AN ABSTRACT OF THE THESIS OF

Mohsen Bahrami for the degree of DOCTOR OF PHILOSOPHY

in Mechanical Engineering presented on January 20, 1982

Title: PERIODIC SYSTEM ANALYSIS WITH APPLICATION TO WIND TURBINES

Redacted for Privacy

Abstract approved:

Dr. William E. Holley

An algorithm is developed to compute the time averaged, output statistics of a linear system with periodic coefficients forced by stationary white noise. The algorithm makes use of Floquet theory and introduces the averaged system dynamics matrix to obtain more favorable numerical performance. The time series approach to averaged statistics is also examined. Application is made to a wind turbine system with atmospheric turbulence.

# Periodic System Analysis
# with
# Application to Wind Turbines

by

Mohsen Bahrami

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctoral of Philosophy

June, 1982

APPROVED:

Redacted for Privacy

Professor of Mechanical Engineering in charge of major

Redacted for Privacy

Head of Department of Mechanical Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented January 20, 1982

Typed by Barbara Hanson for Mohsen Bahrami

In the Name of

Righteousness and Justice

To:

Those who seek knowledge

To be able to better serve the people

# ACKNOWLEDGEMENT

I wish to express my sincere gratitude to all of my teachers. In particular, I would like to show my deep appreciation to my major professor at Oregon State University, Dr. William E. Holley, whose help and guidance has been the main impetus for this accomplishment.

It should be highlighted that I am able to make this achievement because of my parents' (Naheed and Alizera Bahrami) sacrifices, efforts and encouragements. I would also like to thank my brother, Hassan, for his kind company and my brother Mehrdad.

I am also grateful to the staff of Mechanical Engineering Department at Oregon State University for the friendly environment they have created.

The Oregon State University Computer Center was helpful in providing computer time through unsponsored research funds.

A special thanks is extended to Battelle, Pacific Northwest Laboratory for their support of this research during the time period of July 1, 1980 to September 30, 1981, under U. S. Department of Energy Task Agreement DE-AT06-79ET23114.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF SYMBOLS

| Symbol | |
|---|---|
| $A,B,\ldots,S$ | Matrices (including primed & barred letters), unless specified otherwise |
| $\dot{A},\dot{B},\ldots,\dot{S}$ | Time derivatives in case of time varying matrices |
| $T$ | Period of time varying matrices |
| $a,b,\ldots,s,t$ | Scalar (including primed & barred letters), unless specified otherwise |
| $e$ | Neparian number |
| $i$ | $\sqrt{-1}$ |
| $u,v,\ldots,z$ | Vectors (including primed & barred), unless otherwise specified |
| $\dot{u},\dot{v},\ldots,\dot{z}$ | Time derivative in cases of time varying vectors |
| $w$ | White noise |
| $\pi$ | 3.14 |
| $\omega$ | Angular velocity vector |
| $\sum$ | Multiple summation over m,n,... |
| $m,n,\ldots$ | From $-\infty$ to $\infty$, unless other limits specified |

# Periodic System Analysis
# with Application to Wind Turbines

## 0. Introduction

Linear system equations with periodic coefficients have been of interest in the design of helicopter blades in the past, and are attracting increasing attention in present wind turbine research. Most of the work which has dealt with this subject has considered the system input to be deterministic. In many cases, however, the forcing function is of a random nature. The consideration of random inputs is important because it provides a more realistic model of wind properties, the major forcing function for wind turbine response. To fill this apparent gap and provide the basis for a better understanding of the effect of random inputs on linear systems with periodic coefficients, this work undertakes the study of such systems with a statistically stationary forcing function. The stationarity assumption is made because in practice it has been shown to be a useful model for the randomness in the wind.

The thesis is divided into three chapters. Chapter I presents the Floquet Theory, which is the main tool for transforming systems with periodic coefficients into systems with constant system dynamics matrices. Based on this theory an algorithm is developed to achieve this transformation. Chapter II provides discussions on modeling wind as a stationary process and on deriving the stationary processes from white noise. This derivation is used to modify the system with stationary input to an augumented system with white noise input. An algorithm is then developed to relate the input-output stochastic properties of the augmented system assuming it has a constant system matrix. Chapter III presents two models which are analyzed using the algorithms. One is a fifth order system consisting of a single rigid, rotating blade with a flapping degree of freedom. The second is a nineteenth order model of a 2.5 Mw wind turbine. Results and conclusions are also given in this chapter.

Appendix A presents the computer program for the algorithm which is coded in FORTRAN 77 and is implemented on the Oregon State University computer system (CDC CYBER 170, Model 720). A second order example is also presented, and a step by step solution is given. Appendix B contains the numerical values of matrices and parameters of the models discussed in Chapter III.

# I. Equivalence Transformations and Floquet Theory

Engineering problems modeled by linear, time-varying differential equations are difficult to solve, in general. However, in cases where the coefficients are periodic, equivalence transformations can be used to transfer these systems into other linear systems which are easier to study. One such transformation can be deduced from Floquet theory. This theory provides the tool for transferring systems with time varying, periodic coefficients into systems with constant system dynamics matrices.

Section 1 of this chapter discusses equivalence transformations and their properties. Section 2 presents Floquet theory and the equivalence transformation it provides. In Section 3 an algorithm is developed, based on Floquet theory, to carry out the transformation numerically. The first approach in the development of this algorithm was the direct use of the results of Floquet theory. This approach failed due to integration truncation errors when stiff equations were encountered. Modifications were then made, which helped the algorithm to perform more favorably.

For an extensive treatment of the subject the interested reader is referred to V. A. Yakubovich and V. M. Starzhinskii's Linear Differential Equations with Periodic Coefficients [1].

## I-1. Equivalence Transformations:

A linear system with time varying coefficients of the form:

$$\begin{aligned} \dot{x} &= A(t)x + B(t)u \\ y &= C(t)x + D(t)u \end{aligned} \qquad (I.1)$$

is said to be algebraically equivalent to the following linear time varying system:

$$\begin{aligned} \dot{x}' &= A'(t)x' + B'(t)u \\ y &= C'(t)x' + D'(t)u \end{aligned} \qquad (I.2)$$

where

$$x' = P(t)x \tag{I.3}$$

with P(t) a nonsingular differentiable matrix.

and

$$\begin{aligned} A'(t) &= (P(t)A(t) + \dot{P}(t))P^{-1}(t) \\ B'(t) &= P(t)B(t) \\ C'(t) &= C(t)P^{-1}(t) \\ D'(t) &= D(t) \end{aligned} \tag{I.4}$$

The state transformation matrix, P(t), is called an equivalence transformation [2, 3, 4].

At this point the following definitions are given before proceeding to discuss the equivalence transformations in more detail.

Definition I-1- Two linear systems of the form of Eqn. I.1 are said to be zero-state equivalent if they have the same impulse response [5, 6].

Definition I-2- Two linear systems of the form of Eqn. I.1 are said to be zero-input equivalent if and only if for any initial state in one system there exists an initial state in the other system so that the outputs of the systems are identical [7].

With regard to these definitions it is apparent that systems which are algebraically equivalent are also zero-state equivalent and zero-input equivalent. The reverse however, is not true, i.e., two linear time-varying systems which are zero-state equivalent and zero-input equivalent are not necessarily algebraically equivalent. Because of zero-state equivalency, the impulse responses of two algebraically equivalent systems are the same. This implies that the input-output relations are invariant under equivalence transformations. Wiberg [8] shows by means of two theorems that the controllability and the observability properties of time-varying systems are also preserved when these systems undergo equivalence transformations. Wiberg also presents other theorems which help to reduce time varying systems into simpler forms by means of equivalence transformations with additional assumptions on the controllability and observability matrices. It is worth noting that the similarity transformation, which is a powerful tool in studying linear time invariant systems, is also an equivalence transformation.

One should be careful of the fact that equivalence transformations do not preserve the stability properties of time-varying systems in general. However, if a norm of the transformation matrix, P(t) and the same norm of its inverse $P^{-1}(t)$ are bounded (not necessarily with the same bounds) for all time, the transformation preserves the stability properties of the system and is called a Lyapunov transformation. Two systems related with a Lyapunov transformation are said to be topologically equivalent [9].

Note that a similarity transformation is a Lyapunov transformation as well. Hence, in the case of time-invariant linear systems all equivalence transformations are Lyapunov transformations.

## I-2. Theory of Floquet

Floquet theory provides the basis to find an equivalence transformation which transforms linear time-varying systems with periodic coefficients into systems with a constant system dynamics matrix (hereafter simply called the system matrix).

To find this transformation, consider the general form of an nth order, linear, time-varying system with periodic coefficients in its state space representation:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \tag{I.5}$$

Where A, B, C and D are time dependent matrices which are continuous and defined over the time domain $(-\infty, \infty)$.

In addition A(t) is periodic with period T

i.e. $A(t+T) = A(t)$ for all t.

The periodicity of B, C and D does not have any effect on the derivation of the desired transformation. In this work however, these matrices will also be assumed to be constant or periodic with the period T.

In the following, we discuss some properties of linear time-varying systems with periodic coefficients.

Theorem I-1: If a fundamental matrix of the system of the Eqn. I.5 is F(t), then F(t+T) is also a fundamental matrix.

Proof: To show this, first note that:

$$\dot{F}(t) = A(t)F(t)$$

and

$$A(t+T) = A(t)$$

Now

$$\dot{F}(t+T) = A(t+T)F(t+T) = A(t)F(t+T)$$

QED

Because F(t) and F(t+T) are fundamental matrices, both are nonsingular and each can form a basis for the solution space. Thus, there exists a constant nonsingular matrix Q such that [10]

$$F(t+T) = F(t)Q \tag{I.6}$$

Q is sometimes called the monodromy matrix of the fundamental matrix F(t).

Furthermore, a constant matrix $\bar{A}$ can be found [11] so that

$$Q = e^{\bar{A}T} \tag{I.7}$$

which implies

$$F(t+T) = F(t)e^{\overline{A}T} \tag{I.8}$$

It should be noted that the fundamental matrix F(t) is not unique. However, from Theorem I-1 it can be deduced that determining F(t) over one period is sufficient to know its behavior for all time. The nonuniqueness of F(t) results in nonuniqueness of Q and consequently that of $\overline{A}$. But because different Q's derived from different F(t)'s are similar, they have the same eigenvalues. These eigenvalues, $a_j$, j = 1,2,3,...n, are called the characteristic multipliers associated with the periodic matrix A(t). This definition is justified by noting that there exists a solution such that:

$$x(t+T) = a_j\ x(t) \tag{I.9}$$

To show this, consider Eqn. I.6, and assume Q is a diagonal matrix. For each column of F(·) denoted by $F_j(\cdot)$, we have

$$F_j(t+T) = a_j\ F_j(t) \tag{I.10}$$

Because each column of F(·) is a solution vector, there exists a solution x such that

$$x(t+T) = a_j\ x(t) \tag{I.11}$$

The eigenvalues of $\overline{A}$, denoted by $\overline{a}_j$, are called the characteristic exponents of the periodic matrix A(t).

Note that $a_j$ and $\overline{a}_j$ are related as follows:

$$\overline{a}_j = \frac{1}{T}\ \ln(a_j) \tag{I.12}$$

Now, define the matrix P(t) so that

$$P(t) = e^{\overline{A}t}\ F^{-1}(t) \tag{I.13}$$

At this point we can state the Floquet theorem [12]:

Theorem I-2: For the system

$$\dot{x} = A(t)x, \; A(t+T) = A(t) \tag{I.14}$$

any fundamental matrix F(t) is of the form $P^{-1}(t)e^{\overline{A}t}$ where $P^{-1}(t)$ is a periodic function.

Proof: By definition

$$P^{-1}(t) = F(t)e^{-\overline{A}t}$$

thus

$$F(t) = P^{-1}(t)e^{\overline{A}t}$$

Now, to show $P^{-1}(t)$ is periodic, proceed as follows:

$$\begin{aligned} P^{-1}(t+T) &= F(t+T)e^{-\overline{A}(t+T)} \\ &= F(t)e^{\overline{A}T}(e^{-\overline{A}T}e^{-\overline{A}t}) \\ &= P^{-1}(t) \end{aligned}$$

QED

Remark I-1: P(t) is also periodic since

$$\begin{aligned} P(t+T) &= e^{\overline{A}(t+T)} \; F^{-1}(t+T) \\ &= e^{\overline{A}(t+T)} \; e^{-\overline{A}T} \; F^{-1}(t) \\ &= P(t) \end{aligned} \tag{I.15}$$

Remark I-2: Note that proving P(t) and $P^{-1}(t)$ are periodic, and F(t) and $\overline{A}$ are nonsingular, shows that P(t) and $P^{-1}(t)$ are nonsingular.

Theorem I-3: If the matrix P is defined as in I.13, the following system is algebraically equivalent to the system of Eqn. I.5.

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\,\bar{x}(t) + P(t)B(t)u(t) \\ y(t) &= C(t)P^{-1}(t)\bar{x}(t) + D(t)u(t) \end{aligned} \qquad (I.16)$$

Where $\bar{A}$ is a constant matrix as defined in Eqn. I.6.

Proof: See [13].

Remark I-3: Because P(t) and $P^{-1}(t)$ are bounded, the systems of Eqn. I.5 and Eqn. I.16 are also topologically equivalent.

Remark I-4: The following relation exists between P(t) and $\bar{A}$ [14]:

$$\dot{P}(t) = \bar{A}P(t) - P(t)A(t) \qquad (I.17)$$

For the inverse of P, we can write a similar equation

$$\dot{P}^{-1}(t) = A(t)P^{-1}(t) - P^{-1}(\dot{t})\bar{A} \qquad (I.18)$$

We will now discuss the system stability. From remark I-3 we know that the Floquet transformation of Theorem I-2 is a Lyapunov transformation. Chen [15] shows that under this class of transformations, the stability properties of the systems are preserved. Thus, if the system of Eqn. I.16 is bounded-input, bounded-output stable or if it is stable in sense of Lyapunov, then the system of Eqn. I.5 has the same stability properties. This fact facilitates the stability study of linear systems with periodic coefficients. By studying the stability of the transformed system, which has a constant system matrix, enough information is obtained about the behavior of the periodic coefficient system. For a practical example of such a study see Peters and Hohenemser [16].

## I.3. Algorithm Development

To carry out the Floquet transformation numerically, a fundamental matrix, F is needed. It can be obtained by numerical integration as follows:

$$\dot{F}(t) = A(t)F(t) \tag{I.19}$$

with initial conditions

$$F(0) = I \tag{I.20}$$

where the matrix I is the identity matrix. This assumption of initial conditions is not important for the transformation because F(t) is not unique, which results in nonuniqueness of the transformation matrix P(t). The specification of initial conditions simply serves to specify one fundamental matrix with the required properties. With the above assumption, Eqn. I.18 becomes:

$$F(T) = e^{\bar{A}T} \tag{I.21}$$

The next step is to find the eigen-system of F(T), assuming distinct eigenvalues. This gives the modal matrix M, and a diagonal matrix, L, with the eigenvalues, $a_j$, on the diagonal.

Using the similarity transformation yields

$$M^{-1}F(T)M = L \tag{I.22}$$

Now, given L, there exists a diagonal matrix $\bar{A}^*$ such that

$$L = e^{\bar{A}^*T} \tag{I.23}$$

and

$$Me^{\bar{A}^*T}M^{-1} = e^{\bar{A}T} \tag{I.24}$$

Consequently by integrating Eqn. I.19 from time zero, for one period, T, we can find $\overline{A}^*$.

Since $a_j$ are the eigenvalues of F(T) and $\overline{a}_j$ are the eigenvalues of $\overline{A}^*$ we have

$$\overline{a}_j = \frac{\ln(a_j)}{T} \qquad (I.25)$$

The eigenvalues $a_j$ and $\overline{a}_j$ are complex numbers in general. Thus the logorithm is not unique. The following procedures are used to find $\overline{a}_j$. Assume

$$a_j = c_j + ib_j = (c_j{}^2 + b_j{}^2)^{\frac{1}{2}}\ e^{i\ \tan^{-1}(\frac{b_j}{a_j})} \qquad (I.26)$$

and

$$\overline{a}_j = \overline{c}_j + i\overline{b}_j \qquad (I.27)$$

where $i = \sqrt{-1}$, and c, b, $\overline{c}$, and $\overline{b}$ are real numbers. Substituting Eqns. I.26 and I.27 in Eqn. I.25 yields

$$\overline{c}_j = \ln(c_j{}^2 + b_j{}^2)/2T \qquad (I.28)$$

$$\overline{b}_j = [\tan^{-1}(b_j/c_j)]/T \qquad (I.29)$$

Now, $\overline{c}_j$ is unique but $\overline{b}_j$ is not, which causes the matrix $\overline{A}^*$ to be nonunique. The principal branch value will be chosen so that

$$-\pi \leq \tan^{-1} b_j/c_j \leq +\pi$$

At this point, a transformation $P^*(t)$ is found which transforms the system of Eqn. I.5 into a system with $\overline{A}^*$ as its system matrix. From Eqn. I.13

$$P(t) = e^{\overline{A}\ t}\ F^{-1}(t) \qquad (I.30)$$

Introducing the transformation $P^*(t) = M^{-1}P(t)$, and using Eqn. I.24 in Eqn. I.30 yields

$$MP^*(t) = Me^{\bar{A}^* t}M^{-1}F^{-1}(t)$$

or $$P^*(t) = e^{\bar{A}^* t}M^{-1}F^{-1}(t) \qquad (I.31)$$

The following theorem can now be stated.

Theorem I-4: The transformation $P^*(t)$ as defined in Eqn. I.31, transforms the system of Eqn. I.5 into a system with $\bar{A}^*$ as its system matrix, when F(T) has a complete set of eigenvectors.

Proof: Can be easily carried out by direct substitution.

The transformation matrix $P^*(t)$ can be computed at any instant in time since F(t) can be found through numerical integration of Eqn. I.19. Thus its inverse can be calculated, and the matrix $\bar{A}^*$ is computed from the eigenvalues of F(T).

This algorithm was implemented on the Oregon State University computer system and worked successfully for some problems. However, failure occurred in certain cases; specifically when the system matrix was constant with eigenvalues of widely differing magnitudes, and a rather long period was assumed. In this case, numerical truncation errors were very important because the faster modes decayed to small values which were insignificant compared to the slower modes, and the monodromy matrix became ill conditioned. The same problem also occurs in cases where the system matrix is stiff and truly time varying.

To reduce this effect, modifications can be introduced in the above procedure. First, determine the average system matrix

$$A_o = \frac{1}{T}\int_o^T A(t)dt \qquad (I.32)$$

and define

$$A(t) = A_\circ + \delta A(t) \tag{I.33}$$

and $$F(t) = F_\circ(t) + \delta F(t) \tag{I.34}$$

where $F_\circ(t) = e^{A_\circ t}$

Then Eqn. I.19 can be written in the form

$$\dot{F}_\circ(t) + \delta\dot{F}(t) = (A_\circ + \delta A)(F_\circ(t) + \delta F(t)) \tag{I.35}$$

Since $$\dot{F}_\circ(t) = A_\circ F_\circ(t),\ F_\circ(0) = I \tag{I.36}$$

thus $$\begin{aligned} \delta\dot{F}(t) &= A(t)\delta F(t) + \delta A(t)F_\circ(t), \\ \delta F(0) &= 0 \end{aligned} \tag{I.37}$$

These equations are solved separately. To solve Eqn. I.36 the similarity transformation $F_\circ = M_\circ\overline{F}_\circ$ is used, where $M_\circ$ is the modal matrix of $A_\circ$. Assuming a diagonal matrix $L_\circ$ for eigenvalues of $A_\circ$, yields:

$$\dot{\overline{F}}_\circ = L_\circ\overline{F}_\circ(t),\ \overline{F}_\circ(0) = M_\circ^{-1} \tag{I.38}$$

which results in

$$F_\circ(t) = M_\circ e^{L_\circ t} M_\circ^{-1} \tag{I.39}$$

Eqn. I.37 can be solved numerically after substituting Eqn. I.39:

$$\delta\dot{F} = A(t)\ \delta F(t) + \delta A(t)M_\circ e^{L_\circ t}M_\circ^{-1},\ \delta F(0) = 0 \tag{I.40}$$

To find the transformation $P^*(t)$, note from Eqn. I.21 that

$$F(T) = F_\circ(T) + \delta F(T) = e^{\bar{A}T} \tag{I.41}$$

Now the eigensystem of $F(T)$, the modal matrix, M and the eigenvalue matrix, L can be found. From this point on the steps are taken as in the first version of the algorithm, namely:

$$\bar{A}^* = \frac{1}{T} \ln(L) \tag{I.42}$$

and

$$P^*(t) = [[F_\circ(t) + \delta F(t)]Me^{-\bar{A}^* t}]^{-1} \tag{I.43}$$

This procedure performed more favorably and was adopted to carry out the desired transformation of the system.

Theorem I.5: The transformation matrix $P^*(t)$ as given in Eqn. I.43 transforms the system of Eqn. I.5 into a system with $\bar{A}^*$ as its system matrix, assuming a complete set of eigenvectors for the monodromy matrix.

Proof: Use direct substitution.

To show the effect of the modification consider the following system (for complete analysis of the system see the Appendix A):

$$\dot{x} = \begin{bmatrix} -6+\cos\pi t & 5 \\ 5 & -6+\cos\pi t \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \tag{I.44}$$

$$y = [1 \quad 1]x$$

It is possible to find an analytical solution for this system. By doing so one finds that the constant system matrix has eigenvalues of (-11, -1) and the transformation matrix at the end of the period is

$$P^{*}(T) = \begin{bmatrix} \cdot 5 & \cdot 5 \\ -\cdot 5 & \cdot 5 \end{bmatrix} \qquad (I.45)$$

This system can be analyzed by the two algorithms already discussed. When the results of Floquet theory are directly used, the eigenvalues of the constant system matrix are found to be (-12.5780, -1.0000) and the transformation matrix at the end of the period is

$$P^{*}(T) = \begin{bmatrix} \cdot 50000 & \cdot 50000 \\ -\cdot 49998 & \cdot 49998 \end{bmatrix} \qquad (I.46)$$

The modified algorithm however finds the constant system matrix to have eigenvalues of (-10.8600, -1.0000) and the transformation matrix

$$P^{*}(T) = \begin{bmatrix} \cdot 50000 & \cdot 50000 \\ -\cdot 50000 & \cdot 50000 \end{bmatrix} \qquad (I.47)$$

Thus the improvement in accuracy of the modified algorithm is demonstrated.

# II. Power Spectral Density for Periodic Systems

Representing natural physical phenomena by random processes is a common practice in the scientific approach to modeling. In consequence, system models involving differential equations with random inputs have been under extensive study (for example see references [17, 18]). Many of these studies have involved linear systems with constant coefficients. However, the analysis of systems with time varying coefficients is somewhat more involved.

In this chapter, systems with time-varying, periodic coefficients and white noise input are considered. These two assumptions will prove helpful in obtaining statistical information about the output. It should be noted that the assumption of white noise input is not overly restrictive since many random processes can be derived from white noise through a linear filter [19].

Section 1 of this chapter gives some background about random processes. Section 2 introduces an algorithm to calculate the average power spectral density of the output of a system with periodic coefficients and white noise input. Section 3 gives a brief discussion on how the average power spectral density can be obtained from time series data.

## II-1. Mathematical Background

Some of the necessary background discussions and definitions for developing the algorithm are given here and closely follow those given by Papoulis [20] and Crandal [21].

Definition II-1: The autocorrelation function, $R_{xx}(t_1,t_2)$ of a complex random process x is defined as

$$R_{xx}(t_1,t_2) = E[x(t_1)x^{\dagger}(t_2)] \qquad \text{(II.1)}$$

where E is the ensemble average; and $x^{\dagger}(\cdot)$ is the conjugate transpose of $x(\cdot)$. Note that

$$R_{xx}(t_1,t_2) = R^{\dagger}_{xx}(t_2,t_1) \tag{II.2}$$

The cross correlation function of two complex random process is also defined in the same manner:

$$R_{xy}(t_1,t_2) = E[x(t_1)y^{\dagger}(t_2)] \tag{II.3}$$

Definition II-2: The double Fourier transform of the correlation functions is defined through a double integral

$$F_{xx}(f_1,f_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R_{xx}(t_1,t_2)e^{-if_1t_1}\, e^{-if_2t_2}\, dt_2dt_1 \tag{II.4}$$

It is trivial to show that

$$F_{xx}(f_1,f_2) = F_{xx}{}^{\dagger}(f_2,f_1). \tag{II.5}$$

The condition for existance of $F(f_1,f_2)$ is that the correlation be integrable over $[-\infty,\infty]$, which requires:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left|R_{xx}(t_1,t_2)\right| dt_1dt_2 < \infty \tag{II.6}$$

Singularities can exist in the form of

$\delta(f_1-a)$, $\delta(f_2-b)$, or $\delta(f_1-f_2-a+b)$

where a and b are real constants, and $\delta(\cdot)$ is the Dirac delta function.

In the case of a stationary process, the correlation function depends only on the time difference $\tau$, where $\tau = t_1-t_2$. Thus the time argument is suppressed

$$R_{xx}(\tau) = R_{xx}(t+\tau,t). \tag{II.7}$$

The Fourier tranform of $R_{xx}(\tau)$ is called the power spectral density and is denoted by $S_{xx}(\cdot)$, where

$$S_{xx}(f) = \int_{-\infty}^{\infty} R_{xx}(\tau)e^{-if\tau}d\tau \tag{II.8}$$

In this case, the double Fourier transform is singular and given by

$$F_{xx}(f_1,f_2) = 2\pi S_{xx}(f_1)\delta(f_1-f_2) \tag{II.9}$$

In order to simplify the representation of non-stationary random processes, an average autocorrelation function is introduced. When the correlation function is periodic, then

$$R(t_1,t_2) = R(t_1+T,t_2+T) \tag{II.10}$$

where T is the period.

Definition II-3: The average autocorrelation function, $\overline{R}(\tau)$ of the periodic autocorrelation $R(t+\tau,t)$ is defined as

$$\overline{R}(\tau) = \frac{1}{T}\int_{o}^{T} R(t+\tau,t)dt \tag{II.11}$$

The Fourier transform of this function is then called the average power spectral density (APSD). In the next section, the average power spectral density for periodic systems is formulated.

## II-2. Average Power Spectral Density of Periodic Systems

The state-space representation of the problem under consideration is of the form

$$\begin{aligned} \dot{x} &= A(t)x + B(t)w \\ y &= C(t)x \end{aligned} \tag{II.12}$$

where A, B and C are periodic with period T and w is the white noise input with power spectral density Q.

Wong [22] shows that x can be represented in the form

$$x(t) = \phi(t,t_\circ)x(t_\circ) + \int_{t_\circ}^{t} \phi(t,s)B(s)dW(s) \qquad \text{(II.13)}$$

where the transition matrix satisfies

$$\frac{\partial}{\partial t}\phi(t,t_\circ) = A(t)\phi(t,t_\circ),\ \phi(t_\circ,t_\circ) = I \qquad \text{(II.14)}$$

and W is the Brownian motion process which is defined formally by dW/dt = w. The output of the system can be expressed as

$$y = C(t)\phi(t,t_\circ)x(t_\circ) + C(t)\int_{t_\circ}^{t} \phi(t,s)B(s)dW(s) \quad \text{(II.15)}$$

To solve for the statistics of y, the equivalence transformation, $P^*(t)$ developed in the previous chapter is used. Thus, the system of Eqn. II.12 can be transformed into

$$\begin{aligned} \dot{x}^* &= A^*x^* + P^*B\,w \\ y^* &= CP^{-1*}x^* \end{aligned} \qquad \text{(II.16)}$$

where $A^*$ is constant and diagonal and

$$\begin{aligned} y^* = {} & C(t)P^{*-1}\phi^*(t,t_\circ)x^*(t_\circ) \\ & + CP^{*-1}(t)\int_{t_\circ}^{t} \phi^*(t,s)P^*(s)B(s)dW(s) \end{aligned} \qquad \text{(II.17)}$$

with

$$\phi^*(t,s) = e^{A^*(t-s)} \qquad \text{(II.18)}$$

It is now shown that the statistics of the system outputs under an equivalence transformation are invariant. First note that

$$\phi^{*}(t,t_{o}) = P^{*}(t)\phi(t,t_{o})P^{*-1}(t_{o}) \qquad (II.19)$$

The validity of Eqn. II.19 can be easily seen by substituting into Eqn. II.14 Rewriting Eqn. II.17 in terms of $x(t_{o})$ and moving $P^{*-1}(t)$ into the integral yields

$$y^{*} = C\, P^{*-1}(t)\phi^{*}(t,t_{o})P^{*}(t_{o})x(t_{o}) + C\int_{t_{o}}^{t} P^{*-1}(t)\phi^{*}(t,s)P^{*}(s)B(s)dW(s) \qquad (II.20)$$

By use of Eqn. II.19

$$y^{*} = C\phi(t,t_{o})x(t_{o}) + C\int_{t_{o}}^{t} \phi(t,s)B(s)dW_{s} = y \qquad (II.21)$$

Thus the output of the linear periodic system undergoing an equivalence transformation is invariant which implies that the statistics of outputs are also invariant.

The next step is to solve the covariance equation [23J to find the state covariance matrix, V, given by

$$\dot{V} = A^{*}V + VA^{*\dagger} + P^{*}BQ(P^{*}B)^{\dagger} \qquad (II.22)$$

We seek a periodic solution in Fourier series form

$$V(t) = \sum_{m} V_{m}e^{imf_{o}t} \qquad (II.23)$$

where $f_{o} = 2\pi/T$ the fundamental frequency of the system. Thus we expand the last term in Eqn. II.22 in its Fourier series expansion

$$P^*BQ(P^*B)^\dagger = \sum_m G_m\, e^{imf_\circ t} \tag{II.24}$$

substituting Eqns. II.23 and II.24 in Eqn. II.22

$$imf_\circ V_m = A^* V_m + V_m A^{*\dagger} + G_m \tag{II.25}$$

Now by expressing matrices V and G in terms of their components, $V_{jk,m}$ and $G_{jk,m}$ and noting that $A^*$ is diagonal with eigenvalues $a_j^*$ we can find each component of the matrix $V_m$

$$V_{jk,m} = \frac{G_{jk,m}}{imf_\circ - a_j^* - a_k^{*\dagger}} \tag{II.26}$$

At this point the autocorrelation function of the state, x, can be determined. Bryson [23] gives $R_{xx}$ by the following equations.

$$R^1_{x^*x^*}\ (t+\tau,t) = \phi(\tau)V(t) \tag{II.26a}$$

$$\tau \geq o$$

$$R^2_{x^*x^*}\ (t,t+\tau) = V(t)\phi^\dagger(\tau) \tag{II.26b}$$

Superscript 1 and 2 are for distinction of forward or backward correlation in time, respectively. The transition matrix $\phi(\cdot)$ of the system is given by

$$\phi(t) = e^{A^* t} \tag{II.27}$$

Using the Fourier expansion of V(t), Eqns. II.26 can be rewritten, so that

$$R^1_{x^*x^*} = \sum_m e^{A^*\tau}\, V_m\, e^{imf_\circ t} \tag{II.28a}$$

$$R^{2}_{x^*x^*} = \sum_{m} V_m \, e^{imf_\circ t} \, e^{A^{*}\tau^{\dagger}} \qquad \text{(II.28b)}$$

Having $R_{x^*x^*}$ gives the output autocorrelation matrix $R_{yy}$ since

$$y = C\,P^{*-1}\,x^{*}$$

We express the product $CP^{*-1}$ in the Fourier series expansion

$$CP^{*-1}(t) = \sum_{m} H_m \, e^{imf_\circ t} \qquad \text{(II.29)}$$

Therefore

$$y = \sum_{m} H_m \, e^{imf_\circ t} \, x^* \qquad \text{(II.30)}$$

and the autocorrelation of y using Eqn. II.30 is

$$R_{yy}(t+\tau,t) = \sum_{k,m} E[H_k \, e^{ikf_\circ t(t+\tau)} \, x^*(t+\tau)\cdot x^{*\dagger}(t) e^{-imf_\circ t} \, H^{\dagger}_{m}]$$

$$= \sum_{k,m} H_k \, e^{ikf_\circ(t+\tau)} \, R_{x^*x^*} \, e^{-imf_\circ t} \, H^{\dagger}_{m} \qquad \text{(II.31)}$$

using Eqns. II.28a, b, for $R_{x^*x^*}$, yields similar expressions for $R_{yy}$ as follows:

$$R^{1}_{yy} = \sum_{j,k,m} H_k \, e^{[A^{*}+ikf_\circ I]\tau} \, V_m \, H^{\dagger}_{j} \, e^{if_\circ(m+k-j)t} \qquad \text{(II.32a)}$$

$$R^{2}_{yy} = \sum_{j,k,m} H_k V_m \, e^{[A^{*\dagger}-ijf_\circ I]\tau} \, H^{\dagger}_{j} \, e^{if_\circ(k-j+m)t} \qquad \text{(II.32b)}$$

where I is the identity matrix, and $\tau \geq o$. Now, averaging over one period, the average, autocorrelation function as defined in the last section can be written as

$$\bar{R}^1_{yy}(\tau) = \sum_{j,k} H_k \, e^{[A^* + if_\circ kI]\tau} \, V_{j-k} \, H^\dagger_j \tag{II.33a}$$

$$\bar{R}^2_{yy}(-\tau) = \sum_{j,k} H_k \, V_{j-k} \, e^{[A^{*\dagger} - if_\circ jI]\tau} \, H^\dagger_j \tag{II.33b}$$

Now, note that the average power spectral density is the Fourier transform of the average autocorrelation function, and can easily shown to be

$$\bar{S}_{yy}(f) = \sum_{j,k} H_k[A^* + I(kf_\circ - f)i]^{-1} \, G_{j-k}[A^{*\dagger} - I(jf_\circ - f\,)i]^{-1} H^\dagger_j \tag{II.34}$$

Eqn. II.34 thus gives a direct method for computing the average power spectral density. This method was coded in FORTRAN and implemented on the Oregon State University computer system. The program listing is given in Appendix A.

At this point we find the relationship between $F_{yy}(f_1,f_2)$, the double Fourier transform of $R_{yy}$, and the average power spectrum $\bar{S}_{yy}(f)$. For this purpose we rewrite Eqn. II.32a noting

$$t_1 - t_2 = \tau, \quad t_1 = t + \tau, \; t_2 = t$$

and Eqn. II.32b noting

$$t_2 - t_1 = \tau, \; t_1 = t, \; t_2 = t + \tau$$

Thus,

$$R^1_{yy}\,(t_1,t_2) = \sum_{j,k,m} H_j\, e^{[A^* + ijf_\circ I](t_1 - t_2)}$$

$$e^{if_\circ(m + j - k)t_2}\, V_m H_k^\dagger \qquad \text{(II.35a)}$$

$$R^2_{yy}(t_1,t_2) = \sum_{j,k,m} H_j V_m\, e^{[A^* - ikf_\circ I](t_2 - t_1)}$$

$$e^{if_\circ(m + j - k)t_2}\; H_k^\dagger \qquad \text{(II.35b)}$$

or

$$R^1_{yy}(t_1,t_2) = \sum_{j,k,m} H_j\, e^{[A^* + ijf_\circ I]t_1}$$

$$e^{[-A^* + if_\circ(m-k)I]t_2}\; V_m H_k^\dagger \qquad \text{(II.36a)}$$

$$R^2_{yy}(t_1,t_2) = \sum_{j,k,m} H_j\, V_m\, e^{[A^* - kf_\circ I\ t_2}$$

$$e^{[-A^* + if_\circ(m + j)I]t_1}\; H_k^\dagger \qquad \text{(II.36b)}$$

Now, $F_{yy}(f_1,f_2)$ is the double Fourier transform of $R_{yy}$, and can be found by using $R^1_{yy}$ and $R^2_{yy}$, which gives

$$F_{yy}(f_1,f_2) = \sum_{j,k,m} H_j[-A^*+i(jf_\circ-f_1)I]^{-1}\delta[\ (f_\circ(j+m-k)-f_1+f_2)]V_m H_k^\dagger$$

$$+\, H_j V_m[-A^*+i(f_\circ(m+j)-f_1)I]^{-1}\delta[\ (f_\circ(m+j-k)-f_1+f_2)]H_k^\dagger \qquad \text{(II.37)}$$

This equation can be rearranged to get a more compact form

$$F_{yy}(f_1,f_2) = \sum_{j,k,m} H_j[-A^*-i(jf_\circ-f_1)I]V_m H_k^\dagger + \tag{II.38}$$

$$H_j \; V_m[-A^*+i(kf_\circ-f_2)I)]H_k^\dagger \; \delta[\;(f_\circ(m+j-k)-f_1+f_2]$$

The coefficient of the delta function when $f_1 = f_2$ reduces to the average power spectral density given by equation II.34. Thus, the average power spectral density can be interpreted as describing the statistics of the "stationary part" of the system output.

## II-3. Time Series and Average Power Spectral Density

In practice, information about a random variable x is given by its time series, $x(t_0)$, $x(t_1)$ ... $x(t_{n-1})$ where $t_0$, $t_1$, ..., $t_{n-1}$ are the n sampling points in time. An approximation for the power spectral density, the periodogram, is defined [24] using the time series as

$$U(f) = \frac{1}{2\pi n} \sum_{j,k=o}^{n-1} x(t_j)x(t_k)\exp[-i(t_j-t_k)f] \tag{II.39}$$

This equation can be rearranged assuming a constant sampling interval $\tau$ to give

$$U(f) = \frac{1}{2\pi n} \sum_{k=-(n-1)}^{n-1} \sum_{j=o}^{n-1} x(t_j+k\tau)x(t_j)\exp[-ifk\tau] \tag{II.40}$$

The ensemble average of both sides of Eqn. II.40 is now taken so that

$$E[U(f)] = \frac{1}{2\pi n} \sum_{k=-(n-1)}^{n-1} \sum_{j=o}^{n-1} R_{xx}(t_j+k\tau,t_j)\exp[-ifk\tau] \tag{II.41}$$

where

$$R_{xx}(t_1,t_2) = E[x(t_1)x(t_2)]$$

Now, if the autocorrelation function $R_{xx}$ is periodic and the total data time interval is an integer multiple, N, of the period, it follows that

$$\frac{1}{m}\sum_{j=o}^{n-1} R(t_j+\tau,t_j) = \frac{1}{n}\sum_{m=o}^{N-1}\sum_{k=o}^{\frac{n}{N}-1} R(t_m+t_k+\tau,\ t_m+t_k)$$

$$= \frac{1}{N}\sum_{m=o}^{N-1} \frac{N}{n}\sum_{k=o}^{\frac{n}{N}-1} R(t_m+t_k+\tau,\ t_m,t_k)$$

$$= \frac{1}{N}\sum_{m=o}^{N-1} \overline{R}(\tau) = \overline{R}(\tau) \qquad \text{(II.42)}$$

Inserting this result in Eqn. II.41, yields

$$E[U(f)] = \frac{1}{2\pi}\sum_{k=-(n-1)}^{n-1} \overline{R}(k\tau)\exp[-ifk\tau] \qquad \text{(II.43)}$$

In the limit as $n\to\infty$ and $\tau\to o$, the right side of Eqn. II.43 becomes the average power spectral density defined previously. Thus, the ensemble average of the periodogram converges to the average power spectral density as the data record becomes long. This result paves the way for the use of all the standard techniques for smoothing the periodogram [25] to give a better estimate of the average power spectral density.

# III. Examples of Periodic Systems

The recent growing interest in wind turbines in the energy field has motivated the development of advanced structural models for these systems. The more complicated models usually contain periodic time dependent coefficients. This time dependency and the fact that the input of the system is the wind, a random process in nature, causes the practical use of these models to be difficult.

In Sections 1 and 2 of this chapter, two systems are considered. The first system is a rotating, rigid blade with one flapping degree of freedom. This model is represented by a second order differential equation. The wind input is modeled by three first order constant coefficient differential equations. The state space representation of the overall system is, therefore, of fifth order. The second model is a mathematical representation of a 2.5 MW, three-bladed wind turbine. This model considers five degrees of freedom, and the state space form is of order nineteen.

In Section 3, the results and conclusions are discussed and some suggestions for future study are made.

## III-1. Rotating Rigid Wind Turbine Blade with Flapping

In this section, the equation of motion is developed for a single, rigid wind turbine blade, hinged to a massless hub, and restrained by means of a rotational spring. Figure 1 shows the geometry under consideration. The blade is rotating with constant rate of rotation, $f_0$, about the y axis of the xyz inertial reference frame. The flapping motion is about the x' axis of the x'y'z'

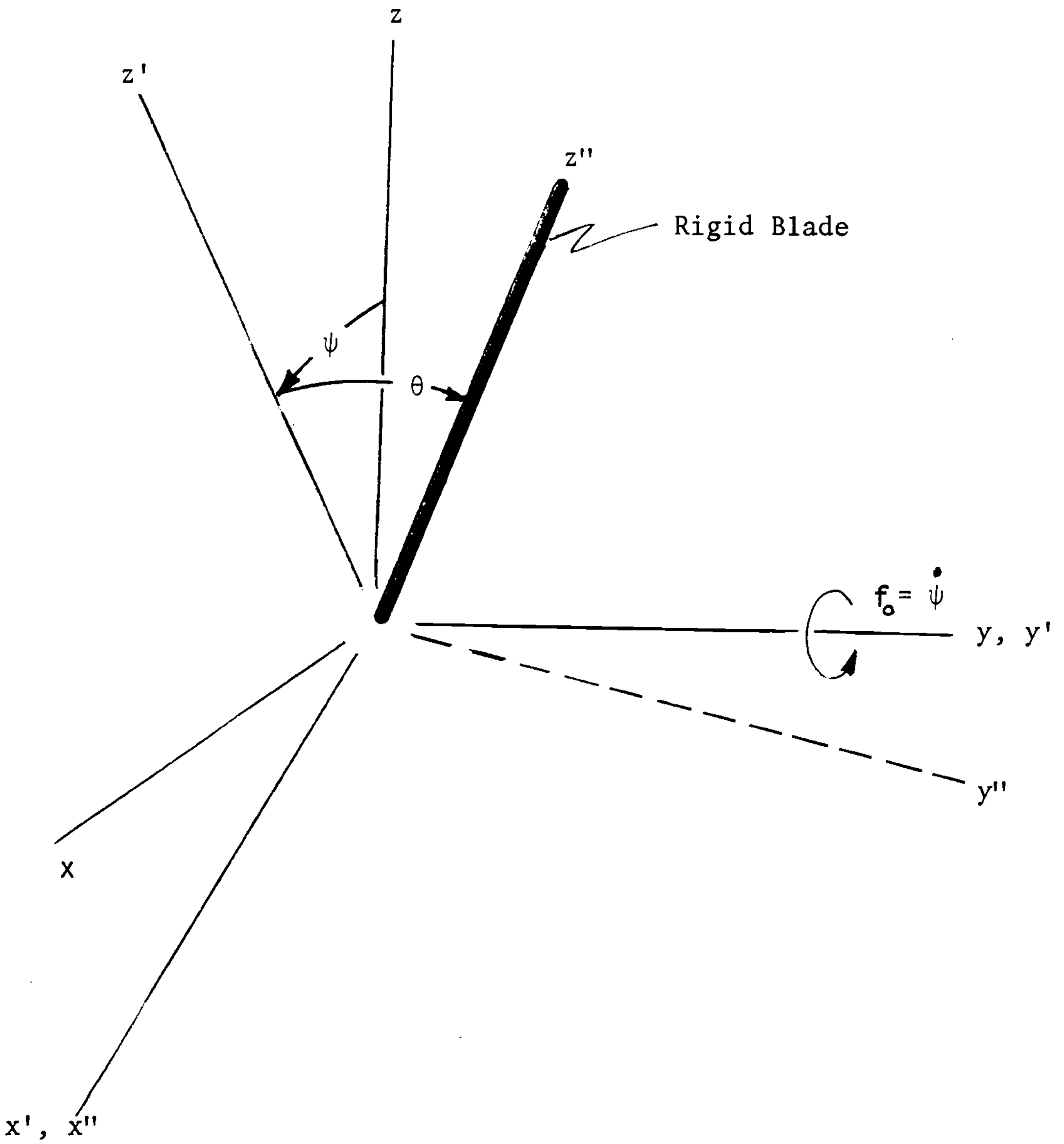


Fig. 1 - Rigid Wind Turbine Blade with Constant Rotation Rate and Small Flapping Angle.

rotating reference frame. The x'y'z' reference frame rotates about y axis with y' and y coincident. The flapping angle, θ, is restrained by a spring with constant stiffness k. The reference frame x"y"z" is fixed to the blade, with x' and x" coincident.

To obtain the equation of motion, the kinetic and potential energy are written in the blade fixed x"y"z" coordinate frame. Thus, the angular velocity and moment of inertia after appropriate transformation are given by

$$\omega'' = \begin{Bmatrix} -\dot{\theta} \\ f_\circ \cos\theta \\ f_\circ \sin\theta \end{Bmatrix} \qquad I'' = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{(III.1)}$$

The kinetic energy, T, is given by

$$T = 1/2\ \omega''^{T}\ I''\ \omega''$$

or substituting for ω" and I" from Eqn. III.1

$$T = 1/2\ (I\ \dot{\theta}^2 + If_\circ^2 \cos^2\theta) \qquad \text{(III.2)}$$

The spring poteneial energy, V, is given by

$$V = 1/2\ k\ \theta^2 \qquad \text{(III.3)}$$

where k is the rotational spring stiffness. Note that the gravitational potential energy of the rotating mass of the blade is neglected.

At this point, Lagrange's equation is used to obtain the equation of motion. Lagrange's equation is given by

$$\frac{d}{dt}\left(\frac{\partial T}{\partial q}\right) - \frac{\partial T}{\partial q} + \frac{\partial V}{\partial q} = Q \qquad \text{(III.4)}$$

where q and Q are the generalized coordinate and generalized force respectively. The coordinate in this case is $\theta$.

Substituting for T and V from Eqns. II.2 and III.3 into Eqn. III.4 results in

$$I\,\ddot{\theta} + (k + I\,f_\circ^2)\,\theta = Q_\theta \qquad \text{(III.5)}$$

where the angle $\theta$ is assumed small and second order terms are neglected.

The generalized force $Q_\theta$ is a moment acting on the blade about the x' axis. This moment is due to atmospheric turbulence. The following discussion for finding $Q_\theta$ follows the work by Holley, et al. [26]

Assuming a cubic aerodynamic force distribution, the force on the rigid blade is given approximately by

$$f_y = \frac{3C}{R^3} \int_\circ^R r(R^2 - r^2)\ (V - r\dot{\theta})dr \qquad \text{(III.6)}$$

where

$f_y$ is the net blade force in y direction

r is the position along the blade

R is the length of the blade

C is a constant aerodynamic parameter

and V is the turbulent velocity of the wind in the vicinity of the rotor disc and is approximated by

$$V(r,\psi,t) = V_y(t) + V_{y,x}(t)\ r \sin\psi + V_{y,z}(t)\ r \cos\psi \qquad \text{(III.7)}$$

The term $V_y$ is the uniform turbulent velocity and the terms $V_{y,x}$ and $V_{y,z}$ are the velocity gradients across the rotor disc. All three of these terms are random. The generalized force $Q_\theta$

in terms of f is given by integrating the relation

$$dQ_\theta = rdf_y$$

so that

$$Q_\theta = \frac{3C}{R^3} \int_o^R r^2(R^2 - r^2)(V - r\dot{\theta})dr \qquad \text{(III.8)}$$

Carrying out the integration and substituting for V from Eqn. III.7 gives

$$Q_\theta = \frac{CR}{I} \left(\frac{3\pi}{16} V_y + \frac{2R}{5} V_{y,x} \sin f_\circ t + \frac{2R}{5} V_{y,z} \cos f_\circ t\right) - \frac{2CR^2}{5I} \dot{\theta}$$

Using this expression for $Q_\theta$ in Eqn. III.5 yields

$$\ddot{\theta} + \frac{2R^2C}{5I} \dot{\theta} + \left(\frac{k}{I} + f_o^2\right)\theta = \frac{CR}{I} \left(\frac{3\pi}{16} V_y + \frac{2R}{5} V_{y,x} \sin f_\circ t + \frac{2R}{5} V_{y,z} \cos f_\circ t\right) \qquad \text{(III.9)}$$

Holley, et al, also give a procedure to approximate the random processes $V_y$, $V_{y,x}$, $V_{y,z}$ by linear differential equations with white noise input. These relations are given by

$$\begin{aligned} \dot{V}_y &= -a_1V_y + b_1w_1 \\ \dot{V}_{y,x} &= -a_2V_{y,x} + b_2w_2 \qquad \text{(III.10)} \\ \dot{V}_{y,z} &= -a_3V_{y,z} + b_3w_3 \end{aligned}$$

for the values of $a_i$ and $b_i$, see Appendix B.

To obtain the state space form of the system model, the following state variables are defined:

$$\begin{aligned} x_1 &= \theta & x_3 &= V_y \\ x_2 &= \dot{\theta} & x_4 &= V_{y,x} \qquad \text{(III.11)} \\ & & x_5 &= V_{y,z} \end{aligned}$$

Now, from Eqn. III.9 and Eqn. III.10, the following matrix form results:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -(\frac{k}{I}+f_{o}^{2}) & -\frac{2CR^{2}}{5I} & \frac{3\pi CR}{16\,I} & \frac{2CR^{2}\sin f_{o}t}{5\;I} & \frac{2CR^{2}\cos f_{o}t}{5\;I} \\ 0 & 0 & -a_{1} & 0 & 0 \\ 0 & 0 & 0 & -a_{2} & 0 \\ 0 & 0 & 0 & 0 & -a_{3} \end{bmatrix} x$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_{1} & 0 & 0 \\ 0 & b_{2} & 0 \\ 0 & 0 & b_{3} \end{bmatrix} \begin{Bmatrix} w_{1} \\ w_{2} \\ w_{3} \end{Bmatrix} \qquad \text{(III.12)}$$

For the output, we define

$$y = [1\ 0\ 0\ 0\ 0] \begin{Bmatrix} x_{1} \\ x_{2} \\ x_{3} \\ x_{4} \\ x_{5} \end{Bmatrix} = \theta \qquad \text{(III.13)}$$

Thus, the model has periodic coefficients in the second row of the system dynamics matrix.

To test the algorithm, another set of state variables is chosen to give a model with constant coefficients. To do this, define the state variables

$$\begin{aligned} x_1 &= \theta & x_3 &= V_y \\ x_2 &= \theta & x_4 &= \cos f_\circ t V_{y,z} + \sin f_\circ t V_{y,x} \\ & & x_5 &= -\sin f_\circ t V_{y,z} + \cos f_\circ t V_{y,x} \end{aligned} \qquad \text{(III.14)}$$

which will result in the following constant coefficient state space form

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{k+If_\circ^2}{I} & -\frac{2CR^2}{5\ I} & \frac{3C\pi R}{16\ I} & \frac{2R^2C}{5\ I} & 0 \\ 0 & 0 & -a_1 & 0 & 0 \\ 0 & 0 & 0 & -a_2 & f_\circ \\ 0 & 0 & 0 & -f_\circ & -a_2 \end{bmatrix} x$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \begin{Bmatrix} w_1' \\ w_2' \\ w_3' \end{Bmatrix} \qquad \text{(III.15)}$$

and $y = [1\ 0\ 0\ 0\ 0]x$ (III.16)

where $w_1'$, $w_2'$, and $w_3'$ are white noise excitations with the same power spectral densities as $w_1$, $w_2$, and $w_3$. The numerical values of the parameters and matrices are given in Appendix B.

These two models were both analyzed using the algorithm developed in the previous chapters. The output power spectral densities for both models are the same (as expected) and shown in Fig. 2. The results show that the algorithm performs satisfactorily. .

## III-2. A Five Degree of Freedom Wind Turbine Model

Thresher, et al [27] use a five degree of freedom model to study a 2.5 MW wind turbine system called the MOD-G, Fig. 3. In their model, the tower is a single cantilever beam element. The three blades are rigid and fixed to the hub. The hub is assumed to be connected with a flexible shaft to a synchronous generator which turns at a constant speed.

Five degrees of freedom are assumed; two displacements at the top of the tower, U and V in the x and y directions, respectively, and two rotations $\phi$ and $\chi$ the pitch and yaw for the axis of rotation of the hub. This axis is assumed to be rigidly connected to the top of the tower. The fifth degree of freedom is the fluctuation $\dot{\psi}$ in the rate of rotation of the rotor. The steady component of this rotation is $f_o$. When the model is represented in state space form and the input random processes are derived from white noise, a nineteenth order model results. The nineteen state variables consist of the five positions and five velocities and the nine additional states given by Thresher, et al, for approximating the wind input from white noise.

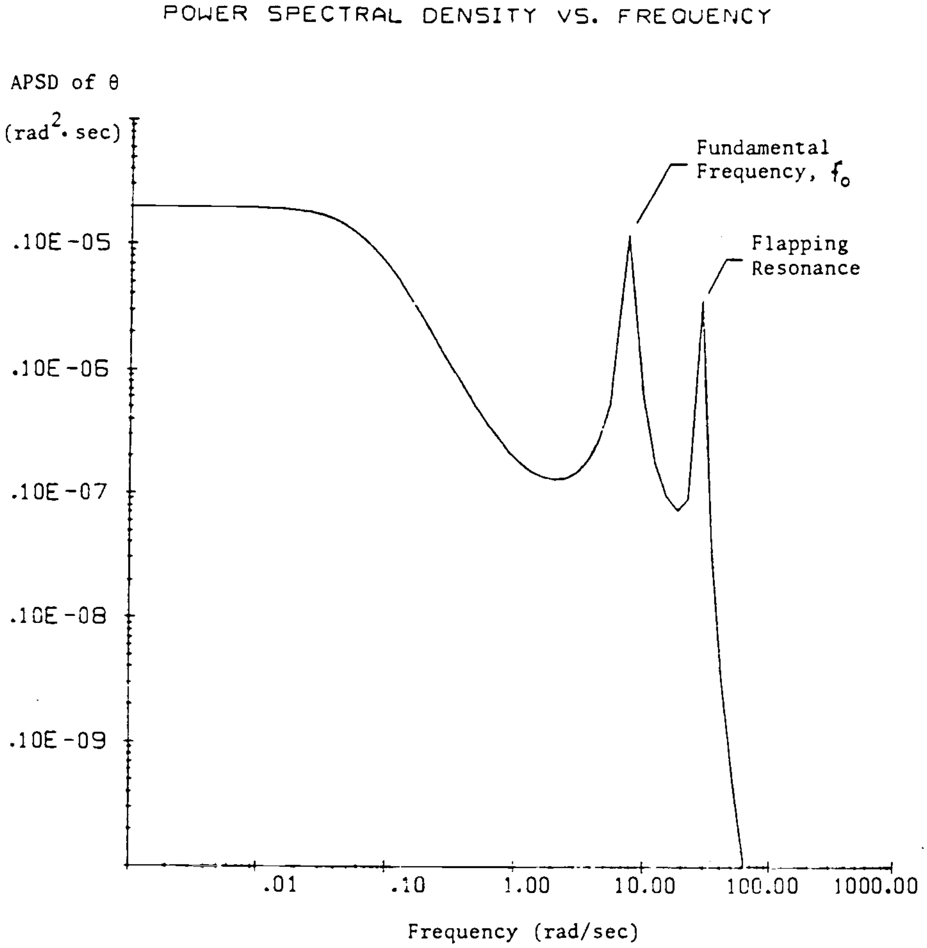


Fig. 2 - Average Power Spectral Density of Flapping Response of a Single Wind Turbine Blade.

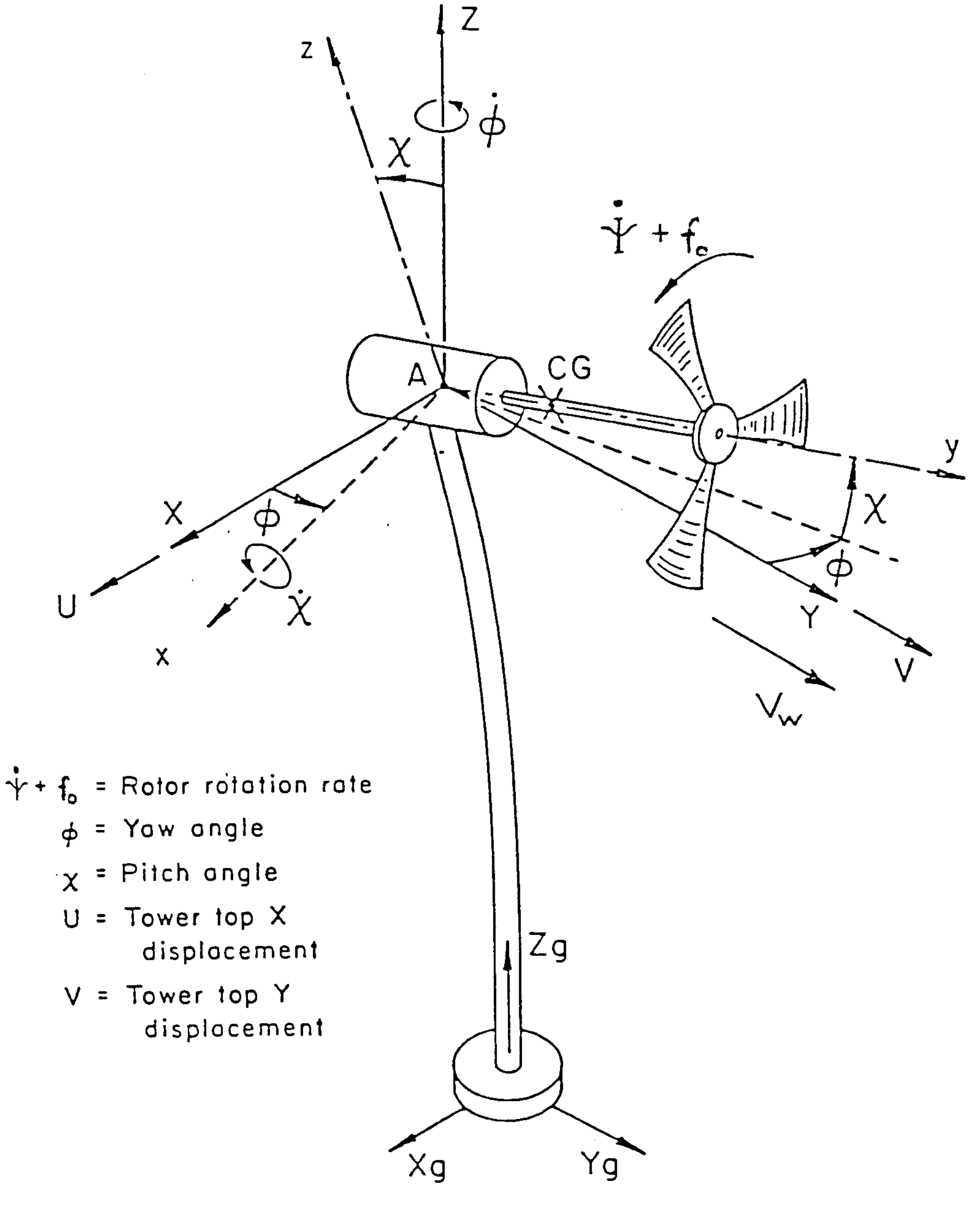


Fig. 3 - Wind Turbine Model with Five Degrees of Freedom

In their report, Thresher, et al, chose the state variables in a manner to give a constant coefficient model. In this work, however, the states are chosen so that the coefficient matrix has periodic terms. For a discussion of the state variables and the coefficient matrices, see Appendix B.

The power spectral density found by Thresher, et al, is shown in Fig. 4 where, for some frequencies, the APSD, calculated using the algorithm developed here, is shown. It should be noted that considerably more computation is required to handle the periodic coefficients; thus, only a few points were computed to verify the accuracy of the algorithm.

## III-3. Conclusions

The algorithm developed in this thesis is a useful tool to find the average power spectral density for systems which can be represented by periodic coefficients in the state space form. The algorithm is also capable of finding the power spectral density for constant coefficient systems.

The numerical modification in the use of Floquet theory results in a more accurate computation of the equivalence transformation. It was shown that the equivalence transformation does not affect the statistics of the output of the system. The average power spectral density was defined and shown to be the same as the coefficient of the Fourier transform of the autocorrelation function when both frequencies are the same. Thus, the average power spectral density represents the stationary part

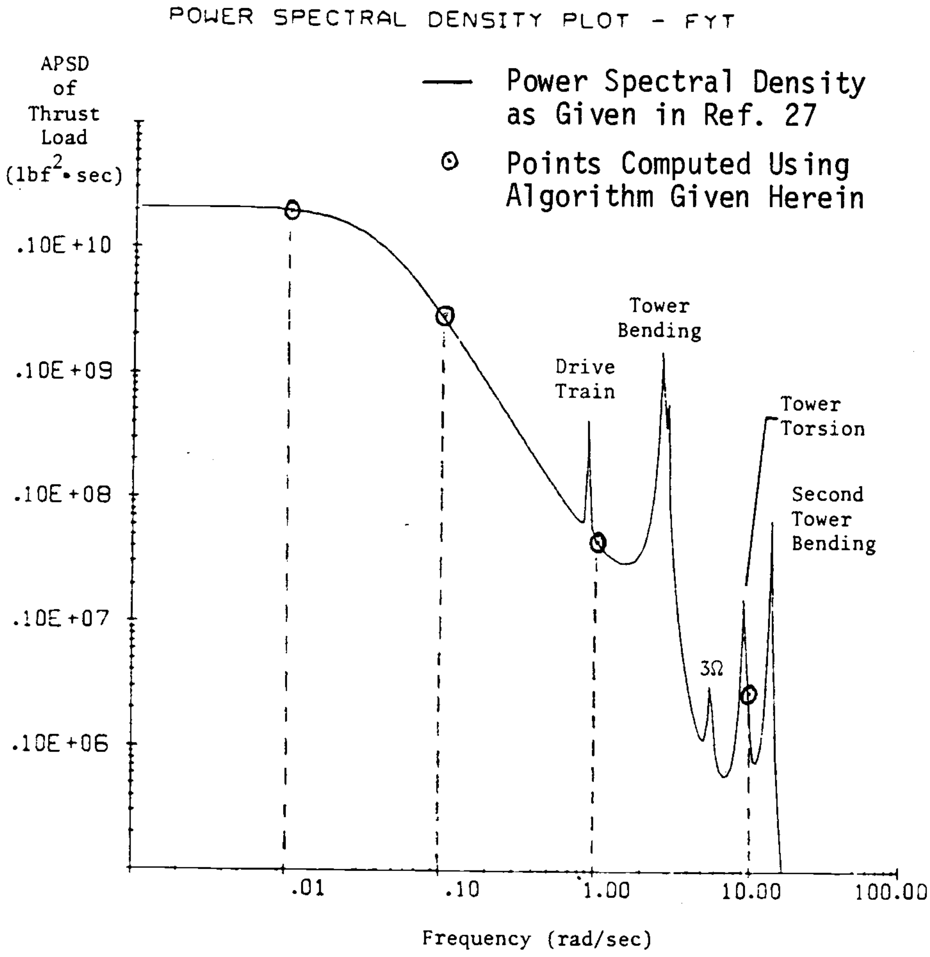


Fig. 4 - Average Power Spectral Density of Thrust Response of a 2.5 MW Wind Turbine System.

of the output statistics and, in the case of stationary systems, is the same as power spectral density. When the system is non-stationary, the average power spectrum provides useful information about the statistical frequency content of the response, and it can also be estimated experimentally using the standard procedures of time series analysis.

## III-4. Suggestions for Future Work

The continuation of this work to find the probability of level crossings and the occurence of maxima will provide a useful tool in designing systems with periodic coefficients for improved fatigue performance.

In the time series analysis aspect of this work, some experimental data should be tested against the numerical results. Also, the derivation of more accurate approximations for the average power spectral density seems in order.

# References

1. Yakubovich, V. A. and Starzhinskii, V. M., Linear Differential Equations with Periodic Coefficients, John Wiley & Sons, 1975, Vol. 1 & 2.

2. Chen, C. T., Introduction to Linear System Theory, Hold, Rinehart and Winston, Inc., 1970, pp. 143-144.

3. Padulo, L., and Arbib, M. A., System Theory, Hemisphere Publishing Corp., 1974, pp. 370-374.

4. Halany, A., Differential Equations, Academic Press, 1966, pp. 402-406.

5. Chen, (1970), loc. cit.

6. Padulo, (1974), op. cit.

7. Meirovitch, L., Methods of Analytical Dynamics, McGraw-Hill Book Co., Inc., 1970, pp. 263-266.

8. Wiberg, D. M., State Space and Linear System, McGraw-Hill Book Co., Inc., 1971, pp. 147-155.

9. Chen, (1970), op. cit.

10. Struble, R. A., Nonlinear Differential Equations, McGraw-Hill Book Co., Inc., 1952, pp. 106-110.

11. Zadeh, L. A. and Desor, C. A., Linear System Theory, McGraw-Hill Book Co., Inc., 1963, pp. 110-115.

12. Padulo, (1974), op. cit.

13. Chen, (1970), op. cit.

14. Brauer, F. and Noheh, J. A., Qualitative Theory of Ordinary Differential Equations, W. A. Benjamin Inc., 1969, pp. 96-100.

15. Chen, (1970), op. cit.

16. Peters, D. A. and Hohenemser, K. H., "Application of the Floquet Transition Matrix to Problems of Lifting Rotor Stability", J. American Helicopter Society, No. 3, Vol. 5, Mar. 1971, pp. 25-33.

17. Soong, T. T., Random Differential Equations in Science and Engineering, Academic Press, 1973, pp. 116-218.

18. Srinivasan, S. K. and Vasvdevan, R., Introduction to Random Differential Equations and Their Applications, American Elsevier Publishing Co., Inc., 1971, pp. 1,100.

19. Kwakernaak, H. and Sivan, R., Linear Optimal Control Systems, John Wiley & Sons, Inc., 1972, pp. 105-107.

20. Papoulis, A., Probability, Random Variables, and Stochastic Process, McGraw-Hill Book Co., Inc., 1965, pp. 430-515.

21. Crandal, S. H. and Mark, W. D., Random Vibration in Mechanical Systems, Academic Press, 1973, pp. 1,102.

22. Wong, E., Stochastic Process in Information and Dynamical Systems, McGraw-Hill Book Co., Inc., 1971, pp. 50-150.

23. Bryson, A. E. and Ho, Y. C., Applied Optimal Control, Hemisphere Publishing Corporation, 1975, pp. 293-400.

24. Bloomfield, P., Fourier Analysis of Time Series: An Introduction, John Wiley & Sons, 1976, pp. 200-210.

25. Bloomfield (1976), loc. cit.

26. Holley, W. E., Thresher, R. W., Lin, S. R., Wind Turbulance Inputs for Horizontal Axis Turbines, Department of Mechanical Engineering, Oregon State University, 1981.

27. Thresher, R. W., and Holley, W. E., The Response Sensitivity of Wind Turbines to Atmospheric Turbulance, Department of Mechanical Engineering, Oregon State University.

# Appendices

# Appendix A

## Computer Program

The computer programs were developed based on the algorithms given in Chapter I and II. Two programs were written, PROGRAM FOURIER and PROGRAM POWSPEC.

### PROGRAM FOURIER

This program finds the Floquet transformation and all the Fourier Coefficients described in Chapter II. This program uses the following routines:

<u>SUBROUTINE SIMAT:</u> This routine performs integration of a matrix differential equation. By using IMSL routine DVERK and subroutine FCN, SIMAT is called by FOURIER

<u>SUBROUTINE FCN:</u> This routine provides the differential equation to be integrated. It is called by SIMAT.

<u>SUBROUTINE EXPAS:</u> Calculate EXP(A*T) where A is diagonal matrix represented by a vector, and T is a scaler variable. This routine is called by FOURIER, SIMAT and FCN.

<u>SUBROUTINE RPRINT:</u> This routine prints NXN real matrices, eleven columns at a time.

<u>SUBROUTINE CPRINI:</u> This routine prints NXN complex matrices, eleven columns at a time, with real parts above imaginary parts.

User should provide the following routines:

<u>SUBROUTINE SYSMAT:</u> This routine should provide the time varying elements of matrix A.

SUBROUTINE BMATRIX: This routine should provide the time varying elements of matrix B.

SUBROUTINE CMATRIX: This routine should provide the time varying elements of matrix C.

INPUT OF PROGRAM FOURIER:

First card: Name of data file (see below). This name should contain less than seven letters.

Second card: User's choice of output, 0, 1 and 2
0 = no printout
1 = partial printout
2 = complete printout

Third card: Number of harmonics desired in Fourier expansion. This number should be less than five.

Fourth card: Accepted termination error for integration routine SIMAT.

Fifth card: The program chooses the number of time segments in one period. Zero on this card would mean the number is satisfactory. Otherwise the desired choice is put on this card. The program calculates this number by choosing the maximum of

$2M$, $M$ = number of Fourier harmonics

or $(T*F)/\pi+1$, $T$ = period
$F$ = The largest imaginary part of the eigenvalues of the average of matrix AC (Averaged part of A)

DATA FILE: This file should be presented to the program with its name given on first card:

Structure of DATA FILE:

First Line should contain:

N = order of system  
NSNPT = No. of inputs  
NOTPT = No. of outputs  
OMGA = fundamental frequency  
ITV = control variable: 0 if there is a time dependent element  
in A, B and C matrices  
1 otherwise

From the second line on the matrices AC, $\phi$, BC, CC (AC, BC and CC are averaged, A, B, and C respectively) should be given line by line.

OUTPUT OF PROGRAM FOURIER

All the input information is printed on file DEDUGF. The fundamental matrix, matric P* and its inverse are printed out at each time segment. All the Fourier coefficients are printed on the same file. All the necessary information for program POWSPEC is output on file CONECTN.

## PROGRAM POWSPEC

This program calculates the Average Power Spectral Density (APSD) and plots the results. COMPLOT package should be provided. The following subroutines are called by this program.

SUBROUTINE CAL: This routine calculates the APSD for a given frequency.

SUBROUTINE PLOTS: This routine plots the calculated APSD's versus the respective frequencies.

## INPUT OF PROGRAM POWSPEC

This program is to be used interactively. See the example and the program for inputs. File CONECTN generated by PROGRAM FOURIER should be provided.

## OUTPUT OF PROGRAM POWSPEC

All ASPD's and their respected frequencies are written on fill DEBUGP. The plot of the ASPD is provided if the user is using a Tektronics 4014 terminal.

In the following pages a second order example is solved and the programs are listed.

PROCEDURE FILE TO CALL AND EXECUTE PROGRAM 'FOURIER':

```
USER,,.
CHARGE,,.
TITLE.  MOSEN
SETTL,100.
ATTACH,IMSL/UN=LIBRARY.
LIBRARY,IMSL.
GET,LGO=FORIERB,SYSB=SYS2TB,DAT2T.
LOAD,LGO,SYSB.
EXECUTE.
REWIND,DEBUGF.
SAVE,DEBUGF=BEBUG2.
SAVE,CONECTN.
```

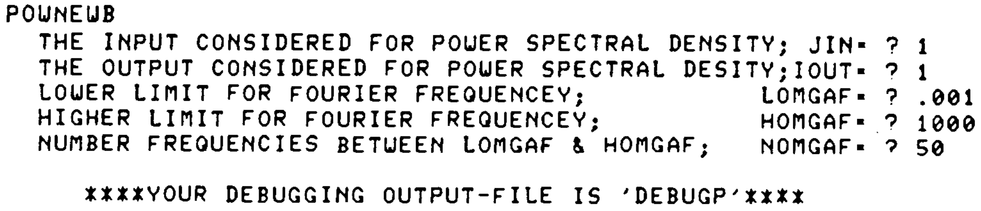
INFORMATION PROVIDED BY CARDS FOR PROGRAM FOURIER:

```
'DAT2T'
2
4
.001
8
```

INFORMATION PROVIDED FO  PROGRAM FOURIER ON DATA-FILE 'DAT2T':

```
2. 1. 1. 3.14 1
-6. 5.
5. -6.
1. 0.
0. 1.
1.
1.
1. 1.
```

INTERACTIVE EXECUTION OF PROGRAM 'POWSPEC' :



```
POWNEWB
  THE INPUT CONSIDERED FOR POWER SPECTRAL DENSITY; JIN= ? 1
  THE OUTPUT CONSIDERED FOR POWER SPECTRAL DESITY;IOUT= ? 1
  LOWER LIMIT FOR FOURIER FREQUENCEY;            LOMGAF= ? .001
  HIGHER LIMIT FOR FOURIER FREQUENCEY;           HOMGAF= ? 1000
  NUMBER FREQUENCIES BETWEEN LOMGAF & HOMGAF;    NOMGAF= ? 50

     ****YOUR DEBUGGING OUTPUT-FILE IS 'DEBUGP'****
```

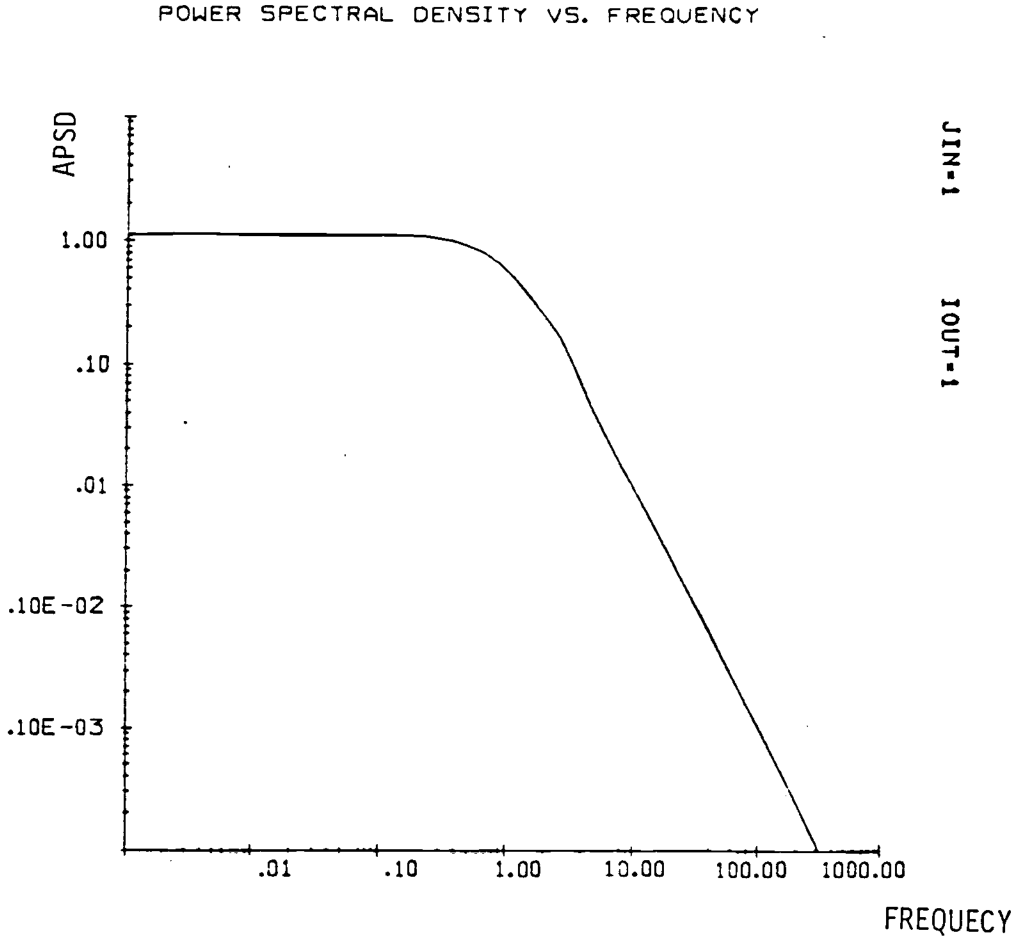


Fig. A-1   APSD For Second Order System.

CONTENTS OF FILE 'DEBUGF' :

```
   N= 2
   NINPT= 1
   NOTPT= 1
   NK= 8
   OMGA= 3.142
   ERR= .00100


   NH= 4



  MATRIX AC; CNSTANT PART OF SYSTEM MATRX:



 COLUMNS  1 TO  2


-6.0000        5.0000
 5.0000       -6.0000


   COMPLEX FORM OF EIGENVECTORS OF MATRIX AC:


 COLUMNS  1 TO  2


 1.0000        1.0000
0.            0.


 1.0000       -1.0000
0.            0.



       EIGENVALUES OF AC:
    -1.0000           0.
    -11.000           0.
   REAL FORME OF EIGENVECTORS OF AC


 COLUMNS  1 TO  2

 1.0000       1.0000
 1.0000      -1.0000


INVERSE OF MATRIX OF EIGENVALUES OF AC


 COLUMNS  1 TO  2

 .50000       .50000
 .50000      -.50000


   MATRIX SI AT TIME (K*PERIOD/NK) ; K= 1


 COLUMNS  1 TO  2

 .52773       .44766
 .44766       .52773


   MATRIX SI AT TIME (K*PERIOD/NK) ; K= 2


 COLUMNS  1 TO  2

 .41974       .41412
```

```
.41412        .41974


  MATRIX SI AT TIME (K*PERIOD/NK) ; K= 3



COLUMNS  1 TO  2


.29594        .29566
.29566        .29594


  MATRIX SI AT TIME (K*PERIOD/NK) ; K= 4



COLUMNS  1 TO  2


.18395        .18393
.18393        .18395


  MATRIX SI AT TIME (K*PERIOD/NK) ; K= 5



COLUMNS  1 TO  2


.11438        .11438
.11438        .11438


  MATRIX SI AT TIME (K*PERIOD/NK) ; K= 6



COLUMNS  1 TO  2
```

```
.81150E-01   .81150E-01
.81150E-01   .81150E-01


  MATRIX SI AT TIME (K*PERIOD/NK) ; K= 7



COLUMNS  1 TO  2


.69375E-01   .69375E-01
.69375E-01   .69375E-01


  MATRIX SI AT TIME (K*PERIOD/NK) ; K= 8



COLUMNS  1 TO  2


.67667E-01   .67667E-01
.67667E-01   .67667E-01


  EIGENVALUES OF SI(TF):
    .13533          0.
    .36908E-09      0.


CONSTAN DIAGONAL SYSTEM MATRIX AFTER TRANSFORMATION :

   -1.0000          0.
   -10.860          0.


COMPLEX FORM OF EIGENVECTORS OF MATRIX  SI AT TIME TF:


COLUMNS  1 TO  2


1.0000      -1.0000
```

```
0.            0.


 1.0000        1.0000
0.            0.






    REAL FORM OF EIGEN-VECTORS OF MATRIX SI AT TIME TF



 COLUMNS  1 TO  2


 1.0000       -1.0000
 1.0000        1.0000


   INVERSE OF MATRIX P AT TIME K*PERIOD/NK; K= 1



 COLUMNS  1 TO  2


 1.2524       -1.2093
 1.2524        1.2093


   MATRIX P:



 COLUMNS  1 TO  2


 .39923        .39923
-.41345        .41345
```

```
   INVERSE OF MATRIX P AT TIME K*PERIOD/NK; K= 2


COLUMNS  1 TO  2

1.3748       -1.2818
1.3748        1.2818


   MATRIX P:


COLUMNS  1 TO  2

 .36369       .36369
-.39007       .39007


   INVERSE OF MATRIX P AT TIME K*PERIOD/NK; K= 3


COLUMNS  1 TO  2

1.2524       -.94219
1.2524        .94219


   MATRIX P:


COLUMNS  1 TO  2

 .39923       .39923
-.53068       .53068
```

```
   INVERSE OF MATRIX P AT TIME K*PERIOD/NK; K= 4


 COLUMNS  1 TO  2


 1.0000       -.57364
 1.0000        .57364


   MATRIX P:


 COLUMNS  1 TO  2


 .50000        .50000
-.87163        .87163


   INVERSE OF MATRIX P AT TIME K*PERIOD/NK; K= 5


 COLUMNS  1 TO  2


 .79845       -.69026
 .79845        .69026


   MATRIX P:


 COLUMNS  1 TO  2


 .62621        .62621
-.72437        .72437
```

```
   INVERSE OF MATRIX P AT TIME K*PERIOD/NK; K= 6


 COLUMNS  1 TO  2


 .72738      -.66208
 .72738       .66208


   MATRIX P:


 COLUMNS  1 TO  2


 .68740       .68740
-.75520       .75520


   INVERSE OF MATRIX P AT TIME K*PERIOD/NK; K= 7


 COLUMNS  1 TO  2


 .79845      -.80337
 .79845       .80337


   MATRIX P:


 COLUMNS  1 TO  2


 .62621       .62621
```

```
-.62238        .62238


   INVERSE OF MATRIX P AT TIME K*PERIOD/NK; K= 8


 COLUMNS  1 TO  2

 1.0000        -1.0000
 1.0000         1.0000


   MATRIX P:


 COLUMNS  1 TO  2

 .50000         .50000
-.50000         .50000


  CONSTATN PART OF MATRIX B:


 COLUMNS  1 TO  1

0.
 1.0000


         MATRIX Q:


 COLUMNS  1 TO  1
```

```
1.0000


    PRODUCT OF P*B*Q*(TRASPOSE OF      (P*B))


   P*B*Q*(TRANS. OF (P*B))) FOR K= 1



COLUMNS   1 TO   2


.15938         .16506
.16506         .17094


   P*B*Q*(TRANS. OF (P*B))) FOR K= 2



COLUMNS   1 TO   2


.13227         .14186
.14186         .15215


   P*B*Q*(TRANS. OF (P*B))) FOR K= 3



COLUMNS   1 TO   2


.15938         .21186
.21186         .28162


   P*B*Q*(TRANS. OF (P*B))) FOR K= 4
```

```
COLUMNS   1 TO   2


.25000         .43582
.43582         .75975


  P*B*Q*(TRANS. OF (P*B))) FOR K= 5



COLUMNS   1 TO   2


.39214         .45361
.45361         .52471


  P*B*Q*(TRANS. OF (P*B))) FOR K= 6



COLUMNS   1 TO   2


.47252         .51913
.51913         .57033


  P*B*Q*(TRANS. OF (P*B))) FOR K= 7



COLUMNS   1 TO   2


.39214         .38974
.38974         .38735


  P*B*Q*(TRANS. OF (P*B))) FOR K= 8
```

```
COLUMNS  1 TO  2

.25000        .25000
.25000        .25000


CONSTANT PART OF MATRIX C:



COLUMNS  1 TO  2

1.0000        1.0000

PRODUCT OF C*(INV. OF P)


(C*(INV. OF P)) FOR K 1



COLUMNS  1 TO  2

2.5048        .15632E-12


(C*(INV. OF P)) FOR K 2



COLUMNS  1 TO  2

2.7496        .24372E-11


(C*(INV. OF P)) FOR K 3
```

```
COLUMNS  1 TO  2

2.5048       .24485E-10

(C*(INV. OF P)) FOR K 4


COLUMNS  1 TO  2

2.0000       .23116E-09

(C*(INV. OF P)) FOR K 5


COLUMNS  1 TO  2

1.5969       .27933E-08

(C*(INV. OF P)) FOR K 6


COLUMNS  1 TO  2

1.4548       .15822E-07

(C*(INV. OF P)) FOR K 7


COLUMNS  1 TO  2
```

```
 1.5969        .31864E-06


 (C*(INV. OF P)) FOR K 8


 COLUMNS  1 TO  2

 2.0000        .36097E-05
    FOURIER COEFFICIENTS OF P*B*Q*(TRANSP SE OFP*B))


 FORFREQUENCY NO. - 0

 COLUMNS  1 TO  2

 .27598        .32088
0.            0.

 .32088        .38711
0.            0.




 FORFREQUENCY NO. - 1

 COLUMNS  1 TO  2

-.43397E-07  -.33009E-01
 .83678E-01   .88384E-01

-.33009E-01  -.85642E-01
 .88384E-01   .92886E-01
 FORFREQUENCY NO. - 2

 COLUMNS   1 TO   2

-.13099E-01   .31033E-02
-.96232E-07  -.21334E-02

 .31033E-02   .35908E-01
-.21334E-02  -.33344E-02




 FCRFREQUENCY NO. - 3

 COLUMNS   1 TO   2

 .63310E-07  -.13445E-01
-.13852E-02  -.59312E-02

-.13445E-01  -.41794E-01
-.59312E-02  -.11657E-01




 FCRFREQUENCY NO. - 4
 COLUMNS   1 TO   2


 .21807E-03   .15817E-01
-.17200E-14  -.21581E-14


 .15817E-01   .45951E-01
-.21581E-14  -.28052E-14




 FORFREQUENCY NO. - 5


 COLUMNS   1 TO   2


 .63310E-07  -.13445E-01
 .13852E-02   .59312E-02


-.13445E-01  -.41794E-01
 .59312E-02   .11657E-01




 FORFREQUENCY NO. - 6


 COLUMNS   1 TO   2


-.13099E-01   .31033E-02
 .96232E-07   .21334E-02


 .31033E-02   .35908E-01
 .21334E-02   .33344E-02
 FORFREQUENCY NO. - 7

 COLUMNS   1 TO   2

-.43397E-07  -.33009E-01
-.83678E-01  -.88384E-01


-.33009E-01  -.85642E-01
-.88384E-01  -.92886E-01




 FORFREQUENCY NO. - 8

 COLUMNS   1 TO   2

 .27598       .32088
-.13737E-13  -.15310E-13


 .32088       .38711
-.15310E-13  -.17528E-13


     FOURIER CCEFFICIENTS OF C*(INV. OF P)


 FORFREQUENCY NO. - 0

 COLUMNS   1 TO   2
 2.0510        .49340E-06
0.            0.




 FORFREQUENCY NO. - 1

 COLUMNS  1 TO  2

-.45283E-06   .47910E-06
-.32236       .30386E-07




 FORFREQUENCY NO. - 2

 COLUMNS  1 TO  2

-.25544E-01   .44927E-06
 .21724E-06   .39484E-07




 FORFREQUENCY NO. - 3

 COLUMNS  1 TO  2

-.52915E-06   .42327E-06
```

```
 .13542E-02   .26431E-07




 FORFREQUENCY NO. - 4

 COLUMNS  1 TO  2

 .10796E-03   .41304E-06
-.45974E-14  -.10956E-19
```

```
 PROGRAM FOURIER
 REAL M(19,19),MINV(19,19)
 DIMENSION P(19,19,90),TEM(19,19),WK(40)
 COMMON/PIN/PINVSI(19,19,90)
 COMMON/BC/ TEMINV(19,19)
 COMMON/SYSMT/ AC(19,19),ASR(19),ASI(19),EASR(19),EASI(19)
+,AT(19,19) ,M,MINV,OMGA,IDEPR
 COMPLEX T1,T2,W(19),Z(19,19),FC(19,19,0:4),PC(19,19,0:8)
 CHARACTER  LABLE*52,LABLE2*40,LABLE3*26,LABLE4*23,NAME*7
 DATA LABLE/↑     INVERSE OF MATRIX P AT TIME K*PERIOD/NK; K=--↑/
 DATA LABLE2/↑     P*B*Q*(TRANS. OF (P*B))) FOR K=--↑/
 DATA LABLE3/↑   (C*(INV. OF P)) FOR K--↑/
 DATA LABLE4/↑   FORFREQUENCY NO. --↑/
 DATA ID/19/
 IDVICE=6
 PRINT*,↑    WHAT IS THE NAME OF YOUR DATA-FILE (<7 LETTERS)↑
 READ*,NAME
 OPEN(5,FILE=NAME)
 REWIND 5
 PRINT*,↑     DO YOU WANT 0) NO OUTPUT↑
 PRINT*,↑                 1) PARTIAL OUTPUT↑
 PRINT*,↑                 2) COMPLETE OUTPUT↑
 READ*,IDEPR
 IF(IDEPR.NE.0) THEN
 PRINT*,↑
+   ↑
 PRINT*,↑    ********YOUR DEBUGUNG OUTPUT FILE IS ≠DEBUGF≠*********↑
 PRINT*,↑
+   ↑
 OPEN(IDVICE,FILE=↑DEBUGF↑)
 REWIND ICVICE
 ENDIF
 C **********************************************************
```

```
      C
      C NK=NO. OF TIME SEGMENT<=16
      C TF:PERIOD PEIODIC SYS. UNDER CONSIDERATION
      C N:ORDER OF SYSTEM
      C NINPT: NO. OF INPUTS.
      C NOTPT: NO. OF OUTPUTS.
      C ERR: SEE CORRESPONDING SUBROUTINES
      C T0:START TIME
      C
      C ******************************************************************
      READ(5,*) N,NINPT,NOTPT,OMGA,ITV
      PRINT*,↑    ↑
      PRINT*,↑     NUMBER OF HARMONICS IN FOURIER SERIES; NH(<5)=↑
      READ*,NH
      PRINT*,↑    ↑
      PRINT*,↑     ACCEPTED TERMINATION ERROR;                 ERR=↑
      READ*, ERR
      PRINT*,↑    ↑
      T0=0.
      PI=ATAN(1.)*4.
      TF=2.*PI/OMGA
      READ(5,*) ((AC(I,J),J=1,N),I=1,N)
      DO 10 I=1,N
        DO 10 J=1,N
          TEM(I,J)=AC(I,J)
10    CONTINUE
      C ******************************************************************
      C
      C EIGENVALUES OF AC  ARE STORED IN ASR(I) ^ ASI(I)
      C MODAL MATRIX OF AC IS STORED IN M(I,J)
      C INVERS OF MODAL MATRIX OF AC IS STORED IN MINV(I,J)
      C
```

```
C  *********************************************************************
      CALL EIGRF(TEM,N,ID,1,W,Z,ID,WK,IER)
      IF(IER.GT.0) THEN
      WRITE(IDVICE,1200) IER
      GO TO 9999
      END IF
      DO 20 J=1,N
        STR=0.0
        DO 15 I=1,N
          SA=CABS(Z(I,J))
          IF(SA.GT.STR) THEN
           STR=SA
           IK=I
          ENDIF
15      CONTINUE
        T1=Z(IK,J)
        DO 20 I=1,N
          Z(I,J)=Z(I,J)/T1
20    CONTINUE
C **********************************************************************
C
C
C THE SPECTRUM NORM IS USED TO SEE IF ANY EIGINVAUE OF A0 IS CCMPLEX
C
C
C **********************************************************************
      FMAX=0.0
      DO 30 I=1,N
         ASR(I)=REAL(W(I))
         ASI(I)=AIMAG(W(I))
         IF(ABS(ASI(I)).GT.FMAX) FMAX=ABS(ASI(I))
30    CONTINUE
```

```
      NK=TF*FMAX/PI+1
      NKT=2*NH
      IF(NK.LT.NKT) NK=NKT
      PRINT*,↑     NUMBER OF THE TIME SEGMENTS;NK IS CHOSEN=↑,NK
      PRINT*,↑     ENTER ZERO(0) IF YOU DO NOT WANT TO CHANGE↑
      PRINT*,↑     OR ENTER YOUR DESIRED NUMBER. YOUR CHOICE↑
      READ*,NKT
      PRINT*,↑    ↑
      IF(NKT.NE.0) NK=NKT
      IF(IDEPR.NE.0) THEN
      WRITE(IDVICE,1100) N,NINPT,NOTPT,NK,OMGA,ERR
      WRITE(IDVICE,↑(↑↑0↑↑,//,↑↑      NH=↑↑,I2,//)↑) NH
      CALL RPRINT(↑0↑,IDVICE,↑    MATRIX AC; CNSTANT PART OF SYSTEM MATRI
     +X:↑,N,N,AC)
      CALL CPRINT(↑0↑,IDVICE,↑     COMPLEX FORM OF EIGENVECTORS OF MATRI
     +X AC:↑,N,N,Z)
      WRITE(IDVICE,1300) (W(I),I=1,N)
      ENDIF
      J=1
50    IF(ASI(J).NE.0.) THEN
       DO 60 K=1,N
         TEMINV(K,J)=REAL(Z(K,J))
60       TEMINV(K,J+1)=AIMAG(Z(K,J))
       J=J+2
      ELSE
       DO 70 K=1,N
70       TEMINV(K,J)=REAL(Z(K,J))
       J=J+1
      ENDIF
      IF(J.LE.N) GO TO 50
      IF(IDEPR.NE.0) THEN
      CALL RPRINT(↑0↑,IDVICE,↑     REAL FORME OF EIGENVECTORS OF AC↑
     ↑,N,N,TEMINV)
      ENDIF
      DO 90   I=1,N
        DO 90   J=1,N
90            M(I,J)=TEMINV(I,J)
      CALL LINV1F(TEMINV,N,ID,TEM,6,WK,IER)
      IF (IER.GT.0) THEN
      WRITE(IDVICE,1400) IER
      GO TO 9999
      END IF
      IF(IDEPR.NE.0) THEN
      CALL RPRINT(↑0↑,IDVICE,↑  INVERSE OF MATRIX OF EIGENVALUES OF AC↑
     ↑,N,N,TEM)
      ENDIF
      DO 100  I=1,N
         DO 100  J=1,N
100   MINV(I,J)=TEM(I,J)
110   CALL SIMAT(T0,TF,N,NK,ERR,ID,IDVICE,ITV,ICH)
      IF(ICH.EQ.100) GO TO 9999
      C ******************************************************************
      C
      C
      C ↑PINVSI↑:USED TO STOR ↑SI↑ MATRIX(FUNDUMENTAL MATRIX OF THE SYSTEM)
      C AT THIS POINT.IT IS PERMANENT STORAGE FOR ↑P↑ CALCULATED LATER
      C ↑ASR↑,↑ASI↑ ARE REAL AND IMAGINARY PARTS OF EIGINVALUES  OF ↑A*↑.
      C↑A*↑,IS CONSTANT SYSTEM MATRIX CALCULATED ACCORDING TO THE FLOQUE TEORY:
      C ↑EXP(A)=SI↑ . DIAGNALIZED FORM OF ↑SI↑ AT TIME ↑TF↑ IS USED.
      C IMSL RCUTINE ↑EIGRF↑ IS USED TC FIND EIGENVALUES OF ↑SI↑ AT TIME ↑TF↑.
      C
      C
      C ******************************************************************
      DO 120 I=1,N
```

```
        DO 120J=1,N
           TEM(I,J)=PINVSI(I,J,NK)
 120  CONTINUE
     CALL EIGRF(TEM,N,ID,1,W,Z,ID,WK,IER)
     IF(IER.GT.0) THEN
     WRITE(IDVICE,1500) IER
     GO TO 9999
     END IF
     DO 140 J=1,N
       STR=0.0
       DO 130 I=1,N
         SA=CABS(Z(I,J))
         IF(SA.GT.STR)THEN
          STR=SA
          IK=I
         ENDIF
130    CONTINUE
       T1=Z(IK,J)
       DO 140 I=1,N
         Z(I,J)=Z(I,J)/T1
140  CONTINUE
     IF(IDEPR.NE.0) THEN
      WRITE(IDVICE,1600)(W(I),I=1,N)
     ENDIF
     C **********************************************************************
     C
     C
     C THE SPECTRUM NORM IS USED TO SEE IF ANY EIGINVAUE OF SI(TF) IS COMPLEX
     C
     C
     C **********************************************************************
     DO 160 I=1,N
```

```
        WR=REAL(W(I))
        WI=AIMAG(W(I))
        T=WR*WR+WI*WI
        ASR(I)=ALOG(T)/(TF*2.)
        ASI(I)=ATAN2(WI,WR)/TF
160   CONTINUE
      WRITE(IDVICE,1700) (ASR(I),ASI(I),I=1,N)
      IF(IDEPR.NE.0)THEN
      CALL CPRINT(↑1↑,IDVICE,↑   COMPLEX FORM OF EIGENVECTORS OF MATRIX
     + SI AT TIME TF:↑,N,N,Z)
      ENDIF
      J=1
170   IF(ASI(J).NE.0.) THEN
       DO 180 K=1,N
        M(K,J)=REAL(Z(K,J))
180     M(K,J+1)=AIMAG(Z(K,J))
       J=J+2
      ELSE
       DO 190 K=1,N
190     M(K,J)=REAL(Z(K,J))
       J=J+1
      ENDIF
      IF(J.LE.N) GO TO 170
      IF(IDEPR.NE.0)THEN
      CALL RPRINT(↑0↑,IDVICE,↑      REAL FORM OF EIGEN-VECTORS OF MATRIX
     + SI AT TIME TF↑,N,N,M)
      ENDIF
      C *************************************************************************
      C
      C
      C CALCULATION OF ↑P↑ AND ITS INVERSE,↑PINVSI↑.
      C ↑P↑ IS THE MATRIX THAT TRANSFORMS THE  PRIODIC COEFFICIET SYSTEM
     C INTO A COSTANT COEFFICIENT ONE.
     C NOTE: AFTER THESE CALCULATIOCNS ↑PINVSI↑ WILL CONTAIN INVERSE OF ↑P↑.
     C
     C
     C **********************************************************************
     DT=(TF-T0)/FLOAT(NK)
     DO 310 K=1,NK
        TSEG=K*DT
        DO 220 I=1,N
        DO 220 J=1,N
        TEMINV(I,J)=0.0
        DO 220 L=1,N
           TEMINV(I,J)=TEMINV(I,J)+PINVSI(I,L,K)*M(L,J)
220      CONTINUE
        CALL EXPAS(TSEG,N,-1)
        J=1
240     IF(ABS(EASI(J)).GT.0.0) THEN
        DO 250 I=1,N
           TEM(I,J)=TEMINV(I,J)*EASR(J)-TEMINV(I,J+1)*EASI(J)
           TEM(I,J+1)=TEMINV(I,J)*EASI(J)+TEMINV(I,J+1)*EASR(J)
250     CONTINUE
        J=J+2
        ELSE
        DO 260 I=1,N
            TEM(I,J)=TEMINV(I,J)*EASR(J)
260     CONTINUE
        J=J+1
        END IF
        IF(J.LE.N) GO TO 240
        IF(IDEPR.EQ.2) THEN
         WRITE(LABLE( 49:50 ),↑(I2)↑) K
        CALL RPRINT(↑1↑,IDVICE,LABLE,N,N,TEM)
        ENDIF
        DO 270 I=1,N
           DO 270 J=1,N
              PINVSI(I,J,K)=TEM(I,J)
270     CONTINUE
     C ********************************************************************
     C
     C
     C CALL IMSL ROUTINE ↑LINV1F↑ TO CALCULATE INVERS OF ↑PINVSI↑:↑P↑.
     C
     C
     C ********************************************************************
        CALL LINV1F(TEM,N,ID,TEMINV,0,WK,IER)
        IF(IER.GT.0) THEN
        WRITE(IDVICE,1800) IER
        GO TO 9999
        END IF
        IF(IDEPR.EQ.2)
    +   CALL RPRINT(↑0↑,IDVICE,↑     MATRIX P:↑,N,N,TEMINV)
        DO 310 I=1,N
           DO 310 J=1,N
              P(I,J,K)=TEMINV(I,J)
310  CONTINUE
     C ********************************************************************
     C
     C
     C  AT THIS POINT THE FOUREIR COEFFICIENTS ARE CALCULATED
     C FOR THE PRODUCTS OF P*B*Q*(TRAS.(P*B))
     C FIRST CALCULATE THE PRCDUCTS.
     C READ ↑Q↑ MATRIX IN AT(I,J)
     C READ ↑B↑ MATRIX IN TEM(I,J)
     C NOTE: MATROX P WILL CONTAIN THE ABOVE PRODUCT.
```

```
      C
      C
      C ****************************************************************
       READ(5,*) ((AT(I,J),J=1,NINPT),I=1,NINPT)
       READ(5,*) ((TEM(I,J),J=1,NINPT),I=1,N)
       CALL RPRINT(↑1↑,IDVICE,↑    CONSTATN PART OF MATRIX B:↑,N,
      +NINPT,TEM)
       CALL RPRINT(↑1↑,IDVICE,↑           MATRIX Q:↑,NINPT
      +,NINPT,AT)
       IF(IDEPR.EQ.2)
      +WRITE(IDVICE,↑(↑↑1↑↑,//,↑↑       PRODUCT OF P*B*Q*(TRASPOSE OF
      +(P*B))↑↑)↑)
       DO 360 K=1,NK
       DO 315  I=1,N
         DO 315 J=1,N
  315        TEMINV(I,J)=0.0
       T=FLOAT(K)*DT
        CALL BMATRIX(T,N)
       DO 320 I=1,N
           DO 320 J=1,N
   320         AC(I,J)=TEM(I,J)+TEMINV(I,J)
       DO 330 I=1,N
       DO 330 J=1,NINPT
       M(I,J)=0.0
       DO 330 L=1,N
  330  M(I,J)=P(I,L,K)*AC(L,J)+M(I,J)
       DO 340 I=1,N
       DO 340 J=1,NINPT
       MINV(I,J)=0.0
       DO 340 L=1,NINPT
  340  MINV(I,J)=M(I,L)*AT(L,J)+MINV(I,J)
       DO 350 I=1,N
      DO 350 J=1,N
      TEMINV(I,J)=0.0
      DO 350 L=1,NINPT
350   TEMINV(I,J)=TEMINV(I,J)+MINV(I,L)*M(J,L)
      IF(IDEPR.EQ.2) THEN
      WRITE(LABLE2(37:38),↑(I2)↑) K
      CALL RPRINT(↑0↑,IDVICE,LABLE2,N,N,TEMINV)
      ENDIF
      DO 360 I=1,N
      DO 360 J=1,N
      P(I,J,K)=TEMINV(I,J)
360   CONTINUE
      C ***********************************************************************
      C
      C
      C CALCULATE THE,PRODUCT OF (C*(INV. OFP))
      C READ ↑C↑ MMATIX IN TEM(I,J)
      C    NOTE PINVSI(I,J,K) WILL CONTAIN (C*(INV. OF P))
      C
      C
      C ***********************************************************************
      READ(5,*) ((TEM(I,J),J=1,N),I=1,NOTPT)
      CALL RPRINT(↑1↑,IDVICE,↑   CCNSTANT PART OF MATRIX C:↑,NOTPT,
     +N,TEM)
      IF(IDEPR.EQ.2)
     +WRITE(IDVICE,↑(↑↑1↑↑,//,↑↑   PROCUCT OF C*(INV. OF P)↑↑)↑)
      DO 370 I=1,N
          DO 370 J=1,N
370          TEMINV(I,J)=0.0
      DO 390 K=1,NK
      T=FLOAT(K)*DT
      CALL CMATRIX(N,T)
      DO 375 I=1,N
         DO 375 J=1,N
375       AC(I,J)=TEM(I,J)+TEMINV(I,J)
      DO 380 I=1,NOTPT
      DO 380 J=1,N
       AT(I,J)=0.0
      DO 380 L=1,N
380   AT(I,J)=AT(I,J)+AC(I,L)*PINVSI(L,J,K)
      IF(IDEPR.EQ.2) THEN
      WRITE(LABLE3(25:26),↑(I2)↑) K
      CALL RPRINT(↑0↑,IDVICE,LABLE3,NOTPT,N,AT)
      ENDIF
      DO 390 I=1,N
      DO 390 J=1,N
390   PINVSI(I,J,K)=AT(I,J)
      C ******************************************************************
      C
      C
      C HERE WE CALCULATE THE FOURIER COEFFICIENTS FOR:
      C       (C*(INV. OF P)),
      C       (P*B)*Q*(TRANSPOSE OF(P*B))
      C
      C FC(I,J,K) IS USED TO SAVE THE F-COEFF. OF FIRST PRODUCT.
      C PC(I,J,K) IS USED TO SAVE THE F-COEFF. OF SECOND PRODUCT.
      C NH IS THE NUMBER OF HARMONIC DESIRED FOR FOURIER SERIES.
      C
      C
      C ******************************************************************
      DO 400 I=1,N
      DO 400 J=1,N
      DO 400 K=0,NH
        PC(I,J,K)=CMPLX(0.0,0.0)
       PC(I,J,NH+K)=CMPLX(0.0,0.0)
400    FC(I,J,K)=CMPLX(0.0,0.0)
     IF(IDEPR.EQ.1.OR.IDEPR.EQ.2)
    +WRITE(IDVICE,↑(↑↑1↑↑,↑↑      FOURIER COEFFICIENTS OF P*B*Q*(TRANSP
    +SE OFP*B))↑↑)↑)
     DO 500 L=0,2*NH
     DO 480 K=1,NK
     T2=CMPLX(0.,-2.*L*K*PI)/FLOAT(NK)
     T1=CEXP(T2)/FLOAT(NK)
     DO 480 I=1,N
     DO 480 J=1,N
480    PC(I,J,L)=PC(I,J,L)+P(I,J,K)*T1
       IF(IDEPR.EQ.1.OR.IDEPR.EQ.2) THEN
        DO 490 IK=1,N
          DO 490 JK=1,N
490         Z(IK,JK)=PC(IK,JK,L)
       WRITE(LABLE4(22:23),↑(I2)↑) L
       CALL CPRINT(↑0↑,IDVICE,LABLE4,N,N,Z)
       ENDIF
500    CONTINUE
     IF(IDEPR.EQ.1.OR.IDEPR.EQ.2)
    +WRITE(IDVICE,↑(↑↑0↑↑,↑↑      FOURIER COEFFICIENTS OF C*(INV. OF F)
    +↑↑)↑)
     DO 560 L=0,NH
     DO 540 K=1,NK
     T2=CMPLX(0.,-2.*L*K*PI)/FLOAT(NK)
     T1=CEXP(T2)/FLOAT(NK)
     DO 540 I=1,NOTPT
     DO 540 J=1,N
540    FC(I,J,L)=FC(I,J,L)+PINVSI(I,J,K)*T1
       IF(IDEPR.EQ.1.OR.IDEPR.EQ.2) THEN
         DO 550 IK=1,N
```

```
           DO 550 JK=1,N
550                Z(IK,JK)=FC(IK,JK,L)
       WRITE(LABLE4(22:23),↑(I2)↑) L
       CALL CPRINT(↑0↑,IDVICE,LABLE4,NOTPT,N,Z)
       ENDIF
560   CONTINUE
      OPEN(7,FILE=↑CONECTN↑)
      REWIND 7
      WRITE(7,*)  N,NOTPT,NH,OMGA
      WRITE(7,*)  (ASR(I),I=1,N)
      WRITE(7,*)  (ASI(I),I=1,N)
      WRITE(7,*)  (((PC(I,J,K),I=1,N),J=1,N),K=0,2*NH)
      WRITE(7,*)  (((FC(I,J,K),I=1,NOTPT),J=1,N),K=0,NH)
      CLOSE (7,STATUS=↑KEEP↑)
      PRINT*,↑     ↑
      PRINT*,↑ ****NECESSARY OUTPUTS FOR PLOTTING PSD ARE IN↑
      PRINT*,↑     LOCAL FILE ↑↑CONECTN↑↑.          ***********↑
      PRINT*,↑      ↑
1100  FORMAT(↑1↑,4X,↑N=↑,I2,/,5X,↑NINPT=↑,I2,/,5X,↑NOTPT=↑,
     ↑I2,/,5X,↑NK=↑,I2,/,5X,↑OMGA=↑,F6.3,/,5X,↑ERR=↑,F7.5)
1200  FORMAT(↑    IER=↑,I4,↑    ERROR IN EIGRF/IMSL↑)
1300  FORMAT(///5X,↑    EIGENVALUES OF AC:↑,/,2(5X,G12.5))
1400  FORMAT(↑    IER=↑,I4,↑    ERROR IN LINV1F/IMSL↑)
1500  FORMAT(↑    IER=↑,I4,↑    ERROR IN EIGRF/IMSL FOR SI(TF)↑)
1600  FORMAT(///,5X,↑EIGENVALUES OF SI(TF):↑,/,2(5X,G12.5))
1700  FORMAT(///,↑  CONSTAN DIAGONAL SYSTEM MATRIX AFTER TRANSFORMATION
     ↑: ↑,//,2(5X,G12.5))
1800  FORMAT(↑   IER=↑,I4,↑    ERROR IN LINV1F/IMSL FOR P↑)
9999  CLOSE(IDVICE,STATUS=↑KEEP↑)
      STOP
      END
      C **********************************************************************
     SUBROUTINE SIMAT(T0,TF,N,NK,ERR,ID,IDVICE,ITV,ICH)
     REAL M(19,19),MINV(19,19)
     DIMENSION S(19),DS(19),W(19,9),C(24),TEM(19,19),AD(19)
     COMMON/PIN/PINVSI(19,19,90)
     COMMON/SYSMT/AC(19,19),ASR(19),ASI(19),EASR(19),EASI(19)
    +,AT(19,19),M,MINV,OMGA,IDEPR
     CHARACTER *45 LABLE
     EXTERNAL FCN
     COMMON /SIFCN/ JN
     DATA LABLE/↑     MATRIX SI AT TIME (K*PERIOD/NK) ; K=--↑/
     C **********************************************************************
     C
     C
     C THIS ROUTINE CALCULATES THE TIME DEPENDANT FUNDUMENTAL MATIX,↑SI↑.
     C NOTE THAT AT THIS POINT ↑PINVSI↑ CONTAINS ↑SI↑.
     C ↑S↑ AND ↑DS↑ ARE USED TEMPORARILY AS COLUMN VECTORS OF ↑SI↑ AND ITS DERIVATIVES
     C RESPECTIVELY. TO SOLV DS=A*S  ROUTINE DVERK FROM IMSL LIBRARY IS CALLED.
     C  SUBROUTINE FCN SHOULD BE PROVIDED BY USE. SEE IMSL MANUAL.
     C
     C
     C **********************************************************************
     WRITE(IDVICE,↑(↑↑1↑↑)↑)
     DT=(TF-T0)/FLOAT(NK)
     DO 5  J=1,N
        DO 5  I=1,N
 5         AT(I,J)=0.0
     DO 30 IC=1,N
        JN=IC
        T=T0
        DO 10 J=1,N
10        S(J)=0.0
        IND=1
```

```
         DO 30 K=1,NK
            TEND=K*DT
            IF(ITV.EQ.1) THEN
            CALL DVERK(N,FCN,T,S,TEND,ERR,IND,C,ID,W,IER)
            IF(IER.GT.0.OR.IND.LT.0) THEN
               WRITE(IDVICE,110) IER,(C(IK),IK=1,24)
             ICH=100
               RETURN
            END IF
           ENDIF
            CALL EXPAS(TEND,N,1)
            I=1
11          IF(ABS(EASI(I)).GT.0.0) THEN
              AD(I)=MINV(I,IC)*EASR(I)+MINV(I+1,IC)*EASI(I)
              AD(I+1)=-MINV(I,IC)*EASI(I)+MINV(I+1,IC)*EASR(I)
              I=I+2
            ELSE
              AD(I)=MINV(I,IC)*EASR(I)
              I=I+1
            END IF
            IF(I.LE.N) GO TO 11
            DO 12 I=1,N
              EASR(I)=0.
              DO 12 J=1,N
12               EASR(I)=EASR(I)+M(I,J)*AD(J)
            DO  20 J=1,N
               PINVSI(J,IC,K)=S(J)+EASR(J)
20          CONTINUE
            T=TEND
30    CONTINUE
      IF(IDEPR.EQ.2) THEN
      DO 50 K=1,NK
      WRITE(LABLE(42:43),↑(I2)↑)  K
         DO 40 I=1,N
            DO 40 J=1,N
40             TEM(I,J)=PINVSI(I,J,K)
50    CALL RPRINT(↑0↑,IDVICE,LABLE,N,N,TEM)
      ENDIF
110   FORMAT(↑     IER=↑,I4,↑      ERRCR IN DVERK/IMSL↑,/,↑
     *C(I) VECTOR:↑,(G12.5))
      RETURN
      END
      C *************************************************************
      SUBROUTINE FCN(N,T,S,DS)
      REAL M(19,19),MINV(19,19)
      DIMENSION S(N),DS(N),AD(19),A(19,19)
      COMMON /SIFCN/ JN
      COMMON/SYSMT/AC(19,19),ASR(19),ASI(19),EASR(19),EASI(19)
     +,AT(19,19),M,MINV,OMGA,IDEPR
      C *************************************************************
      C
      C
      C THIS RCUTINE IS FOR USE BY IMSL ROUTINE ↑DVERK↑. SHOULD APPEAR IN EXTERNAL
      C STATMENT OF SUBRCUTINE ↑SIMAT↑THISE ROUTINE PROIDES DRIVATIVES OF
      C VECTOR ↑S↑; ↑DS↑.  SUBROUTINE ↑SYSMAT↑ GIVING ↑A↑ MATRIX OF THE SYSTEM
      C SHOULD BE PROVIDED.
      C
      C
      C *************************************************************
      CALL EXPAS(T,N,1)
      I=1
10    IF(ABS(EASI(I)).NE.0.0) THEN
        AD(I)=MINV(I,JN)*EASR(I)+MINV(I+1,JN)*EASI(I)
        AD(I+1)=-MINV(I,JN)*EASI(I)+MINV(I+1,JN)*EASR(I)
        I=I+2
      ELSE
        AD(I)=MINV(I,JN)*EASR(I)
        I=I+1
      END IF
      IF(I.LE.N) GO TO 10
      DO 20 I=1,N
        EASR(I)=0.
        DO 20 J=1,N
20         EASR(I)=EASR(I)+M(I,J)*AD(J)
      CALL SYSMAT(T,N)
      DO 40 I=1,N
        DO 40 J=1,N
40         A(I,J)=AC(I,J)+AT(I,J)
      DO 60 I=1,N
         DS(I)=0.
         DO 50 J=1,N
            DS(I)=DS(I)+A(I,J)*S(J)+AT(I,J)*EASR(J)
50       CONTINUE
60    CONTINUE
      RETURN
      END
      C **********************************************************************
      SUBROUTINE EXPAS(T,N,ISIGN)
      REAL M(19,19),MINV(19,19)
      COMMON/SYSMT/ AC(19,19),ASR(19),ASI(19),EASR(19),EASI(19)
     +,AT(19,19),M,MINV,OMGA,IDEPR
      COMPLEX W,W1
      C **********************************************************************
      C
      C
      C THIS SUBROUTINE CALCULATES EXP(ISIGN*A*T) AT TIME T AS USED BY USER  WHEN
      C ↑SI↑ MATRIX WAS BEIG CALCULATED. FOR EACH NEW TIME THE ROUTINE
      C  SHOULD BE CALLED.A IS IN DIAGCNAL FORM ASSUMING THRE IS NO JORAN BLOCK.
      C
      C
      C ***********************************************************************
      DO 3 I=1,N
       IF(ASI(I).NE.0.)THEN
        W=CMPLX(ISIGN*T*ASR(I),ISIGN*T*ASI(I))
        W1=CEXP(W)
        EASI(I)=AIMAG(W1)
        EASR(I)=REAL(W1)
       ELSE
        EASI(I)=0.
        EASR(I)=EXP(ISIGN*ASR(I)*T)
       END IF
3     CONTINUE
      RETURN
      END
      C ***********************************************************************
      SUBROUTINE RPRINT(IPAGE,IDVICE,LABLE,NR,NC,T)
      DIMENSION T(19,19)
      CHARACTER IPAGE,LABLE*(*),FMT*14,TITLE*21
      DATA FMT,TITLE/↑(--(1X,G12.5))↑,↑   COLUMNS -- TO --↑/
      WRITE(IDVICE,↑(A///A//)↑) IPAGE,LABLE
      K=1
      ICO=1
10    M=ICO*10
      IF(M-NC)30,30,20
20     M=NC
30     WRITE(TITLE(12:13),↑(I2)↑) K
       WRITE(TITLE(18:19),↑(I2)↑) M
      WRITE(FMT(2:3),↑(I2)↑) (M-(ICO-1)*10)
      WRITE(IDVICE,↑(//A //)↑) TITLE
      WRITE(IDVICE,FMT)  ((T(I,J),J=K,M),I=1,NR)
      IF(M.EQ.NC) RETURN
      K=M+1
      ICO=ICO+1
      GO TO 10
      END
      C ************************************************************************
      SUBROUTINE CPRINT(IPAGE,IDVICE,LABLE,NR,NC,T)
      COMPLEX T(19,19)
      CHARACTER IPAGE,LABLE*(*),FMT*14,TITLE*21
      DATA FMT,TITLE/↑(--(1X,G12.5))↑,↑   COLUMNS -- TO --↑/
      WRITE(IDVICE,↑(A///A/)↑) IFAGE,LABLE
      K=1
      ICO=1
10    M=ICO*10
      IF (M-NC)30,30,20
20    M=NC
30    WRITE(TITLE(12:13),↑(I2)↑) K
      WRITE(TITLE(18:19),↑(I2)↑) M
      WRITE(FMT(2:3),↑(I2)↑) (M-(ICO-1)*10)
      WRITE(IDVICE,↑(/A//)↑) TITLE
      DO 40 I=1,NR
      WRITE(IDVICE,FMT)(REAL(T(I,J)),J=K,M),(AIMAG(T(I,J)),J=K,M)
40    WRITE(IDVICE,↑(/)↑)
      WRITE(IDVICE,↑(//)↑)
      IF(M.EQ.NC) RETURN
      K=M+1
      ICO=ICO+1
      GO TO 10
      END
      PROGRAM POWSPEC
      REAL LOMGAF
      DIMENSION ASR(19),ASI(19),PSD(400),OMGAF(400),OMGAFT1(400)
     +,PSDT1(400)
      COMPLEX W(19),FC(19,19,0:4),PC(19,19,0:8),PSDC(400),PSDC1(400)
      CHARACTER LABLE*3,LABLE1*25
      DATA LABLE1/'   RESONANT FREQUENCEY*****'/
       C ********************************************************************************
       C
       C   AT THIS POINT CALCULATE THE POWER SPECTRAL DENSITY FOR JIN INPUT AND IOUT OUTPUT
       C JIN : THE INPUT DESIRED IN CALCULAYION OF POWER SPECTAL DENSITY(PSD).
       C IOUT : THE OUTPUT DESIRED IN CALCULATION OF POWER SPECTRAL DENSITY.
       C LOMGAF: THE LOWER LIMIT OF FOURIER FREQUENCY IN CALCULATING PSD.
       C HOMGAF: THE UPPER LIMIT OF FOURIER FREQUENCEY IN CALCULATING PSD.
       C NOMGAF: THE NO. OF FREQUENCEYS CONSIDERD IN CALCULATING PSD.
       C
       C ********************************************************************************
       OPEN(7,FILE='CONECTN')
      OPEN(8,FILE='DEBUGP')
      REWIND 7
      REWIND 8
      READ(7,*) N,NOTPT,NH,OMGA
      READ(7,*)(ASR(I),I=1,N)
      READ(7,*)(ASI(I),I=1,N)
      READ(7,*)(((PC(I,J,K),I=1,N),J=1,N),K=0,2*NH)
      READ(7,*)(((FC(I,J,K),I=1,NOTPT),J=1,N),K=0,NH)
      CLOSE(7,STATUS='KEEP')
100   PRINT*,' THE INPUT CONSIDERED FOR POWER SPECTRAL DENSITY; JIN=',
      READ*,JIN
      PRINT*,' THE OUTPUT CONSIDERED FOR POWER SPECTRAL DESITY;IOUT=',
      READ*, IOUT
      PRINT*,' LOWER LIMIT FOR FOURIER FREQUENCEY;          LOMGAF=',
      READ*, LOMGAF
      PRINT*,' HIGHER LIMIT FOR FOURIER FREQUENCEY;         HOMGAF=',
      READ*, HOMGAF
      PRINT*,' NUMBER FREQUENCIES BETWEEN LOMGAF & HOMGAF;   NOMGAF=',
      READ*, NOMGAF
      PRINT*,'    '
      PRINT*,'    ****YOUR DEBUGGING OUTPUT-FILE IS ''DEBUGP''*****'
      PRINT*,'    '
      WRITE(8,'(''1'',///,5X,''  JIN='',I2,''   IOUT='',I2)')JIN,IOUT
      WRITE(8,1100) N,NH,OMGA,LOMGAF,HOMGAF,NOMGAF
      DO  120 I=1,N
        W(I)=CMPLX(ASR(I),ASI(I))
120   CONTINUE
      ALOGLF=ALOG10(LOMGAF)
      DOMGAF=(ALOG10(HOMGAF)-ALOGLF)/FLOAT(NOMGAF-1)
      DO 130 I=1,NOMGAF
130     OMGAF(I)=(I-1)*DOMGAF+ALOGLF
      NRF=0
      DO 150 IH=-NH,NH
      DO 150 J=1,N
        OMT=ABS(IH*OMGA+ASI(J))
        IF(OMT.LE.LOMGAF.OR.OMT.GT.HOMGAF) GO TO 150
        NRF=NRF+1
        OMGAFT1(NRF)=ALOG10(OMT)
        IF(NRF.EQ.400) THEN
         PRINT*,'  NO. OF RESONANT FREQUENCEY MORE THAN 400.
     +   UPDATE DIMENSION STATMENT FOR OMGAFT1(400)'
        ENDIF
150   CONTINUE
      IF(NRF.GT.0) THEN
      CALL CAL(IOUT,JIN,N,NH,OMGA,NRF,OMGAFT1,W,FC,PC,FSDT1,PSDC1)
      ENDIF
      CALL CAL(IOUT,JIN,N,NH,OMGA,NOMGAF,OMGAF,W,FC,PC,PSD,PSDC)
      IF(NRF.GT.0) THEN
      K=1
      DO 160 I=1,NRF
      DO 160 J=2,NOMGAF
      IF(PSD(J).LT.PSDT1(I).AND.PSD(J-1).LT.PSDT1(I).
     +AND.OMGAF(J).GT.OMGAFT1(I).AND.OMGAF(J-1).LT.OMGAFT1(I))THEN
        PSD(J-1)=PSDT1(I)
        OMGAF(J-1)=OMGAFT1(I)
        PSDC(J-1)=PSDC1(I)
        PSDT1(K)=J-1
        K=K+1
      ENDIF
160   CONTINUE
      ENDIF
      KPSDT1=0
      IF(NRF.GT.0) THEN
      J=1
      KPSDT1=PSDT1(J)
      ENDIF
      WRITE(8,1200)
      DO 170 I=1,NOMGAF
        IF(KPSDT1.EQ.I) THEN
        WRITE(8,1300)PSDC(I),OMGAF(I),LABLE1
        J=J+1
        KPSDT1=PSDT1(J)
        ELSE
        WRITE(8,1300) PSDC(I),OMGAF(I)
        ENDIF
170   CONTINUE
      CALL PLOTS(PSD,NOMGAF,LOMGAF,HOMGAF,OMGAF,JIN,IOUT)
      PRINT*,"       DO YOU WISH TO FIND ANOTHER PSD ?"
      PRINT*,'          ENTER  YES OR NO.'
      READ(*,'(A)')  LABLE
      IF(LABLE.EQ.'YES')GO TO 100
      CLOSE(8,STATUS='KEEP')
 1100 FORMAT(//'      N=',I2,/,'      NH=',I2,/,'      OMGA=',G12.4
     +,/,'      LOMGAF=',G12.4,/,'      HOMGAF=',G12.4,/,
     +'      NOMGAF=',I3)
 1200 FORMAT(///,5X,'   COMPLEX FORM OF PSD',10X,'  FREQUENCIES',//)
 1300 FORMAT(2(3X,G12.4),9X,G12.4,A)
      STOP
      END
 SUBROUTINE CAL(ICUT,JIN,N,NH,OM,NOMF,OMF,W,FC,PC,PSD,PSDC)
 CCMPLEX W(19),WCA(19),WCB(19),WCC(19),FC(19,19,0:4),PC(19,19,0:8)
+,PSDC(400),TL,TK,AI,A,B
 DIMENSION PSD(400),CMF(400)
 AI=CMPLX(0.,1.)
C ******************************************************************

C THIS ROUTINE CALCULATES THE AVERAGE POWER SPECTRAL DENSITY ACCORDING TO
C THE FCRMULATICN GIVEN IN CHAPTER 3.
C IOUT,JIN: THE ELEMENT OF APSD ON ROW IOUT AND CCLUMN JIN.
C N: ORDER OF SYSTEM
C NH: NC. CF FCURIER FREQUENCIES
C CM: FUNDUMENTAL FREQUENCY.
C NOMF: NO. CF FREQUENCIES FOR WHICHAPSD IS DESIRED
C CMF: VECTOR CONTAINING THE FREQUENCIES FOR WHICH APSD IS DESIRED
C W: COMLEX FORM OF THE EIGENVAUES OF THE SYSTEM
C FC: MATRIX OF FCURIER COEFFICIENTS OF (C*INV(P))
C PC: MATRIX OF FOURIER COEFFICIENT OF (P*B*Q*(TRANS(P*B)))
C FSD: REAL PART CF APSD
C FSDC: COMPLEX FORM OF APSD
C
C ROUTINE PLCT IS NEEDED TO PLOT THE REAL PART CF ASPD
C
C******************************************************************
  DC 650 LF=1,NCMF
   FSDC(LF)=CMPLX(0.,0.)
   CMF1=10.**CMF(LF)
   DO 650 IK=-NH,NH
    K=ABS(IK)
    DO 650 IL=-NH,NH
      L=ABS(IL)
      IF(ABS(IL-IK).LE.2*NH) THEN
       ILK=ABS(IL-IK)
      TL=CMPLX(0.,-IL*OM+OMF1)
      TK=CMPLX(0.,IK*OM-OMF1)
      DO 50 I=1,N
50     WCB(I)=1./(CONJG(W(I))+TL)
      I=1
      IF(IL.LE.0) THEN
100    IF(AIMAG(W(I)).EQ.0.) THEN
        WCA(I)=WCB(I)*FC(JIN,I,L)
        I=I+1
       ELSE
        A=(WCB(I)+WCB(I+1))/2.
        B=(WCB(I)-WCB(I+1))/2.
        WCA(I)=A*FC(JIN,I,L)-AI*B*FC(JIN,I+1,L)
        WCA(I+1)=A*FC(JIN,I+1,L)+AI*B*FC(JIN,I,L)
        I=I+2
       ENDIF
       IF(I.LE.N)GO TO 100
      ELSE
150    IF(AIMAG(W(I)).EQ.0.) THEN
        WCA(I)=WCB(I)*CCNJG(FC(JIN,I,L))
        I=I+1
       ELSE
        A=(WCB(I)+WCB(I+1))/2.
        B=(WCB(I)-WCB(I+1))/2.
        WCA(I)=A*CONJG(FC(JIN,I,L))-AI*B*CONJG(FC(JIN,I+1,L))
        WCA(I+1)=A*CCNJG(FC(JIN,I+1,L))+AI*B*CONJG(FC(JIN,I,L))
        I=I+2
       ENDIF
       IF(I.LE.N)GO TO 150
      ENDIF
      IF(IL-IK.LT.0) THEN
       DO 200 I=1,N
```

```
          WCB(I)=CMPLX(0.,0.)
          DO 200 J=1,N
200          WCB(I)=WCB(I)+CONJG(PC(I,J,ILK))*WCA(J)
       ELSE
        DO 250 I=1,N
          WCB(I)=CMPLX(0.,0.)
         DO 250 J=1,N
250         WCB(I)=WCB(I)+PC(I,J,ILK)*WCA(J)
       ENDIF
       DO 300 I=1,N
300       WCC(I)=1./(W(I)+TK)
       I=1
350    IF(AIMAG(W(I)).EQ.0.) THEN
        WCA(I)=WCC(I)*WCB(I)
        I=I+1
       ELSE
        B=(WCC(I)-WCC(I+1))/2.
        A=(WCC(I)+WCC(I+1))/2.
        WCA(I+1)=A*WCB(I+1)+AI*B*WCB(I)
        WCA(I)=A*WCB(I)-AI*B*WCB(I+1)
        I=I+2
       ENDIF
       IF(I.LE.N)GO TO 350
        IF(IK.LT.0) THEN
         DO 450 J=1,N
450       PSDC(LF)=PSDC(LF)+CONJG(FC(IOUT,J,K))*WCA(J)
        ELSE
         DO 500 J=1,N
500        PSDC(LF)=PSDC(LF)+FC(IOUT,J,K)*WCA(J)
        ENDIF
        PSD(LF)=REAL(PSDC(LF))
      ENDIF
```

```
650   CONTINUE
      RETURN
      END
      SUBROUTINE PLOTS(PSD,NOMGAF,LOMGAF,HOMGAF,W1,JIN,ICUT)
      REAL LOMGAF
      DIMENSION PSD(400),W1(400)
      CHARACTER *10 LABLE(4)
      DATA LABLE/'POWER SPEC','TRAL DENSI','TY VS. FRE','QUENCY    '/
C *********************************************************************
C THIS ROUTINE PLOTS A REAL VECTOR ON TEK-TEMINAL 4010
C PSD: THE VECTOR TO BE PLOTTED
C NOMGAF: X-AXIS COMPONENT OF THE VECTOR TO BE PLOTTED( FREQUENCIES)
C LOMGAF: SMALLEST ELEMENT OF W1
C HOMGAF: THE LARGEST ELEMENT OF W1
C NOMGAF: DIMENSION OF W1
C JIN,IOUT: INTEGERS TO IDENTIFY THE PLOT
C
C ********************************************************************
      CALL PLOTYPE(1)
      CALL TKTYPE(4010)
      W=8.67
      H=6.5
      XIN=5.
      XM=1.0
      XMIN=ALOG10(LOMGAF)
      XMAX=ALOG10(HOMGAF)
      XF=XIN/(XMAX-XMIN)
      YIN=5.
      YM=0.75
      CALL TEKPAUS
      PRINT*,'      JIN=',JIN,'      IOUT=',IOUT
      CALL SIZE(W,H)
      TMAX=PSD(1)
      DO 100 I=2,NOMGAF
100     IF(PSD(I).GT.TMAX) TMAX=PSD(I)
```

```
      TMIN=TMAX/100000.
      CALL RANGEL(TMIN,TMAX,TMINR,TMAXR)
      YMIN=ALOG10(TMINR)
      YMAX=ALOG10(TMAXR)
      YF=YIN/(YMAX-YMIN)
      HB=YM+YIN+YF*YMIN+XF*XMIN
      VB=XM-XF*XMIN+YF*YMIN
      CALL ROTATE(90.)
      CALL SCALE(XF,YF,HB,VB,XMIN,YMIN)
      X=XMIN+.2/XF
      Y=YMAX+(YM-.1)/YF
      CALL SYMBOL(X,Y,0.,.1,40,LABLE)
      CALL AXISL(LOMGAF,HOMGAF,LOMGAF,TMINR,TMAXR,TMINR,1.,1.
     +                           ,0,0,1,1,1.,1.,0.1,3)
      CALL WINDOW(XMIN,YMIN,XMAX,YMAX)
      CALL VECTORS
      IP=0
      DO 200 I=1,NOMGAF
       X=W1(I)
       Y=ALOG10(PSD(I))
       CALL PLOT(X,Y,IP,0)
200    IP=1
      CALL PLOTEND
      CALL WINDOW(0.,0.,0.,0.)
      RETURN
      END
```

# Appendix B

# Numerical Values

## B-1. Numerical Values for 5th Order Systems

The values of parameters used are as follows:

$I = 602$ lbf s /rad $\qquad c = 6.9$ lbf s/ft

$K = 46.10$ lbf $\qquad R = 16.67$ ft

$f_0 = 7.65$ rad/sec $\qquad V_W =$

For the parameters $a_i$ and $b_i$ the procedure given in the paper by Holley et. al.[1], was used,

$a_1 = -.073 \qquad b_1 = 2.578$

$a_1 = .488 \qquad b_2 = b_3 = 1.55$

power spectral density matrix, Q of the input white rouse is a diagonal matrix with diagonal elements all equal to .085 and the constant system equations are:

$$\{\dot{x}\} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -823.5 & -1.28 & .113 & 1.28 & 0 \\ 0 & 0 & -.073 & 0 & 0 \\ 0 & 0 & 0 & -.488 & 7.68 \\ 0 & 0 & 0 & -7.68 & -.488 \end{bmatrix} \{x\}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2.578 & 0 & 0 \\ 0 & 1.55 & 0 \\ 0 & 0 & 1.55 \end{bmatrix} \{w\}$$

$$\{y\} = [1 \quad 0 \quad 0 \quad 0 \quad 0] \{x\}$$

[1] Holley, W. E., Thresher, R. W., Lin, S. R.; Wind Turbulance Inputs For Horizontal Axis Wind Turbine; Department of Mechanical Engineering, Oregon State University, 1981.

The periodic system equations are:

$$\{\dot{x}\} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -832.505 & -1.28 & .113 & 1.28\sin(7.68t) & 1.28\cos(7.68t) \\ 0 & 0 & -.073 & 0 & 0 \\ 0 & 0 & 0 & -.488 & 0 \\ 0 & 0 & 0 & 0 & -.488 \end{bmatrix} \{x\} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2.578 & 0 & 0 \\ 0 & 1.55 & 0 \\ 0 & 0 & 1.55 \end{bmatrix} \{w\}$$

$$\{y\} = [1 \quad 0 \quad 0 \quad 0 \quad 0]\ \{x\}$$

## B-2. Numerical Values for 19th Order Systems

The state variables are defined as follows for periodic coefficient system:

$x_1$ = U = horizontal displacement of the top of the tower.<br>
$x_2$ = V = lateral displacement of the top of the tower.<br>
$x_3$ = $\phi$ = nacelle yaw angle.<br>
$x_4$ = $\chi$ = nacelle pitch angle.<br>
$x_5$ = $\psi$ = azimuthal angular position of a rotor blade.<br>
$x_6$ = $\dot{x}_1$<br>
$x_7$ = $\dot{x}_2$<br>
$x_8$ = $\dot{x}_3$<br>
$x_9$ = $\dot{x}_4$<br>
$x_{10}$= $\dot{x}_5$<br>
$x_{11}$= $V_x$ = uniform wind velocity, x-direction.<br>
$x_{12}$= $V_y$ = uniform wind velocity, y-direction.<br>
$x_{13}$= $V_z$ = uniform wind velocity, z-direction.<br>
$x_{14}$= $V_{y,x}$ = velocity gradient across rotor disk.<br>
$x_{15}$= $V_{y,z}$ = velocity gradient across rotor disk.<br>
$x_{16}$= $\gamma_{zx}$ = swirl about mean wind axis.

$$\left.\begin{array}{l} x_{17} = \varepsilon_{zx} = \\ x_{18} = \bar{\gamma}_{zx} = \end{array}\right\}\text{in plane shear strain rates}$$

$x_{19}$= $\varepsilon_{zx}$ = in plane dilation.

Power spectral density of white noise excitations are Q = .11837 sec.

The power spectral density matrix of the white nois is diagonal with diagonal elements equal to .11835. The system matrices are given as follows:

Matrix A

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1.0000 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1.0000 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0000 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0000 |
| -9.2064 | 0 | -657.32 | -.14303 | 0 | .299E-3 | 0 | -3.6384 | -24.998 | 0 |
| 0 | -18.750 | -.15849 | -886.57 | 0 | .793E-2 | -.0828 | -26.459 | 3.9691 | 1.8183 |
| -.676 | 0 | -84.770 | -.0202 | 0 | .289E-3 | 0 | -.50749 | -3.3886 | 0 |
| 0 | -2.0194 | .141 | -165.77 | 0 | -.705E-3 | .741E-3 | 2.3503 | -.35256 | -.0163 |
| 0 | 0 | 0 | 0 | -.80993 | 0 | -.139E-2 | 0 | 0 | -.0162 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| -.299E-3 | 0 | .613E-2 | 10.759 | 4.3348 | 0 | | | 0 |
| -.793E-2 | .23917 | -.237E-2 | -.34153 | 11.833 | 1.8183 | a(t) | | -.57574 |
| -.289E-3 | 0 | .101E-2 | 1.5123 | .732 | 0 | | | 0 |
| .705E-3 | -.214E-2 | .210E-3 | .0303 | -1.0511 | -.163 | (see below) | | .515 |
| 0 | .404E-2 | 0 | 0 | 0 | -.162 | | | -.101 |
| -.0862 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | -.0401 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | -.0862 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | -.0937 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | -.0937 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | -.13147 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | -.168 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | -.168 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -2.8245 |

Matrix B

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2.3767 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.5615 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2.3767 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & .0111 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & .0111 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & .0109 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .994E-2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .994E-2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0275
\end{bmatrix}$$

Matrix C

$$\begin{bmatrix}
0 & .6589E+6 & 0 & .4372E+8 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$[a(t)] = \begin{bmatrix}
-.8615\ \cos(5.498t) - .4253\ \sin(5.498t) & -.8615\ \sin(5.498t) + .4253\ \cos(5.498t) \\
.2910\ \cos(5.498t) - .8462\ \sin(5.498t) & .2910\ \sin(5.458t) + .8462\ \cos(5.498t) \\
-.1088\ \cos(5.498t) - .0384\ \sin(5.498t) & -.1088\ \sin(5.498t) + .0384\ \cos(5.498t) \\
-.0259\ \cos(5.498t) + .0752\ \sin(5.498t) & -.0259\ \cos(5.498t) - .0752\ \cos(5.498t) \\
0 & 0
\end{bmatrix}$$