CE790
Directed Research

A Report on Nonstationary Response of Nonlinear MDOF Systems using Equivalent Linearization and Compact Analytical Model of the Excitation Process

by Baris Erkus [1]
supervised by Prof. Dr. Sami F. Masri

27 October 2003

[1]Ph.D. Student, Civil Engineering Dept., University of Southern California, erkus@usc.edu

Contents

Abstract

This report is a fruit of the effort to learn the fundamental concepts of random processes and random vibrations theory under the course "CE790, Directed Research". The boundaries, however, extended to more advanced topics such as equivalent linearization and nonstationary responses. The goal was to review the paper by Smyth and Masri on nonstationary analysis of nonlinear structures and implement one of the examples given in that paper using Mathematica. The main challenge was not the main theory as may be expected. Indeed, the paper is very clear to understand and the theory is easy to follow. The challenge was the implementation in Mathematica. The inefficiency of Mathematica in writing highly numerical codes was extremely restricted the author from further progress. The preparation of this report took more time than expected, however, the author has gained a lot of knowledge on the topic under the supervision of Prof. Dr. Sami F. Masri for a stand-alone research that will be carried out later.

This report consists of two parts: The first part is a brief review of theory of random processes. The explanations are straightforward and very helpful for a review. The second section is the review of the theory explained in the aforementioned paper. Although it looks similar to the original paper, some parts uses different nomenclature and, details of some of the sections are added. In addition to the original theory, a simplification of the method proposed is given. This simplified approach is easier to implement than the original approach. Finally a Mathematica code is given as an example of the application of the method.

The only benefit of Mathematica was the use of built-in approximation function, namely "`Fit`". Other than that, the differential equation solver of Mathematica was not able to solve the equations obtained and, therefore, a Runge-Kutta solver is coded. At the time of preparation of this document a new version of Mathematica (Version 5.0.0) is released. This version is capable of solving multi single degree differential equations using Runge-Kutta method. However, it may be useful to use other programming languages such as Matlab for the numerical computation to avoid the other problems of Mathematica, which were explained in this report.

The author wish to thank to Prof. Masri, for his supervision. The work was extremely helpful to the student to understand the concepts that appear very frequently in the field, and it will definitely be extended in the future.

Chapter 1

A Review of Random Processes

1.1 Introduction

This chapter reviews the basic concepts of probability theory and random processes to establish a foundation for the stochastic analysis of linear and nonlinear systems. In particular, response statistics of a system under a stationary excitation is investigated. Roberts and Spanos (1990) is used as the main reference. Definitions given herein are considerably brief and, interested reader is directed to textbooks on these topics such as Papoulis (1984) for more explanations.

1.2 Probability

Theory of probability is characterized with three fundamental axioms.

Axiom 1.2.1. *Let A be a random event and $P(A)$ be the probability that event A occurs. Then,*

$$0 \leqslant P(A) \leqslant 1 \tag{1.1}$$

Axiom 1.2.2. *Let S be the event that any event in the sample space occurs, that is certain event. Then,*

$$P(S) = 1 \tag{1.2}$$

Axiom 1.2.3. *Let A and B are mutually exclusive. Then*

$$P(A \cup B) = P(A) + P(B) \text{ and } P(A \cap B) = \varnothing \tag{1.3}$$

If A and B are not mutually exclusive. Then

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{1.4}$$

Probability of an event may be defined as a mapping of the possible events of sample space into $[0, 1]$.

1.3 Random Variables

For a complete set of exclusive events, each event can be associated with a number, say η, which is called as random variable. In some cases η takes only a finite number of distinct values

if underlying events are finite. It may be also continuous and may take any number between $-\infty$ and $+\infty$ if underlying events are infinite.

In most of the problems, use of more than one random variable associated with different sample spaces is required. These random variables are formulated as a vector, $\boldsymbol{\eta} = [\eta_1, \eta_2, \cdots, \eta_n]$.

Definition 1.3.1. *Joint distribution function of a random vector* $\boldsymbol{\eta}$ *is defined as*

$$F_{\boldsymbol{\eta}}(\boldsymbol{x}) = P_{\boldsymbol{\eta}}(\boldsymbol{\eta} \leqslant \boldsymbol{x}). \tag{1.5}$$

Properties of joint distribution function can be summarized as follows: $F_{\boldsymbol{\eta}}(\boldsymbol{x}) \to 0$ as any element of $\boldsymbol{x}$ approaches zero and $F_{\boldsymbol{\eta}}(\boldsymbol{x}) \to 1$ as all elements of $\boldsymbol{x}$ approach to $+\infty$. Also, $F_{\boldsymbol{\eta}}(\boldsymbol{x})$ increases monotonically in all variables.

Definition 1.3.2. *Joint density function of a random vector* $\boldsymbol{\eta}$ *is defined as*

$$f_{\boldsymbol{\eta}}(\boldsymbol{x}) = f(x_1, x_2, \cdots, x_n) = \frac{\partial^n F_{\boldsymbol{\eta}}(x_1, x_2, \cdots, x_n)}{\partial x_1 \partial x_2 \cdots \partial x_n} \tag{1.6}$$

Due to the probability axioms, $f_{\boldsymbol{\eta}}(\boldsymbol{x}) \geqslant 0$. Also, the probability that the random vector $\boldsymbol{\eta}$ falls in some region R is equal to

$$P_{\boldsymbol{\eta}}(\boldsymbol{\eta} \in R) = \int_R f_{\boldsymbol{\eta}}(\boldsymbol{x}) d\boldsymbol{x}. \tag{1.7}$$

If the individual random variables are independent, i.e.

$$P_{\boldsymbol{\eta}}(\boldsymbol{\eta} \geqslant \boldsymbol{x}) = P_{\boldsymbol{\eta}}(\eta_1 \geqslant x_1) P_{\boldsymbol{\eta}}(\eta_2 \geqslant x_2) \cdots P_{\boldsymbol{\eta}}(\eta_n \geqslant x_n), \tag{1.8}$$

then

$$F_{\boldsymbol{\eta}}(\boldsymbol{x}) = F_1(x_1) F_2(x_2) \cdots F_n(x_n) \tag{1.9}$$

$$f_{\boldsymbol{\eta}}(\boldsymbol{x}) = f_1(x_1) f_2(x_2) \cdots f_n(x_n) \tag{1.10}$$

where $F_k(x_k)$ and $f_k(x_k)$ are distribution and density functions of variable η_k, respectively.

Consider two random variables, η and ρ related through the equation

$$\rho = g(\eta). \tag{1.11}$$

Then, the density function $f_\rho(y)$ of ρ is related with the density function $f_\eta(x)$ of η as follows:

$$f_\rho(y) = \sum_{i=1}^{m} f_\eta(x_i) \left| \frac{d}{dx} g(x_i) \right| \tag{1.12}$$

where x_i, $(i = 1, 2, \cdots, m)$ are the roots of $y = g(x)$.

Equation (1.11) can also be used for random vectors. If two random vectors are related with the matrix transformation given by

$$\boldsymbol{\rho} = \boldsymbol{g}(\boldsymbol{\eta}), \tag{1.13}$$

then the relation between two density functions is given as follows:

$$f_{\boldsymbol{\rho}}(\boldsymbol{y}) = f_{\boldsymbol{\eta}}(\boldsymbol{x}) |\boldsymbol{J}|_{\boldsymbol{x}=\boldsymbol{g}(\boldsymbol{y})}. \tag{1.14}$$

where,

$$\boldsymbol{J} = \begin{vmatrix} \frac{\partial g_1}{\partial y_1} & \frac{\partial g_1}{\partial y_2} & \cdots & \frac{\partial g_1}{\partial y_n} \\ \vdots & & & \\ \frac{\partial g_n}{\partial y_1} & \frac{\partial g_n}{\partial y_2} & \cdots & \frac{\partial g_n}{\partial y_n} \end{vmatrix}. \tag{1.15}$$

Definition 1.3.3. *Lets* $\boldsymbol{\eta}$ *be a random vector and* $\boldsymbol{g}(\boldsymbol{\eta})$ *is a matrix function of* $\boldsymbol{\eta}$*. Expectation of* $\boldsymbol{g}(\boldsymbol{\eta})$ *is defined by*

$$E[\boldsymbol{g}(\boldsymbol{\eta})] = \int_{\boldsymbol{x}=-\infty}^{\boldsymbol{x}=+\infty} \boldsymbol{g}(\boldsymbol{\eta}) f_{\boldsymbol{\eta}}(\boldsymbol{x}) d\boldsymbol{x} \tag{1.16}$$

Definition 1.3.4. *Covariance matrix of* $\boldsymbol{\eta}$ *is defined as*

$$\boldsymbol{K} = E[(\boldsymbol{\eta} - \boldsymbol{m})(\boldsymbol{\eta} - \boldsymbol{m})^T] \tag{1.17}$$

where $\boldsymbol{m} = E[\boldsymbol{\eta}]$*. In this definition* $\boldsymbol{g}(\boldsymbol{\eta}) = (\boldsymbol{\eta} - \boldsymbol{m})(\boldsymbol{\eta} - \boldsymbol{m})^T$*.*

Diagonal terms of $\boldsymbol{K}$ are called as variances of each of the individual random variable:

$$var(\eta_i) = \sigma_i^2 = E[(\eta_i - m_i)(\eta_i - m_i)] = E[(\eta_i - m_i)^2] \tag{1.18}$$

Definition 1.3.5. *Correlation (or autocorrelation) matrix of* $\boldsymbol{\eta}$ *is defined as*

$$\boldsymbol{R} = E[\boldsymbol{\eta}\boldsymbol{\eta}^T] \tag{1.19}$$

In this definition $\boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\eta}\boldsymbol{\eta}^T$*.*

Detailed treatment of covariance and correlation matrices can be found in Stark and Woods (2002), pages 251-258.

Definition 1.3.6. *Characteristic function is defined as*

$$\Theta_{\boldsymbol{\eta}}(\boldsymbol{\omega}) = E[e^{i\boldsymbol{\omega}^T\boldsymbol{\eta}}] = \int_{\boldsymbol{x}=-\infty}^{\boldsymbol{x}=+\infty} e^{i\boldsymbol{\omega}^T\boldsymbol{\eta}} f_{\boldsymbol{\eta}}(\boldsymbol{x}) d\boldsymbol{x}. \tag{1.20}$$

In this definition $\boldsymbol{g}(\boldsymbol{\eta}) = g(\boldsymbol{\eta}) = e^{i\boldsymbol{\omega}^T\boldsymbol{\eta}}$

Clearly characteristic function is multi dimensional Fourier transform of $f_{\boldsymbol{\eta}}(\boldsymbol{x})$[1].

1.4 The Gaussian Distribution

The Gaussian probability distribution of a random vector $\boldsymbol{\eta}$ is given by

$$f_{\boldsymbol{\eta}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{K}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^T\boldsymbol{K}^{-1}(\boldsymbol{x}-\boldsymbol{m})}. \tag{1.21}$$

The characteristic function is given by

$$\Theta_{\boldsymbol{\eta}}(\boldsymbol{\omega}) = e^{i\boldsymbol{m}^T\omega - \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{K}\boldsymbol{\omega}}. \tag{1.22}$$

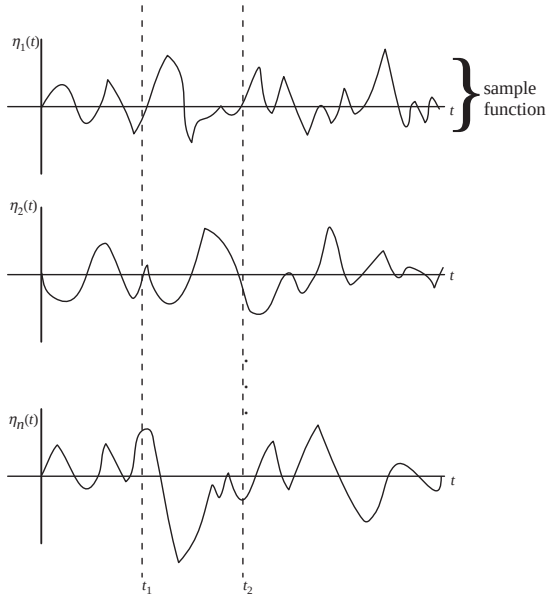


Figure 1.1: Ensemble of Random Process

1.5 Random Processes

In the previous section, a random variable is defined as a number that is associated with an event. In most of the cases, outcome of an event is a function of several variables. This function is called a realization or a sample function. The collection of all possible outcomes is called the ensemble of the random process. An example of a random process with the n-sample functions of a single variable is given by figure 1.1, which is the case for most of the engineering problems.

Now, lets take the points from each sample function for a fixed time, $t = t_1$. The collection of these n points is actually the set of possible outcomes of a random variable as each outcome is associated with the value of the function at this particular time i.e. $\eta = \eta(t_1)$. As one can observe, for the random process given by figure 1.1, there are infinite number of random variables which forms a random vector with an infinite dimension. Hence, all the definitions given in the previous sections for random variables are applicable for the random process given by figure 1.1. For example, random variable that is obtained by fixing the time to t_1 has a probability distribution. Also the random variable at time t_2 has a probability distribution. There is also a joint probability distribution of these two random variables, which results the 2-by-2 covariance and autocorrelation matrices.

In this report, the notation used for a random vectors will be also used for the random processes, $\boldsymbol{\eta}$. However, we should keep in mind that random vector elements are finite and random process elements are infinite. Therefore while we talk about the i^{th} element of a random vector, we mention the random variable at time t for random processes and denote it as $\boldsymbol{\eta}(t)$.

[1]Definition of the Fourier Transform is not given here as it will be defined later in this report

1.5.1 Stationary Random Processes

Different from a random vector, a random process has an infinite number of probability and distribution functions, which makes the interpretation of the random processes difficult as one cannot deal with infinite number of distinct probability definitions. A good practice is to use some assumptions to reduce such a complex problem into a simpler one. For example, one may assume that random variables at any times t_i and t_j have the same probability definitions. Furthermore assume that joint probability distribution of the two random variables at any times t_i and t_j are same as the joint probability of the two random variables at any times t_k and t_l, if $t_i - t_j = t_k - t_l$. A random process, where above assumptions hold, is known as stationary random process of order two. It follows that for a second order stationary random process any two random variables at time t_i, t_j, t_k and t_l have the same mean and the correlation between t_i and t_j is same as the correlation between t_k and t_l.

The question at this point is that, is a random process stationary if random variables at time t_i, t_j, t_k and t_l have the same mean and the correlation between the random variables at times t_i and t_j is same as the correlation between the random variables at times t_k and t_l when $t_i - t_j = t_k - t_l$? The answer is no; as for such a process, probability distributions of the random variables at t_i, t_j, t_k and t_l and joint probabilities defined above may be different. The latter random process is known as wide sense stationary (WSS), while the former one is known as strict sense stationary (SSS). Here more formal definitions of these terms are given.

Definition 1.5.1. *A random process is called strict-sense stationary (SSS) if its spectral properties are invariant to a time shift, i.e. the two random processes $\boldsymbol{\eta}(t)$ and $\boldsymbol{\eta}(t+c)$ have the same statistics for any c.*

First-order statistics of SSS processes are independent of time and can be abbreviated as $f(\boldsymbol{\eta}; t) = f(\boldsymbol{\eta})$. Similarly, joint probability function $f(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2; t_1 + c, t_2 + c)$ is independent of c and is abbreviated as

$$f(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2; t_1, t_2) = f(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2; \tau), \ \tau = t_1 - t_2. \tag{1.23}$$

As natural consequences of these properties, for a SSS process, mean of the random variable is constant for all time, t and abbreviated as $E[\boldsymbol{\eta}(t)] = \boldsymbol{m}(t) = m$. Correlation and covariance of the two random variables $\boldsymbol{\eta}(t_1)$ and $\boldsymbol{\eta}(t_2)$ depends only $\tau = t_1 - t_2$ and abbreviated as $R(\tau)$ and $K(\tau)$. The relation between these parameters is

$$R(\tau) = K(\tau) + m^2 \tag{1.24}$$

Definition 1.5.2. *A random process is called wide-sense stationary (WSS) if its mean is constant*

$$E[\boldsymbol{\eta}(t)] = \boldsymbol{m}(t) = m \tag{1.25}$$

and the correlation between the random variables at times t_1 and t_2 depends only only $\tau = t_1 - t_2$ i.e.

$$E[\boldsymbol{\eta}(t_1)\boldsymbol{\eta}^T(t_2)] = R(t_1 - t_2) = R(\tau). \tag{1.26}$$

These assumptions are not valid for real-life events, yet they form a sound background for the analysis of more complex problems.

Definition 1.5.3. *The temporal averages of a sample function is defined as*

$$< \boldsymbol{\eta}(t) >= \lim_{T\to\infty} \frac{1}{T} \int_0^T \boldsymbol{\eta}(t)dt, \tag{1.27}$$

$$< \boldsymbol{\eta}(t)\boldsymbol{\eta}(t+\tau) >= \lim_{T\to\infty} \frac{1}{T-\tau} \int_0^{T-\tau} \boldsymbol{\eta}(t)\boldsymbol{\eta}(t+\tau)dt. \tag{1.28}$$

It should be noted that the temporal averages of a sample function is different then the mean value and correlation of SSS and WSS random processes. The mean value and correlation are computed using whole set of sample functions and known as ensemble averages. A special case occur when the temporal averages of all sample functions and the ensemble averages of a stationary random process are equal.

Definition 1.5.4. *A stationary random process is called ergodic in the mean if*

$$< \boldsymbol{\eta}(t) >= E[\boldsymbol{\eta}(t)] = m, \text{ for all } i = 1, \cdots, n. \tag{1.29}$$

Definition 1.5.5. *A stationary random process is called ergodic in the correlation if*

$$< \boldsymbol{\eta}(t)\boldsymbol{\eta}(t+\tau) >= E[\boldsymbol{\eta}(t)\boldsymbol{\eta}(t+\tau)] = R(\tau), \text{ for } i = 1, \cdots, n. \tag{1.30}$$

1.5.2 Cross-Correlation Function

Definition 1.5.6. *Cross-correlation function of two random processes is defined as*

$$R_{\boldsymbol{\eta}\boldsymbol{\rho}}(t_1, t_2) = E[\boldsymbol{\eta}(t_1)\boldsymbol{\rho}(t_2)]. \tag{1.31}$$

If the processes $\boldsymbol{\eta}(t)$ and $\boldsymbol{\rho}(t)$ are stationary, the cross-correlation function depends only on τ i.e.,

$$R_{\boldsymbol{\eta}\boldsymbol{\rho}}(\tau) = E[\boldsymbol{\eta}(t+\tau)\boldsymbol{\rho}(t)]. \tag{1.32}$$

It should be noted that the the order of the subscripts determine the definition of cross-correlation. For example, the cross-correlation

$$R_{\boldsymbol{\rho}\boldsymbol{\eta}}(\tau) = E[\boldsymbol{\rho}(t+\tau)\boldsymbol{\eta}(t)] \tag{1.33}$$

is different than $R_{\boldsymbol{\eta}\boldsymbol{\rho}}(\tau)$. In fact, for stationary random processes $R_{\boldsymbol{\eta}\boldsymbol{\rho}}(\tau) = R_{\boldsymbol{\rho}\boldsymbol{\eta}}(-\tau)$ (Wirsching et al. (1995))

1.5.3 Derivatives of Stationary Processes

The cross-correlation between a stationary random process and its derivative is of special interest. consider the derivative of correlation function,

$$\begin{aligned} \frac{d}{d\tau}R_{\boldsymbol{\eta}}(\tau) &= \frac{d}{d\tau}E[\eta(t+\tau)\boldsymbol{\eta}(t)] \\ &= E[\frac{d}{d\tau}\boldsymbol{\eta}(t+\tau)\boldsymbol{\eta}(t)] \\ &= E[\dot{\boldsymbol{\eta}}(t+\tau)\boldsymbol{\eta}(t)] \\ &= R_{\dot{\boldsymbol{\eta}}\boldsymbol{\eta}}(\tau). \end{aligned} \tag{1.34}$$

Therefore the first derivative of correlation function is the cross-correlation function between $\boldsymbol{\eta}(t)$ and $\dot{\boldsymbol{\eta}}(t)$. Correlation of the derivative of the stationary random process is found as

$$\begin{aligned}\frac{d^2}{d\tau^2}R_{\boldsymbol{\eta}}(\tau) &= \frac{d}{d\tau}E[\dot{\boldsymbol{\eta}}(t+\tau)\boldsymbol{\eta}(t)] \\ &= \frac{d}{d\tau}E[\dot{\boldsymbol{\eta}}(t)\boldsymbol{\eta}(t-\tau)] \\ &= E[\dot{\boldsymbol{\eta}}(t)(-\dot{\boldsymbol{\eta}}(t-\tau))] \\ &= -R_{\dot{\boldsymbol{\eta}}}(\tau). \end{aligned} \tag{1.35}$$

1.5.4 Spectral properties of Stationary Random Processes

The correlation function of a stationary random process is a time-domain function. It is useful to define this function in terms of some eigenfunctions such as sines and cosines. For this purpose Fourier transformation is used. Here, Fourier transformation that will be used throughout this paper is given first as there are different definitions of Fourier transformation in various literature.

Definition 1.5.7. *The Fourier transform of a function $f(t)$ is defined as*

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt. \tag{1.36}$$

Similarly the inverse Fourier transform is defined as

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega. \tag{1.37}$$

Definition 1.5.8. *The power spectral density (psd) function of a stationary random process is the Fourier transform of correlation function,*

$$S(\omega) = \int_{-\infty}^{\infty} R(\tau)e^{-i\omega\tau}d\tau. \tag{1.38}$$

The correlation function is found taking the inverse Fourier transform of the psd function,

$$R(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)e^{i\omega\tau}d\omega. \tag{1.39}$$

If the random process is real, $R(\tau)$ is real and even. Similarly $S(\omega)$ is real and even. In this case, the Fourier transform pair reduces to

$$S(\omega) = \int_{-\infty}^{\infty} R(\tau)cos(\omega\tau)d\tau = 2\int_{0}^{\infty} R(\tau)cos(\omega\tau)d\tau \tag{1.40}$$

$$R(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)cos(\omega\tau)d\omega = \frac{1}{\pi}\int_{0}^{\infty} S(\omega)cos(\omega\tau)d\omega. \tag{1.41}$$

Equations (1.38), (1.39), (1.40) and (1.41) are known as Wiener-Khinchine relations.

It is observed that given the psd, mean square value of a stationary random process can be found as

$$R(0) = \frac{1}{\pi}\int_{0}^{\infty} S(\omega)d\omega. \tag{1.42}$$

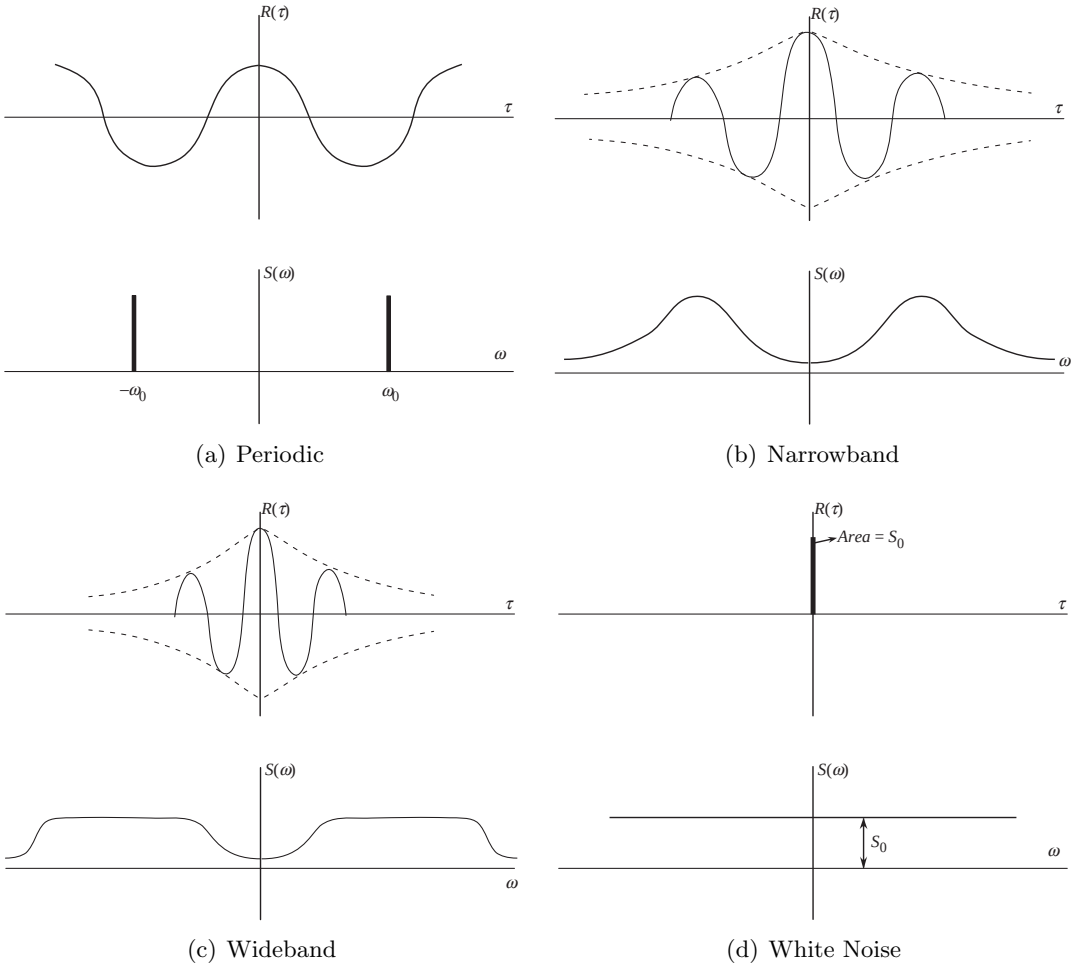


Figure 1.2: Classification of the processes according to the frequency content

Definition 1.5.9. *The cross-spectral density function of two random processes is defined as*

$$S_{\boldsymbol{\eta\rho}}(\omega) = \int_{-\infty}^{\infty} R_{\boldsymbol{\eta\rho}}(\tau) e^{-i\omega\tau} d\tau. \tag{1.43}$$

The cross-correlation function is found taking the inverse Fourier transform of the cross-spectral function,

$$R_{\boldsymbol{\eta\rho}}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{\boldsymbol{\eta\rho}}(\omega) e^{i\omega\tau} d\omega. \tag{1.44}$$

Cross-spectral density function is generally complex even when both processes are real. For all cases, following relations hold:

$$S_{\boldsymbol{\eta\rho}}(\omega) = S_{\boldsymbol{\rho\eta}}(\omega) \tag{1.45}$$

$$R_{\boldsymbol{\eta\rho}}(\tau) = R_{\boldsymbol{\rho\eta}}(\tau). \tag{1.46}$$

1.5.5 Narrowband, Wideband and White Noise Processes

Power spectral density provides information about the frequency content of correlation. Four types of psd function is of interest. If the correlation is a periodic function which can be represented by one frequency, psd is a delta function as shown in figure 1.2(a). A process is called narrowband if its psd has significant values over a narrow frequency range around a central frequency as shown in figure 1.2(b). A process is called as wideband if its psd has significant values over a wide range of frequencies as shown in figure 1.2(c). A common idealization of wideband processes is to assume $S(\omega) = S_0$ as shown in figure 1.2(d). In this case, the correlation is a delta function implying that nor correlation for $\tau \neq 0$. The correlation is computed as

$$\begin{aligned} R(\tau) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S_0 e^{i\omega\tau} d\omega \\ &= \frac{1}{2\pi} S_0 \int_{-\infty}^{\infty} e^{i\omega\tau} d\omega \\ &= \frac{1}{2\pi} S_0 (2\pi\delta(\tau)) \\ &= S_0\delta(\tau). \end{aligned} \tag{1.47}$$

1.6 Stochastic Response of Linear Systems

It is known that the response of a linear system, $x(t)$ to an arbitrary excitation, $f(t)$ can be found through convolution

$$\boldsymbol{x}(t) = \int_0^t f(\lambda) g(t-\lambda) d\lambda \tag{1.48}$$

where $g(t)$ is the impulse response. It can be also shown that (Meirovitch (1986)) equation (1.48) can be written as

$$\boldsymbol{x}(t) = \int_{-\infty}^{\infty} f(\lambda) g(t-\lambda) d\lambda = \int_{-\infty}^{\infty} f(t-\lambda) g(\lambda) d\lambda. \tag{1.49}$$

It is also known that that the input-output relation in the frequency domain is given by

$$X(\omega) = G(\omega) F(\omega) \tag{1.50}$$

where $X(\omega)$, $G(\omega)$ and $F(\omega)$ are the Fourier transform of input, impulse response function and the excitation, respectively.

Now lets assume that the excitation is a stationary random process with known statistics. It is known that the response is also a stationary random process. Lets the mean value of the excitation be m_f. The mean of the response is

$$m_x = m_f \int_{-\infty}^{\infty} g(\lambda) d\lambda = m_f G(0). \tag{1.51}$$

The response autocorrelation function can be derived using the definition given by equation 1.26 as follows:

$$R_x(\tau) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\lambda_1) g(\lambda_2) R_f(\tau + \lambda_1 - \lambda_2) d\lambda_1 d\lambda_2 \tag{1.52}$$

where R_f is the correlation of the excitation. Power spectral density of the response is the Fourier transform of the response correlation. A useful relation between the psds of input and output is given by

$$S_x(\omega) = |G(\omega)|^2 S_f(\omega). \tag{1.53}$$

Finally the mean square response can be obtained setting $\tau = 0$.

$$R_x(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(\omega)|^2 S_f(\omega). \tag{1.54}$$

The detailed derivations of these relations can be found in Meirovitch (1986) and Papoulis (1984).

Chapter 2

Nonstationary Response of Nonlinear MDOF Systems

2.1 Introduction

In this chapter, the method proposed by Smyth and Masri (2002) for the stochastic analysis of a class of nonlinear multi-degree-of-freedom systems subjected to nonstationary random excitation is reviewed. The covariance kernel of the nonstationary excitation is decomposed into its eigenvalues and eigenvectors, which is known as Karhunen-Loeve (K-L) decomposition. Then, covariance eigenvectors are approximated by orthogonal Chebyshev polynomials, which results a more compact form of the covariance matrix. The final excitation covariance is used in the equivalent linearization analysis of the nonlinear structure. A Duffing oscillator is considered as an example. As the nonstationary excitation covariance, the example given by Traina et al. (1986) is used.

In this study, most of the numerical computations are carried out using the software tool Mathematica. Therefore, throughout this report details of the Mathematica code used will be given when necessary.

2.2 A Compact form of Nonstationary Excitation Covariance

In this section, a compact form of a nonstationary excitation covariance matrix is derived. First, the covariance matrix is decomposed into its eigenvalues and eigenvectors. Then, eigenvectors are represented in terms of orthogonal Chebyshev polynomials using least-squares method. Interested reader is referred to the work by Traina et al. (1986) for a detailed treatment of this decomposition.

2.2.1 Eigenvector Expansion of Covariance Matrix

Assume that the covariance matrix of a random excitation is given by an N by N matrix, $\boldsymbol{\mathcal{C}}$. Let $\lambda_1 \leq \cdots \leq \lambda_N$ and $\mathbf{p}_1, \ldots, \mathbf{p}_N$ be its eigenvalues and corresponding eigenvectors, respectively. Using K-L expansion [1], covariance matrix $\boldsymbol{\mathcal{C}}$ can be approximated in terms of its first k eigenvalues

[1] For continuous covariance functions, K-L is based on eigenvalues and eigenfunctions. See e.g. Stark and Woods (2002) for a review of K-L expansion.

and eigenvectors as

$$\boldsymbol{\mathcal{C}}(t_1, t_2) \approx \boldsymbol{\mathcal{C}}_k(t_1, t_2) = \sum_{i=1}^{k} \lambda_i \mathbf{p}_i(t_1) \mathbf{p}_i^{\mathrm{T}}(t_2) \tag{2.1}$$

where $\boldsymbol{\mathcal{C}}_k$ represents the approximated covariance matrix. It is shown by Traina et al. (1986) that magnitude of the error introduced by this approximation is bounded and can be reduced to a relatively small amount by increasing k.

2.2.2 Orthogonal Chebyshev Polynomials Representation of Covariance Matrix

The eigenvectors of covariance matrix can be approximated by orthogonal polynomials as

$$\mathbf{p}_i(t) \approx \sum_{j=0}^{m_i-1} H_{ij} T_j(s), \tag{2.2}$$

where

$$s = \frac{2t}{t_{\max}} - 1, \tag{2.3}$$

T_i's are orthogonal Chebyshev polynomials of first kind, H_{ij}'s are polynomial coefficients and m_i is the number of the polynomials used in the series fit of $\mathbf{p}_i$. Chebyshev polynomials of first kind are given as

$$T_0 = 1, \quad T_1(s) = s, \quad T_2(s) = 2s^2 - 1, \quad T_3(s) = 4s^3 - 3s, \quad \ldots \tag{2.4}$$

Chebyshev polynomials can also be computed using

$$T_j(s) = \cos(j \arccos(s)), \quad s = 0, 1, 2, \ldots. \tag{2.5}$$

It is clear from equation (2.5) that each of the Chebyshev polynomials are defined for $0 \leq s \leq 1$, which is different than the domain of the eigenvectors, i.e. t. Therefore, to compute the approximated value, first the $t-$domain should be converted to $s-$domain using equation (2.3). Consider, for example, figure (2.1). The approximated value of the point A is computed using s_{A}.

The polynomial coefficients, H_{ij} can be computed using least-squares method (see e.g. Traina et al., 1986).

The above procedure reduces the covariance kernel from a numerical matrix form into a polynomial matrix function form, which is very suitable for mathematical manipulation. The final form of the covariance is given by

$$\boldsymbol{\mathcal{C}}_k(s_1, s_2) = \sum_{i=1}^{k} \lambda_i \sum_{j=0}^{m_i-1} \sum_{l=0}^{m_i-1} H_{ij} H_{il} T_j(s_1) T_l(s_2). \tag{2.6}$$

It should be noted that the method given above approximates a numerical matrix as a function, and hence is a numerical method. Therefore, accuracy is very sensitive to the number of the eigenvectors, number of the Chebyshev polynomials used and the method employed to compute the polynomial coefficients. The software Mathematica has a built-in function, "`Fit`", that can be used to compute the coefficients of the polynomial fit. This function computes a leastsquares fit to a vector of data as a linear combination of a set of user defined functions. In our problem, these functions are the Chebyshev polynomials given by (2.5). Mathematica also has a built-in function called "`ChebyshevT`" to compute the Chebyshev polynomials. (See the Mathematica source code given in this report for an implementation of these functions)

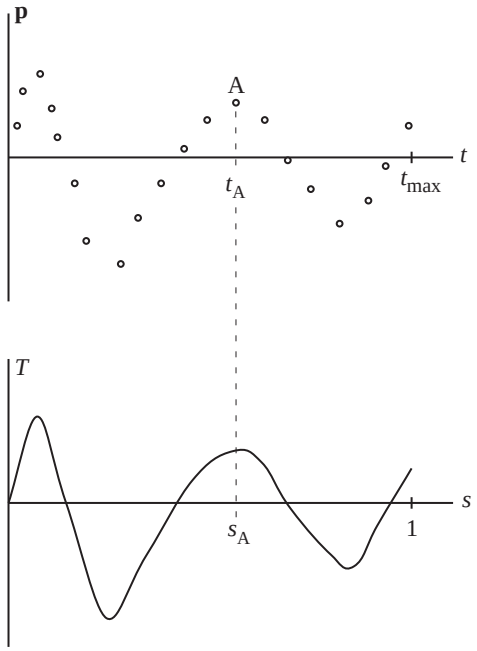


Figure 2.1: Eigenvector and its polynomial approximation

2.3 Statical Linearization Method for Nonstationary Excitation

In the previous section, the nonstationary excitation is reduced to a form suitable for the stochastic analysis. Analysis of linear structures excited by a nonstationary excitation is straightforward. However, nonlinear structures cannot be immediately analyzed. In the paper mentioned above, the authors use a linearization method to obtain a locally linear model of the nonlinear structure. As mentioned in the paper, this part of the work is mainly adapted from Roberts and Spanos (1990). Interested reader is referred to this book and other related sources for detailed information on the linearization technique used. In this section, a brief review of the linearization technique is given.

Consider an $n-$degree-of-freedom-system defined by

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} + \mathbf{g}(\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{F} \tag{2.7}$$

where, **M, C, K** are $n \times n$ mass, damping and stiffness matrices, respectively, $\mathbf{x}$ is the $n \times 1$ displacement vector, $\mathbf{F}$ is the $n \times 1$ excitation vector and $\mathbf{g}$ is a $n \times 1$ nonlinearity function that has displacement and velocity as parameters. In this equation, displacement and the excitation vectors are random processes.

For the sake of simplicity, we wish to progress on zero-mean random processes as manipulation of zero-mean random processes are easier than non-zero-mean random processes. For this purpose, we will define zero-mean displacement and zero-mean excitation vectors as follows:

$$\mathbf{x}_0 = \mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}} \tag{2.8a}$$

$$\mathbf{F}_0 = \mathbf{F} - \boldsymbol{\mu}_{\mathbf{F}} \tag{2.8b}$$

where $\boldsymbol{\mu}_{\mathbf{x}}$ and $\boldsymbol{\mu}_{\mathbf{F}}$ are $n \times 1$ time-dependent mean of displacement and excitation. Substituting

these equations will yield

$$\mathbf{M}\ddot{\mathbf{x}}_0 + \mathbf{C}\dot{\mathbf{x}}_0 + \mathbf{K}\mathbf{x}_0 + \mathbf{M}\ddot{\boldsymbol{\mu}}_\mathbf{x} + \mathbf{C}\dot{\boldsymbol{\mu}}_\mathbf{x} + \mathbf{K}\boldsymbol{\mu}_\mathbf{x} + \mathbf{g}(\mathbf{x}_0 + \boldsymbol{\mu}_\mathbf{x}, \dot{\mathbf{x}}_0 + \dot{\boldsymbol{\mu}}_\mathbf{x}) = \mathbf{F}_0 + \boldsymbol{\mu}_\mathbf{F}. \tag{2.9}$$

If we take the expectation of both sides of the above equation,

$$\mathbf{M}\ddot{\boldsymbol{\mu}}_\mathbf{x} + \mathbf{C}\dot{\boldsymbol{\mu}}_\mathbf{x} + \mathbf{K}\boldsymbol{\mu}_\mathbf{x} + \mathbf{E}[\mathbf{g}(\mathbf{x}_0 + \boldsymbol{\mu}_\mathbf{x}, \dot{\mathbf{x}}_0 + \dot{\boldsymbol{\mu}}_\mathbf{x})] = \boldsymbol{\mu}_\mathbf{F}. \tag{2.10}$$

If we subtract equation (2.10) from equation (2.9), we obtain

$$\mathbf{M}\ddot{\mathbf{x}}_0 + \mathbf{C}\dot{\mathbf{x}}_0 + \mathbf{K}\mathbf{x}_0 + \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{F}_0 \tag{2.11}$$

where

$$\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{g}(\mathbf{x}_0 + \boldsymbol{\mu}_\mathbf{x}, \dot{\mathbf{x}}_0 + \dot{\boldsymbol{\mu}}_\mathbf{x}) - \mathbf{E}[\mathbf{g}(\mathbf{x}_0 + \boldsymbol{\mu}_\mathbf{x}, \dot{\mathbf{x}}_0 + \dot{\boldsymbol{\mu}}_\mathbf{x})]. \tag{2.12}$$

Now that the equation of motion is in terms of zero-mean response quantities. An equivalent linear system can be represented casting the nonlinear term $\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}})$ into equivalent time dependent stiffness and damping matrices as follows

$$\mathbf{M}\ddot{\mathbf{x}}_0 + (\mathbf{C} + \mathbf{C}_\mathrm{e})\dot{\mathbf{x}}_0 + (\mathbf{K} + \mathbf{K}_\mathrm{e})\mathbf{x}_0 = \mathbf{F}_0 \tag{2.13}$$

where $\mathbf{C}_\mathrm{e}$ and $\mathbf{K}_\mathrm{e}$ are $n \times n$ time dependent damping and stiffness matrices of the equivalent linear system. The error introduced by equation (2.13) is

$$\begin{aligned} \boldsymbol{\varepsilon} &= \mathbf{C}\dot{\mathbf{x}}_0 + \mathbf{K}\mathbf{x}_0 + \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}) - (\mathbf{C} + \mathbf{C}_\mathrm{e})\dot{\mathbf{x}}_0 - (\mathbf{K} + \mathbf{K}_\mathrm{e})\mathbf{x}_0 \\ &= \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}) - \mathbf{C}_\mathrm{e}\dot{\mathbf{x}}_0 - \mathbf{K}_\mathrm{e}\mathbf{x}_0. \end{aligned} \tag{2.14}$$

To determine $\mathbf{C}_e$ and $\mathbf{K}_e$, we need to minimize the error. Since the error is a vector, minimizing a norm of the error will be more convenient. A frequently used norm is Euclidean norm which is defined by $\|\boldsymbol{\varepsilon}\|_2 = (\boldsymbol{\varepsilon}^\mathrm{T}\boldsymbol{\varepsilon})^{1/2}$. We will minimize the expectation of the square Euclidean norm, i.e. $\mathbb{E}\left[\boldsymbol{\varepsilon}^\mathrm{T}\boldsymbol{\varepsilon}\right] = \mathbb{E}\left[\varepsilon_1^2 + \cdots + \varepsilon_n^2\right]$. This minimization problem can also be written as

$$\min_{c_{ij}, k_{ij}} \left(\sum_{l=1}^{n} D_l^2 \right) \qquad i, j, l = 1, 2, \ldots, n \tag{2.15}$$

where $D_i^2 = \mathbb{E}\left[\varepsilon_i^2\right]$ or

$$D_i^2 = \mathbb{E}\left[\left(h_i - \sum_{j=1}^{n} (c_{ij}^e \dot{x}_j + k_{ij}^e x_j) \right)^2 \right] \qquad i, j = 1, 2, \ldots, n \tag{2.16}$$

where h_i is the i^{th} element of $\mathbf{h}(\dot{\mathbf{x}}, \mathbf{x})$ and, c_{ij}^e and k_{ij}^e are the (i, j) elements of the $\mathbf{C}_e$ and $\mathbf{K}_e$ respectively. The necessary conditions for (2.15) to be true are

$$\frac{\partial}{\partial c_{ij}^e}(D_i^2) = 0 \qquad j = 1, 2, \ldots, n \tag{2.17a}$$

$$\frac{\partial}{\partial k_{ij}^e}(D_i^2) = 0 \qquad j = 1, 2, \ldots, n. \tag{2.17b}$$

If we substitute (2.16) into equations (2.17) and represent the results in matrix form, we obtain

$$\mathbb{E}\left[h_i\hat{\mathbf{x}}\right] = \mathbb{E}\left[\hat{\mathbf{x}}\hat{\mathbf{x}}^{\mathrm{T}}\right]\begin{bmatrix}\mathbf{c}_i^e\\ \mathbf{k}_i^e\end{bmatrix} \qquad j = 1, 2, \ldots, n \tag{2.18}$$

Using the property

$$\mathbb{E}\left[f(\boldsymbol{\eta})\boldsymbol{\eta}\right] = \mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^{\mathrm{T}}\right]\mathbb{E}\left[\boldsymbol{\nabla} f(\boldsymbol{\eta})\right] \tag{2.19}$$

where

$$\boldsymbol{\nabla} = \left[\frac{\partial}{\partial\eta_1}, \cdots, \frac{\partial}{\partial\eta_n}\right]^{\mathrm{T}}, \tag{2.20}$$

we can rewrite the equation (2.18) as

$$\mathbb{E}\left[\hat{\mathbf{x}}\hat{\mathbf{x}}^{\mathrm{T}}\right]\mathbb{E}\left[\begin{bmatrix}\dfrac{\partial h_i}{\partial\dot{\mathbf{x}}}\\ \dfrac{\partial h_i}{\partial\mathbf{x}}\end{bmatrix}\right] = \mathbb{E}\left[\hat{\mathbf{x}}\hat{\mathbf{x}}^{\mathrm{T}}\right]\begin{bmatrix}\mathbf{c}_i^e\\ \mathbf{k}_i^e\end{bmatrix} \qquad i = 1, 2, \ldots, n \tag{2.21}$$

which gives the final expressions for the $\mathbf{C}_e$ and $\mathbf{K}_e$ as

$$c_{ij}^e = \mathbb{E}\left[\frac{\partial h_i}{\partial\dot{x}_j}\right] \tag{2.22a}$$

$$k_{ij}^e = \mathbb{E}\left[\frac{\partial h_i}{\partial x_j}\right]. \tag{2.22b}$$

These relations are first introduced by Kazakov (1965). They are mainly for linearization purposes and can be used for other linearization problems that may appear in the fields other than random vibration analysis.

An important observation about equation (2.22) is the time-dependence of $\mathbf{C}_e$ and $\mathbf{K}_e$ for nonstationary excitation. Therefore, they are solved with the equations that define the system response.

2.4 Linearized Equations for Compact Form of the Excitation Covariance Kernel

In the previous section, we presented a general form of the solution of a linearized model of a nonlinear system to a nonstationary excitation. In this section, we will use the compact form of the excitation covariance to derive the equations for the response of the linearized system.

Lets define the equation of motion given by equation (2.13) in its state-space form as follows:

$$\dot{\mathbf{z}}(t) = \mathbf{G}(t)\mathbf{z}(t) + \mathbf{f}(t) \tag{2.23}$$

where

$$\begin{aligned}\mathbf{G}(t) &= \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{M}^{-1}(\mathbf{K}+\mathbf{K}_e(t)) & -\mathbf{M}^{-1}(\mathbf{C}+\mathbf{C}_e(t))\end{bmatrix},\\ \mathbf{f}(t) &= \begin{bmatrix}\mathbf{0}\\ -\mathbf{M}^{-1}\mathbf{F}_0(t)\end{bmatrix}\end{aligned} \tag{2.24}$$

and the states are $\mathbf{z}(t) = [\mathbf{x}_0^{\mathrm{T}}(t) \quad \dot{\mathbf{x}}_0^{\mathrm{T}}(t)]^{\mathrm{T}}$. The solution to the equation (2.23) can be expressed as

$$\mathbf{z}(t) = \mathbf{Y}(t)\mathbf{z}(0) + \mathbf{Y}(t)\int_0^t \mathbf{Y}^{-1}(\tau)\mathbf{f}(\tau)\mathrm{d}\tau \tag{2.25}$$

using standard linear system theory (see e.g. Reid, 1983). Also lets define the covariance of the states as $\mathbf{V}(t) = \mathbb{E}\left[\mathbf{z}(t)\mathbf{z}^{\mathrm{T}}(t)\right]$. If we substitute equation (2.25) into this equation, solution will be very difficult. Instead we will take the time-derivative of the covariance as

$$\dot{\mathbf{V}}(t) = \mathbb{E}\left[\dot{\mathbf{z}}(t)\mathbf{z}^{\mathrm{T}}(t)\right] + \mathbb{E}\left[\mathbf{z}(t)\dot{\mathbf{z}}^{\mathrm{T}}(t)\right] \tag{2.26}$$

and, if we substitute equation (2.23), it yields

$$\dot{\mathbf{V}}(t) = \mathbf{G}(t)\mathbf{V}(t) + \mathbf{V}(t)\mathbf{G}^{\mathrm{T}}(t) + \mathbf{U}(t) + \mathbf{U}^{\mathrm{T}}(t) \tag{2.27}$$

where

$$\begin{aligned}
\mathbf{U}(t) &= \mathbb{E}\left[\mathbf{z}(t)\mathbf{f}^{\mathrm{T}}(t)\right] \\
&= \mathbf{Y}(t)\mathbb{E}\left[\mathbf{z}(0)\mathbf{f}^{\mathrm{T}}(t)\right] + \mathbf{Y}(t)\int_0^t \mathbf{Y}^{-1}(\tau)\mathbb{E}\left[\mathbf{f}(\tau)\mathbf{f}^{\mathrm{T}}(t)\right]\mathrm{d}\tau &(2.28)\\
\mathbf{U}(t) &= \mathbf{Y}(t)\int_0^t \mathbf{Y}^{-1}(\tau)\mathbf{W}_{\mathbf{f}}(\tau,t)\mathrm{d}\tau &(2.29)
\end{aligned}$$

where $\mathbf{W}_{\mathbf{f}}(\tau,t) = \mathbb{E}\left[\mathbf{f}(\tau)\mathbf{f}^{\mathrm{T}}(t)\right]$ and we also take $\mathbb{E}\left[\mathbf{z}(0)\mathbf{f}^{\mathrm{T}}(t)\right] = 0$. Observing that

$$\begin{aligned}
\mathbb{E}\left[\mathbf{f}(\tau)\mathbf{f}^{\mathrm{T}}(t)\right] &= \mathbb{E}\left[\begin{bmatrix}\mathbf{0}\\ -\mathbf{M}^{-1}\mathbf{F}_0(\tau)\end{bmatrix}\begin{bmatrix}\mathbf{0} & -\mathbf{F}_0^{\mathrm{T}}(t)(\mathbf{M}^{-1})^{\mathrm{T}}\end{bmatrix}\right] &(2.30)\\
&= \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbb{E}\left[\mathbf{M}^{-1}\mathbf{F}_0(\tau)\mathbf{F}_0^{\mathrm{T}}(t)(\mathbf{M}^{-1})^{\mathrm{T}}\right]\end{bmatrix} &(2.31)\\
\mathbf{W}_{\mathbf{f}}(\tau,t) &= \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}^{-1}\mathbf{W}_{\mathbf{F}}(\tau,t)(\mathbf{M}^{-1})^{\mathrm{T}}\end{bmatrix} &(2.32)
\end{aligned}$$

where $\mathbf{W}_{\mathbf{F}}(\tau,t) = \mathbb{E}\left[\mathbf{F}_0(\tau)\mathbf{F}_0^{\mathrm{T}}(t)\right] = \mathbf{W}_{\mathbf{F}}(t,\tau)$. Now lets assume that the excitation force vector can be represented as $\mathbf{F} = \mathbf{e}s(t)$ where $\mathbf{e}$ is a constant vector and, the covariance of the single input process is

$$\boldsymbol{\mathcal{C}}_{ss}(t_1,t_2) = \mathbb{E}\left[(s(t_1) - \mu_s(t_1))((s(t_2) - \mu_s(t_2))\right]. \tag{2.33}$$

One can easily show that for this special case covariance of $\mathbf{f}$ becomes

$$\mathbf{W}_{\mathbf{f}}(t_1,t_2) = \mathbf{R}\boldsymbol{\mathcal{C}}_{ss}(t_1,t_2) \tag{2.34}$$

where

$$\mathbf{R} = \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}^{-1}(\mathbf{e}\mathbf{e}^{\mathrm{T}})(\mathbf{M}^{-1})^{\mathrm{T}}\end{bmatrix}. \tag{2.35}$$

Now we can employ the Chebyshev polynomial representation of the covariance matrix for equation (2.34) and obtain

$$\mathbf{W}_{\mathbf{f}}(t_1,t_2) \approx \mathbf{R}\sum_{i=1}^{k}\lambda_i\sum_{j=0}^{m_i-1}\sum_{l=0}^{m_i-1}H_{ij}H_{il}T_j(s_1)T_l(s_2) \tag{2.36}$$

where s_1 and s_2 are computed using equation (2.3). We can now write the $\mathbf{U}$ term in terms of Chebyshev polynomial as

$$\begin{aligned}
\mathbf{U}(t) &= \mathbf{Y}(t)\int_0^t \mathbf{Y}^{-1}(\tau)\mathbf{R}\sum_{i=1}^{k}\lambda_i \sum_{j=0}^{m_i-1}\sum_{l=0}^{m_i-1} H_{ij}H_{il}T_j(s_t)T_l(s_\tau)\mathrm{d}\tau \\
&= \sum_{i=1}^{k}\lambda_i \sum_{j=0}^{m_i-1}\sum_{l=0}^{m_i-1} H_{ij}H_{il}T_j(s_t)\mathbf{Y}(t)\int_0^t \mathbf{Y}^{-1}(\tau)\mathbf{R}T_l(s_\tau)\mathrm{d}\tau \\
\mathbf{U}(t) &= \sum_{i=1}^{k}\lambda_i \sum_{j=0}^{m_i-1}\sum_{l=0}^{m_i-1} H_{ij}H_{il}T_j(s_t)\mathbf{U}_l(t) \qquad (2.37)
\end{aligned}$$

where

$$\mathbf{U}_l(t) = \mathbf{Y}(t)\int_0^t \mathbf{Y}^{-1}(\tau)\mathbf{R}T_l(s_\tau)\mathrm{d}\tau \quad l = 1, 2, \cdots, m_i - 1. \qquad (2.38)$$

To solve this equation with the equation (2.26), we need to take derivative of (2.38) so that we can obtain a differential equation without the integral as follows:

$$\begin{aligned}
\dot{\mathbf{U}}_l(t) &= \dot{\mathbf{Y}}(t)\mathbf{Y}^{-1}(t)\mathbf{U}_l(t) + \mathbf{Y}(t)\mathbf{Y}^{-1}(t)\mathbf{R}T_l(s_t) \\
\dot{\mathbf{U}}_l(t) &= \mathbf{G}(t)\mathbf{U}_l(t) + \mathbf{R}T_l(s_t) \quad l = 1, 2, \cdots, m_i - 1 \qquad (2.39)
\end{aligned}$$

Finally we need to solve equations (2.39) along with the equations (2.22) and (2.27).

Another way to represent same problem to avoid the complexity, is simply to use the original form of the covariance given by equation (2.1) in the equation (2.6). This will give

$$\begin{aligned}
\mathbf{U}(t) &= \mathbf{Y}(t)\int_0^t \mathbf{Y}^{-1}(\tau)\mathbf{R}\sum_{i=1}^{k}\lambda_i\mathbf{p}_i(t)\mathbf{p}_i^{\mathrm{T}}(\tau)\mathrm{d}\tau \\
&= \sum_{i=1}^{k}\lambda_i\mathbf{p}_i(t)\mathbf{Y}(t)\int_0^t \mathbf{Y}^{-1}(\tau)\mathbf{R}\mathbf{p}_i^{\mathrm{T}}(\tau)\mathrm{d}\tau \\
\mathbf{U}(t) &= \sum_{i=1}^{k}\lambda_i\mathbf{p}_i(t)\mathbf{U}_i(t) \qquad (2.40)
\end{aligned}$$

where

$$\mathbf{U}_i(t) = \mathbf{Y}(t)\int_0^t \mathbf{Y}^{-1}(\tau)\mathbf{R}\mathbf{p}_i^{\mathrm{T}}(\tau)\mathrm{d}\tau \quad l = i, 2, \cdots, k \qquad (2.41)$$

and derivative of this equation with respect to time gives

$$\dot{\mathbf{U}}_i(t) = \mathbf{G}(t)\mathbf{U}_i(t) + \mathbf{R}\mathbf{p}_i^{\mathrm{T}}(t) \quad i = 1, 2, \cdots, k. \qquad (2.42)$$

2.5 A Numerical Example: Duffing Oscillator

We will examine a SDOF Duffing oscillator as an example. We will use the nonstationary excitation covariance given by Traina et al. (1986). First, we will derive the governing differential equations for the state covariance matrix. Then we will give an explanation of the numerical implementation of the problem. And finally, we will give a Mathematica code, written particularly for this problem, which in fact is a good example for implementation of more difficult problems.

2.5.1 Governing Equations for Duffing Oscillator

The nonlinear equations of motion of a simple Duffing oscillator is given by

$$\ddot{x}(t) + 2\zeta\dot{x}(t) + x(t) + \lambda x^3(t) = s(t) \tag{2.43}$$

One should note that for this specific case, $\mathbf{M} = 1$, $\mathbf{C} = 2\zeta$, $\mathbf{K} = 1$ and $\mathbf{h}(x, \dot{x}) = \lambda q^3$. The equivalent linear system is then given by

$$\ddot{x}(t) + 2\zeta\dot{x}(t) + (1 + k^e)x(t) = s(t) \tag{2.44}$$

where

$$k^e = \mathbb{E}\left[\frac{\partial h}{\partial x}\right] = \mathbb{E}\left[3\lambda x^2\right] = 3\lambda\mathbb{E}\left[x^2\right] \tag{2.45}$$

and, the state-space form becomes

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -(1 + 3\lambda\mathbb{E}\left[x^2\right]) & 2\zeta\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\end{bmatrix} + \begin{bmatrix}0\\ s\end{bmatrix}. \tag{2.46}$$

For this system $\mathbf{V}$ is a 2×2 matrix and $\mathbf{V}(1,1) = \mathbb{E}\left[x^2\right]$ and $\mathbf{V}(2,2) = \mathbb{E}\left[\dot{x}^2\right]$. Therefore, one can write the equation (2.27) as

$$\begin{bmatrix}\dot{X}_1 & \dot{X}_2\\ \dot{X}_2 & \dot{X}_3\end{bmatrix} = \begin{bmatrix}0 & 1\\ -(1 + 3\lambda X_1) & 2\zeta\end{bmatrix}\begin{bmatrix}X_1 & X_2\\ X_2 & X_3\end{bmatrix} + \begin{bmatrix}0 & -(1 + 3\lambda X_1)\\ 1 & 2\zeta\end{bmatrix} + \begin{bmatrix}X_4 & X_5\\ X_6 & X_7\end{bmatrix} + \begin{bmatrix}X_4 & X_6\\ X_5 & X_7\end{bmatrix} \tag{2.47}$$

where

$$X_1 = \mathbf{V}(1,1), \quad X_2 = \mathbf{V}(2,1) = \mathbf{V}(1,2), \quad X_3 = \mathbf{U}(1,1), \quad X_4 = \mathbf{U}(1,2), \quad X_5 = \mathbf{U}(2,1), \quad X_7 = \mathbf{U}(2,2). \tag{2.48}$$

Finally we have three nonlinear differential equations:

$$\dot{X}_1 = 2X_2 + 2X_4 \tag{2.49a}$$
$$\dot{X}_2 = X_3 - (1 + 3\lambda X_1)X_1 - 2\zeta X_2 + X_5 + X_6 \tag{2.49b}$$
$$\dot{X}_3 = -2(1 + 3\lambda X_1)X_2 - 4\zeta X_3 + 2X_7 \tag{2.49c}$$

One should also note that some of the variables shown here are zeros since

$$e = 1 \Rightarrow \mathbf{R} = \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}. \tag{2.50}$$

To be able to solve these equations we need to solve for $\mathbf{U}$, i.e., X_4, X_5, X_6, X_7. $\mathbf{U}$ is computed from (2.37) and (2.39), which includes differential equations also. Therefore we have two groups of differential equations to solve.

Above differential equations are obtained particularly for a Duffing Oscillator. To be aplicabel to a general MDOF system, in the following numerical example we will employ the matrix form of the differential equations given by 2.27, (2.41) and (2.42).

2.5.2 Random Excitation and Covariance Kernel

As the covariance excitation, we use the covariance function given by Traina et al. (1986), which is defined as follows. Consider a random process defined by

$$x(t) = g(t)\Pi(t)n(t) \tag{2.51}$$

where

$$g(t) = \exp(-b_1 t) - \exp(-b_2 t) \quad and \quad b_1, b_2 \in \mathbb{R} \tag{2.52}$$

$$\Pi(t) = \begin{cases} 0 & : \quad t < 0 \\ 1 & : \quad 0 < t < T_m \\ 0 & : \quad T_m < t \end{cases} \tag{2.53}$$

$$n(t) : N(0, \sqrt{\frac{2\pi S_0}{\Delta t}}) \tag{2.54}$$

where $N(m, \sigma)$ is Gaussian probability density with mean m and variance σ^2. The corresponding covariance becomes

$$\mathcal{C}(t_1, t_2) = g(t_1)g(t_2)\Pi(t_1)\Pi(t_2)R_n(t_2 - t_1) \tag{2.55}$$

where $R_n(\tau)$ is the autocorrelation function and given by

$$R_n(\tau) = \begin{cases} \sigma_0^2 \left[\frac{2}{3} - \left(\frac{\tau}{\Delta t}\right)^2 + \frac{1}{2}\left(\frac{|\tau|}{\Delta t}\right)^3\right] & : \quad |\tau| < \Delta t \\ \sigma_0^2 \left[\frac{4}{3} - 2\frac{|\tau|}{\Delta t} + \left(\frac{\tau}{\Delta t}\right)^2 - \frac{1}{6}\left(\frac{|\tau|}{\delta t}\right)^3\right] & : \quad \Delta t \leqslant |\tau| < 2\Delta t \\ 0 & : \quad 2\Delta t \leqslant |\tau| \end{cases} \tag{2.56}$$

where $\tau = t_2 - t_1$. In this example we take $b_1 = 1$, $b_2 = 0.1$, $T_m = 20.1$, $\Delta t = 1$ and $S_0 = 1$.

2.5.3 Numerical Solution Technique and Mathematica Source code

It is immediately observed that governing equations are nonlinear differential equations. As analytical solution may be difficult to obtain, we solve them numerically. A fourth-order Runge-Kutta (RK4) method is used to solve these equations. We will use the matrix form of the differential equations to illustrate applicability of the method to the MDOF systems. We will use the formulation given by (2.41) and (2.42) instead of using the original formulation given in the paper. First we give a very brief summary of the RK4 method. Then we will give the source code that was implemented in Mathemeatica and descriptions of the statements.

Fourth Order Runge-Kutta Method

Consider a set of first order differential equations:

$$\dot{x} = f(t, x, y) \tag{2.57a}$$

$$\dot{y} = g(t, x, y). \tag{2.57b}$$

given the initial values t_0, x_0 and y_0, the values of x and y for the next time step $t_1 = t_0 + \Delta t$ are computed from

$$x_1 = x_0 + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.58a}$$

$$y_1 = y_0 + \frac{\Delta t}{6}(m_1 + 2m_2 + 2m_3 + m_4) \tag{2.58b}$$

where

$$k_1 = f(t_0, x_n, y_n) \tag{2.59a}$$

$$m_1 = g(t_0, x_n, y_n) \tag{2.59b}$$

$$k_2 = f(t_0 + \frac{\Delta t}{2}, x_0 + \frac{\Delta t}{2}k_1, y_0 + \frac{\Delta t}{2}m_1) \tag{2.59c}$$

$$m_2 = g(t_0 + \frac{\Delta t}{2}, x_0 + \frac{\Delta t}{2}k_1, y_0 + \frac{\Delta t}{2}m_1) \tag{2.59d}$$

$$k_3 = f(t_0 + \frac{\Delta t}{2}, x_0 + \frac{\Delta t}{2}k_2, y_0 + \frac{\Delta t}{2}m_2) \tag{2.59e}$$

$$m_3 = g(t_0 + \frac{\Delta t}{2}, x_0 + \frac{\Delta t}{2}k_2, y_0 + \frac{\Delta t}{2}m_2) \tag{2.59f}$$

$$k_4 = f(t_0 + (\Delta t), x_0 + (\Delta t)k_3, y_0 + (\Delta t)m_3) \tag{2.59g}$$

$$m_4 = g(t_0 + (\Delta t), x_0 + (\Delta t)k_3, y_0 + (\Delta t)m_3). \tag{2.59h}$$

This formulation can easily be extended to system with more than two differential equations, which is the case for our problem.

In this particular example, we set the time step to 0.1. We used first 20 eigenvalues and 25-order Chebyshev approximation of these eigenvalues. Clearly these values can be modified to obtain better approximation of the covariance kernel leading to better results.

Source Code

```
(*THIS MATHEMATICA FILE COMPUTES THE NONSTATIONARY RESPONSE OF A DUFFING OSC.*)
(*IT USES A FOURTH ORDER RUNGE KUTTA SOLVER TO SOLVE THE DIFFERENTIAL EQUATIONS*)
(*OBTAINED FROM THE LINEARIZATION OF THE SYSTEM*)

Clear["Global`*"];
(*************************************************************************)
(*COVARINACE FUNCTION (TRAINA ET AL., 1986) *)
covar01[t1_, t2_] := Module[{b1 = 1, b2 = 0.1, Tm = 20.1,
tmax = Tm, dt = 1, S0 = 1,g1, g2, pi1, pi2, tau, sigmasqr, fun1, fun2, Rntau},

tau = t2 - t1;
sigmasqr = 2*Pi*S0/dt;

g1 = Exp[-b1*t1] - Exp[-b2*t1];
g2 = Exp[-b1*t2] - Exp[-b2*t2];

pi1 = If[(0 <= t1) && (t1 <= Tm), 1, 0];
```

```
pi2 = If[(0 <= t2) && (t2 <= Tm), 1, 0];

fun1 = sigmasqr*(2/3 - (tau/dt)^2 + 1/2*(Abs[tau]/dt)^3);
fun2 = sigmasqr*(4/3 - 2*Abs[tau]/dt + (tau/dt)^2 - 1/6*(Abs[tau]/dt)^3);

Rntau = If[Abs[tau] < dt, fun1,
If[(dt <= Abs[tau]) && (Abs[tau] < 2*dt), fun2,
If[Abs[tau] >= 2*dt, 0, ERR01]]];

g1*g2*pi1*pi2*Rntau];
(*****************************************************************************)
(*****************************************************************************)
(*SOME PLOTS OF THE COVARIANCE FUNCTION*)
(*Plot3D[Evaluate[covar01[t1, t2]], {t1, 0, 20}, {t2, 0, 20},
PlotRange -> All]*)

tmax = 20.1; tmin = 0.1; dt = 0.1; t = Range[tmin, tmax, dt]; len = Length[t];
tnew = Table[2*(t[[i]] - tmin)/(tmax - tmin) - 1, {i, 1, len}];

covar = Transpose[Table[Table[covar01[t1, t2], {t1, tmin, tmax, dt}],
{t2, tmin, tmax, dt}]];

(*ListPlot3D[covar,
MeshRange -> {{tmin, tmax}, {tmin, tmax}}, PlotRange -> All]*)
(*****************************************************************************)
(*****************************************************************************)
(*EIGENVALUES AND EIGENVECTORS OF THE COVARIANCE*)

evals = Eigenvalues[covar];
evecs = Eigenvectors[covar];
(*****************************************************************************)
(*****************************************************************************)
(*COMPUTATION OF THE CHEBYSHEV POLYNOMIAL APPROXIMATION OF THE COVARIANCE*)
maxevec =  20;
evecfit = Table[0, {i, maxevec}] ;

Do[
tempevec1 = Table[{t[[i]], evecs[[enum, i]]}, {i, 1, len} ];
tempevec2 = Table[{tnew[[i]], evecs[[enum, i]]}, {i, 1, len} ];
NN = 25;
basefun2 = Table[ChebyshevT[i, x], {i, 0, NN}];
evecfit2 = Fit[tempevec2, basefun2, x];
tt = 2*(x - tmin)/(tmax - tmin) - 1;
evecfit[[enum]] = (evecfit2 /. x -> tt);
Block[{$DisplayFunction = Identity}, g1 = ListPlot[tempevec1,
PlotJoined -> True, PlotRange -> All,
```

```
PlotStyle -> {Thickness[0.002], Dashing[{0.02}]}];
g2 = Plot[evecfit[[enum]], {x, tmin, tmax}, PlotRange -> All]];
(*Show[g1, g2, PlotRange -> All]; *)
, {enum, maxevec}]

(*apprcovar = Sum[evals[[k]]*(evecfit[[k]] /. x -> t1)
*(evecfit[[k]] /. x -> t2), {k, maxevec}];*)

(*EVALUATE THIS STATEMENT TO SEE THE APPROXIMATE COVARIANCE MATRIX*)
(*Plot3D[apprcovar, {t1, tmin, tmax}, {t2, tmin, tmax}, PlotRange -> All]*)
(*************************************************************************)
(*************************************************************************)
(*DUFFING OSCILLATOR*)
dr = 0.1; lam = 0.1; dof = 2 ;
R = {{0, 0}, {0, 1}};
V = Table[0, {i, len}, {j, dof}, {k, dof}] ;

Ubasis = Table[0, {i, len}, {j, maxevec}, {k, dof}, {i, dof}] ;
UbasisF1 = Table[0, {j, maxevec}, {k, dof}, {i, dof}] ;
UbasisF2 = Table[0, {j, maxevec}, {k, dof}, {i, dof}] ;
UbasisF3 = Table[0, {j, maxevec}, {k, dof}, {i, dof}] ;
UbasisF4 = Table[0, {j, maxevec}, {k, dof}, {i, dof}] ;

Tjs = Table[0, {i, maxevec}];

(*RK4 SOLVER*)
Do[ step = k; tini = t[[k]];

(*F1*)
time = tini;
Ubasval = Ubasis[[step]];
Vval = V[[step]];
G = {{0, 1}, {-(1 + 3*lam*Vval[[1, 1]]), -2*dr}};
Do [Tjs[[i]] = (evecfit[[i]] /. x -> time), {i, maxevec}];
U = Sum[evals[[i]]*Tjs[[i]]*Ubasval[[i]], {i, maxevec}];
VF1 = G.Vval + Vval.Transpose[G] + U + Transpose[U];
Do[UbasisF1[[j]] = G.Ubasval[[j]] + R*Tjs[[j]], {j, maxevec}];

(*F2*)
time = tini + dt/2;
Ubasval = Ubasis[[step]] + dt/2*UbasisF1;
Vval = V[[step]] + dt/2*VF1;
G = {{0, 1}, {-(1 + 3*lam*Vval[[1, 1]]), -2*dr}};
Do [Tjs[[i]] = (evecfit[[i]] /. x -> time), {i, maxevec}];
U = Sum[evals[[i]]*Tjs[[i]]*Ubasval[[i]], {i, maxevec}];
VF2 = G.Vval + Vval.Transpose[G] + U + Transpose[U];
```

```
Do[UbasisF2[[j]] = G.Ubasval[[j]] + R*Tjs[[j]], {j, maxevec}];

(*F3*)
time = tini + dt/2;
Ubasval = Ubasis[[step]] + dt/2*UbasisF2;
Vval = V[[step]] + dt/2*VF2;
G = {{0, 1}, {-(1 + 3*lam*Vval[[1, 1]]), -2*dr}};
Do [Tjs[[i]] = (evecfit[[i]] /. x -> time), {i, maxevec}];
U = Sum[evals[[i]]*Tjs[[i]]*Ubasval[[i]], {i, maxevec}];
VF3 = G.Vval + Vval.Transpose[G] + U + Transpose[U];
Do[UbasisF3[[j]] = G.Ubasval[[j]] + R*Tjs[[j]], {j, maxevec}];

(*F4*)
time = tini + dt;
Ubasval = Ubasis[[step]] + dt*UbasisF3;
Vval = V[[step]] + dt*VF3;
G = {{0, 1}, {-(1 + 3*lam*Vval[[1, 1]]), -2*dr}};
Do [Tjs[[i]] = (evecfit[[i]] /. x -> time), {i, maxevec}];
U = Sum[evals[[i]]*Tjs[[i]]*Ubasval[[i]], {i, maxevec}];
VF4 = G.Vval + Vval.Transpose[G] + U + Transpose[U];
Do[UbasisF4[[j]] = G.Ubasval[[j]] + R*Tjs[[j]], {j, maxevec}];

Vnext = V[[step]] + dt/6*(VF1 + VF2 + VF3 + VF4); V[[step + 1]] = Vnext;
Ubasisnext = Ubasis[[step]] + dt/6*(UbasisF1 + UbasisF2 + UbasisF3 + UbasisF4);
Ubasis[[step + 1]] = Ubasisnext;, {k, len - 1}]

(*PLOT THE RESULTS*)
dat = Table[{t[[i]], Sqrt[V[[i, 1, 1]]]}, {i, len}];
ListPlot[dat,PlotRange -> All, PlotStyle -> Thickness[0.01],
PlotLabel -> "Sqrt(E[x1^2])",
TextStyle -> {Helvetica, FontSize -> 12},
AxesLabel -> {"t", ""}, PlotJoined -> True]

dat = Table[{t[[i]], Sqrt[V[[i, 2, 2]]]}, {i, len}];
ListPlot[dat,PlotRange -> All, PlotStyle -> Thickness[0.01],
PlotLabel -> "Sqrt(E[x1^2])",
TextStyle -> {Helvetica, FontSize -> 12},
AxesLabel -> {"t", ""}, PlotJoined -> True]
(**************************************************************************)
```

2.5.4 Monte-Carlo Simulation and Mathematica Source Code

To investigate the validity of the method and the code, we also carried out a Monte-Carlo Simulation. 10000 synthetic excitation is created based on random process definition given above. Then, the Duffing oscillator is analyzed using Runge-Kutta method for a time step $\Delta t = 0.01$.

Source

```
(*THIS CODE CREATES A SET OF SYNTHETIC EXCITATION AND ANALYZE
THE DUFFING OSCILLATOR FOR EACH OF THE EXCITATION *)

Clear["Global`*"];
(*********************************************************************)
Needs["Statistics`ContinuousDistributions`"];
Needs["Statistics`DataManipulation`"];
Needs["Graphics`Graphics`"];
(*********************************************************************)
(*********************************************************************)
(* SYSTEM PARAMETERS*)
dt = 1;  t0 = 0.1; dt = 1; Tm = 20.1; S0 = 1; sigma0 =
  Sqrt[2*Pi*S0/dt]; m = 0;
b1 = 1; b2 = 0.1;
t = Table[i, {i, t0, Tm, dt}]; t = Flatten[{0, t}]; size = Length[t];
newdt = 0.01; newt = Table[i, {i, 0, Tm, newdt}]; newsize = Length[newt];
dr = 0.1; lam = 0.1;
nsim = 3;
sumx = Table[0, {i, 1, newsize}]; sumv = Table[0, {i, 1, newsize}];
(*********************************************************************)
(*********************************************************************)
(* CREATE THE EXCITATION AND ANALYZE THE SYSTEM *)
Do[

    dist = NormalDistribution[m, sigma0];
tempn = RandomArray[dist, size];
datn = Table[ {t[[i]], tempn[[i]]}, {i, 1, size}];
datn[[1]] = {0, 0};
interfun = Interpolation[datn, InterpolationOrder -> 1];
interdat = Table[interfun[newt[[i]]], {i, 1, newsize}];
fung = Table[{newt[[i]],
(Exp[-b1*newt[[i]] ] - Exp[-b2*newt[[i]]])}, {i, 1, newsize}];
    (*ListPlot[fung, PlotJoined -> True];*)

funn = Table[{newt[[i]], interdat[[i]]}, {i, 1, newsize}];
    (*ListPlot[funn, PlotJoined -> True];*)
excit = Table[(Exp[-b1*newt[[i]] ] - Exp[-b2*newt[[i]]])*interdat[[i]], {i,
          1, newsize}];
(*ListPlot[Table[{newt[[i]], excit[[i]]}, {i, 1, newsize}],
        PlotJoined -> True]*)

    (****************************************)
    (* RK4 Solution of the duffing Oscillator *)
```

```
    x = Table[0, {i, 1, newsize}]; v = Table[0, {i, 1, newsize}];

Do [
      (*f1 and g1*)
      tim = newt[[k]];
      disp = x[[k]];
      vel = v[[k]];
      exc = excit[[k]];
      f1 = exc - 2*lam*vel - disp - lam*disp^3;
      g1 = vel;

      (*f2 and g2*)
      tim = newt[[k]] + newdt/2;
      disp = x[[k]] + newdt/2*f1;
      vel = v[[k]] + newdt/2*g1;
      exc = (excit[[k]] + excit[[k + 1]])/2;
      f2 = exc - 2*lam*vel - disp - lam*disp^3;
      g2 = vel;

      (*f3 and g3*)
      tim = newt[[k]] + newdt/2;
      disp = x[[k]] + newdt/2*f2;
      vel = v[[k]] + newdt/2*g2;
      exc = (excit[[k]] + excit[[k + 1]])/2;
      f3 = exc - 2*lam*vel - disp - lam*disp^3;
      g3 = vel;

      (*f4 and g4*)
      tim = newt[[k]] + newdt;
      disp = x[[k]] + newdt*f3; vel = v[[k]] + newdt*g3;
      exc = excit[[k + 1]];
      f4 = exc - 2*lam*vel - disp - lam*disp^3;
      g4 = vel;

      v[[k + 1]] = v[[k]] + (newdt/6)*(f1 + 2*f2 + 2*f3 + f4);
      x[[k + 1]] = x[[k]] + (newdt/6)*(g1 + 2*g2 + 2*g3 + g4);

      , {k, newsize - 1}];
      (****************************************)

    sumx = sumx + x^2; sumv = sumv + v^2;
(*ListPlot[Table[{newt[[i]], x[[i]]}, {i, 1, newsize}],
          PlotJoined -> True];*)
(*ListPlot[Table[{newt[[i]], v[[i]]}, {i, 1, newsize}],
          PlotJoined -> True];*)
```

```
    , {sim, nsim}];
(*********************************************************************)
(*********************************************************************)
(* PLOT THE RMS VALUES *)
ListPlot[Table[{i, Sqrt[sumx[[i]]/nsim]}, {i, 1, newsize}],
    PlotJoined -> True];
ListPlot[Table[{i, Sqrt[sumv[[i]]/nsim]}, {i, 1, newsize}],
    PlotJoined -> True];
(*********************************************************************)

Save ["c:\montecarlo", "sumv, sumx, newt, newsize, nsim"];
```

Remarks on Mathematica

Above file was written and compiled in Mathematica Version 4.0.0. The author believes that Mathematica is not the best programming language that can be used to implement this problem. The beneficial part of Mathematica is the function "`Fit`", which is very efficient. However, Mathematica is not suitable for numerical computation. Its matrix support is very primitive and difficult to understand. Moreover it does not have the capabilities of a standard programming language like Fortran, C or Matlab. It does not have exception handling and debugging capabilities at all, which makes the error tracing almost impossible. The author strongly recommends other programming languages for this problem. For polynomial fit, Mathematica can be used and, approximated polynomial can be transferred to the programming language used. Another problem with Mathematica is that it is not a WYSIWYG (what you see is what you get) program. It is a very high-level programming language (higher that Matlab). The source code that appear on the output (screen) may not be the code that Mathematica kernel processing. The version used is not a stable version and cause too much trouble with the code and modifies or corrupts it. The author recommends to save the Mathematica codes into simple ASCII files ("`.txt`" files) instead of working with original Mathematica files.

2.5.5 Results

Figure (2.2) shows the covariance kernel used. For this covariance, the root mean square displacement and the velocity are obtained as shown in figures 2.3(a) and 2.3(b).

As can be from the results, the equivalent linearization method can predict the shape very well but not the magnitudes. The reason for that is obviously the limited number of eigenvectors used and low order polynomial approximation. For example, in the original paper, authors use 100 eigenvalues and 200-order approximation. Also, the a smaller time step should be chosen for the RK4 solution. Another possibility is an error in the source code. The code should also be reviewed and should be executed in more stable versions of Mathematica.

One suggestion is to work on the polynomial fit separately to obtain a better approximation of the covariance kernel. After that the RK4 solver can be improved for better results.

Although, looks complex, the author believes that this method is quite straightforward and stays as a powerful tool for the analysis of the nonlinear structures excited by nonstationary excitation.

2.5.6 Conclusion

In this section, we reviewed the method proposed by Smyth and Masri (2002) for the analysis of nonlinear MDOF systems excited by nonstationary excitation. A Mathematica code is presented for the analysis of a Duffing oscillator. The results of the method is compared with a Monte-Carlo simulation. Suggestions for the improvement of the method are given.

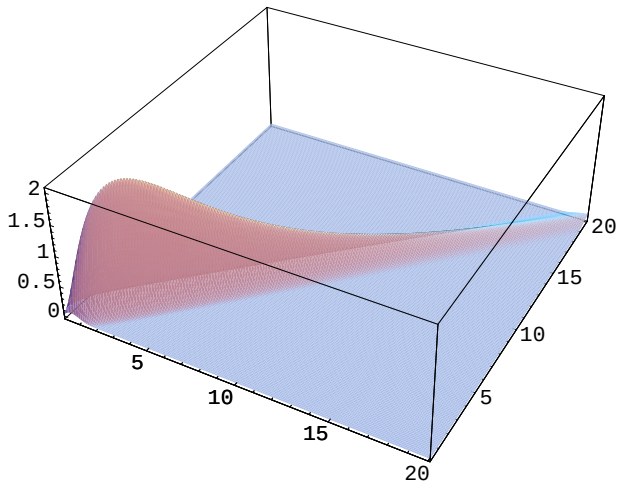
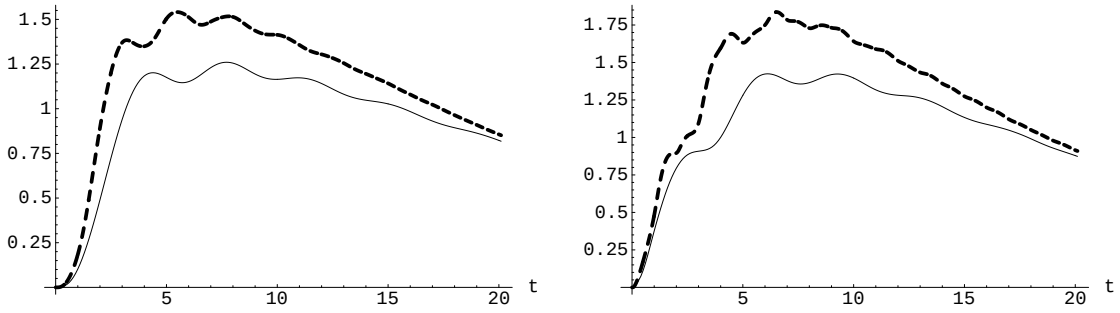


Figure 2.2: Covariance kernel of the excitation



(a) Displacement (b) Velocity

Figure 2.3: Root mean square values of the responses (Dash: Monte-Carlo Simulation, Solid: Equivalent Linearization)

Bibliography

Kazakov, I. E. (1965). "Generalization of the method of statistical linearization to multidimensional systems." *Automatic Remote Control*, 26, 1201–1206.

Meirovitch, L. (1986). *Elements of Vibration Analysis*. McGraw-Hill, New York, second edition.

Papoulis, A. (1984). *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York, second edition.

Reid, J. G. (1983). *Linear System Fundamentals: Continuous and Discrete, Classic and Modern*. McGraw Hill, New York.

Roberts, J. B. and Spanos, P. D. (1990). *Random Vibrations and Statistical Linearization*. John Wiley & Sons, New Jersey.

Smyth, A. W. and Masri, S. F. (2002). "Nonstationary response of nonlinear systems using equivalent linearization with a compact analytical form of the excitation process." *Probabilistic Engineering Mechanics*, 17(1), 97–108.

Stark, H. and Woods, J. W. (2002). *Probability and Random Processes with Applications to Signal Processing*. Prentice-Hall, New Jersey, third edition.

Traina, M. I., Miller, R. K., and Masri, S. F. (1986). "Orthogonal decomposition and transmission of nonstationary random process." *Probabilistic Engineering Mechanics*, 1(3), 136–149.

Wirsching, P. H., Paez, T. L., and Ortiz, K. (1995). *Random Vibrations*. John Wiley & Sons, New York.