may be expressed in a form such as

$$\min_{k,\alpha,\beta>0} \sum_l \left(W_{\text{req}}(\omega_l) - W_{\text{approx}}(w)\right)^2 g(\omega_l)$$

The above is only given as an illustration on how the problem may be formulated and solved without getting very much inside the problem. Other more elaborate formulations can be easily envisaged. For instance, an interesting alternative may be the following optimization problem

$$\begin{gathered}\text{minimize } \epsilon \text{ subjected to} \\ (1-\epsilon)\, W_{\text{req}} < W_{\text{approx}} < (1+\epsilon)\, W_{\text{req}}\end{gathered}$$

which tries to bound as tightly as possible the solution W_{approx} between an upper and lower bounds. From above considerations, it is clear that there should be no special difficulty in obtaining an approximate representation of the

[2]The system is *stable* if all roots of the denominator polinomial have negative real parts, and will be of *minimum phase* if the same property holds for the numerator polinomial.

[3]It is clear that, for the system to be physically realizable, the order of the denominator polinomial must be greater or equal than that of the numerator.

Frequency (Hz)	PSD (g^2/Hz)	Transfer function (g)
$0-100$	20 dB/dec	20 dB/dec
$100-400$	0.2	$\sqrt{0.2}$
$400-\infty$	-20 dB/dec	-20 dB/dec

Table 1: An example of PSD spectrum

auxiliary system. And it is to be known that the approximation can be, in principle, as accurate as needed, simply by increasing the order of the system, that is, the order of the numerator and denominator polynomials. Once the coefficients of these polynomials are known, the description of the system in state space form can be obtained by standard techniques.

Of course, there are specific powerful algorithms which automate the process. One of these is given in [7] and is called "complex cepstrum". An implementation of this algorithm in MATLAB can be found in [6], in a routine called FITMAG. In fact, this was the algorithm that we have used in the examples which follows.

The procedure described above can be used in a straight manner for single source of excitation[4]. It is still applicable when multiple excitation sources are present, provided they are uncorrelated, that is, their PSD is a diagonal matrix. This case will lead to as many auxiliary systems as there are inputs, all of them merged diagonally into a single system.

3.1.2 Multiple and correlated sources of excitation

The case of multiple and correlated sources of excitation leads to a non diagonal (although still symmetric and positive definite) PSD matrix. This case is much more difficult to handle than the former one. The auxiliary system will take the original form (7). Assuming that there are p sources of excitation and that $w_{\text{white\_noise}}$ is a process of PSD equal to the identity matrix, I_{pp}, the transfer matrix of the system from the input $w_{\text{white\_noise}}$ to the output z is

$$H_{zp}(s) = C_{zn}(sI - A_{nn})^{-1} B_{np}$$

and the PSD matrix of the output z is given by the real, symmetric and positive definite matrix

$$W_{\text{approx}}(\omega) = H_{zp}(i\omega)\, H_{zp}(i\omega)^H$$

where the superscript $()^H$ means the transpose conjugate. Now the problem can be formulated in the following way: given a symmetric positive definite matrix $W_{\text{req}}(\omega)$, find the matrices A_{nn}, B_{np} and C_{zn} matrices, so that the scalar

$$\left\| W_{\text{req}}(\omega) - W_{\text{approx}}(\omega) \right\|$$

is minimum. Here, $\|.\|$ means any kind of suitable norm, usually $\|.\|_2$ or $\|.\|_\infty$.

This problem, as stated above, is considerably more difficult than the previous one. The reason is twofold:

- It is not easy to formulate the problem in terms of the coefficients of the Transfer Matrix, as was done in the previous case, and
- The representation of a system in state space form, that is, by means of the matrices A_{nn}, B_{np} and C_{zn}, being easier, is not unique.

These two difficulties may become important for the optimizer if the number of correlated inputs is large.

3.2 Examples of application

As a first example of application of the above theoretical discussion, consider a PSD spectrum given by the law shown in Table 1.

[4] It should be clear that a "single source of excitation" by no means imply a "single force". It means that all the forces acting on the structure are correlated.

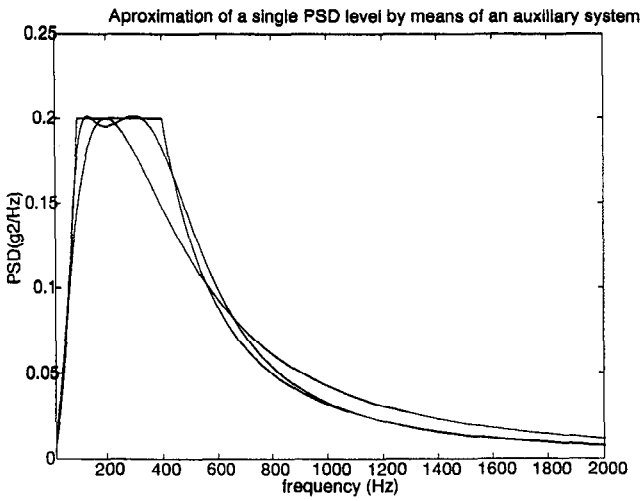

Figure 1: Approximation of a one PSD level spectrum by means of an auxiliary system.

A very first examination of the required $h(\omega)$ law reveals that this function may be approximated with a zero at $\omega = 0$ and poles at $\omega = 2\pi \times 100$ and at $\omega = 2\pi \times 400$ rad/s. Furthermore, the transfer function must decrease at large frequencies at a rate of 20 dB/dec, that is, as $1/s$. The resulting transfer function is

$$h(s)_{\text{approx}} = k \frac{s}{(s + 200\pi)(s + 800\pi)}$$

where the value of the gain k is calculated so as to give a value of $\sqrt{0.2}$ at a frequency of $\sqrt{200\pi \times 800\pi}$, resulting in $k = \frac{200\pi}{\sqrt{0.2}}$. The transfer function is finally given by

$$h_{\text{approx}} = \frac{200\pi}{\sqrt{0.2}} \frac{s}{(s + 400\pi)(s + 800\pi)} \tag{9}$$

If a higher order approximation is required, we can use directly the command FITMAG available from MATLAB ([6]). In this case it is used a weighting factor of 5 in the range of constant PSD, and it is required a system of four states. Figure 1 shows the required spectrum, and the PSD provided by the transfer functions given in (9), and the four order system obtained from MATLAB. It is clear that the approximation is excellent for the four order system, but for practical purposes the two order system gives an acceptable PSD representation.

It is to be noted that the four order system transfer function obtained is not minimum phase. In fact, the original result provided by FITMAG was indeed minimum phase, but it had a small feedthrough D matrix, which has been eliminated. The resulting PSD is practically unaffected.

A state space representation of this system is given by:

$$\begin{aligned} A_{nn} &= 10^3 \times \begin{bmatrix} -2.3272 & -1.2611 & 1.5324 & 0.0156 \\ 1.2611 & 0.0000 & 0.0011 & 0.0000 \\ -1.5324 & 0.0011 & -1.6782 & -1.2720 \\ 0.0156 & 0.0000 & 1.2720 & -0.0003 \end{bmatrix} \\ B_{np} &= \begin{bmatrix} -34.1365 & 0.0295 & -9.9223 & 0.1275 \end{bmatrix}^T \\ C_{pn} &= \begin{bmatrix} -34.1365 & -0.0295 & 9.9223 & 0.1275 \end{bmatrix} \end{aligned}$$

As a second example, we will take the PSD spectrum given in Table 2.

Frequency (Hz)	PSD (g^2/Hz)	Transfer function (g)
25-200	20 dB/dec	20 dB/dec
200-400	1	1
400-800	20 dB/dec	20 dB/dec
800-1000	4	2
1000-2000	-20 dB/dec	-20 dB/dec

Table 2: Two level PSD spectrum

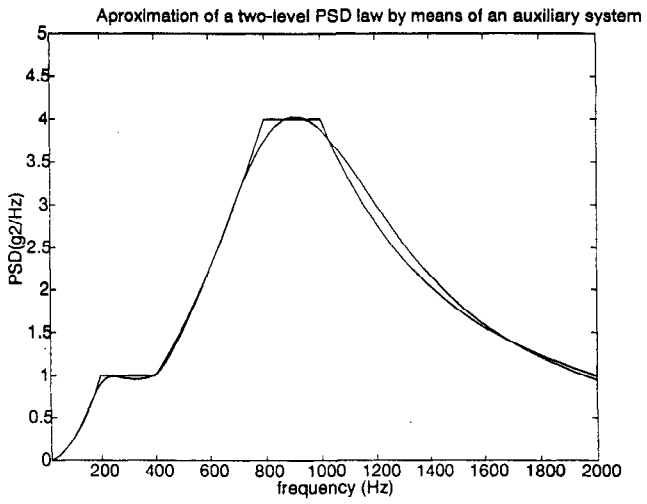

Figure 2: Approximation of a two level PSD spectrum by means of an eighth order auxiliary system

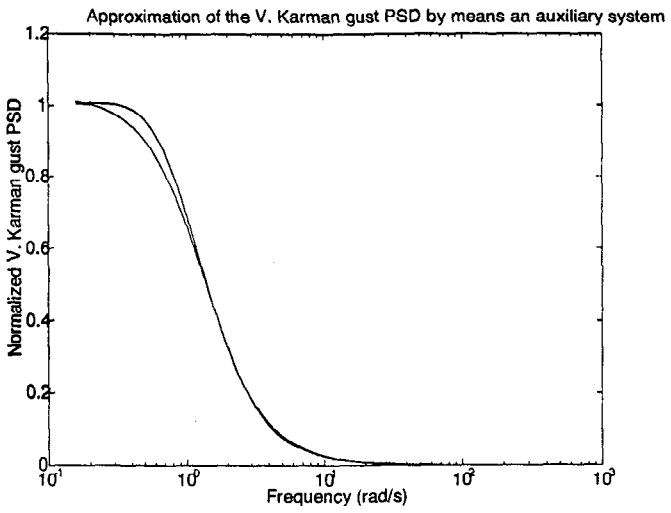

Figure 3: Approximation of Von Karman gust PSD by a second order linear dynamic system.

Using command FITMAG from MATLAB [6], and requiring a system of 8 states, the approximation found is, in state space representation

$$A_{nn} = 10^4 \times \begin{bmatrix} -0.5895 & -0.5299 & -0.2283 & 0.1401 & -0.0016 & 0.2150 & 0.2184 & -0.0130 \\ 0.5299 & -0.0021 & -0.0086 & 0.0114 & -0.0001 & 0.0141 & 0.0143 & -0.0008 \\ 0.2283 & -0.0086 & -0.0503 & 0.1640 & -0.0009 & 0.1321 & 0.1337 & -0.0077 \\ 0.1401 & -0.0114 & -0.1640 & -0.0931 & 0.0021 & -0.2400 & -0.2452 & 0.0154 \\ 0.0016 & -0.0001 & -0.0009 & -0.0021 & -0.0001 & 0.5841 & 0.2510 & -0.0021 \\ 0.2150 & -0.0141 & -0.1321 & -0.2400 & -0.5841 & -2.2000 & -2.3216 & 0.2290 \\ 0.2184 & -0.0143 & -0.1337 & -0.2452 & -0.2510 & -2.3216 & -2.4536 & 0.2488 \\ -0.0130 & 0.0008 & 0.0077 & 0.0154 & 0.0021 & 0.2290 & 0.2488 & -0.0520 \end{bmatrix}$$

$$B_{np} = \begin{bmatrix} -108.0515 & 6.0625 & 15.7287 & 14.2541 & 0.1459 & 19.8795 & 20.1871 & -1.1919 \end{bmatrix}^T$$

$$C_{pn} = \begin{bmatrix} -108.0515 & -6.0625 & -15.7287 & 14.2541 & -0.1459 & 19.8795 & 20.1871 & -1.1919 \end{bmatrix}$$

Figure 2 represents the required and approximate systems. It can be seen that the adjustment is excellent.

As a final example, let us consider the gust velocity PSD given by the Von Karman law. For vertical and lateral gusts, it is defined by the following mathematical expression

$$W_{\text{Von Karman}}(\omega) = \frac{1 + \frac{8}{3}1,339\omega^2}{\left(1 + (1.339\omega)^2\right)^{11/6}}$$

where $W_{\text{Von Karman}}(\omega) = \Phi(\Omega)\frac{\pi}{L\sigma_w^2}$, σ_w is the RMS gist velocity, L is a length scale of turbulence, and $\omega = \Omega L$. This PSD function cannot be generated by a linear system since the exponents of ω are even not integers. However, an approximation may be found by the same procedure as before. The results is

$$h(s)_{\text{approx}} = \frac{1.9557s + 21.5164}{s^2 + 17.0638s + 21.2553}$$

The resulting PSD curves are shown in Figure 3

4 Application to structural models

4.1 Dynamic equations of a structural system (frequency response problem).

Following [5] the dynamic equilibrium equations of a structure, expressed in modal coordinates can be written as

$$\ddot{u}_h + B_{hh}\dot{u}_h + K_{hh}u_h = B_{hp}f_p$$

and in state space form as

$$\dot{x}_e = A_{ee}x_e + P_{ep}f_p \tag{10}$$

where x is the state vector,

$$x_e = \left\{ \begin{array}{c} u_h \\ \dot{u}_h \end{array} \right\} \tag{11}$$

of order $2n_h$, u_h is the $n_h \times 1$ vector of generalized modal coordinates, A_{ee} is the system matrix,

$$A_{ee} = \left[\begin{array}{cc} 0_{hh} & I_{hh} \\ -\Omega_{hh}^2 & -G_{hh}\Omega_{hh} \end{array} \right] \tag{12}$$

where 0_{hh} and I_{hh} are zero and identity matrices of order h respectively,

$$\Omega_{hh} = \text{diag}\,(\omega_1, \omega_2, \ldots, \omega_h) \tag{13}$$

being ω_i the natural frequency of the i-th mode, and

$$G_{hh} = \text{diag}\,(g_1, g_2, \ldots, g_h) \tag{14}$$

where g_i corresponds to the modal structural damping associated to mode i. The above assumes that the natural modes are normalized with respect to the mass matrix. Finally the load influence matrix P_{ep} is

$$P_{ep} = \left[\begin{array}{c} 0_{hp} \\ P_{hp} \end{array} \right] \tag{15}$$

The response variables, that is those structural output variables in which the user is interested, can usually be expressed as a linear function of displacements, and less often in terms of velocities and accelerations. To simplify the notation we will assume that the accelerations are not present in the response variables. In this case, the response variables can be expressed in the form

$$z_k = C_{ke}x_e$$

The matrix C_{ke} can be recovered directly from MSC/NASTRAN as shown in [5].

4.2 Auxiliary system

From the guidelines given in 3, the auxiliary system may be expressed in the form

$$\begin{aligned} \dot{x}_n &= A_{nn}x_x + B_{np}w_{\text{white\_noise}} \\ f_p &= C_{pn}x_n \end{aligned}$$

4.3 Complete system

The equations of the complete system are

$$\begin{aligned} \dot{x}_s &= A_{ss}x_s + B_{sp}w_{\text{white\_noise}} \\ z_k &= C_{ks}x_s \end{aligned}$$

where $x_s = \left[\begin{array}{cc} x_e & x_n \end{array} \right]$ is the complete state vector,

$$\begin{aligned} A_{ss} &= \left[\begin{array}{cc} A_{ee} & B_{ep}C_{pn} \\ 0_{en} & A_{nn} \end{array} \right] \\ B_{sp} &= \left[\begin{array}{c} 0_{ep} \\ B_{np} \end{array} \right] \\ C_{ks} &= \left[\begin{array}{cc} C_{ks} & 0_{kn} \end{array} \right] \end{aligned}$$

are the state, load influence and output influence matrices respectively.

The Lyapunov equation (3) can be solved explicitly if the eigenvalues and eigenvectors of the system are known (see for instance [3] and [5]). Although the eigenvalues and eigenvectors of the complete system can be calculated directly by MSC/NASTRAN, it is much better to obtain them by the following procedure. In this way, they are obtained in a known order, and this is necessary for the procedure to work.

Let Λ_{ee} and E_{ee} be the eigenvalues and eigenvectors of the structural state matrix A_{ee}, i.e. they verify the equation

$$A_{ee}E_{ee} = E_{ee}\Lambda_{ee} \tag{16}$$

It can be easily verified that they can be obtained as

$$\Lambda_{ee} = \left[\begin{array}{cc} \Lambda_{hh} & 0_{hh} \\ 0_{hh} & \Lambda_{hh}^H \end{array} \right] \tag{17}$$

and

$$E_{ee} = \left[\begin{array}{cc} I_{hh} & I_{hh} \\ \Lambda_{hh} & \Lambda_{hh}^H \end{array} \right] \tag{18}$$

the superindex H meaning, as before, complex conjugate transpose. The diagonal matrix Λ_{hh} can be easily calculated from natural frequencies and modal dampings as

$$\Lambda_{hh} = \Omega_{hh}\left(-\frac{1}{2}G_{hh} + i\sqrt{I_{hh} - \frac{1}{4}G_{hh}^2} \right) \tag{19}$$

Now, let Λ_{nn} and E_{nn} be the eigenvalues and eigenvectors of the auxiliary system, that is, they verify

$$A_{nn}E_{nn} = E_{nn}\Lambda_{nn}$$

The eigenvalues and eigenvectors of the complete system can be constructed from

$$\begin{aligned} \Lambda_{ss} &= \left[\begin{array}{cc} \Lambda_{ee} & 0 \\ 0 & \Lambda_{nn} \end{array} \right] \\ E_{ss} &= \left[\begin{array}{cc} E_{ee} & X_{en} \\ 0_{ne} & E_{nn} \end{array} \right] \end{aligned} \tag{20}$$

where the matrix X_{en} is given by

$$X_{en} = E_{ee}\bar{X}_{en}$$

and the i, j element of the matrix $\bar{X}_{en}$ is given by

$$(\bar{X}_{en})_{ij} = -\frac{(E_{ee}^{-1} B_{ep} C_{pn} E_{nn})_{ij}}{(\Lambda_{ee})_{ii} - (\Lambda_{nn})_{jj}}$$

This result can be verified simply by substitution.

Once the eigenvalues and eigenvectors of the complete system are known, the solution to the Lyapunov equation can be calculated in the same way as in [5], that is, the solution of

$$A_{ss} X_{ss} + X_{ss} A_{ss}^T + P_{sp} W_{pp} P_{sp}^T = 0 \tag{21}$$

is given by

$$X_{ss} = E_{ss} \bar{X}_{ss} E_{ss}^H \tag{22}$$

where

$$(\bar{X}_{ss})_{ij} = -\frac{(\bar{W}_{ss})_{ij}}{(\Lambda_{ss})_{ii} + (\Lambda_{ss}^H)_{jj}} \tag{23}$$

and

$$\bar{W}_{ss} = E_{ss}^{-1} P_{sp} W_{pp} P_{sp}^T E_{ss}^{-H}$$

The steady state variance of the structural responses is given by

$$Z_{kk} = C_{ks} X_{ss} C_{ks}^T \tag{24}$$

and the steady state mean square values of the response,

$$\overline{z_k^2} = \text{diag}\left(C_{ks} X_{ss} C_{ks}^T\right) \tag{25}$$

Finally, the steady state RMS value of the structural response is given by

$$z_{\text{rms}} = \sqrt{\frac{\text{diag}\left(C_{ks} X_{ss} C_{ks}^T\right)}{2}} \tag{26}$$

The factor 2 comes from the fact that in the above derivation it was supposed that the frequency spanned from $-\infty$ to $+\infty$, while in structural applications it is more usual to restrict the frequencies to take positive values only.

The expression (26) is the final results of our analysis. They give the *exact* solution for the RMS values of the responses of a structure submitted to a random excitation with PSD given by the symmetric positive definite matrix $W_{pp}(\omega)$.The whole procedure can now be summarized as follows:

1. Obtain the matrix of modal responses for the desired output variables. This will give the matrix C_{ke}
2. Obtain the eigenvalues and eigenvectors of the state matrix from the natural frequencies and modal dampings, by means of equations (19), (17) and (18)
3. Obtain the eigenvalues Λ_{nn} and eigenvectors E_{nn} of the auxiliary system. These can be obtained by use of the Module CEAD in MSC/NASTRAN.
4. Obtain the eigenvalues Λ_{ss} and eigenvectors E_{ss} of the complete system from (20)
5. Obtain the state variance matrix X_{ss} by means of equations (23) and (22)
6. Perform the matrix operation (26) to obtain the RMS values of the desired outputs.

5 Programming within MSC/NASTRAN

Following the operations sequence shown in preceding paragraph, a DMAP alter has been prepared for RANDOM problems solved within modal formulation. This formulation is more usual than the direct one when dealing with medium/high size structures. Particularly the ALTER is written for MSC/NASTRAN version 69, SOL 11.

The PSD matrix W_{pp} of the external forces (see [5] and preceding section of this paper) must be introduced by the user via DMI cards on the Bulk Data Section. This matrix is square and it has a size equal to the number of independent excitations to the structure. Moreover, the matrices that define the auxiliary system: A_{nn}, B_{np}, and C_{pn} have also to be introduced on the Bulk Data Section by means of DMI cards.

With respect to the capabilities of our DMAP ALTER, it allows to recover RMS values of stresses, element forces and displacements. For the rest of output variables: SPC or MPC forces and so on, a few lines should be added to the ALTER. In particular, MPC forces can be calculated for instance using the theory given in [4].

The displacements are printed in the .f06 file in the standard format of static solutions of MSC/NASTRAN, while the stresses and element forces are printed in the .f06 file in matrix form (using module MATPRN), and an auxiliary index table is also printed to allow the identification of the different items (stresses or forces). It is impossible to print these results in the standard format of MSC/NASTRAN. This inconvenient is produced by the poor capability of DMAP programming language when managing DATABLOCKS (stresses and element forces are stored on datablocks while displacements are stored both in matrix and in datablock form).

For the postprocessing, the stresses and the element forces matrix are output (via OUPTPUT4) in ASCII form to an external file, so that they can be used by a postprocessor. Then, a simple external computer program is needed to translate the stress/force results into postprocessor format. In the authors' case the postprocessor used was MSC/ARIES which eases considerably the above process, since the results file is written in ASCII format (.anf extension). Following this procedure, it was possible to visualize the results in the same way as it is usually made in static analysis, and color contour plots of RMS displacements and stresses have been obtained without difficulty. This allows the user to have a clear map of the stress (or displacement or force) level over the structure, so critical zones are easily identified. This capability is not implemented in the standard RANDOM solution of MSC/NASTRAN.

6 Examples

The DMAP Alter sequence has been validated, comparing its results with the outputs obtained from MSC/NASTRAN standard solution SOL 111. A relatively complex model is used (it represents an Optical monitoring camera, mounted onto the INTEGRAL ESA satellite), and it is shown in Figure 1. The size of the model is about 6000 degrees of freedom, and about 1100 elements (most of them beam and shell types). The comparison has been made for stresses, both in terms of accuracy and computation time. The external input consists of an unique random excitation introduced as a concentrated force, whose PSD spectrum correspond to the two PSD level example shown in section 3.2. This spectrum has a level of 1 $\frac{g^2}{Hz}$ from 200 to 400 Hz, and a second one of 4 $\frac{g^2}{Hz}$ from 800 to 1000 Hz. The frequency range of the spectrum is 25 - 2000 Hz.

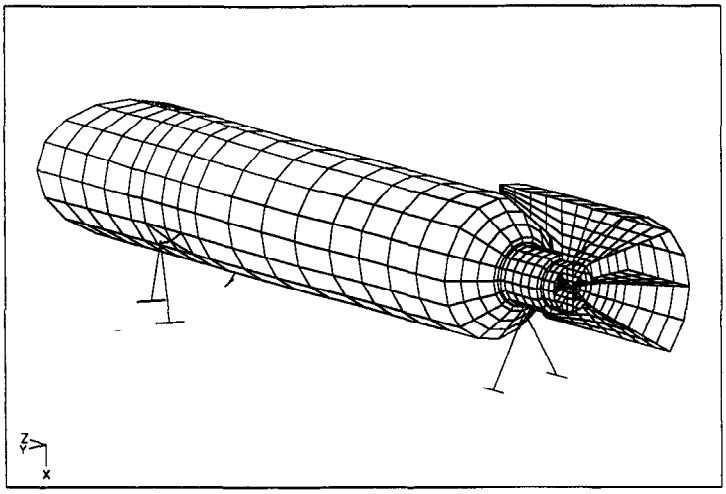

Figure 4: Finite Element Model of the Optical Monitoring Camera

	Number of Frequencies	Interpolation of Frequencies	CPU Time (sec.)	Average Error on RMS results
Std. MSC/NASTRAN	50	Logarithmic	42.2	12 %
Std. MSC/NASTRAN	100	Logarithmic	45.6	4 %
Std. MSC/NASTRAN	200	Logarithmic	53.5	0.1 %
Std. MSC/NASTRAN	400	Logarithmic	72.1	Taken as errorless
This Method	N.A.	N.A.	46.9	1 %

Table 3: Results Comparison for the OMC Model

The comparison for RMS stresses is shown in Table 3. When using the standard solution of MSC/NASTRAN, the user has to define a set of frequencies that the program uses to calculate the frequency response of the output variables, and then a numerical integration is performed to derive the RMS values (this was explained in section 1). In this case FREQ2 cards were used, that means that the frequencies are in log equally spaced along the total frequency range (25-2000 Hz). As it can be seen in the table it was necessary to choose a relatively high number of frequencies, to get accurate results. RMS results are practically the same for 200 and 400 frequencies, so for the calculation of the last column of the table (Average error on RMS results), the results correspondent to 400 frequencies are considered exact.

However, with our method no set of frequencies has to be defined when using it (this is an advantage), and the obtained RMS results are almost exact. The matrices A_{nn}, B_{np}, and C_{pn} introduced in the analysis are the ones shown in section 3.2. The small error of the method is originated because the approximation of the PSD spectrum by means of an auxiliary system, is not exact, although this error or difference can be minimized as much as desired, just increasing the order of the auxiliary system, or even better by penalizing more heavily the approximation error in the area of the first natural frequencies. Also it has to be taken into account, that the Finite Element Model was deliberately chosen so as having its lowest natural frequencies, at the first constant level of the spectrum where the approximation of the auxiliary system is poorer. Anyway, as it is shown in the table, the proposed method presents an optimum Accuracy/CPU Time ratio. For displacements and element forces, similar checking have been done, and completely similar results were obtained.

Another comparison was done using a bigger Finite Element Model: a plate model constructed with linear solid elements TETRA. This model is shown in Figure 5. The number of elements is about 23000. The plate is simply

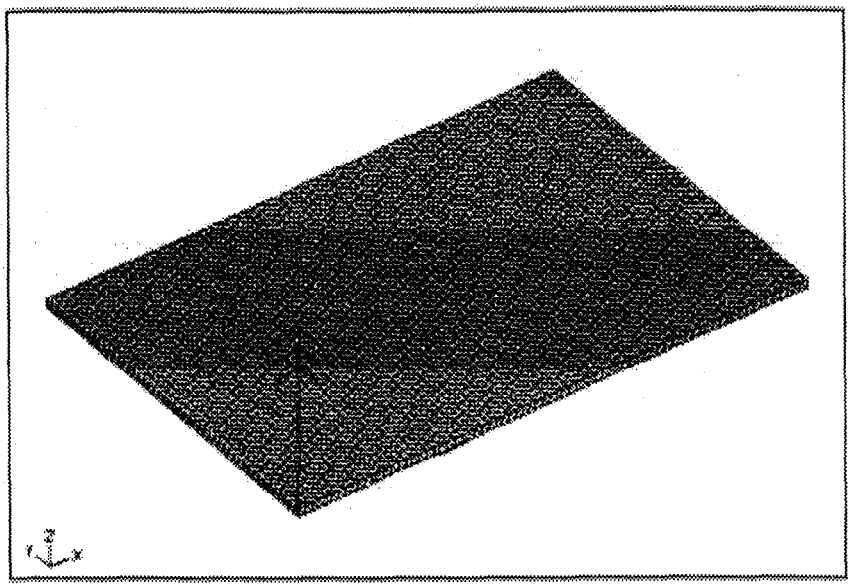

Figure 5: Finite Element Model of the plate

supported in its four corners, and the external RANDOM input consists of a vertical concentrated force at its center. The PSD spectrum correspond to the single PSD level example shown in section 3.2. This spectrum has a constant level of 0.2 $\frac{g^2}{Hz}$ from 100 to 400 Hz, and the complete frequency range of the spectrum is 20 - 2000 Hz.

For the frequencies needed by MSC/NASTRAN to calculate the frequency response of the output variables, FREQ2 cards were used also in this example, that means that the frequencies are log equally spaced along the total frequency range (20-2000 Hz). RMS results are practically the same for 200 and 300 frequencies, so for the calculation of the two last columns of the table (Average and Maximum error on RMS results), the results correspondent to 300 frequencies are considered exact. As it is shown in table 4, the standard solution of MSC/NASTRAN is clearly limited in the number of structural response variables requested by the user, and the number of frequencies that the user defines for the calculation of the transfer function of the output variables. When this limit is exceeded, the program aborts because of lack of disk space. As it is shown for 50 frequencies, that makes the RMS results obtained to be not sufficiently exact (average error of 17.5 %), MSC/NASTRAN is not able to finish a run with 120000 RMS stresses requested, that is still a relatively moderate number of outputs. If a higher number of frequencies is defined to get more accurate results, this limit number of stresses requests decreases drastically, as shown in the Table.

However using our method there is actually no limitation in the number of responses requested (a run with 4×10^5 requests finished successfully, and this number is not a limit at all), and these results are almost exact (only an average error of 0.2 %). Again the results contain a small error, that it is originated because the approximation of the PSD spectrum by means of the auxiliary system, is not exact. In this case, the matrices A_{nn}, B_{np}, and C_{pn} introduced in the analysis, are the ones correspondent to the fourth order system (see section 3.2). The CPU times shown in the table are very much penalized because of the DMAP limitations (this is carefully explained in next section). All these checkings were made with MSC/NASTRAN version 69 for Windows NT, running on a PC Pentium Pro of 4 Gbytes of disk space, and 128 Mbytes of RAM memory.

6.1 A comment on computational effort

It is worthwhile to point out that in the matrix operation (26) only the *diagonal* of the matrix $C_{ke}X_{ee}C_{ke}^T$ is needed, and so it is unnecessary to perform explicitly the matrix product. In fact, this matrix may be quite large in practical

	Number of Frequencies	Number of σ RMS requested	Interpolation of Frequencies	CPU Time (sec.)	Average Error on RMS Results
Standard MSC/NASTRAN	50	105000	Logarithmic	264.8	17.5 %
Standard MSC/NASTRAN	50	120000	Logarithmic	Aborted	-
Standard MSC/NASTRAN	100	60000	Logarithmic	251.7	1.4 %
Standard MSC/NASTRAN	100	75000	Logarithmic	Aborted	-
Standard MSC/NASTRAN	200	30000	Logarithmic	240.6	0.4 %
Standard MSC/NASTRAN	200	37500	Logarithmic	Aborted	-
Standard MSC/NASTRAN	300	24000	Logarithmic	257.4	Taken as errorless
Standard MSC/NASTRAN	300	30000	Logarithmic	Aborted	-
This Method	N. A.	100000	N.A.	313.2	0.2 %
This Method	N. A.	200000	N.A.	656.4	0.2 %
This Method	N. A.	400000	N.A.	1836.8	0.2 %

Table 4: Results Comparison for the Solid Plate Model

applications. If n_k is the number of structural responses requested, and n_h is the number of elastic modes included in the analysis, the operation (26) needs a total of about $2n_k n_h\,(2n_h + 1)$ flops, whereas performing the matrix product first and then extracting the diagonal would take about $2n_k n_h\,(2n_h + n_k)$ flops, that is about n_k times higher. Since n_k may be very large, the difference in CPU time would be very significant. Also, it has to be taken into account that the matrix $C_{ke}X_{ee}C_{ke}^T$ is $n_k \times n_k$ size, so for large n_k (4×10^5 for instance in our plate example), the space disk consumed to store the matrix is prohibitive.

Given that the matrix operation (26) cannot be made in the optimum way described above, since no DMAP instruction performs this specific operation, and the second posibility is very unefficient, requiring an enormous disk space available, an alternative procedure was followed. The optimum compromise we were able to find, was to perform the matrix operation (26) by blocks of some predefined size m (default value in the DMAP ALTER is 100), and extracting the diagonal of each resulting matrix of size $m \times m$. This operation is repeated until the full diagonal is constructed. This process is still much slower than the one actually needed, that is, obtaining the diagonal directly. This is becuase many intermediate operations (partitioning and merging matrices) have to be done, and must be neccesarily done using DO loops. Therefore, the time consumed by the computer in the compilation and interpretation of these unnecessary instructions, and above all the DO loops, causes a dramatic increase in the CPU time, much higher to that strictly requested to perform the operations themselves. From the matemathical point of view, the number of operations needed by our compromise to perform the final matrix product is $2n_k n_h\,(2n_h + m)$, that is $\frac{2n_h+m}{2n_h+1}$ higher that the number required with the optimum procedure described above. For instance for the plate example shown in preceding section, the number of modes used in the analysis is 30, so the above ratio becomes $\frac{60+100}{60+1} \simeq 2.5$. So it is clear that the procedure we follow, is strongly penalized in time because makes a considerable higher number of operations, and has to use DO loops.

Taking into account that this final matrix product (26) consumes most of the CPU time, it can be concluded that if a DMAP instruction that performed the exact operation needed was available, the CPU times consumed by our method would be clearly inferior to those shown in tables 3 and 4. Anyway the procedure proposed in this paper is more efficient than the conventional one, because the number of mathematical operations needed is lower by orders of magnitude.

7 Future extensions

The capabilities of the procedure described in this paper can be substantially increased by further extensions. Description of these extensions that can be easily incorporated is given below.

- *Extension to direct formulation.* The procedure described as such, works only with modal formulation for dynamic problems. By far, this is the most usual approach. However, the algorithm can be easily reformulated in terms of M_{aa}, B_{aa} and K_{aa} (a meaning the degrees of freedom selected in ASET) instead of the corresponding modal matrices. Note however that the eigenvalues and eigenvectors of the corresponding system matrix A_{ee} must be calculated explicitly by using the CEAD module. Once Λ_{ee} and T_{ee} are calculated in this way, the rest

of the procedure is the same.

- *Non diagonal matrices.* The user might be interested in including damping or stiffness matrices via DMIG cards. In this case, the mass, stiffness and/or damping matrices might not be diagonal. The consequences of this fact are twofold: first, the eigenvalues and eigenvectors of the resulting system matrix A_{ee} must be calculated explicitly, and second, the eigenvalues/vectors will not generally be arranged in the same order than that of the natural frequencies and modes of the structural model. Since the structural modal responses are ordered columnwise according to the structural modes, the user must be careful in ensuring that the order of system matrix eigenvectors is consistent.

8 Conclusions

An efficient method for computation *exact* RMS values of any output (displacement, acceleration, stress, internal force, constraint force and so on) of an structure submitted to random loads. The method allows input random PSD spectrum of any shape, and the number of independent excitations (correlated or not) is unlimited. The method can be easily appended to the MSC/NASTRAN Frequency Response solution, by means of a DMAP ALTER. This ALTER allows the calculation of RMS values of displacements, stresses and element forces. Any other conceivable output variable, that can be expressed as linear function of displacements and velocities, may also be calculated requiring a minor modification of the DMAP Alter. DMAP results have been compared to those provided by MSC/NASTRAN standard method, and have been proved to be the correct in all cases.

Main advantages of the described method are:

- The procedure does not require the calculation of the frequency response function, obtention of the PSD of each output, and piecewise numerical integration of the PSD curves (this is the standard procedure that Finite Element Solvers follow for the obtention of RMS results). Thus the procedure is by far much more efficient than the standard one, given that the number of mathematical operations needed is inferior in orders of magnitude.
- Owing to this fact, there is no limitation on the number of structural response variables requested by the user, as it happens in fact with the standard procedure when managing large size models, or when an high number of frequencies are specified to obtain accurate results (usually problems of disk space appears, as was shown in section 6).
- The results are requested in the input file, using simply the CASE CONTROL command STRESS = ALL, n, and not by means of the XYPRINT, XYPEAK, etc. commands.
- The DMAP allows the postprocessing of RMS results exactly in the same manner as if they were obtained from a conventional static analysis. Therefore contour plots of the RMS results over the structure can be obtained, easing the analyst work.
- Extensions of the capabilities of this procedure to more general problems are easy and require only minor modifications.

9 References

[1]. H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*, John Wiley and Sons, Inc., New York, 1972.

[2] J.L. Junkins and Y. Kim. *Introduction to the Dynamics and Control of Flexible Structures.* AIAA, 1992.

[3] Anonymous, MATLAB *Control Systems Toolbox*, User's Guide. The Mathworks Inc.

[4] E. Fuente and J. San Millán. "Calculation within MSC/NASTRAN of the forces transmitted by the Multipoint Constraints (MPC) and the Forces Generated in Suport Constraints". MSC World Users Conference, Newport Beach (CA), June 5th, 1996.

[5] E. Fuente and J. San Millán. "An efficient and exact solution for Random Vibration Analysis using MSC/NASTRAN. Part I: White noise spectrum". MSC Aerospace Users Conference, Newport Beach (CA), Nov 17-20th, 1997.

[6] G.J. Balas, J.C. Doyle et al. *μ-Analysis and Synthesis TOOLBOX for use with MATLAB*, The Mathworks Inc., 1993

[7] A.V. Oppenheim, R.W. Schaffer. *Digital Signal Processing*, Prentice Hall, New Jersey, 1975 (pp. 513)