NASA Contractor Report 159356

ROTOR DYNAMIC SIMULATION AND SYSTEM IDENTIFICATION METHODS FOR APPLICATION TO VACUUM WHIRL DATA

A. Berman
N. Giansante
W. G. Flannelly

KAMAN AEROSPACE CORPORATION
Bloomfield, Connecticut 06002

CONTRACT NAS 1 13710
September 1980

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

SYMBOLS*

A	blade cross-sectional area, coefficient matrix
B_1^*, B_2^*, C_1, C_1^*	blade cross-sectional integrals (see Ref. 3)
BF	vector of applied forces to blade, defined after Equation (34)
BIN, BDAM, BSPR	matrices in hub equations, defined after Equation (34)
BIRI, BIRID, BIRIO, BIRIDH, BIRIIH	matrices defined after Equation (34)
$C_{H_x}, C_{H_y}, C_{\alpha_x}, C_{\alpha_y}$	effective hub damping coefficients
CIB	blade coordinate transformation matrix, defined after Equation (34)
COIR, COIH, CODR, CODH, COR	blade equation matrices, defined after Equation (34)
DYYI, DYYII, DZZII, etc	definite integrals defined in Appendix A
E	Young's modulus
E_1	$= e_A EAK_A{}^2 - EB_2^*$
E_v	effective in-plane stiffness $= EI_z' - (EI_z'-EI_y')\theta^2 - e_A{}^2 EA$
E_w	effective out-of-plane stiffness $= EI_y' + (EI_z'-EI_y')\theta^2 - e_A{}^2 EA\theta$
E_ϕ	effective torsional stiffness $= GJ - K_A{}^4 EA\theta'^2 + EB_1^*\theta'^2 + K_A{}^2\,\Omega^2\;\tau_I\phi'$
e	mass centroid offset from elastic axis, positive when centroid is forward
e_A	area centroid offset from elastic axis, positive when centroid is forward

---
* Most symbols relating to blade parameters are consistent with the notation of Reference 3.

$F_{H_x}, F_{H_y}, F_{H_z}, F_{\alpha_x}, F_{\alpha_y}$	applied forces and moments at hub
FNL	vector of nonlinear terms, defined after Equation (34)
FR	vector of steady forces due to offsets, defined after Equation (34)
G	shear modulus
g_v, g_w, g_Φ	blade inplane, out of plane, torsion damping, force/unit length/unit velocity
HC,HF,HK	hub damping, force, and stiffness matrices, defined after Equation (34)
I	as used in EI, appropriate area moment of inertia
IB	index referring to a particular blade of the rotor
$I_{y'}, I_{z'}$	blade section moments of inertia from y' and z' axes
$I_{\alpha_x}, I_{\alpha_y}$	effective moments of inertia of hub
K_A	area radius of gyration of blade cross-section
K_m, K_{m_1}, K_{m_2}	mass radius of gyration of blade cross-section, polar, from chord, from axis through c.g. perpendicular to chord.
K_{H_x}, K_{H_y},etc	effective stiffness of hub
L_u, L_v, L_w	components of applied forces to blade in u,v,w coordinate system.
m	blade mass per unit length
m_{H_x}, m_{H_y},etc	effective hub masses
$\bar{M}$	vector of elements of mass matrix
$\bar{M}_A$	vector of elements of approximate mass matrix

SYMBOLS (Continued)

NB	number of blades
NY,NZ,NP	number of in-plane, out-of-plane, torsion modes, respectively
NT	total number of modes used = NY + NZ + NP
NX	number of blade stations
$\bar{r}$	right-hand side vector
R	value of x at blade tip, blade radius
RIOC	inverse of blade inertial coefficient matrix, COIR
SIB	blade coordinate transformation matrix, defined after Equation (34)
t	time
T	tension, also kinetic energy
TM	hub inertial matrix, defined after Equation (34)
u,v,w	elastic displacements in radial, in-plane, and out-of-plane directions
$\bar{v},\bar{w},\bar{\phi}$	vector components of coupled blade normal modes, ψ
w_i	weighting factor on i-th variable
W	weighting matrix
x	blade station, measured from hub
x,y,z	blade displacement from undeformed blade coordinates
x_H,y_H,z_H	coordinates of hub in inertial reference system, Figure 2
x_R,y_R,z_R	non-rotating blade coordinates with origin at hub, Figure 2
y_i,z_i,ϕ_i	generalized coordinates, amplitudes of i-th in-plane, out-of-plane, and torsion modes in Galerkin method, functions of time only
Y_i,Z_i,Φ_i	modal functions used in Galerkin method, function of x only

YI,ZI,PI	integrals defined in Appendix A
Y_{z_p}	vector of blade generalized coordinates
α_x, α_y	pitch and roll angles of hub
β_{pc}	precone angle
ΔE	$EI_{z'} - EI_{y'} - e_A{}^2 EA$
ΔK	$K_{m_2}{}^2 - K_{m_1}{}^2$
$\overline{\Delta m}$	vector of changes in elements of mass matrix
η	blade section coordinate
θ	built-in twist
ξ	dummy variable for blade station
τ	centrifugal tension integral = $\int_x^R m\xi d\xi$
ϕ	elastic twist about elastic axis
$\bar{\phi}$	vector torsional component of coupled blade normal mode
ϕ_i	generalized coordinate, amplitude of i-th torsion mode in Galerkin method
Φ_i	i-th torsional mode used in Galerkin method
ψ	blade azimuth
Ψ	vector of coupled blade normal modes
ω	blade natural frequency
ω_f	frequency of forcing function
Ω	blade rotational speed

$\int$	for simplicity, often used to indicate $\int_x^R (\ )dx$
$(\dot{\ })$	$\frac{\partial}{\partial t}(\ )$
$(\ )'$	$\frac{\partial}{\partial x}(\ )$

INTRODUCTION

The analysis of rotor dynamic and aeroelastic phenomena and the resulting capability to control and modify undesirable characteristics requires an understanding of the dynamics and aerodynamics of the rotor blade. Much of the theoretical and experimental research efforts have centered on the aerodynamic aspects of the problem. Of the recent work done in the field of rotor dynamics, most has been directed toward particular phenomena using idealized blade models. Little effort has been devoted to the development of methods of analyzing the dynamic characteristics of actual rotors.

The ability to analyze and predict the dynamic characteristics of a rotor blade has rarely been adequately tested. Non-rotating tests and rotating tests in the atmosphere omit the extreme structural operating conditions associated with the large centrifugal forces or involve significant aerodynamic effects which cannot be analytically removed. One attempt (Reference 1) to test an idealized rotor model in a vacuum chamber resulted in the conclusion that the state-of-the-art of rotor dynamic analysis was not adequate for even a simple solid homogeneous uniform blade with a rectangular cross-section.

There are reasons why there are considerable uncertainties in the mathematical modeling of a rotor blade. In addition to the extreme centrifugal field effects, the major problem lies in the representation of the blade section properties. The state-of-the-art methods (for example, Reference 3) apply to blades with homogeneous sections. In actuality, a typical rotor blade will contain many of the following features: a tapered, twisted hollow spar; bonded thin skinned pockets with ribs or a honeycomb filler; leading edge balance weights; a bonded anti-icing boot; inboard stiffeners; multiple hinges; root cutout. The analytic determination of "effective stiffness", "elastic axis", and "structural damping coefficient" are, at best, intuitive approximations.

The vacuum chamber rotor testing planned at Langley Research Center offers a unique opportunity to significantly advance the state-of-the-art of rotor analytic modeling and rotor dynamic analysis. The purpose of the work presented in this report is to develop tools to augment the aforementioned testing program. Two specific computer programs have been developed. The V22 program has been developed to simulate the tests, including all the necessary special characteristics such as hub forcing, and independent rotational and forcing frequencies, including the non-rotating condition. In addition, the program was designed to be used as a research tool and emphasizes operational flexibility and ease of data input and solution controls.

The other program, ROTSI, is an attempt to use measured data to help identify better approximations to the mass and offset parameters of the rotor blade. The method is an adaptation of the method of incomplete models which has been used with success for other related structural problems.

The analytical developments necessary to implement these tools are derived and discussed in this report. The programs, operators guides, descriptions of special features, and illustrative computational results are also presented.

The major part of this work was completed in 1977, prior to the actual vacuum chamber tests. After the testing was performed an analysis of this data was carried out and is reported in Appendix D.

The contract research effort which has led to the results in this report was financially supported by the Structures Laboratory, USARTL (AVRADCOM).

EQUATIONS OF MOTION

A comprehensive development of the equations of motion of a rotor blade was first published by Houbolt and Brooks (Reference 2) in 1958. The equations were reformulated by Hodges and Dowell (Reference 3). Their major contributions were the improved generality, including nonlinear terms, and the independent verification of the earlier work. There being no need to rederive these equations again, the rotor equations used in this study were based on those given in Reference 3.

The addition of hub degrees of freedom necessitated the development of the additional terms in the blade equations and the development of the equations of motion of the hub itself which includes the effects of the blades.

The development of the equations of motion of the blades and hub, the application of the Galerkin method, the method of solution, and some of the major features of the program implementing these solutions is presented in the following sections.

ROTOR EQUATIONS

As suggested in Reference 3, the tension, T, and the longitudinal deflection, u, shall be eliminated from the equations. Using the nomenclature as shown in Figure 1 and considering θ and ϕ to be small with ϕ ignored compared to θ in the nonlinear terms, the equation for the tension in the blade becomes: (Equation 62 of Reference 3)

$$T = EA\{u' + \frac{v'^2}{2} + \frac{w'^2}{2} + K_A^2\theta'\phi' - e_A(v'' + w''\theta)\} \tag{1}$$

Integrating with respect to x and solving for u yields:

$$u = \int_0^x u'd\xi = \int_0^x \{\frac{T}{EA} - K_A^2\theta'\phi' + e_A(v'' + \theta_w'')\}d\xi - \int_0^x (\frac{v'^2}{2} + \frac{w'^2}{2})d\xi$$

with boundary condition $u(0) = 0$ (2)

From Reference 3 the equation (Equation 61a) for the elastic displacement in the x direction is:

$$T' = -L_u - m(\Omega^2 x + 2\Omega\dot{v}) \tag{3}$$

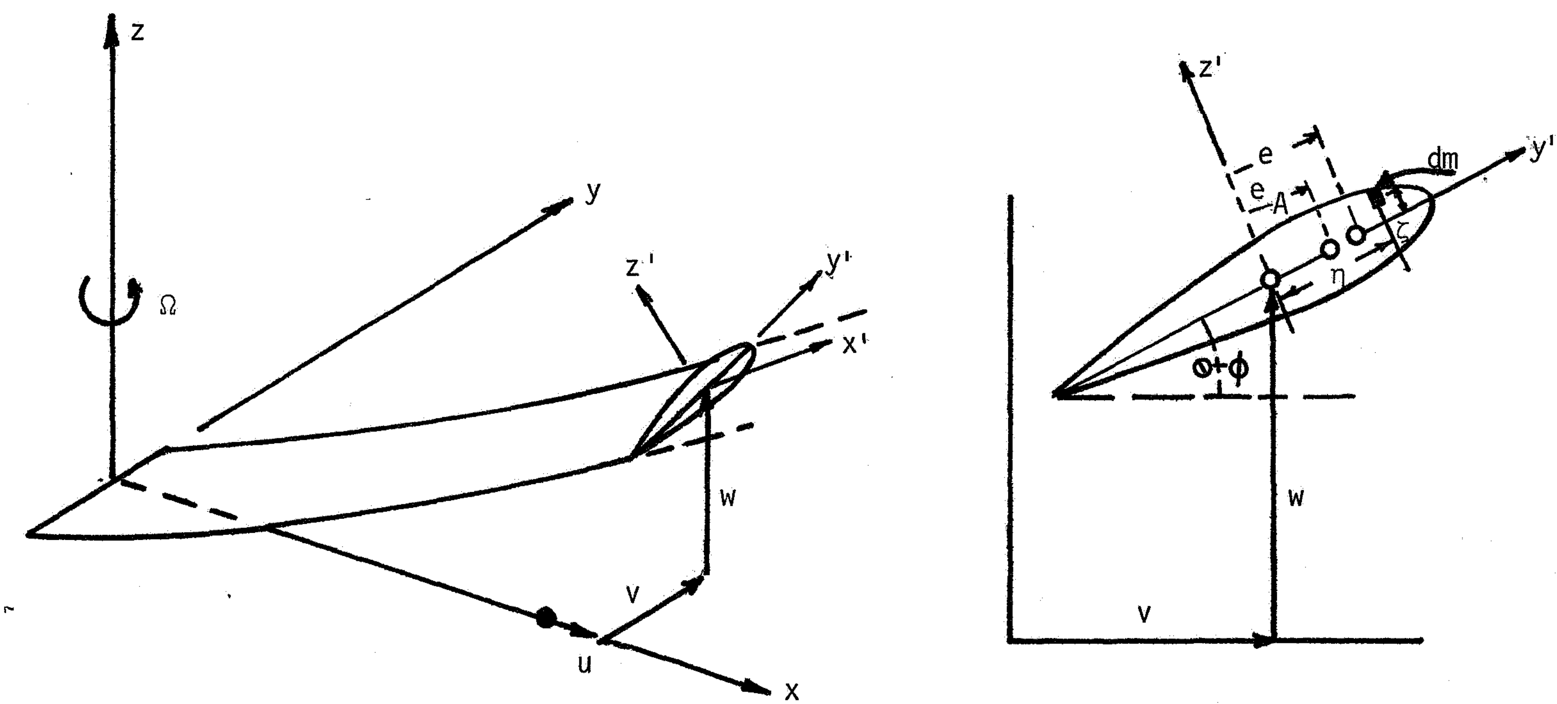


Figure 1. Blade Coordinate System

Integrating Equation (3) and using $L_u = 0$, $T(R) = 0$ and $\tau \equiv \int_x^R m\xi d\xi$ the resulting equation is:

$$T = \Omega^2\tau + 2\Omega\int_x^R m\dot{v}d\xi \tag{4}$$

Equation (3) and (4) and an expression for u' developed from Equation (1) are substituted into the Equations (61b), (61c), (61d) of Reference 3, the equations for the in-plane, out-of-plane, and torsion become (where third and higher order terms have been neglected):

$$\{E_v v'' - 2\Omega e_A\int_x^R m\dot{v}d\xi + \Delta E\theta_w'' - EC_1^*\theta\phi'' + E_1\theta'\phi' - e_A\Omega^2\tau\}'' - \Omega^2\tau v'' + \Omega^2 mxv'$$

$$- \Omega^2 mv + m\ddot{v} - 2\Omega me\dot{v}' + 2\Omega m\dot{v}v' - 2\Omega\int_x^R m\dot{v}d\xi v'' - 2\Omega m\beta_{pc}\dot{w} - 2\Omega me\theta\dot{w}'$$

$$- me\theta\ddot{\phi} - \{me(\Omega^2 x + 2\Omega\dot{v})\}' + 4\Omega^2 m\int_o^x \frac{1}{EA}\int_x^R m\ddot{v}d\xi\, d\xi - 2\Omega m\int_o^x K_A{}^2\theta'\dot{\phi}'d\xi$$

$$+ 2m\Omega\int_o^x e_A\dot{v}''d\xi + 2m\Omega\int_o^x e_A\theta\dot{w}''d\xi - 2m\Omega\int_o^x \dot{v}'v'd\xi - 2m\Omega\int_o^x \dot{w}'w'd\xi = L_v + m\Omega^2 e \tag{5}$$

$$\{\Delta E\theta v'' - 2\Omega e_A\theta\int_x^R m\dot{v}d\xi + E_w w'' + EC_1^*\phi'' + E_1\theta\theta'\phi' - \Omega^2 e_A\tau\phi - \Omega^2 e_A\tau\theta\}''$$

$$+ 2\Omega m\beta_{pc}\dot{v} - \Omega^2\tau w'' + \Omega^2 mxw' + m\ddot{w} + 2\Omega m\dot{v}w' - 2\Omega\int_x^R m\dot{v}d\xi w'' + me\ddot{\phi}$$

$$-\{me(2\Omega\theta\dot{v} + \Omega^2 x\phi + \Omega^2 x\theta)\}' = L_w - m\Omega^2\beta_{pc}x \tag{6}$$

$$-\{E_1\theta'v'' + 2\Omega K_A{}^2\theta'\int_x^R m\dot{v}d\xi + E_1\theta\theta'w'' + E_\phi\phi' + \Omega^2 K_A{}^2\tau\theta'\}' + \Omega^2 e_A\tau\theta v''$$

$$- \Omega^2 mex\theta v' + \Omega^2 me\theta v - me\theta\ddot{v} - \Omega^2 e_A\tau w'' + \Omega^2 mexw' + me\ddot{w} + \Omega^2 m\Delta K\phi$$

$$+ mK_m{}^2\ddot{\phi} + \{-EC_1^*\theta v'' + EC_1^*w'' + EC_1\phi''\}'' = M\phi - \Omega^2 m\Delta K\theta - \Omega^2 me\beta_{pc}x \tag{7}$$

These equations contain spatial derivatives of physical parameters which would be difficult to evaluate numerically. Integrating each equation twice between the limits x to R will eliminate this problem. Using the variable x as the lower limit is the more convenient because of the boundary conditions at the tip of the blade. For example, consider the double integration of functions f"(x) and f'(x) as follows:

$$\int_x^R \int_x^R f''(x)dxdx = f'(R)(R - x) - f(R) + f(x)$$

and

$$\int_x^R \int_x^R f'(x)dxdx = f(R)(R - x) - \int_x^R f(x)dx$$

Following the Galenkin (Ritz) procedure, arbitrary functions for the blade elastic displacements are substituted into the previous equations as follows:

$$v(x_1 t) = \sum_i y(t)Y_i(x) \equiv \sum_i y_i Yi$$

$$w(x_1 t) = \sum_j z_j(t)Z_j(x) \equiv \sum_j z_j Z_j$$

$$\phi(x_1 t) = \sum_k \phi_k(t)\Phi_k(x) \equiv \sum_k \phi_k \Phi_k$$

where $Y_i(x)$, $Z_j(x)$, $\Phi_k(x)$ are modal functions which satisfy the boundary conditions and $y_i(t)$, $z_j(t)$, $\phi_k(t)$ are time dependent generalized co-ordinates. The modal functions are completely general and are not restricted to normal mode shapes.

In the following equations the short-hand notation $\int = \int_x^R (\ )d\xi$ is used for simplicity.

$$\sum_i\{\ddot{y}_i(\int\int mY_i + 4\Omega^2\int\int m\int_o^x \frac{1}{EA}\int mY_i) + 2\Omega\dot{y}_i[\int\int m\int_o^x e_A Y_i'' - \int\int meY_i'$$

$$- e_A\int mY_i - \int meY_i - (R - x)(meY_i)_R] + y_i[E_v Y_i'' - \Omega^2(\int\int\tau Y_i''$$

$$- \int\int mxY_i' + \int\int mY_i)]\} + \sum_j\{2\Omega\dot{z}_j(\int\int m\int_o^x e_A\theta Z_j'' - \int\int me\theta Z_j' - \beta_{pc}\int\int mZ_j)$$

$$+ z_j(\Delta E\theta Z''_j)\} + \sum_k\{\ddot{\phi}_k(-\int\int me\theta\Phi_k) - 2\Omega\dot{\phi}_k(\int\int m\int_o^x K_A{}^2\theta\Phi_k') + \phi_k(- EC_1{}^*\theta\Phi_k''$$

$$+ E_1\theta'\Phi_k')\} + 2\Omega\{\int\int m\dot{v}v' - \int\int v''\int m\dot{v} - \int\int m\int_o^x \dot{v}'v' - \int\int m\int_o^x \dot{w}'w'$$

$$= \int\int(L_v - m\Omega^2 e) - \Omega^2(\int mex - e_A\tau - R(me)_R(R - x)) \tag{8}$$

$$\sum_i\{2\Omega\dot{y}_i[\beta_{pc}\int\int mY_i + \int me\theta Y_i - e_A\theta\int mY_i - (R - x)(me\theta Y_i)_R] + y_i(\Delta E\theta Y_i'')\}$$

$$+ \sum_j\{\ddot{z}_j(\int\int mZ_j) + z_i[E_w Z_j'' - \Omega^2(\int\int\tau Z_j'' - \int\int mxZ_j')]\} + \sum_k\{\ddot{\phi}_k(\int\int me\Phi_k)$$

$$+ \phi_k[EC_1{}^*\Phi_k'' + E_1\theta\theta'\Phi_k' + \Omega^2(\int mex\Phi_k - e_A\tau\Phi_k - R(R - x)(me\Phi_k)_R)]\}$$

$$+ 2\Omega\{\int\int m\dot{v}w' - \int\int w''\int m\dot{v} = \int\int(L_w - \Omega^2\beta_{pc}mx) - \Omega^2[\int mex\theta - e_A\tau\theta$$

$$- R(me\theta)_R(R - x)] \tag{9}$$

$$\sum_i\{- \ddot{y}_i(\int\int me\theta Y_i) + 2\Omega\dot{y}_i[\int(k_A{}^2\theta\int mY_i)] + y_i[\int E_1\theta' Y_i'' - EC_1{}^*\theta Y_i''$$

$$+ \Omega^2(\int\int e_A\tau\theta Y_i'' - \int\int mex\theta Y_i' + \int\int me\theta Y_i)]\} + \sum_j\{\ddot{z}_j(\int\int meZ_j)$$

$$+ z_i[\int E_1\theta\theta' Z_j'' + EC_1{}^*Z_j'' - \Omega^2(\int\int e_A\tau Z_j'' - \int mexZ_j')]\} + \sum_k\{\ddot{\phi}_k(\int\int mk_m{}^2\Phi_k)$$

$$+ \phi_k\ (EC_1\Phi_k'' + \int E_\phi\Phi_k' + \Omega^2\int\int m\Delta K\Phi_k)\} = \int\int[M_\phi - \Omega^2(m\Delta K\theta + \beta_{pc}mex)]$$

$$- \Omega^2\int k_A{}^2\tau\theta' \tag{10}$$

ADDITION OF HUB MOTIONS

In this section the linear effects of the hub degrees of freedom are evaluated and will be combined with the blade equations.

The coordinate of a point on a blade in the nonrotating hub system, as shown in Figure 2, can be defined in terms of r, the undeformed reference line along the blade span as follows (including the major linear terms).

$$x_R = r\cos\psi - [v + \eta\cos(\theta + \phi)]\sin\psi$$

$$y_R = r\sin\psi + [(v + \eta\cos(\theta + \phi)]\cos\psi$$

$$z_R = r\beta_{pc} + w + \eta\sin(\theta + \phi) \qquad (11)$$

Assuming small angles for θ and ϕ in Equations (11), including hub displacements and angular motions α_x and α_y about the respective axes, the linear expression for the inertial coordinates for a point on the blade become:

$$x = x_H + r\cos\psi - (\eta + v)\sin\psi + (r\beta_{pc} + \eta\theta)\alpha_y$$

$$y = y_H + r\sin\psi + (\eta + v)\cos\psi - (r\beta_{pc} + \eta\theta)\alpha_x$$

$$z = z_H + r\beta_{pc} + \eta(\theta + \phi) + w + (r\sin\psi + \eta\cos\psi)\alpha_x$$

$$- (r\cos\psi - \eta\sin\psi)\alpha_y \qquad (12)$$

Accelerations of the inertial coordinates are derived from Equation (12) and are used in the formulation of the hub equations, below:

$$\ddot{x} = \ddot{x}_H - \Omega^2(r\cos\psi - \eta\sin\psi) - \ddot{v}\sin\psi - 2\Omega\dot{v}\cos\psi + \Omega^2 v\sin\psi$$

$$+ \eta\theta\ddot{\phi}\sin\psi + 2\eta\Omega\theta\dot{\phi}\cos\psi + (r\beta_{pc} + \eta\theta)\ddot{\alpha}_y \qquad (13)$$

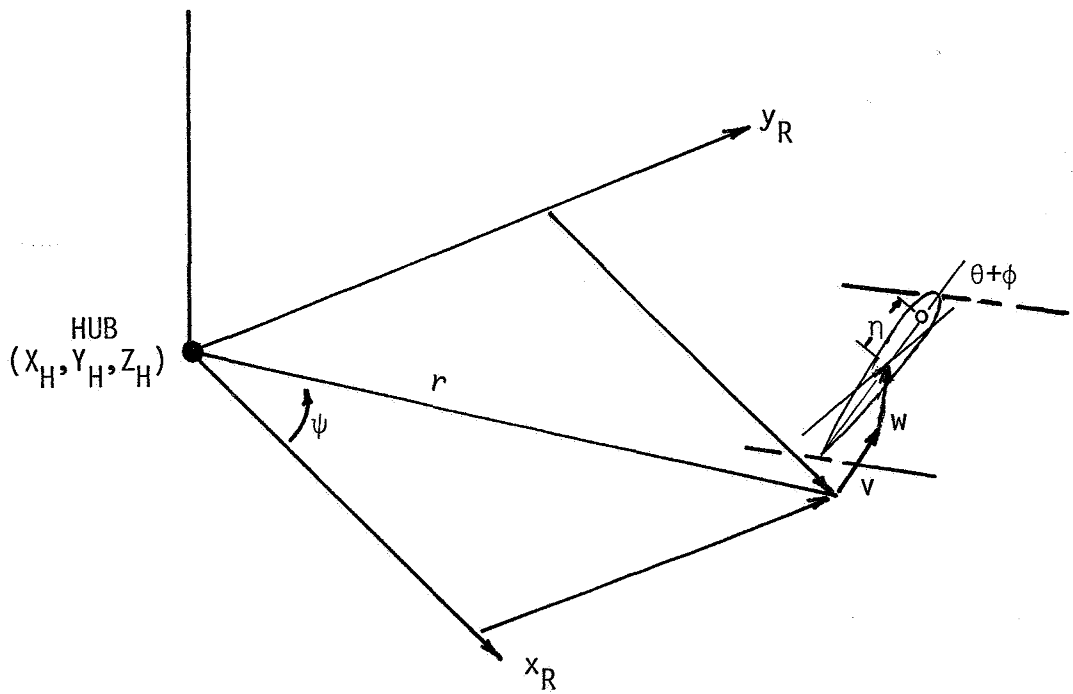


Figure 2. Point on Blade Referenced to Non-Rotating Hub Coordinate System

$$\ddot{y} = \ddot{y}_H - \Omega^2(r \sin\psi + \eta\cos\psi) + \ddot{v}\cos\psi - 2\Omega\dot{v}\sin\psi - \Omega^2 v\cos\psi$$

$$- \eta\theta\ddot{\phi}\cos\psi + 2\eta\Omega\theta\dot{\phi}\sin\psi - (r\beta_{pc} + \eta\theta)\ddot{\alpha}_x \tag{14}$$

$$\ddot{z} = \ddot{z}_H + \ddot{w} + \eta\ddot{\phi} - \Omega^2(r\sin\psi + \eta\cos\psi)\alpha_x + 2\Omega(r\cos\psi - \eta\sin\psi)\dot{\alpha}_x$$

$$+ (r\sin\psi + \eta\cos\psi)\ddot{\alpha}_x + \Omega^2(r\cos\psi - \eta\sin\psi)\alpha_y$$

$$+ 2\Omega(r\sin\psi + \eta\cos\psi)\dot{\alpha}_y - (r\cos\psi - \eta\sin\psi)\ddot{\alpha}_y \tag{15}$$

Applying LaGrange's equation, the <u>additional terms</u> in the equations for the elastic displacements v, w, ϕ <u>due to hub motions</u> become:

<u>v Equation</u>

$$- \ddot{x}_H \sin\psi \int\!\!\int m + \ddot{y}_H \cos\psi \int\!\!\int m + (\ddot{\alpha}_x \cos\psi + \ddot{\alpha}_y \sin\psi)(\beta_{pc} \int\!\!\int mx + \int\!\!\int me\theta) \tag{16}$$

<u>w Equation</u>

$$\ddot{z}_H \int\!\!\int m + (\ddot{\alpha}_x - \Omega^2\alpha_x + 2\Omega\dot{\alpha}_y)(\sin\psi \int\!\!\int mx + \cos\psi \int\!\!\int me)$$

$$- (\ddot{\alpha}_y - \Omega^2\alpha_y - 2\Omega\dot{\alpha}_x)(\cos\psi \int\!\!\int mx - \sin\psi \int\!\!\int me) \tag{17}$$

<u>ϕ Equation</u>

$$[(\ddot{x}_H \sin\psi - \ddot{y}_H \cos\psi) + \Omega(\dot{x}_H \cos\psi - \dot{y}_H \sin\psi)]\int\!\!\int me\theta + \ddot{z}_H \int\!\!\int me$$

$$+ (\ddot{\alpha}_x - \Omega^2\alpha_x + 2\Omega\dot{\alpha}_y)(\sin\psi \int\!\!\int mex + \cos\psi \int\!\!\int mK^2_{m_2}) - (\ddot{\alpha}_y - \Omega^2\alpha_y$$

$$- 2\Omega\dot{\alpha}_x)(\cos\psi \int\!\!\int mex - \sin\psi \int\!\!\int mK^2_{m_2}) + [(\ddot{\alpha}_x \cos\psi + \ddot{\alpha}_y \sin\psi)$$

$$- \Omega(\dot{\alpha}_x \sin\psi - \dot{\alpha}_y \cos\psi)](\beta_{pc} \int\!\!\int mex\theta + \int\!\!\int mK^2_{m_2}\theta) \tag{18}$$

where $\int \equiv \int_x^R (\ )d\xi$

FINAL BLADE EQUATIONS OF MOTION

Combining the respective equations given in (8)-(10) and (16)-(18) yields the equations of motion for the elastic displacements v, w and ϕ.

v Equation

$$\sum_i\{\ddot{y}_i[\int\int mY_i + 4\Omega^2\int\int m\int_o^x \frac{1}{EA}\int mY_i] + 2\Omega\dot{y}_i[\int\int m\int_o^x e_A Y_i'' - \int\int meY_i' - e_A\int mY_i + \int meY_i$$

$$- (R - x)(meY_i)_R] + y_i[E_v Y_i'' - \Omega^2(\int\int \tau Y_i'' - \int\int mxY_i' + \int\int mY_i)]\}$$

$$+ \sum_j\{2\Omega\dot{z}_j[\int\int m\int_o^x e_A\theta Z_j'' - \int\int me\theta Z_j' - \beta_{pc}\int\int mZ_j] + z_j(\Delta E\theta Z_j'')\}$$

$$- \sum_k\{\ddot{\phi}_k\int\int me\theta\Phi_k + 2\Omega\dot{\phi}_k\int\int m\int_o^x K_A^2\theta'\Phi_k' + \phi_k(EC_1{*}\theta\Phi_k'' - E_1\theta'\Phi_k')\} - \ddot{x}_H \sin\psi\int\int m$$

$$\ddot{y}_H \cos\psi \int\int m + \ddot{\alpha}_x \cos\psi(\beta_{pc}\int\int mx + \int\int me\theta) + \ddot{\alpha}_y \sin\psi(\beta_{pc}\int\int mx + \int\int me\theta)$$

$$+ 2\Omega\{\int\int m\dot{v}v' - \int\int(v''\int m\dot{v}) - \int\int m\int_o^x \dot{v}'v' - \int\int m\int_o^x \dot{w}'w'\} = \int\int L_v + \Omega^2\int\int me$$

$$- \Omega^2\int mex + \Omega^2[e_A\tau + (me)_R R(R - x)] \qquad (19)$$

w Equation

$$\sum_i\{2\Omega\dot{y}_i[\beta_{pc}\int\int mY_i + \int me\theta Y_i - E_A\theta\int mY_i - (R - x)(me\theta Y_i)_R] + y_i\Delta E\theta Y_i''\}$$

$$+ \sum_j\{\ddot{z}_j\int\int mZ_j + z_j[E_w Z_j'' - \Omega^2(\int\int \tau Z_j'' - \int\int mxZ_j')]\} + \sum_k\{\ddot{\phi}_k\int\int me\Phi_k$$

$$+ \phi_k[EC_1{*}\Phi_k'' + E_1\theta\theta'\Phi_k' + \Omega^2(\int mex\Phi_k - e_A\tau\Phi_k - R(R - x)(me\Phi_k)_R)]\}$$

$$+ \ddot{z}_H\int\int m + \ddot{\alpha}_x(\sin\psi\int\int mx + \cos\psi\int\int me) + 2\Omega\dot{\alpha}_x(\cos\psi\int\int mx - \sin\psi\int\int me)$$

$$- \Omega^2\alpha_x(\sin\psi\int\int mx + \cos\psi\int\int me - \ddot{\alpha}_y(\cos\psi\int\int mx - \sin\psi\int\int me)$$

$$+ 2\Omega\dot{\alpha}_y(\sin\psi\int\int mx + \cos\psi\int\int me) + \Omega^2\alpha_y(\cos\psi\int\int mx - \sin\psi\int\int me)$$

$$+ 2\Omega\{\int\int m\dot{v}w' - \int\int(w''\int m\dot{v})\} = \int\int L_w - \Omega^2\beta_{pc}\int\int mx - \Omega^2\int mex\theta + \Omega^2[e_A\tau\theta$$

$$+ R(me\theta)_R(R-x)] \tag{20}$$

<u>ϕ Equation</u>

$$\sum_i\{ - \ddot{y}_i\int\int me\theta Y_i + 2\Omega y_i[\int(K_A{}^2\theta'\int mY_i)] + y_i[\int E_1\theta' Y_i'' - EC_1{}^*\theta Y_i''$$

$$+ \Omega^2(\int\int e_A\tau\theta Y_i'' - \int\int mex\theta Y_i' + \int\int me\theta Y_i)]\} + \sum_j\{\ddot{z}_j\int\int meZ_j + z_j[\int E_1\theta\theta' Z_j''$$

$$+ EC_1{}^*Z_j'' - \Omega^2(\int\int e_A\tau Z_j'' - \int\int mexZ_j')]\} + \sum_k\{\ddot{\phi}_k\int\int mK^2{}_m\Phi_k + \phi_k[EC_1\Phi_k''$$

$$+ \int E_\phi\Phi_k' + \Omega^2(\int\int m\Delta K\Phi)] + \ddot{x}_H\sin\psi\int\int me\theta + \Omega\dot{x}_H\cos\psi\int\int me\theta$$

$$- \ddot{y}_H\cos\psi\int\int me\theta - \Omega\dot{y}_H\sin\psi\int\int me\theta + \ddot{z}_H\int\int me + \ddot{\alpha}_x[\sin\psi\int\int mex + \cos\psi(\int\int mK^2_{m_2}$$

$$+ \beta_{pc}\int\int mex\theta + \int\int mk^2_{m_2}\theta)] + \Omega\dot{\alpha}_x[2\cos\psi\int\int mex - \sin\psi(2\int\int mK^2_{m_2} + \beta_{pc}\int\int mex\theta$$

$$+ \int\int mK^2_{m_2}\theta)] - \Omega^2\alpha_x(\sin\psi\int\int mex + \cos\psi\int\int mK^2_{m_2}) + \ddot{\alpha}_y[-\cos\psi\int\int mex$$

$$+ \sin\psi(\int\int mK^2_{m_2} + \beta_{pc}\int\int mex\theta + \int\int mK^2_{m_2}\theta)] + \Omega\dot{\alpha}_y[2\sin\psi\int\int mex + \cos\psi(2\int\int mK^2_{m_2}$$

$$+ \beta_{pc}\int\int mex\theta + \int\int mK^2_{m_2}\theta)] + \Omega^2\alpha_y(\cos\psi\int\int mex - \sin\psi\int\int mK^2_{m_2})$$

$$= \int\int M_\phi - \Omega^2(\int\int m\Delta K\theta + \beta_{pc}\int\int mex + \int K_A{}^2\tau\theta') \tag{21}$$

The Galerkin (Ritz) method of effecting approximate solutions of differential equations applied to the previous equations requires a set of averaging integrals. Equations (19) - (21) for v, w and ϕ are multiplied by Y_i, Z_j, and Φ_k, respectively, where i = 1, NY; j = 1, NZ and k = 1, NP and each resulting equation is integrated from 0 to R. This procedure yields NT equations (NT = NY + NZ + NP) which may be solved for the generalized coordinates.

$$\sum_{J=1}^{NY} \{\ddot{y}_J[DYYII(I,J,1) + 4\Omega^2 DYSI(I,J,1)] + 2\Omega\dot{y}_J[DYSI(I,J,2) - DYYII(I,J,5)$$

$$- DYF(I,J,2) + DYYI(I,J,2) - DYF(I,J,1)] + y_J[DYF(I,J,3)$$

$$- \Omega^2 (DYYII(I,J,7) - DYYII(I,J,4) + DYYII(I,J,1))]\}$$

$$+ \sum_{J=1}^{NZ} \{2\Omega\dot{z}_J[DYSI(I,J,3) - DYZII(I,J,5) - \beta_{pc}DYZII(I,J,1)] + z_J DYF(I,J,4)\}$$

$$+ \sum_{J=1}^{NP} \{-\ddot{\phi}_J DYPII(I,J,3) - 2\Omega\dot{\phi}_J DYSI(I,J,4) + \phi_J DYF(I,J,5)\}$$

$$- \ddot{x}_H \sin\psi\, DYMII(I,1) + \ddot{y}_H \cos\psi\, DYMII(I,1) + \ddot{\alpha}_x \cos\psi[DYMII(I,5)$$

$$+ \beta_{pc}DYMII(I,2)] + \ddot{\alpha}_y \sin\psi[DYMII(I,5) + \beta_{pc}DYMII(I,2)]$$

$$+ 2\Omega\{\int_0^R Y_I \int\int m\dot{v}v' - \int_0^R Y_I \int\int(v''\int m\dot{v}) - \int_0^R Y_I \int\int m \int_0^x \dot{v}'v' - \int_0^R Y_I \int\int m \int_0^x \dot{w}'w'\}$$

$$= \int_0^R Y_I \int\int L_v + \Omega^2\{DYMII(I,3) - DYMI(I,4) + DYF(I,1,6)\} \qquad (22)$$

where I = 1 to NY; thus, there is one equation for each in-plane mode. Similarly, for the w equation:

$$\sum_{J=1}^{NY} \{2\Omega\dot{y}_J[DZYI(I,J,3) - DZF(I,J,1) + \beta_{pc}DZYII(I,J,1)] + y_J DZF(I,J,2)\}$$

$$+ \sum_{J=1}^{NZ} \{\ddot{z}_J\, DZZII(I,J,1) + z_J[DZF(I,J,3) - \Omega^2 \langle DZZII(I,J,6) - DZZII(I,J,3)\rangle]\}$$

$$+ \sum_{J=1}^{NP} \{\ddot{\phi}_J DZPII(I,J,1) + \phi_J[DZF(I,J,4) + \Omega^2 (DZPI(I,J,2) - DZF(I,J,6))]\}$$

$$+ \ddot{z}_H DZMII(I,1) + \ddot{\alpha}_x[\sin\psi\, DZMII(I,2) + \cos\psi\, DZMII(I,3)]$$

$$+ 2\Omega\dot{\alpha}_x[\cos\psi\, DZMII(I,2) - \sin\psi\, DZMII(I,3)] - \Omega^2\alpha_x[\sin\psi\, DZMII(I,2)$$

$$+ \cos\psi\, DZMII(I,3)] - \ddot{\alpha}_y[\cos\psi\, DZMII(I,2) - \sin\psi\, DZMII(I,3)]$$

$$+ 2\Omega\dot{\alpha}_y[\sin\psi\, DZMII(I,2) + \cos\psi\, DZMII(I,3)] + \Omega^2\,\alpha_y[\cos\psi\, DZMII(I,2)$$

$$- \sin\psi\, DZMII(I,3)] + 2\Omega\{\int_0^R Z_I \int\int m\dot{v}w' - \int_0^R Z_I \int\int (w''\int m\dot{v})\} = \int_0^R Z_I \int\int L_w$$

$$- \Omega^2[DZMI(I,6) - DZF(I,1,5) + \beta_{pc} DZMII(I,2)] \qquad (23)$$

where I = 1 to NZ; following the same procedure, w, the ϕ equation is

$$\sum_{J=1}^{NY} \{- \ddot{y}_J DPYII(I,J,3) + 2\Omega\dot{y}_J DPSI(I,J,5) + y_J[DPYI(I,J,9) - DPF(I,J,1)$$

$$+ \Omega^2(DPYII(I,J,8) - DPYII(I,J,6) + DPYII(I,J,3))]\}$$

$$+ \sum_{J=1}^{NZ} \{\ddot{z}_J DPZII(I,J,2) + z_J[DPZI(I,J,8) + DPF(I,J,3) - \Omega^2\,(DPZII(I,J,7)$$

$$- DPZII(I,J,4))]\}$$

$$+ \sum_{J=1}^{NP} \{\ddot{\phi}_J DPPII(I,J,4) + \phi_J\,[DPF(I,J,3) + DPPI(I,J,6) + \Omega^2(DPPII(I,J,5)$$

$$+ DPPI(I,J,7))]\} + \{\ddot{x}_H \sin\psi + \Omega\dot{x}_H \cos\psi - \ddot{y}_H \cos\psi - \Omega\dot{y}_H \sin\psi$$

$$+ \ddot{z}_H\} DPMII(I,3) + \ddot{\alpha}_x[\sin\psi\, DPMII(I,4) + \cos\psi(DPMII(I,7) + DPMII(I,8)$$

$$+ \beta_{pc} DPMII(I,6))] + 2\Omega\dot{\alpha}_x[\cos\psi DPMII(I,4) - \sin\psi(DPMII(I,7) + \frac{1}{2} DPMII(I,8)$$

$$+ \frac{1}{2}\beta_{pc} DPMII(I,6)\,] - \Omega^2\alpha_x[\sin\psi\, DPMII(I,4) + \cos\psi\, DPMII(I,7)]$$

$$+ \ddot{\alpha}_y[- \cos\psi \, DPMII(I,4) + \sin\psi \, (DPMII(I,7) + DPMII(I,8) + \beta_{pc} DPMII(I,6))]$$

$$+ 2\Omega\dot{\alpha}_y[\sin\psi \, DPMII(I,4) + \cos\psi \, (DPMII(I,7) + \frac{1}{2} DPMII(I,8)$$

$$+ \frac{1}{2} \beta_{pc} DPMII(I,6)] + \Omega^2\alpha_y[\cos\psi \, DPMII(I,4) - \sin\psi \, DPMII(I,7)]$$

$$= \int_0^R \Phi_I \int\int M_\phi - \Omega^2[DPMII(I,9) + \beta_{pc} DPMII(I,4) + DPMI (I,10)] \qquad (24)$$

where I = 1 *to* NP. The coefficients shown in Equations (22), (23) and (24) are defined in Appendix A.

Equations (22), (23) and (24) may be written in partitioned matrix form as shown on the following pages.

In order to include a simple structural damping representation, terms of the form $g_v\dot{v}$, $g_w\dot{w}$, $g_\Phi\dot{\phi}$ were added to Equations (5), (6), (7) resulting in the integrals DYD, DZD, DPD which appear in the following pages and are defined in Appendix A.

$$
\begin{bmatrix}
DYYII(I,J,1) + 4\Omega^2 DYSI(I,J,1) & 0 & -\,DYPII(I,J,3) \\
0 & DZZII(I,J,1) & DZPII(I,J,1) \\
-\,DPYII(I,J,3) & DPZII(I,J,2) & DPPII(I,J,4)
\end{bmatrix}
\begin{bmatrix} \ddot{y}_J \\ \ddot{z}_J \\ \ddot{\phi}_J \end{bmatrix}
=
$$

$$
-\begin{bmatrix}
-\,\sin\psi DYMII(I,1) & \cos\psi DYMII(I,1) & 0 & \cos\psi[DYMII(I,5) + \beta_{pc} DYMII(I,2)] & \sin\psi[DYMII(I,5) + \beta_{pc} DYMII(I,2)] \\
0 & 0 & DZMII(I,1) & \sin\psi DZMII(I,2) + \cos\psi DZMII(I,3) & -\,\cos\psi DZMII(I,2) + \sin\psi DZMII(I,3) \\
\sin\psi DPMII(I,3) & -\cos\psi DPMII(I,3) & DPMII(I,3) & \sin\psi DPMII(I,4) + \cos\psi[DPMII(I,7) + DPMII(I,7) + \beta_{pc} DPMII(I,6) & -\,\cos\psi DPMII(I,4) + \sin\psi[DPMII(I,7) + DPMII(I,8) + \beta_{pc} DPMII(I,6)]
\end{bmatrix}
\begin{bmatrix} \ddot{x}_H \\ \ddot{y}_H \\ \ddot{z}_H \\ \ddot{\alpha}_x \\ \ddot{\alpha}_y \end{bmatrix}
$$

$$
+\left\{ -2\Omega
\begin{bmatrix}
DYSI(I,I,2)-DYYII(I,J,5) - DYF(I,J,2)-DYYI(I,J,2) + DYF(I,J,1) & DYSI(I,J,3)-DYZII(I,J,5) - \beta_{pc} DYZII(I,J,1) & -DYSI(I,J,4) \\
DZYI(I,J,3)-DZF(I,J,1) + \beta_{pc} DZYII(I,J,1) & 0 & 0 \\
DPSI(I,J,S) & 0 & 0
\end{bmatrix}
+
\begin{bmatrix}
DYD & 0 & 0 \\
0 & DZD & 0 \\
0 & 0 & DPD
\end{bmatrix}
\right\}
\begin{bmatrix} \dot{y}_J \\ \dot{z}_J \\ \dot{\phi}_J \end{bmatrix}
$$

$$
-2\Omega \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \cos\psi \mathrm{DZMII(I,2)} - \sin\psi \mathrm{DZMII(I,3)} & \sin\psi \mathrm{DZMII(I,2)} \; \cos\psi \mathrm{DZMII(I,3)} \\
\frac{1}{2}\cos\psi \mathrm{DPMII(I,3)} & -\frac{1}{2}\sin\psi \mathrm{DPMII(I,3)} & 0 & \cos\psi \mathrm{DPMII(I,4)} - \sin\psi[\mathrm{DPMII(I,7)} + \frac{1}{2}\mathrm{DPMII(I,8)} + \frac{1}{2}\mathrm{DPMII(I,6)} & \sin\psi \mathrm{DPMII(I,4)} + \cos\psi[\mathrm{DPMII(I,7)} + \frac{1}{2}\mathrm{DPMII(I,8)} + \frac{1}{2}\mathrm{DPMII(I,6)}
\end{bmatrix}
\begin{bmatrix} \dot{x}_H \\ \dot{y}_H \\ \dot{z}_H \\ \dot{\alpha}_x \\ \dot{\alpha}_y \end{bmatrix}
$$

$$
- \begin{bmatrix}
\mathrm{DYF(I,I,3)} - \Omega^2[\mathrm{DYYII(I,J,1)} - \mathrm{DYYII(I,J,4)} + \mathrm{DYYII(I,J,7)} & \mathrm{DYF(I,J,4)} & \mathrm{DYF(I,J,5)} \\
\mathrm{DZF(I,J,2)} & \mathrm{DZF(I,J,3)} + \Omega^2[\mathrm{DZZII(I,J,3)} + \mathrm{DZZII(I,J,6)}] & \mathrm{DZF(I,J,4)} + \Omega^2[\mathrm{DZPI(I,J,2)} - \mathrm{DZF(I,\ J,6)}] \\
\mathrm{DPYI(I,J,9)} - \mathrm{DPF(I,J,1)} + \Omega^2[\mathrm{DPYII(I,J,8)} - \mathrm{DPYII(I,J,6)} + \mathrm{DPYII(I,J,3)}] & \mathrm{DPZI(I,J,8)} + \mathrm{DPF(I,J,2)} + \Omega^2[\mathrm{DPZII(I,J,4)} - \mathrm{DPZII(I,J,7)}] & \mathrm{DPF(I,J,3)} + \mathrm{DPPI(I,J,6)} + \Omega^2[\mathrm{DPPII(I,J,5)} + \mathrm{DPPI(I,J,7)}]
\end{bmatrix}
\begin{bmatrix} \underline{y_J} \\ \underline{z_J} \\ \phi_J \end{bmatrix}
$$

$$
-\Omega^2 \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\sin\psi \mathrm{DZMII(I,2)} - \cos\psi \mathrm{DZMII(I,3)} & \cos\psi \mathrm{DZMII(I,2)} - \sin\psi \mathrm{DZMII(I,3)} \\
0 & 0 & 0 & -\sin\psi \mathrm{DPMII(I,4)} - \cos\psi \mathrm{DPMII(I,7)} & \cos\psi \mathrm{DPMII(I,4)} - \sin\psi \mathrm{DPMII(I,7)}
\end{bmatrix}
\begin{bmatrix} x_H \\ y_H \\ z_H \\ \alpha_x \\ \alpha_y \end{bmatrix}
$$

$$
+\begin{bmatrix}
\int_0^R Y_I \int_x^R \int_x^R L_v - \Omega^2[-DYMII(I,3) + DYMI(I,4) - DYF(I,1,6)] \\
\hline
\int_0^R Z_I \int_x^R \int_x^R L_w - \Omega^2[DZMI(I,6) - DZF(I,1,5) + \beta_{pc}DZMII(I,2)] \\
\hline
\int_0^R \Phi_I \int_x^R \int_x^R M_\phi - \Omega^2[DPMII(I,9) + \beta_{pc}DPMII(I,4)]
\end{bmatrix}
$$

$$
+2\Omega \begin{bmatrix}
\int_0^R Y_I[- \int_x^R \int_x^R m\dot{v}v' + \int_x^R \int_x^R v''\int_0^x m\dot{v} + \int_x^R \int_x^R m\int_0^x \dot{v}'v' + \int_x^R \int_x^R m\int_0^x w'\dot{w}'] \\
\hline
\int_0^R Z_I[- \int_x^R \int_x^R m\dot{v}w' + \int_x^R \int_x^R (w''\int m\dot{v})] \\
\hline
0
\end{bmatrix} \quad (25)
$$

HUB EQUATIONS

Terms In Hub Equations Due to Blade Motions

The kinetic energy of a rotor blade may be expressed as follows:

$$T = \frac{1}{2} \int_0^R (\dot{x}^2 + \dot{y}^2 + \dot{z}^2)dm$$

Assuming a spring-mass-damper model of the hub in each of the three orthogonal directions, and torsional models with respect to the body axes, the hub equations of motion including blade effects are:

$$m_{H_x}\ddot{x}_H + C_{H_x}\dot{x}_H + K_{H_x}x_H + \sum_{b=1}^{NB}\int_o^R m\ddot{x}d\xi = F_{H_x}$$

$$m_{H_y}\ddot{y}_H + C_{H_y}\dot{y}_H + K_{H_y}y_H + \sum_{b=1}^{NB}\int_o^R m\ddot{y}d\xi = F_{H_x}$$

$$m_{H_z}\ddot{z}_H + C_{H_z}\dot{z}_H + K_{H_z}z_H + \sum_{b=1}^{NB}\int_o^R m\ddot{z}d\xi = F_{H_x}$$

$$I_{\alpha_x}\ddot{\alpha}_x + C_{\alpha_x}\dot{\alpha}_x + K_{\alpha_x}\alpha_x + \sum_{b=1}^{NB}\int_o^R m\,\{-\ddot{y}(r\beta_{pc} + \eta\theta) + \ddot{z}(r\sin\psi + \eta\cos\psi)\}d\xi$$

$$= F_{\alpha_x}$$

$$I_{\alpha_y}\ddot{\alpha}_y + C_{\alpha_y}\dot{\alpha}_y + K_{\alpha_y}\alpha_y + \sum_{b=1}^{NB}\int_o^R m\{\ddot{x}(r\beta_{pc} + \eta\theta) - \ddot{z}(r\cos\psi - \eta\sin\psi)\}d\xi$$

$$= F_{\alpha_y}$$

Substituting the expressions for the accelerations of the inertial coordinates from Equations (13)-(15),performing the integration with respect to chord and blade span and assuming two of more symmetrical blades, the previous equations become:

x_H Equation

$$m_{H_x}\ddot{x}_H + C_{H_x}\dot{x}_H + K_{H_x}x_H + NB[MI(1,1)\ddot{x}_H] + \sum_{IB=1}^{NB}\{-\int_o^R m\ddot{v}\sin\psi - 2\Omega\int_o^R m\dot{v}\cos\psi$$

$$+ \Omega^2\int_o^R mv\sin\psi + \int_o^R me\theta\ddot{\phi}\sin\psi + 2\Omega\int_o^R me\theta\dot{\phi}\cos\psi + (\beta_{pc}MI(1,2) + MI(1,5))\ddot{\alpha}_y\}$$

$$= F_{H_x} \tag{26}$$

y_H Equation

$$m_{H_y}\ddot{y}_H + C_{H_y}\dot{y}_H + K_{H_y}y_H + NB[MI(1,1)\ddot{y}_H] + \sum_{IB=1}^{NB}\{\int_0^R m\ddot{v}\cos\psi - 2\Omega\int_0^R m\dot{v}\sin\psi$$

$$- \Omega^2\int_0^R mv\cos\psi - \int_0^R me\theta\ddot{\phi}\cos\psi + 2\Omega\int_0^R me\theta\dot{\phi}\sin\psi - (\beta_{pc}MI(1,2) + MI(1,5))\ddot{\alpha}_x\}$$

$$= F_{H_y} \tag{27}$$

z_H Equation

$$m_{H_z}\ddot{z}_H + C_{H_z}\dot{z}_H + K_{H_z}z_H + NB[MI(1,1)\ddot{z}_H] + \sum_{IB=1}^{NB}\{\int_0^R m\ddot{w} + \int_0^R me\ddot{\phi}\} = F_{H_z} \tag{28}$$

α_x Equation

$$I_{\alpha_x}\ddot{\alpha}_x + C_{\alpha_x}\dot{\alpha}_x + K_{\alpha_x}\alpha_x - NB\{\beta_{pc}MI(1,2) + MI(1,5)\}\ddot{y}_H + \sum_{IB=1}^{NB}\{-(\beta_{pc}\int_0^R mx\ddot{v}$$

$$+ \int_0^R me\theta\ddot{v})\cos\psi + 2\Omega(\beta_{pc}\int_0^R mx\dot{v} + \int_0^R me\theta\dot{v})\sin\psi + \Omega^2(\beta_{pc}\int_0^R mxv$$

$$+ \int_0^R me\theta v)\cos\psi + (\beta_{pc}\int_0^R mxe\theta\ddot{\phi} + \int_0^R me^2\theta^2\ddot{\phi})\cos\psi - 2\Omega(\beta_{pc}\int_0^R mxe\theta\dot{\phi}$$

$$+ \int_0^R me^2\theta^2\dot{\phi})\sin\psi + (\beta_{pc}^2\int_0^R mx^2 + 2\beta_{pc}\int_0^R mex\theta + \int_0^R me^2\theta^2)\ddot{\alpha}_x$$

$$+ \sin\psi\int_0^R mx\ddot{w} + \cos\psi\int_0^R me\ddot{w} + \sin\psi\int_0^R mx\ddot{\phi} + \cos\psi\int_0^R me\ddot{\phi} - \Omega^2(\sin^2\psi\int_0^R mx^2$$

$$+ 2\sin\psi\cos\psi\int_0^R mex + \cos^2\psi\int_0^R me^2)\alpha_x + 2\Omega[\sin\psi\cos\psi\int_0^R mx^2 - (\sin^2\psi$$

$$- \cos^2\psi)\int_0^R mex - \sin\psi\cos\psi\int_0^R me^2]\dot{\alpha}_x + (\sin^2\psi\int_0^R mx^2 + 2\sin\psi\cos\psi\int_0^R mex$$

$$+ \cos^2\psi\int_0^R me^2)\ddot{\alpha}_x + \Omega^2(\sin\psi\cos\psi\int_0^R mx^2 - (\sin^2\psi - \cos^2\psi)\int_0^R mex$$

$$- \sin\psi\cos\psi\int_0^R me^2)\alpha_y + 2\Omega(\sin^2\psi\int_0^R mx^2 + 2\sin\psi\cos\psi\int_0^R mex + \cos^2\psi\int_0^R me^2)\dot{\alpha}_y$$

$$- \Omega^2(\sin\psi\cos\psi\int_0^R mx^2 - (\sin^2\psi - \cos^2\psi)\int mex - \sin\psi\cos\psi\int me^2)\ddot{\alpha}_y\} = F_{\alpha_y} \quad (29)$$

α_y Equation

$$I_{\alpha_y}\ddot{\alpha}_y + C_{\alpha_y}\dot{\alpha}_y + K_{\alpha_y}\alpha_y + NB\{\beta_{pc}MI(1,2) + MI(1,5)\}\ddot{x}_H + \sum_{IB=1}^{NB}\{ - (\beta_{pc}\int_0^R mx\ddot{v}$$

$$+ \int_0^R me\theta\ddot{v})\sin\psi - 2\Omega(\beta_{pc}\int_0^R mx\dot{v} + \int_0^R me\theta\dot{v})\cos\psi + \Omega^2(\beta_{pc}\int_0^R mxv$$

$$+ \int me\theta v)\sin\psi + (\beta_{pc}\int_0^R mex\theta\ddot{\phi} + \int_0^R me^2\theta^2\ddot{\phi})\sin\psi + 2\Omega(\beta_{pc}\int_0^R mex\theta\dot{\phi}$$

$$+ \int_0^R me^2\theta^2\dot{\phi})\cos\psi + (\beta_{pc}{}^2\int_0^R mx^2 + 2\beta_{pc}\int_0^R mex\theta + \int_0^R me^2\theta^2)\ddot{\alpha}_y - \int_0^R mx\ddot{w}\cos\psi$$

$$+ \int_0^R me\ddot{w}\sin\psi - \int_0^R mx\ddot{\phi}\cos\psi + \int_0^R me\ddot{\phi}\sin\psi + \Omega^2(\sin\psi\cos\psi\int_0^R mx^2 + \cos^2\psi\int_0^R mex$$

$$- \sin^2\psi\int_0^R mex - \sin\psi\cos\psi\int_0^R me^2)\alpha_x + 2\Omega(- \cos^2\psi\int_0^R mx^2 + 2\sin\psi\cos\psi\int_0^R mex$$

$$- \sin^2\psi\int_0^R me^2)\dot{\alpha}_x - (\sin\psi\cos\psi\int_0^R mx^2 + \cos^2\psi\int_0^R mex - \sin^2\psi\int_0^R mex$$

$$- \sin\psi\cos\psi\int_0^R me^2)\ddot{\alpha}_x + \Omega^2(- \cos^2\psi\int_0^R mx^2 + 2\sin\psi\cos\psi\int_0^R mex - \sin^2\psi\int_0^R me)\alpha_y$$

$$- 2\Omega(\sin\psi\cos\psi\int_0^R mx^2 + \cos^2\psi\int_0^R mex - \sin^2\psi\int_0^R mex - \sin\psi\cos\psi\int_0^R me^2)\dot{\alpha}_y$$

$$- (- \cos^2\psi\int_0^R mx^2 + 2\sin\psi\cos\psi\int_0^R mex - \sin^2\psi\int_0^R me^2)\ddot{\alpha}_y\} = F_{\alpha_y} \quad (30)$$

Considering only the hub translational equations of motion and following a similar procedure as applied to the blade equations arbitrary functions for the elastic displacements are substituted into Equations (26)-(28) yielding:

x_H Equation

$$m_{H_x}\ddot{x}_H + C_{H_x}\dot{x}_H + K_{H_x}x_H + NB[MI(1,1)\ddot{x}_H] - \sum_{IB=1}^{NB} \sin\psi_{IB} \sum_{J=1}^{NY} YI(I,J,1)\ddot{y}_{J,IB}$$

$$+ \sum_{IB=1}^{NB} \sin\psi_{IB} \sum_{J=1}^{NP} PI(I,J,3)\ddot{\phi}_{J,IB} + 2\Omega \sum_{IB=1}^{NB} \cos\psi_{IB} \sum_{J=1}^{NY} YI(I,J,1)\dot{y}_{J,IB}$$

$$+ \Omega^2 \sum_{IB=1}^{NB} \sin\psi_{IB} \sum_{J=1}^{NY} YI(I,J,1)y_{J,IB} + 2\Omega \sum_{IB=1}^{NB} \cos\psi_{IB} \sum_{J=1}^{NY} PI(I,J,3)\dot{\phi}_{J,IB}$$

$$+ NB[\beta_{pc}MI(1,2) + MI(1,5)]\ddot{\alpha}_y = F_{H_x} \qquad (31)$$

y_H Equation

$$m_{H_y}\ddot{y}_H + C_{H_y}\dot{y}_H + K_{H_y}y_H + NB[MI(1,1)\ddot{y}_H] + \sum_{IB=1}^{NB} \cos\psi_{IB} \sum_{J=1}^{NY} YI(1,J,1)\ddot{y}_{J,IB}$$

$$- \sum_{IB=1}^{NB} \cos\psi_{IB} \sum_{J=1}^{NP} PI(1,J,3)\ddot{\phi}_{J,IB} - 2\Omega \sum_{IB=1}^{NB} \sin\psi_{IB} \sum_{J=1}^{NY} YI(1,J,1)\dot{y}_{J,IB}$$

$$- \Omega^2 \sum_{IB=1}^{NB} \cos\psi_{IB} \sum_{J=1}^{NY} YI(1,J,1)y_{J,IB} + 2\Omega \sum_{IB=1}^{NB} \sin\psi_{IB} \sum_{J=1}^{NP} PI(1,J,3)\dot{\phi}_{J,IB}$$

$$+ NB[\beta_{pc}MI(1,2) + MI(1,5)]\ddot{\alpha}_x = F_{H_y} \qquad (32)$$

z_H Equation

$$m_{H_z}\ddot{z}_H + C_{H_z}\dot{z}_H + K_{H_z}z_H + NB[MI(1,1)\ddot{z}_H] + \sum_{IB=1}^{NB} \sum_{J=1}^{NZ} ZI(1,J,1)\ddot{z}_{J,IB}$$

$$+ \sum_{IB=1}^{NB} \sum_{J=1}^{NP} PI(1,J,1)\ddot{\phi}_{J,IB} = F_{H_z} \qquad (33)$$

Equations (31)-(33) may be solved for the hub accelerations and written in matrix form:

$$
\begin{bmatrix} m_{H_x}+NB\cdot MI(1,1) & 0 & 0 \\ 0 & m_{H_y}+NB\cdot MI(1,1) & 0 \\ 0 & 0 & m_{H_3}+NB\cdot MI(1,1) \end{bmatrix} \begin{bmatrix} \ddot{x}_H \\ \ddot{y}_H \\ \ddot{z}_H \end{bmatrix}
$$

$$
= \sum_{IB=1}^{NB} \begin{bmatrix} \sin\psi_{IB} & 0 & 0 \\ 0 & \cos\psi_{IB} & 0 \\ 0 & & 1 \end{bmatrix} \begin{bmatrix} YI(1,J,1) & 0 & -PI(1,J,3) \\ -YI(1,J,1) & 0 & PI(1,J,3) \\ 0 & -ZI(1,J,1) & -PI(1,J,1) \end{bmatrix} \begin{bmatrix} \ddot{y}_J \\ \ddot{z}_J \\ \ddot{\phi}_J \end{bmatrix}_{IB}
$$

$$
+ \sum_{IB=1}^{NB} 2\Omega \begin{bmatrix} \cos\psi_{IB} & 0 & 0 \\ 0 & \sin\psi_{IB} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} YI(1,J,1) & 0 & -PI(1,J,3) \\ YI(1,J,1) & 0 & -PI(1,J,3) \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{y}_J \\ \dot{z}_J \\ \dot{\phi}_J \end{bmatrix}_{IB}
$$

$$
+ \sum_{IB=1}^{NB} \Omega^2 \begin{bmatrix} \sin\psi_{IB} & 0 & 0 \\ 0 & \cos\psi_{IB} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -YI(1,J,1) & 0 & 0 \\ YI(1,J,1) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_J \\ z_J \\ \phi_J \end{bmatrix}_{IB}
$$

$$
- \begin{bmatrix} C_{H_x} & 0 & 0 \\ 0 & C_{H_y} & 0 \\ 0 & 0 & C_{H_z} \end{bmatrix} \begin{bmatrix} \dot{x}_H \\ \dot{y}_H \\ \dot{z}_H \end{bmatrix} - \begin{bmatrix} K_{H_x} & 0 & 0 \\ 0 & K_{H_y} & 0 \\ 0 & 0 & K_{H_z} \end{bmatrix} \begin{bmatrix} x_H \\ y_H \\ z_H \end{bmatrix} + \begin{bmatrix} F_{H_X} \\ F_{H_y} \\ F_{H_z} \end{bmatrix} \qquad (34)
$$

METHOD OF SOLUTION

The coefficient matrices of Equation (25) with the hub angular motions α_x and α_y omitted may be defined thusly:

$$[COIR] = \left[\begin{array}{c|c|c} DYYII(I,J,1)+4\Omega^2 DYSI(I,J,1) & 0 & -DYPII(I,J,3) \\ \hline 0 & DZZII(I,J,1) & DZPII(I,J,1) \\ \hline -DPYII(I,J,3) & DPZII(I,J,2) & DPPII(I,J,4) \end{array}\right]$$

$$[COIH][SIB] = -\left[\begin{array}{c|c|c} -\sin\psi DYMII(I,1) & \cos\psi DYMII(I,1) & 0 \\ \hline 0 & 0 & DZMII(I,1) \\ \hline \sin\psi DPMII(I,3) & -\cos\psi DPMII(I,3) & DPMII(I,3) \end{array}\right]$$

$$[CODR] = -\left[\begin{array}{l|l|l} DYD+2\Omega\{-DYYI(I,J,2) & 2\Omega\{DYSI(I,J,3) & -2\Omega DYSI(I,J,4) \\ -DYYII(I,J,5) & -DYZII(I,J,5) & \\ +DYF(I,J,1)-DYF(I,J,2) & -\beta_{pc} DYZII(I,J,1)\} & \\ +DYSI(I,J,2)\} & & \\ \hline 2\Omega\{DZYI(I,J,3)-DZF(I,J,1) & 0 & 0 \\ +\beta_{pc} DZYII(I,J,1)\} & & \\ \hline 2\Omega\ DPSI(I,J,5) & 0 & 0 \end{array}\right]$$

$$[CODH][CIB] = -\ \Omega\left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \cos\psi DPMII(I,3) & -\sin\psi DPMII(I,3) & 0 \end{array}\right]$$

$$[COR] = -\left[\begin{array}{c|c|c}
\begin{array}{l}DYF(I,J,3)\\ -\Omega^2\{DYYII(I,J,7)\\ -DYYII(I,J,4)\\ +DYYII(I,J,1)\}\end{array} & DYF(I,J,4) & DYF(I,J,5) \\ \hline
DZF(I,J,2) & \begin{array}{l}DZF(I,J,3)\\ +\Omega^2\{DZZII(I,J,3)\\ -DZZII(I,J,6)\}\end{array} & \begin{array}{l}DZF(I,J,4)\ +\\ +\Omega^2\{DZPI(I,J,2)\\ -DZF(I,J,6)\}\end{array} \\ \hline
\begin{array}{l}DPYI(I,J,9)\\ -DPF(I,J,1)\\ +\Omega^2\{DPYII(I,J,3)\\ -DPYII(I,J,6)\\ +DPYII(I,J,8)\}\end{array} & \begin{array}{l}DPZI(I,J,8)\\ +DPF(I,J,2)\\ +\Omega^2\{DPZII(I,J,4)\\ -DPZII(I,J,7)\}\end{array} & \begin{array}{l}DPF(I,J,3)+DPPI(I,J,6)\\ +\Omega^2\{DPPII(I,J,5)\\ +DPPI(I,J,7)\}\end{array}
\end{array}\right]$$

$$\{FR\} = -\Omega^2\left[\begin{array}{l}
-DYMII(I,3) + DYMI(I,4) - DYF(I,1,6) \\ \hline
DZMI(I,6) - DZF(I,1,5) + \beta_{pc}DZMII(I,2) \\ \hline
DPMII(I,9) + \beta_{pc}DPMII(I,4)
\end{array}\right]$$

$$\{BF\} = \left[\begin{array}{l}
DYALII \\ \hline
DZALII \\ \hline
DPALII
\end{array}\right]$$

$$\{FNL\} = 2\Omega\left[\begin{array}{c}
\int_0^R Y_I\{-\int_x^R\int_x^R m\dot{v}v' + \int_x^R\int_x^R v''\int_x^R m\dot{v} + \int_x^R\int_x^R m\int_0^x \dot{v}'v' + \int_x^R\int_x^R m\int_0^x \dot{w}'w' \\ \hline
\int_0^R Z_I\{-\int_x^R\int_x^R m\dot{v}w' + \int_x^R\int_x^R w''\int_x^R m\dot{v}\} \\ \hline
0
\end{array}\right]$$

$$[SIB] = \begin{bmatrix} \sin\psi_{IB} & 0 & 0 \\ & \cos\psi_{IB} & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[CIB] = \begin{bmatrix} \cos\psi_{IB} & 0 & 0 \\ 0 & \sin\psi_{IB} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[RIOC] = [COIR]^{-1}$$

$$\{Y_{Z_P}\} = \begin{bmatrix} y \\ --- \\ z \\ --- \\ \phi \end{bmatrix} \qquad \{x_H\} = \begin{bmatrix} x_H \\ y_H \\ z_H \end{bmatrix}$$

Similarly, the coefficients in the hub equations of motion, Equation (27) may be defined as:

$$[TM] = \begin{bmatrix} m_{H_x}+NB \bullet MI(1,1) & 0 & 0 \\ 0 & m_{H_y}+NB \bullet MI(1,1) & 0 \\ 0 & 0 & m_{H_z}+NB \bullet MI(1,1) \end{bmatrix}$$

$$[BIN] = \begin{bmatrix} YI(1,J,1) & 0 & -PI(1,J,3) \\ -YI(1,J,1) & 0 & PI(1,J,3) \\ 0 & -ZI(1,J,1) & -PI(1,J,1) \end{bmatrix}$$

$$[BDAM] = 2\Omega \begin{bmatrix} YI(1,J,1) & 0 & -PI(1,J,3) \\ YI(1,J,1) & 0 & -PI(1,J,3) \\ 0 & 0 & 0 \end{bmatrix}$$

$$[BSPR] = \Omega^2 \begin{bmatrix} -YI(1,J,1) & 0 & 0 \\ YI(1,J,1) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[HC] = - \begin{bmatrix} C_{H_x} & 0 & 0 \\ 0 & C_{H_y} & 0 \\ 0 & 0 & C_{H_z} \end{bmatrix}$$

$$[HK] = - \begin{bmatrix} K_{H_x} & 0 & 0 \\ 0 & K_{H_y} & 0 \\ 0 & 0 & K_{H_z} \end{bmatrix}$$

$$\{HF\} = \begin{bmatrix} F_{H_x} \\ F_{H_y} \\ F_{H_z} \end{bmatrix}$$

$$[BIRI] = [BIN][RIOC]$$

$$[BIRID] = [BIRI][CODR]$$

$$[BIRIO] = [BIRI][COR] + [BSPR]$$

$$[BIRIDH] = [BIRI][CODH]$$

$$[BIRIIH] = [BIRI][COIH]$$

Using the previous definitions, and assuming a sinusoidal forcing function, Equation (25) may be written as:

$$\{\ddot{Y}_{Z_P}\}_{IB} = ([RIOC]([CODR]\{\dot{Y}_{Z_P}\}_{IB} + [COR]\{Y_{Z_P}\}_{IB} + \{FR\}_{IB} + \{BF\}\sin\omega_F t$$

$$+ \{FNL\}_{IB} + [COIH][SIB]_{IB}\{\ddot{x}_H\} + [CODH][CIB]_{IB}\{\dot{x}_H\}) \quad (35)$$

Equation (34) for the hub accelerations is written as:

$$[TM]\{\ddot{x}_H\} = \sum_{IB=1}^{NB} [SIB]_{IB}[BIN]\{\ddot{Y}_{Z_P}\}_{IB} + \sum_{IB=1}^{NB} [CIB]_{IB}[BDAMP]\{\dot{Y}_{Z_P}\}_{IB}$$

$$+ \sum_{IB=1}^{NB} [SIB][BSPR]\{Y_{Z_P}\}_{IB} + [HC]\{\dot{x}_H\} + [HK]\{x_H\} + \{HF\} \quad (36)$$

Solving for the blade accelerations from Equation (35) and substituting the result into Equation (36) removes the inertial coupling in the system and allows solution of the hub accelerations directly.

$$
\begin{aligned}
\{\ddot{x}_H\} = & \left([TM] - \sum_{IB=1}^{NB} [SIB]_{IB}[BIRIIH][SIB]_{IB}\right)^{-1} \Bigg(\Bigg([HC] \\
& + \sum_{IB=1}^{NB} [SIB]_{IB} [BIRIDH][CIB]\Bigg)\{\dot{x}_H\} + [HK]\{x_H\} + \{H_F\} \\
& + \sum_{IB=1}^{NB} \Big(([SIB]_{IB}[BIRID] + [CIB]_{IB}[BDAM])\{\dot{Y}_{Z_P}\}_{IB} \\
& + [SIB]_{IB}[BIRIO]\{Y_{Z_P}\}_{IB} + [SIB]_{IB}[BIRI](\{FR\}_{IB} + \{BF\}\sin w_F t \\
& + \{FNL\}_{IB})\Big)\Bigg)
\end{aligned}
\qquad (37)
$$

Solution of Equation (37) is effected by use of a fourth order Runge-Kutta timewise integration technique. Once the hub responses are obtained for a particular time increment, Equation (35) is solved for the blade motions. These blade motions are, in turn, substituted into Equation (37) to yield the hub responses for the subsequent time increments. This procedure is continued until the total time interval of interest is reached.

PROGRAM FEATURES - V22

The V22 program, developed to implement the solutions of the equations developed above, was designed to achieve the flexibility and ease of use necessary to make it a useful research tool. The details of the necessary and optional inputs are described in Appendix B. Some of the major features of the program are outlined in this section.

1. General input - The input data, in most cases, may be input in any order. Certain data is optional as input and need not be entered unless used. In running successive cases, only changed data need be input.

2. No. of Blades - One to four blades may be specified. With a hub, a minimum of two is required.

3. Modal input - The method of solution (Galerkin's method) uses separate in-plane, out-of-plane, and torsion "modes" as generalized degrees of freedom. They need not be normal modes (and thus need not be changed for changes in parameters and rotor speed). The equations contain the modal displacement as well as the first and second derivatives. Only the second derivative and the root slope of each mode is required as input. The program integrates and normalizes each mode to a value of unit displacement at the tip. Modes which are representative of the expected normal mode shapes are suggested.

4. Frequencies - Rotational and forcing frequencies are input independently. A frequency sweep may be simulated with a single card for each discrete frequency. $\Omega = 0$ is allowed.

5. Hub data - The hub is represented by a single degree of freedom spring, mass, damper in each direction. These parameters may be easily changed with forcing frequency to simulate actual hub impedances. Optionally 0, 1, 2 or 3 directions of motion are allowed. Sinusoidal forcing in any of these directions may be specified.

6. Blade forces - Optional forces may be applied at any blade station. An optional 1- cos type excitation for a specified fraction of one revolution is available.

7. Floquet option - If this option is selected, the program automatically produces a Floquet transition matrix by performing one (force) cycle for each initial condition. A further option ignores the steady effects due to such quantities as twist and precone.

8. Periodic solution - A periodic solution is obtained through the Floquet matrix which allows the solution for the initial conditions which will result in periodicity.

9. Nonlinear options - All, in-plane only, or no nonlinear effects may be optionally included in the solution.

10. Solution controls - The integration procedure used includes error checks and automatically selects appropriate sized integration increments. The user specifies quantities such as the number of cycles, error bound, variable to be tested for error, initial condition (unless periodic solution is specified).

SYSTEM IDENTIFICATION

The mass parameters of any continuous structure are not amenable to direct verification. An operational rotor blade is subjected to very large centrifugal forces and undergoes a highly coupled motion which includes deformation of the elastic axis in and out of the plane of rotation and torsional deformations about this axis. Under these conditions, the adequacy of the mass parameters which are based on a fictitious homogeneous section are in some doubt. While there is no way of directly measuring these parameters, the relationship between them and the normal modes, which are at least conceptionally measurable, are well understood.

The method of incomplete models (References 4 and 5), which addresses the problem, has been adapted to the specific set of rotor blade parameters. This formulation determines the minimum changes required in the intuitively derived set of mass parameters to make them compatible with the measured modes. There are other related developments and features of the implementation program which will yield valuable information regarding the adequacy of the analytical model. These are derived and discussed in this section.

THEORETICAL BACKGROUND

Consider a discrete element dynamic model of a continuous structure. One part of this model is a mass matrix, M. If Ψ_k is a vector representing the k-th normal mode, there exists a necessary orthogonality relationship as follows:

$$\Psi_k^T M \Psi_n = 0 \qquad k \neq n \tag{38}$$

If the modal vectors are considered to be known, and the masses unknown, this equation can be rewritten as a set of linear equations:

$$A\bar{M} = 0 \tag{39}$$

where A is a matrix whose elements are products of the elements of the modal vectors, and $\bar{M}$ is a vector made up of the unknown elements of the mass matrix. There will be one equation for each unique pair of modes and one unknown for each of the elements of $\bar{M}$. The problem is formulated so that the symmetrical off-diagonal elements in the (symmetrical) mass matrix appear only once in the mass vector, $\bar{M}$.

Since the scalar product $\Psi_k^T M \Psi_n$ is identical to $\Psi_n^T M \Psi_k$ there will be NM(NM-1)/2 equations, where NM is the number of modes. If N is the number of coordinates, the number of unknowns may be between N and N(N+1)/2 where the first corresponds to a pure diagonal matrix and the upper limit corresponds to a fully populated mass matrix. As discussed in References 4 and 5 it is usual and desirable to have many more unknowns than equations. There are, thus, an infinite number of solutions which will satisfy Equation (39).

It is, of course, desired to obtain that solution which is the most representative of the actual structure. This objective may be achieved by finding, of those mass matrices which satisfies Equation (39), and (38), that which is closest to an analytically derived model of the structure. That is to say, determine the smallest possible changes in the analytical mass matrix necessary to orthogonalize the measured modes. This may be done as follows. Let $\bar{M}_A$ be a vector which is made up of the elements of the analytical (or approximate) mass matrix and then write $\bar{M} = \bar{M}_A + \overline{\Delta M}$, where $\overline{\Delta M}$ represents the required changes in $\bar{M}_A$. Substituting into Equation (39) yields:

$$A\overline{\Delta M} = - A\bar{M}_A \tag{40}$$

As discussed in Reference 5, the use of the matrix pseudoinverse yields a solution which has the minimum sum of the squares of the individual elements, i.e., $\overline{\Delta M}^T \overline{\Delta M} = \text{min}$. This solution may be written:

$$\overline{\Delta M}_{min} = - A^T(AA^T)^{-1}A\bar{M}_A \tag{41}$$

The application to the specific rotor blade problem is given below, where certain other more detailed considerations of minimization and other constraints are discussed.

<u>ROTOR BLADE APPLICATION</u>

The normal modes of a rotor blade are conveniently expressed in terms of the in-plane, out-of-plane, and torsional components as follows:

$$\Psi_K = \begin{bmatrix} \bar{v} \\ \bar{w} \\ \bar{\phi} \end{bmatrix}_k$$

where $\bar{v}$, $\bar{w}$, and $\bar{\phi}$ are vectors, each having NX elements, when NX is the number of blade stations used in the analysis and test.

The mass matrix, as can be seen from the acceleration terms of Equations (5), (6), and (7) may be conveniently partitioned, where each of the partitions is a diagonal matrix of order NX. The rotor blade form of Equation (38) then may be written:

$$[\bar{v}^T\ \bar{w}^T\ \bar{\phi}^T]_k \left[\begin{array}{c|c|c} m_i & 0 & -(me\theta)_i \\ \hline 0 & m_i & (me)_i \\ \hline -(me\theta)_i & (me)_i & (mkm^2)_i \end{array}\right] \begin{bmatrix} \bar{v} \\ \bar{w} \\ \bar{\phi} \end{bmatrix}_n = 0 \qquad k \neq n \tag{42}$$

The elements of these diagonal partitions (i = 1, 2,...NX) represent a "lumped mass" (rather than a "distributed mass") formulation of the problem, which is inherent in the matrix representation.

Treating the modal displacements as knowns and the mass parameters as unknowns, the analogy of Equation (39) becomes:

$$\left[\begin{array}{c|c|c|c} v_{k_i} v_{n_i} & w_{k_i} \phi_{n_i} & -\,v_{k_i} \phi_{n_i} & k_i \phi_{n_i} \\ +w_{k_i} w_{n_i} & +w_{n_i} \phi_{k_i} & -v_{n_i} \phi_{k_i} & \end{array}\right] \begin{bmatrix} \bar{m} \\ \bar{me} \\ \overline{me\theta} \\ \overline{mk_m^2} \end{bmatrix} = 0 \tag{43}$$

where, typically, v_{k_i} represents the in-plane displacement of mode k at station i. Each partition of the matrix A has NM(NM-1)/2 rows (one for each pair of modes, k < n) and NX columns, one for each station (i = 1, 2...NX). This, there are NM(NM-1)/2 equations and 4 • NX unknowns (in vector $\bar{M}$).

As above, let $\bar{M} = \bar{M}_A + \overline{\Delta M}$, then Equation (43) is:

$$A\overline{\Delta M} = - A\bar{M}_A \quad (44)$$

This equation may be solved for minimum $\overline{\Delta M}$ as in Equation (41). However, if there are significant differences in size between elements of M_A it would not be appropriate to simply minimize the sum of the squares of the magnitudes of the changes. This procedure could result in excessively large percentage changes in the very small elements, even though these same changes would be quite small compared to the larger elements.

It is possible, through a simple modification in the method to minimize the sum of the squares of the percentage changes, which is a more reasonable criteria. In addition, it is also possible to allow the analyst to indicate a level of confidence in each element, so that items with higher confidence will tend to change least. The result is a solution which has a weighted sum of squares of the elements at a minimum.

Let the i-th element of $\bar{M}_A$ be designated $(\bar{M}_A)_i$ and the corresponding assigned weighting factor (confidence level) be w_i. Form a diagonal matrix W such that $W_{ii} = w_i/(\bar{M}_A)_i$. Then the elements of $W\overline{\Delta M}$ are

$$(W\overline{\Delta M})_i = w_i(\overline{\Delta M})_i/(\bar{M}_A)_i$$

which is the function that should be minimized. This is achieved by making $W\overline{\Delta M}$ the unknown in Equation (44) by inserting $I = W^{-1}W$ as follows:

$$AW^{-1}\, W\overline{\Delta M} = - A\bar{M}_A \quad (45)$$

Then, as above:

$$(W\overline{\Delta M})_{min} = - W^{-1}A^T\{AW^{-2}A^T\}^{-1} A\bar{M}_A$$

and

$$\bar{M} = \bar{M}_A - W^{-2}A^T\{AW^{-2}A^T\}^{-1} A\bar{M}_A \quad (46)$$

such that:

$$\overline{\Delta M}^T W^2\overline{\Delta M} = \min$$

MASS CONSTRAINTS

Since the number of equations is generally much less than the number of unknowns, it is possible to add equations to Equation (43) which will impose constraints on the mass parameters. In the method as implemented, five optional constraints are available. These each maintain the following mass characteristics at the same value they have in $\bar{M}_A$. These constraints refer to: total mass, radial static moment (cg), chordwise static moment (cg), flapping moment of inertia, and feathering moment of inertia. These five constraints result in the following equations added to Equation (43):

$$\left[\begin{array}{c|c|c|c} 1,1,1\ldots & 0 & 0 & 0 \\ x_1,x_2,x_3\ldots & 0 & 0 & 0 \\ 0 & 1,1,1\ldots & 0 & 0 \\ x_1^2,x_2^2\ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1,1,1\ldots \end{array}\right] \begin{bmatrix} \bar{m} \\ \overline{me} \\ \overline{me}\theta \\ \overline{mk}_m^2 \end{bmatrix} = \begin{bmatrix} \Sigma m_{A_i} \\ \Sigma X_i m_{A_i} \\ \Sigma (me)_{A_i} \\ \Sigma X_i^2 m_{A_i} \\ \Sigma (mk_m^2)_{A_i} \end{bmatrix} \tag{47}$$

The solution then becomes:

$$\bar{M} = \bar{M}_A - W^{-2}A^T\{A^TW^{-2}A^T\}^{-1}\{A\bar{M}_A - \bar{r}\} \tag{48}$$

where $\bar{r}$ is the right-hand side vector of Equation (43) augmented by that of Equation (47).

Thus it is possible to find the necessary changes in the mass matrix to make the modes orthogonal, such that the weighted sum of squares of the percentage changes is a minimum and the specified mass characteristics remain invariant.

ROTATIONAL SPEED EFFECTS

The mass matrix discussed above is independent of the blade rotational speed, Ω. The natural frequencies and the mode shapes, however, do change as the rotational speed is changed. The analysis, as presented, is valid for any single Ω including the nonrotating condition, $\Omega = 0$.

The fact that the modes change with Ω provides an opportunity for obtaining additional information within a fixed range of forcing frequencies over that available for a conventional nonrotating structure. If several modes are measured at each of several values of Ω, the same mass matrix must make the modes at any one Ω orthogonal.

Thus, the method above has been modified to accept modes at different values of Ω and to set up an equation for each pair of modes at each Ω. For example, if the first three modes were identified at three Ω's, there would be nine equations which would provide information about the mass matrix.

MODE CHANGES

The measured data, even if exact, is not sufficient to uniquely identify an analytical model and thus intuitive decisions are required of the user of this method. Some of these decisions have been described above. In addition to finding the necessary mass model changes, consideration should be given to the unavoidable errors in the measured modes. It is of interest to determine the minimum changes that would be required in the modes to achieve orthogonality using the analytical mass matrix. Methods of this general type have been suggested in the literature from time to time (References 6, 7, and 8). The method developed and implemented in this study uses techniques very similar to those for the mass identification, above.

If the modes are placed in order of decreasing confidence (usually in order of increasing natural frequency), the method assumes the first is correct, changes the second to make it orthogonal to the first, then changes the third to make it orthogonal to the first and the corrected second mode, and similarly for all higher modes. The changes are the minimum sum of squares of the percentage changes of each element as discussed above.

The first equation may be written:

$$\Psi_1^T M(\Psi_2 + \Delta\Psi_2) = 0$$

or

$$A\Delta\Psi_2 = - A\Psi_2 \qquad (49)$$

where $A = \Psi_1^T M$ is a 1 x 3•NX matrix. The next equation then is:

$$A\Delta\Psi_3 = - A\Psi_3 \tag{50}$$

where:

$$A = \begin{bmatrix} \Psi_1^T \\ \Psi_2^T + \Delta\Psi_2^T \end{bmatrix} M$$ and A is a 2 x 3•NX matrix.

The equations for $\Delta\Psi_M$ results in an A matrix of order M-1 x 3 NX. The procedure used for solving these equations is the same as that described above without any weighting function, w, assigned to the individual elements.

PROGRAM FEATURES - ROTSI

This program has been designed to provide maximum flexibility as a research tool. The theoretical basis has been described in the previous paragraphs. The Users Guide with detailed input instructions is in Appendix B. This section will briefly outline several of the major features and capabilities of the program.

1. Normalization - the modes may be normalized so the diagonal elements of the generalized mass matrix are unity.

2. Add modes - after a computation is completed, additional modes may be added and further operations may be performed.

3. Rotational speed - modes of more than one rotational speed may be included (for mass identification) and the proper pairing takes place automatically.

4. Random errors - modes may be polluted with random errors with specified random or bias errors for sensitivity analyses.

5. Modal changes - necessary mode changes as described above with constant mass matrix may be determined.

6. Limited mode changes - modes may be changed as above but with limits specified for each mode. Truncation or scaling options are available.

7. Mass changes - weighted minimum mass changes may be obtained as described above.

8. Invariant stations - the mass parameters at selected stations may be held invariant.

9. Invariant parameters - mass, static moments, moments of inertia may optionally be maintained invariant during mass identification.

10. Sequential operations - the various options may be executed sequentially, for example, one may first change all the modes up to some specified percentages and then finish the correction by modifying the mass matrix.

METHOD APPLICATIONS

The two programs were continually checked for validity and reasonableness during their development. All features were at least qualitatively verified. The programs were then used to approximately simulate the tests to be carried out in the vacuum chamber at the Langley Research Center. These applications are described below.

SIMULATION DATA

The system simulated consisted of two blades and a hub with a vertical degree of freedom. The system was excited by a vertical force at the hub.

Each blade was represented by 17 stations. The parameters are shown in Table 1 which is taken from an actual computer run. The units are all in the lb-in-sec system.

Tables 2, 3, and 4 show the modes used as generalized degrees of freedom. These modes were developed from an approximate cantilever eigenvalue analysis. The one in-plane, three out-of-plane, and one torsional mode represent all the modes expected to have natural frequencies below 12/rev at Ω = 25 rad/sec. The tables illustrate the second and first derivative and the displacements after normalization.

The hub was arbitrarily represented by a mass of .6 lb-sec^2/in and a spring rate of 20,000 lb/in. This implies a rigid rotor vertical natural frequency of 111. rad/sec or 4.44/rev at Ω = 25 rad/sec.

Tables 5 and 6 give the blade and hub matrices as described in the section on Method of Solution and Equations (36) and (37).

SIMULATION COMPUTATIONS

Simulated frequency sweeps were carried out at Ω = 0, 20, and 25 rad/sec. The Floquet option was used to obtain precise periodic responses to sinusoidal excitation at the hub. The objective of the simulated test was to locate the frequencies at which hub vertical antiresonances occur. At this frequency, cantilever conditions exist and since damping is light the displacement will be a good approximation to the coupled cantilever normal modes of the blades. Since discrete frequency inputs are required, a coarse sweep was first carried out, followed by necessary points at small frequency intervals to identify the point of zero hub displacement.

TABLE 1. BLADE PROPERTIES

IO = 1,2 BLADE PROPERTIES

2 BLADES PRECONE = 0.0 THETA 0 = 0.0 DAMPING (V,W,P) 0.0 0.0

	X	M	E	SMALL EA	KM1	KM2	KA	THETA PRIME
1	0.0	6.470E-03	-2.925E 00	-2.850E 00	1.265E 00	5.557E 00	5.699E 00	-4.850E-04
2	8.000E 00	5.900E-03	-2.275E 00	-2.163E 00	1.281E 00	5.819E 00	5.958E 00	-4.850E-04
3	1.200E 01	4.450E-03	-1.407E 00	-1.269E 00	1.219E 00	6.025E 00	6.147E 00	-4.850E-04
4	2.000E 01	3.190E-03	-1.030E 00	-8.750E-01	1.057E 00	5.820E 00	5.915E 00	-4.850E-04
5	4.000E 01	2.320E-03	-9.250E-01	-7.250E-01	9.130E-01	6.018E 00	6.087E 00	-4.850E-04
6	6.000E 01	1.720E-03	-1.100E 00	-8.500E-01	8.010E-01	6.475E 00	6.524E 00	-4.850E-04
7	8.000E 01	1.650E-03	-9.050E-01	-5.630E-01	8.160E-01	6.260E 00	6.313E 00	-4.850E-04
8	1.000E 02	1.490E-03	-7.900E-01	-3.750E-01	8.080E-01	6.186E 00	6.239E 00	-4.850E-04
9	1.200E 02	1.340E-03	-7.000E-01	-1.500E-01	7.910E-01	6.082E 00	6.133E 00	-4.850E-04
10	1.400E 02	1.310E-03	-4.130E-01	2.300E-01	8.000E-01	5.785E 00	5.840E 00	-4.850E-04
11	1.600E 02	1.420E-03	3.250E-01	7.000E-01	7.840E-01	5.394E 00	5.451E 00	-4.850E-04
12	1.800E 02	1.550E-03	9.620E-01	1.050E 00	7.640E-01	5.066E 00	5.123E 00	-4.850E-04
13	2.000E 02	1.540E-03	1.037E 00	1.200E 00	7.670E-01	4.937E 00	4.996E 00	-4.850E-04
14	2.200E 02	1.540E-03	1.025E 00	1.187E 00	7.670E-01	4.971E 00	5.030E 00	-4.850E-04
15	2.400E 02	1.590E-03	1.087E 00	1.287E 00	7.560E-01	4.914E 00	4.972E 00	-4.850E-04
16	2.600E 02	1.620E-03	1.162E 00	1.405E 00	7.480E-01	4.899E 00	4.956E 00	-4.850E-04
17	2.680E 02	1.620E-03	1.162E 00	1.405E 00	7.480E-01	4.899E 00	4.956E 00	-4.850E-04

	EI OP	EI IP	GJ	EA	EB1*	EB2*	EC1	EC1*
1	4.500E 08	9.000E 09	2.400E 08	2.440E 08	0.0	0.0	0.0	0.0
2	4.000E 08	8.250E 09	1.850E 08	2.225E 08	0.0	0.0	0.0	0.0
3	2.530E 08	6.030E 09	1.550E 08	1.649E 08	0.0	0.0	0.0	0.0
4	1.280E 08	4.000E 09	9.500E 07	1.144E 08	0.0	0.0	0.0	0.0
5	9.900E 07	3.250E 09	4.700E 07	8.310E 07	0.0	0.0	0.0	0.0
6	7.000E 07	2.650E 09	3.350E 07	6.060E 07	0.0	0.0	0.0	0.0
7	4.000E 07	2.350E 09	3.300E 07	5.750E 07	0.0	0.0	0.0	0.0
8	4.000E 07	2.040E 09	3.120E 07	5.250E 07	0.0	0.0	0.0	0.0
9	3.500E 07	1.720E 09	3.500E 07	4.750E 07	0.0	0.0	0.0	0.0
10	3.000E 07	1.460E 09	3.150E 07	4.630E 07	0.0	0.0	0.0	0.0
11	3.000E 07	1.250E 09	3.300E 07	4.580E 07	0.0	0.0	0.0	0.0
12	3.000E 07	1.070E 09	3.450E 07	4.630E 07	0.0	0.0	0.0	0.0
13	3.000E 07	9.700E 08	3.200E 07	4.600E 07	0.0	0.0	0.0	0.0
14	3.000E 07	1.000E 09	3.400E 07	4.600E 07	0.0	0.0	0.0	0.0
15	3.000E 07	1.000E 09	3.300E 07	4.750E 07	0.0	0.0	0.0	0.0
16	3.130E 07	1.000E 09	3.300E 07	4.900E 07	0.0	0.0	0.0	0.0
17	3.250E 07	1.010E 09	3.300E 07	4.900E 07	0.0	0.0	0.0	0.0

TABLE 2. IN-PLANE MODES

IO = 3 IN-PLANE MODES

SECOND DERIVATIVES

1	1.504E-05
2	1.580E-05
3	2.114E-05
4	3.034E-05
5	3.301E-05
6	3.518E-05
7	3.393E-05
8	3.268E-05
9	3.176E-05
10	2.942E-05
11	2.566E-05
12	2.089E-05
13	1.454E-05
14	7.446E-06
15	2.708E-06
16	2.557E-07
17	0.0

(C) FIRST DERIV (NORMALIZED)

1	0.0
2	1.234E-04
3	1.972E-04
4	4.032E-04
5	1.037E-03
6	1.719E-03
7	2.410E-03
8	3.076E-03
9	3.720E-03
10	4.332E-03
11	4.883E-03
12	5.348E-03
13	5.703E-03
14	5.922E-03
15	6.024E-03
16	6.054E-03
17	6.055E-03

(C) MODE SHAPES

1	0.0
2	4.934E-04
3	1.135E-03
4	3.536E-03
5	1.793E-02
6	4.549E-02
7	8.677E-02
8	1.416E-01
9	2.096E-01
10	2.901E-01
11	3.823E-01
12	4.846E-01
13	5.951E-01
14	7.113E-01
15	8.308E-01
16	9.516E-01
17	1.000E 00

TABLE 3. OUT-OF-PLANE MODES

IO = 4 OUT-OF-PLANE MODES

SECOND DERIVATIVES

1	1.802E-05	-4.294E-05	1.909E-04
2	1.792E-05	-4.191E-05	1.780E-04
3	2.657E-05	-6.097E-05	2.528E-04
4	4.590E-05	-1.012E-04	3.900E-04
5	4.056E-05	-7.579E-05	1.832E-04
6	3.784E-05	-4.479E-05	-7.925E-05
7	4.318E-05	2.610E-06	-4.891E-04
8	2.808E-05	6.047E-05	-5.814E-04
9	2.124E-05	1.152E-04	-5.069E-04
10	1.691E-05	1.713E-04	-1.989E-04
11	1.188E-05	1.904E-04	2.875E-04
12	8.304E-06	1.872E-04	7.154E-04
13	5.526E-06	1.575E-04	9.073E-04
14	3.170E-06	1.074E-04	7.839E-04
15	1.278E-06	4.914E-05	4.176E-04
16	1.137E-07	4.592E-06	4.200E-05
17	0.0	0.0	0.0

(C) FIRST DERIV (NORMALIZED)

1	0.0	0.0	0.0
2	1.437E-04	-3.394E-04	1.475E-03
3	2.327E-04	-5.452E-04	2.337E-03
4	5.226E-04	-1.194E-03	4.908E-03
5	1.387E-03	-2.963E-03	1.064E-02
6	2.171E-03	-4.169E-03	1.168E-02
7	2.981E-03	-4.591E-03	5.996E-03
8	3.694E-03	-3.960E-03	-4.710E-03
9	4.187E-03	-2.203E-03	-1.559E-02
10	4.569E-03	6.615E-04	-2.265E-02
11	4.857E-03	4.278E-03	-2.176E-02
12	5.058E-03	8.054E-03	-1.174E-02
13	5.197E-03	1.150E-02	4.491E-03
14	5.284E-03	1.415E-02	2.140E-02
15	5.328E-03	1.571E-02	3.342E-02
16	5.342E-03	1.625E-02	3.801E-02
17	5.343E-03	1.627E-02	3.818E-02

(C) MODE SHAPES

1	0.0	0.0	0.0
2	5.749E-04	-1.358E-03	5.901E-03
3	1.328E-03	-3.127E-03	1.353E-02
4	4.349E-03	-1.008E-02	4.251E-02
5	2.345E-02	-5.165E-02	1.980E-01
6	5.903E-02	-1.230E-01	4.212E-01
7	1.106E-01	-2.106E-01	5.979E-01
8	1.773E-01	-2.961E-01	6.108E-01
9	2.561E-01	-3.577E-01	4.078E-01
10	3.437E-01	-3.732E-01	2.534E-02
11	4.379E-01	-3.238E-01	-4.188E-01
12	5.371E-01	-2.004E-01	-7.538E-01
13	6.396E-01	-4.882E-03	-8.263E-01
14	7.444E-01	2.516E-01	-5.673E-01
15	8.506E-01	5.503E-01	-1.910E-02
16	9.573E-01	8.699E-01	6.952E-01
17	1.000E 00	1.000E 00	1.000E 00

TABLE 4. TORSION MODE

IO = 5 TORSION MODES

SECOND DERIVATIVES

1	1.207E-04
2	1.207E-04
3	1.207E-04
4	1.207E-04
5	1.207E-04
6	1.063E-05
7	-7.173E-06
8	-2.343E-05
9	-2.749E-05
10	-3.228E-05
11	-3.299E-05
12	-3.299E-05
13	-3.299E-05
14	-3.299E-05
15	-3.299E-05
16	-5.642E-05
17	0.0

(C) FIRST DERIV (NORMALIZED)

1	0.0
2	9.659E-04
3	1.449E-03
4	2.415E-03
5	4.829E-03
6	6.143E-03
7	6.178E-03
8	5.872E-03
9	5.362E-03
10	4.765E-03
11	4.112E-03
12	3.452E-03
13	2.792E-03
14	2.132E-03
15	1.472E-03
16	5.783E-04
17	3.526E-04

(C) MODE SHAPES

1	0.0
2	3.864E-03
3	8.693E-03
4	2.415E-02
5	9.659E-02
6	2.063E-01
7	3.295E-01
8	4.500E-01
9	5.624E-01
10	6.636E-01
11	7.524E-01
12	8.280E-01
13	8.905E-01
14	9.397E-01
15	9.758E-01
16	9.963E-01
17	1.000E 00

TABLE 5. THE BLADE INERTIAL, DAMPING, STIFFNESS MATRICES, AND INVERSE OF THE INERTIAL MATRIX AT Ω = 25 RAD/SEC (SEE EQ. 36, 37)

COIR

3.842E 02	0.0	0.0	0.0	5.527E 01
0.0	4.659E 02	2.379E 02	-2.544E 01	5.633E 02
0.0	-4.712E 02	-1.848E 02	9.288E 01	-5.558E 02
0.0	6.047E 02	1.278E 02	-1.883E 02	6.554E 02
1.128E 02	1.005E 03	5.848E 02	-8.914E 01	3.080E 04

CODR

6.604E 02	-1.493E 01	-3.420E 01	-2.353E 01	-4.104E 02
-6.850E 01	0.0	0.0	0.0	0.0
3.819E 01	0.0	0.0	0.0	0.0
-2.737E 01	0.0	0.0	0.0	0.0
6.686E 02	0.0	0.0	0.0	0.0

COR

-1.803E 06	6.378E 04	8.455E 05	3.021E 06	8.414E 04
1.294E 05	-4.525E 05	-1.113E 06	-1.739E 06	-4.470E 06
-9.931E 04	4.499E 05	5.408E 05	-1.227E 06	2.074E 06
5.645E 04	-5.593E 05	3.336E 05	3.681E 06	-5.062E 05
9.821E 03	-9.167E 05	-1.632E 06	4.679E 05	-1.179E 09

RIOC

2.604E-03	-1.773E-06	-2.421E-05	-9.396E-06	-4.879E-06
7.526E-06	1.408E-02	2.540E-02	1.064E-02	-2.563E-05
8.327E-06	-2.001E-02	-4.458E-02	-1.928E-02	-2.836E-05
-4.841E-06	3.168E-02	5.188E-02	1.599E-02	1.648E-05
-9.959E-06	1.233E-05	1.683E-04	6.531E-05	3.391E-05

TABLE 6. HUB MATRICES (SEE EQ. 36, 37)

BIRIIH

1.915E-01	-1.915E-01	-8.149E-08
-1.915E-01	1.915E-01	8.149E-08
1.313E-04	-1.313E-04	3.109E-01

BIRID

2.465E-01	-5.578E-03	-1.277E-02	-8.790E-03	-1.533E-01
-2.465E-01	5.578E-03	1.277E-02	8.790E-03	1.533E-01
5.712E-02	-4.236E-07	-9.700E-07	-6.675E-07	-1.164E-05

BIRIO

-7.632E 02	2.381E 01	3.156E 02	1.128E 03	2.748E 02
7.632E 02	-2.381E 01	-3.156E 02	-1.128E 03	-2.748E 02
2.440E-01	1.329E-02	-2.770E-02	3.819E 03	3.364E 03

BIRIDH

6.286E-03	-6.286E-03	0.0
-6.286E-03	6.286E-03	0.0
2.920E-03	-2.920E-03	0.0

BIRI

3.736E-04	-7.553E-08	-1.032E-06	-4.002E-07	-2.078E-07
-3.736E-04	7.553E-08	1.032E-06	4.002E-07	2.078E-07
2.837E-08	-2.920E-03	-5.237E-03	-2.087E-03	-9.654E-08

Note that since the responses are the steady-state periodic responses to sin $\omega_f t$ forcing, the response at $\omega_f t = 90°$ is the "real" or in-phase component and the response at $\omega_f t = 0°$ is the "imaginary" or out-of-phase component. Figures 3-12 illustrate the hub responses in the vicinity of the antiresonant frequencies. In most cases, the imaginary component is too small to be observed and is not plotted. These figures also illustrate the system natural frequencies.

At each antiresonant frequency the amplitudes of the generalized coordinates were determined and normalized on the largest component. These represent cantilever coupled modes and are summarized in Table 7. A Campbell diagram displaying these frequencies is given in Figure 13.

The actual mode shapes in each of the three directions are shown in Figures 14-19. Figures 14 and 15 are the in-plane and torsion component shapes. Since only one of each was used as a degree of freedom in the simulation, these shapes are the same for all the coupled normal modes obtained. The magnitudes are given in Table 7. The out-of-plane bending was represented by three modes and different combinations appear for each normal mode. Figures 16-19 illustrate these shapes for all the modes referenced in Table 7. The amplitude of these normalized modes is the sum of the z_1, z_2, z_3 components given in the table. The small but noticeable effect of rotor speed is illustrated in these figures.

SYSTEM IDENTIFICATION

In order to test and illustrate the ROTSI methods and program, the data obtained in the simulation runs, above, was treated as if it were actual test data. The analytical model was first intuitively reduced to an eight station lumped mass model as shown on Table 8.

Several combinations of these modes were used for mass identification. A sample output is shown in Table 9 where the original parameter, the modified parameter and the percentage changes are given. Table 10 summarizes the sample analyses that were carried out showing mean absolute percent changes of the four parameters: m, e, θ, K_m. The results are not satisfactory as shown. In addition to these cases, other combinations of modes at different rotational speeds have yielded very large percentage change requirements.

Since similar analyses on other structures using as many as ten modes and 150 unknowns have been successfully carried out, the large changes required for all but the simplest combinations is surprising. However, there are two significant considerations which may shed some light on this problem.

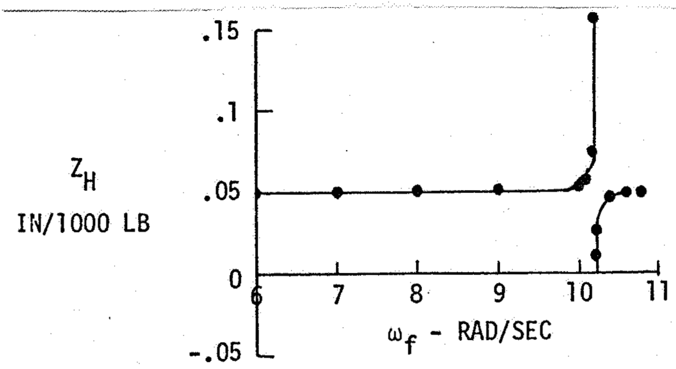


Figure 3. Hub Vertical Deflection vs Forcing Frequency, $\Omega = 0$. 1st OP Cantilever = 10.19 Rad/Sec

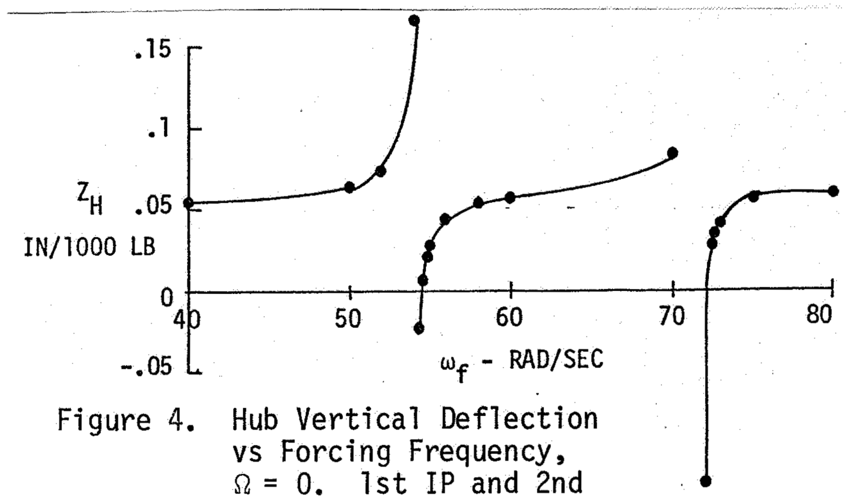


Figure 4. Hub Vertical Deflection vs Forcing Frequency, $\Omega = 0$. 1st IP and 2nd OP Frequencies = 54.55, 74.20 Rad/Sec

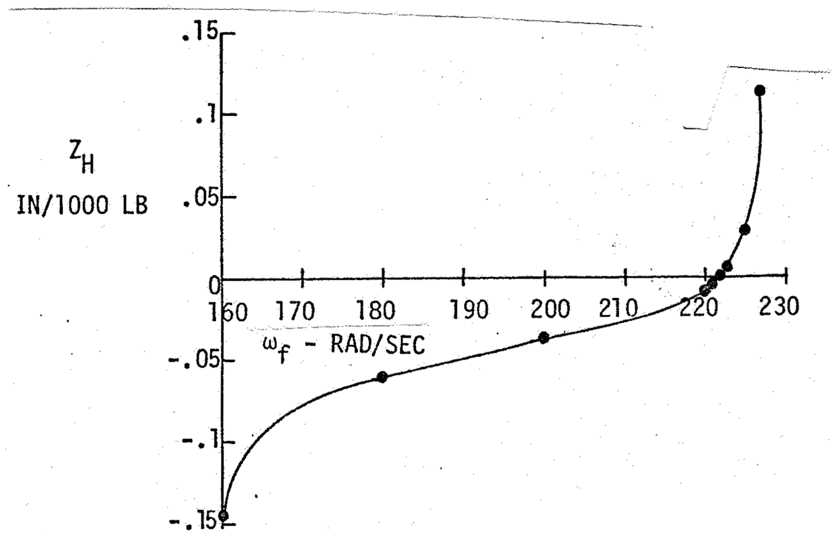


Figure 5. Hub Vertical Deflection vs Forcing Frequency, $\Omega = 0$. 3rd OP Frequency = 222 Rad/Sec

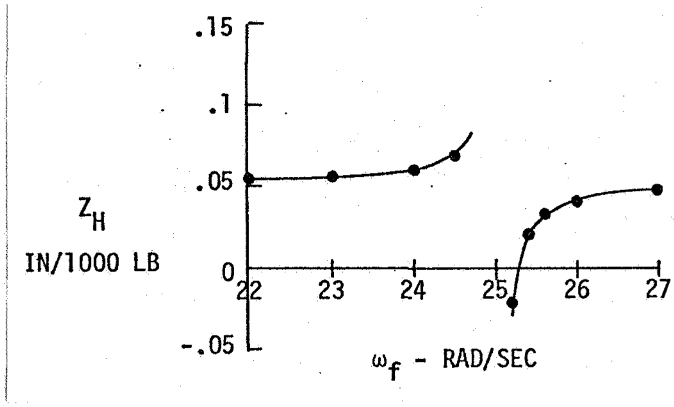


Figure 6. Hub Vertical Deflection vs Forcing Frequency, Ω = 20 Rad/Sec. 1st OP Frequency = 25.25 Rad/Sec

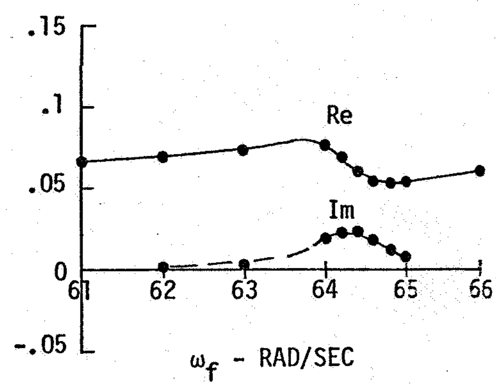


Figure 7. Hub Vertical Deflection vs Forcing Frequency, Ω = 20 Rad/Sec. An Apparent Highly Damped Response in Vicinity of 1st IP Frequency

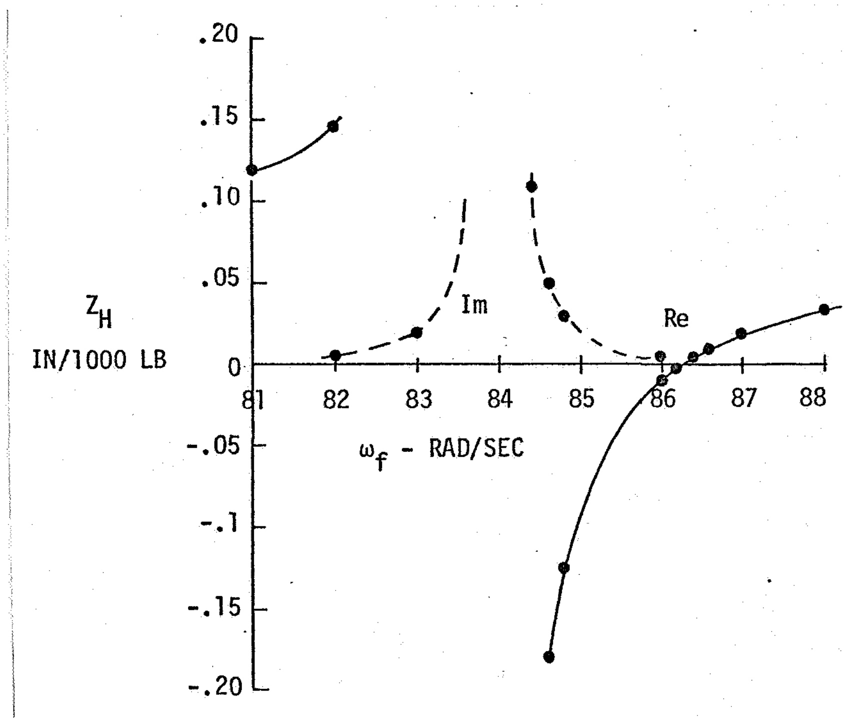


Figure 8. Hub Vertical Deflection vs Forcing Frequency, Ω = 20 Rad/Sec. 2nd OP Frequency = 86.25 Rad/Sec

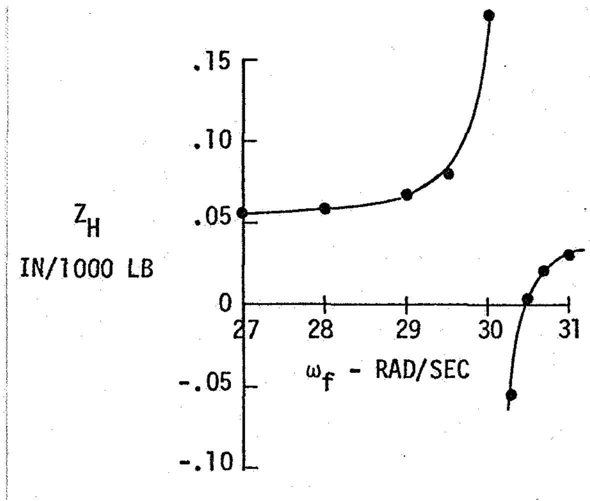


Figure 9. Hub Vertical Deflection vs Forcing Frequency, Ω = 25. 1st OP Frequency = 30.49 Rad/Sec

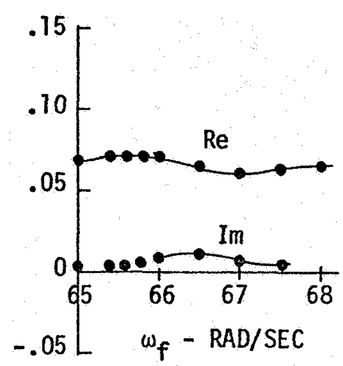


Figure 10. Hub Vertical Deflection vs Forcing Frequency, Ω = 25. Apparent Highly Damped Response in Vicinity of 1st IP Frequency

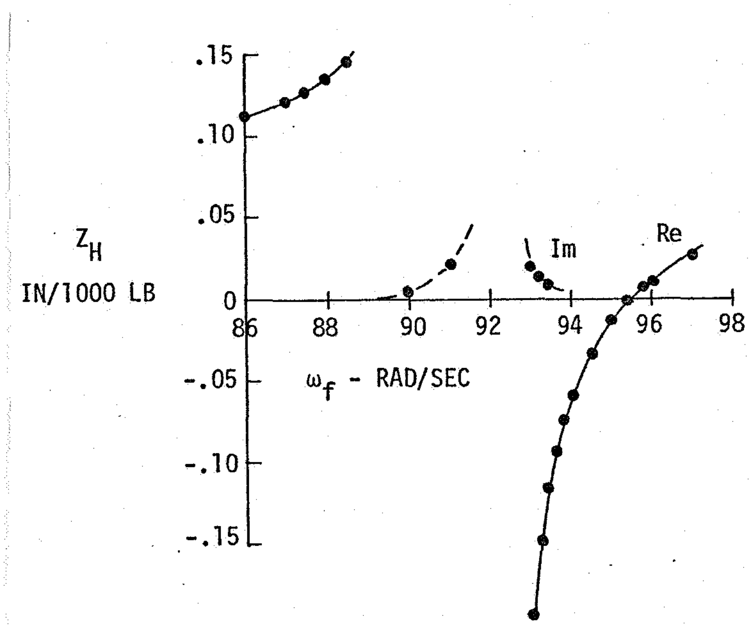


Figure 11. Hub Vertical Deflection vs Forcing Frequency, Ω = 25. 2nd OP Frequency = 95.52 Rad/Sec

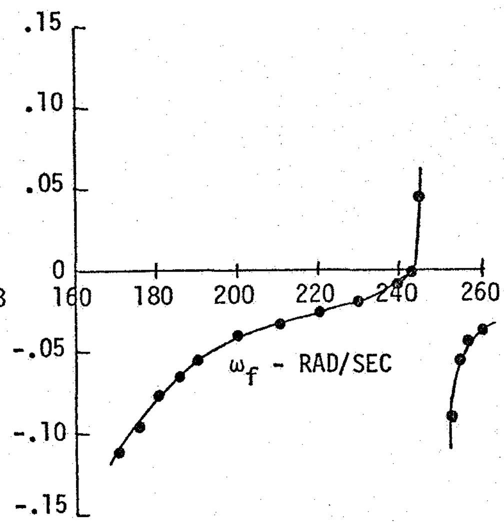


Figure 12. Hub Vertical Deflection vs Forcing Frequency, Ω = 25. 3rd OP Frequency = 243.3 Rad/Sec

TABLE 7. CANTILEVER NORMAL MODES

Type	Ω (Rad/Sec)	ω	y	z_1	z_2	z_3	Ψ
1st OP	0	10.19	.0655	1.0	.0868	-.0100	.000097
	20	25.25	.0408	1.0	.0020	-.0013	.000049
	25	30.49	.0354	1.0	-.0198	.0013	.000037
1st IP	0	54.55	1.0	-.3393	.8503	-.0537	.000801
2nd OP	0	74.20	-1.928	-.3015	1.0	-.0561	.000348
	20	86.25	-.6268	-.2863	1.0	-.0448	.000845
	25	95.52	-.4180	-.2839	1.0	-.0379	.00104
3rd OP	0	222.0	.1569	.3240	.4024	1.0	.003650
	25	243.3	-.131	.287	-.359	1.0	.000756

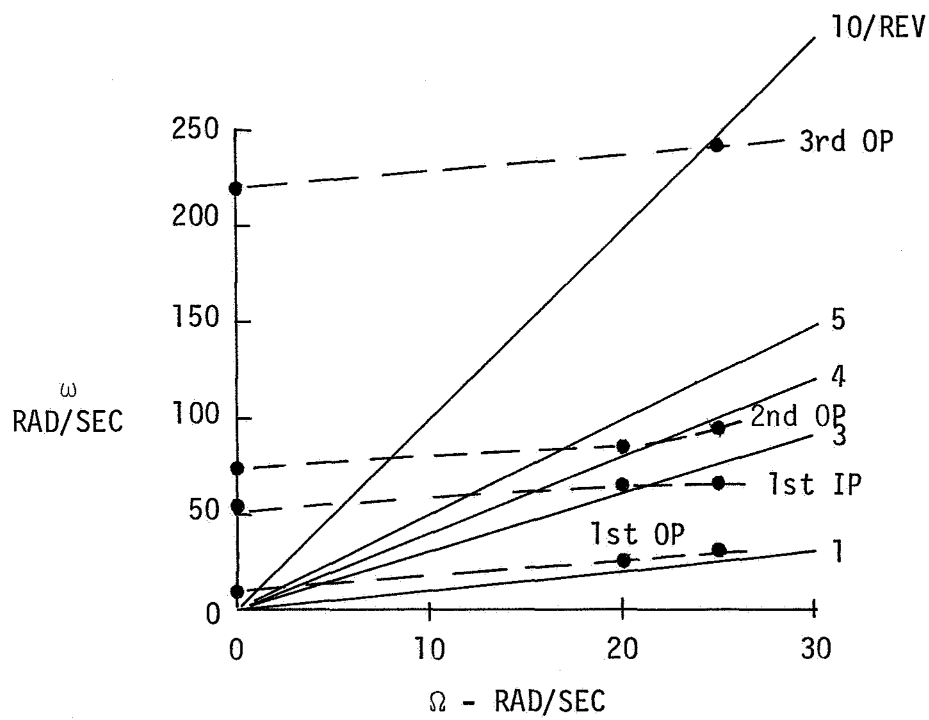


Figure 13. Campbell Diagram Illustrating Natural Frequencies Obtained During Simulated Frequency Sweep

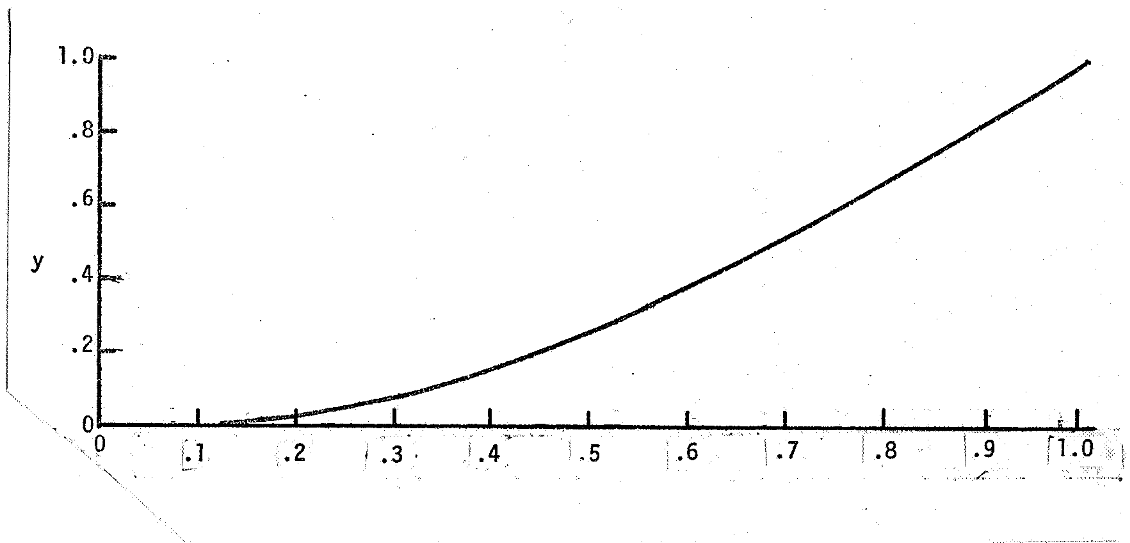


Figure 14. In-Plane Mode Shape for All Frequencies

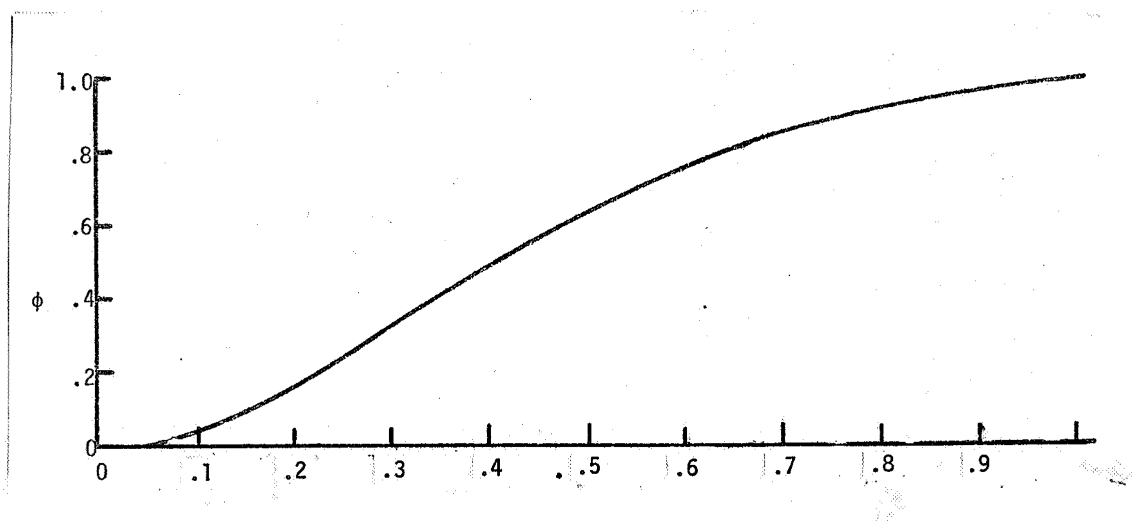


Figure 15. Torsional Mode Shape for All Frequencies

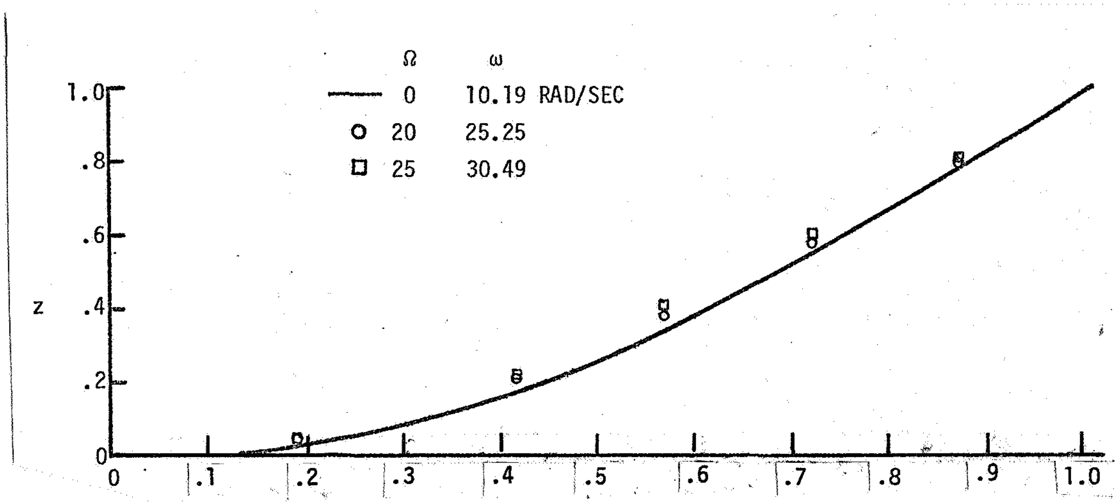


Figure 16. Out-of-Plane Shapes From 1st OP Coupled Modes

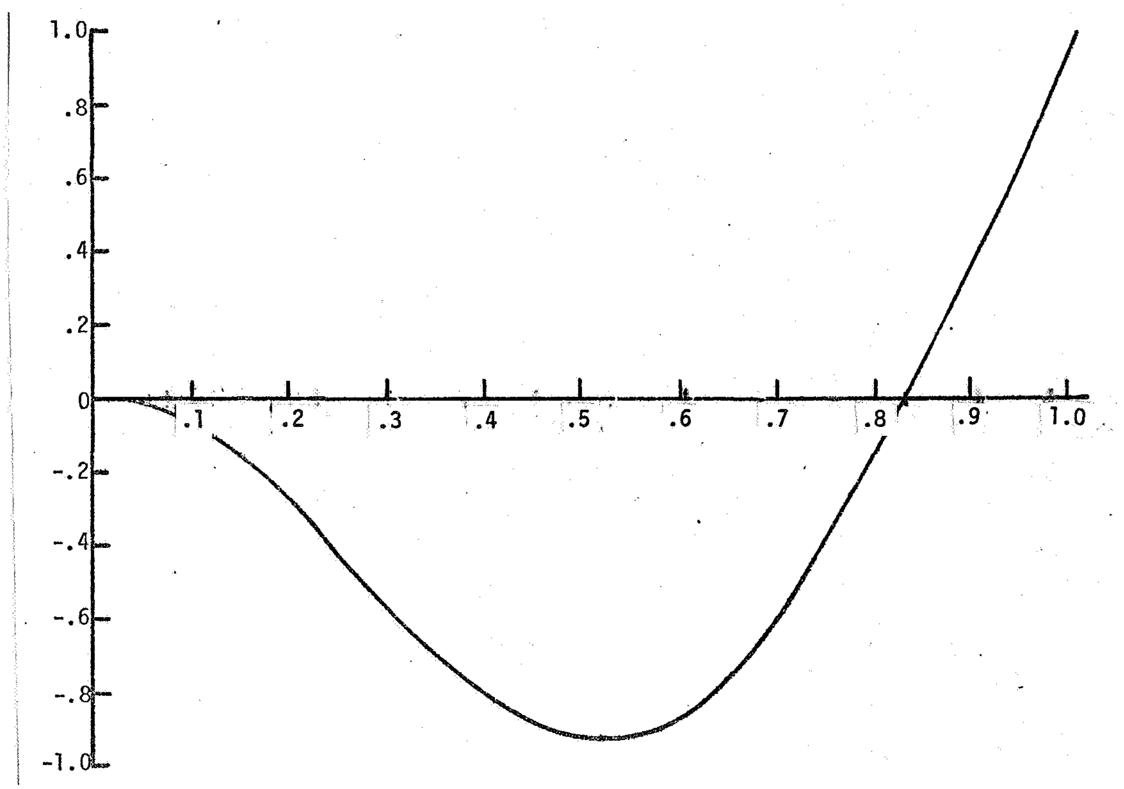


Figure 17. Out-of-Plane Shapes From 1st IP Coupled Modes

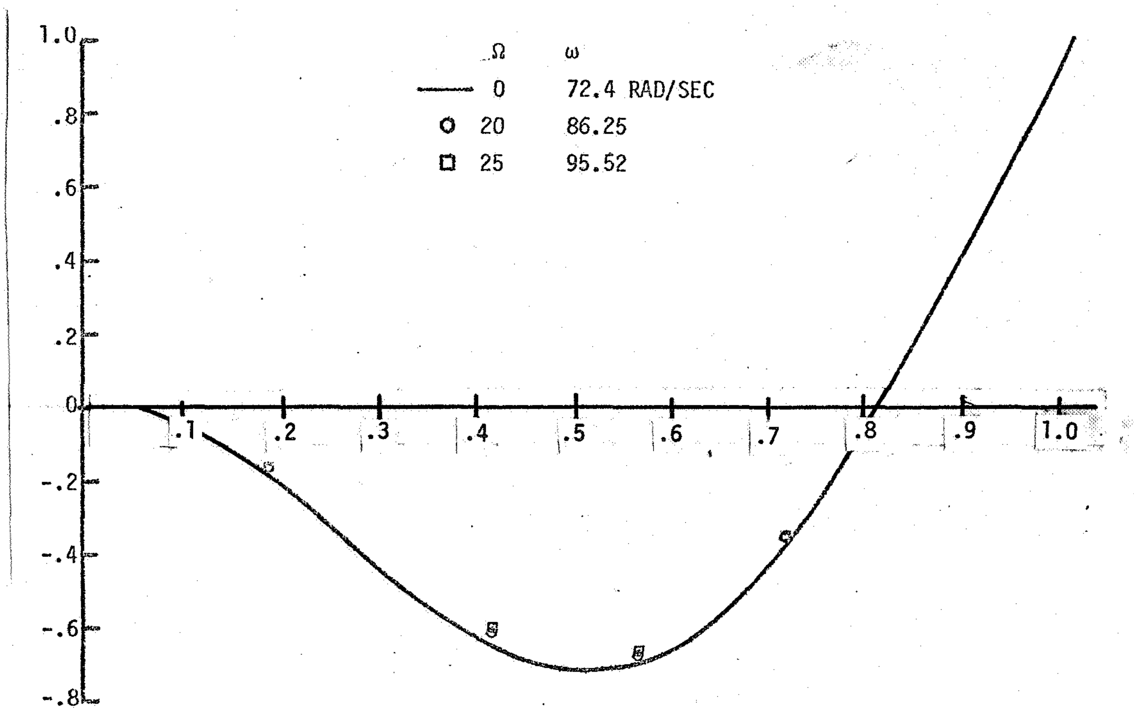


Figure 18. Out-of-Plane Shapes From 2nd OP Coupled Modes

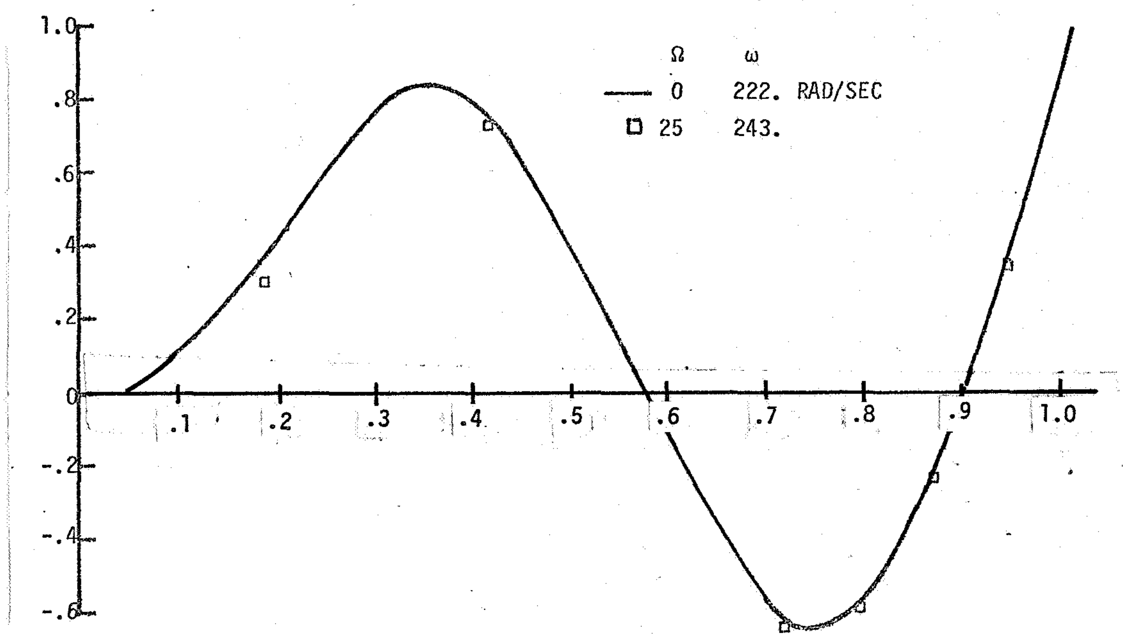


Figure 19. Out-of-Plane Shapes From 3rd OP Coupled Modes

TABLE 8. EIGHT STATION LUMPED MASS MODEL

I	STA	W	M	W	E	W	TH	W	KM
1	50.000	1.	1.445E-01	1.	-1.010E 00	1.	-2.430E-02	1.	6.310E 00
2	110.000	1.	7.300E-02	1.	-7.450E-01	1.	-5.340E-02	1.	6.190E 00
3	150.000	1.	5.540E-02	1.	-4.400E-02	1.	-7.280E-02	1.	5.650E 00
4	190.000	1.	4.580E-02	1.	1.000E 00	1.	-9.220E-02	1.	5.060E 00
5	210.000	1.	3.080E-02	1.	1.030E 00	1.	-1.020E-01	1.	5.010E 00
6	230.000	1.	3.130E-02	1.	1.060E 00	1.	-1.120E-01	1.	5.000E 00
7	250.000	1.	3.500E-02	1.	1.130E 00	1.	-1.210E-01	1.	4.960E 00
8	268.000	1.	1.000E-02	1.	1.160E 00	1.	-1.300E-01	1.	4.960E 00

TABLE 9. SAMPLE PARAMETER IDENTIFICATION OUTPUT

I	ORIG M	NEW M	PCT	ORIG E	NEW E	PCT
1	1.445E-01	1.443E-01	-0.2	-1.010E 00	-1.012E 00	0.2
2	7.300E-02	7.191E-02	-1.5	-7.450E-01	-7.562E-01	1.5
3	5.540E-02	5.415E-02	-2.3	-4.400E-02	-4.502E-02	2.3
4	4.580E-02	4.511E-02	-1.5	1.000E 00	1.015E 00	1.5
5	3.080E-02	3.072E-02	-0.3	1.030E 00	1.033E 00	0.3
6	3.130E-02	3.160E-02	1.0	1.060E 00	1.050E 00	-1.0
7	3.500E-02	3.605E-02	3.0	1.130E 00	1.097E 00	-2.9
8	1.000E-02	1.015E-02	1.5	1.160E 00	1.143E 00	-1.4

ORIG TH	NEW TH	PCT	ORIG KM	NEW KM	PCT
-2.430E-02	-2.430E-02	0.0	6.310E 00	6.315E 00	0.1
-5.340E-02	-5.340E-02	0.0	6.190E 00	6.237E 00	0.8
-7.280E-02	-7.280E-02	0.0	5.650E 00	5.715E 00	1.1
-9.220E-02	-9.220E-02	-0.0	5.060E 00	5.098E 00	0.8
-1.020E-01	-1.020E-01	-0.0	5.010E 00	5.017E 00	0.1
-1.120E-01	-1.120E-01	-0.0	5.000E 00	4.976E 00	-0.5
-1.210E-01	-1.210E-01	-0.0	4.960E 00	4.887E 00	-1.5
-1.300E-01	-1.300E-01	-0.0	4.960E 00	4.924E 00	-0.7

TABLE 10. SUMMARY OF MASS IDENTIFICATION RESULTS

Case No.	Input Modes Ω = 0: 1	2	3	4	20: 1	2	25 Rad/Sec: 1	2	3	Maximum Change (%)	Mean (%) Change	Comments
1	x	x								.7	.3	1 Eq., 24 unknowns
1a	x	x								1.5	.6	5 mass constraints, 6 Equations
2	x	x	x							-	-	very large changes
3	x	x		x						25.5	9.0	3 Equations
3a	x	x		x						26.4	9.0	mass const, 4 Equations
3b	x	x		x						24.7	9.2	5 mass constraints, 8 Equations
4	x	x	x	x						379.0	65.0	mode 3 apparently inconsistent
5					x	x				1.2	.6	
6							x	x		3.0	1.2	
7							x	x	x	13.6	3.8	3 Equations
7a							x	x	x	250.0	43.0	5 mass constraints, 8 Equations
8					x	x	x	x		307.0	45.0	2 Equations
9	x	x			x	x	x	x		412.0	51.0	3 Equations

(1) Only five generalized coordinates (modes) were used in the simulation. The torsional mode participated only slightly in any of the normal modes, thus there are essentially only four degrees of freedom in the problem. Whenever the number of equations approaches four, the necessary changes can be expected to become large. This situation, of course, will not exist in a real test and, thus, it is expected that the analysis of actual test data may be considerably more successful. It is possible to use the simulation program using up to 11 degrees of freedom and it is expected that the results of such an analysis would be considerably improved.

(2) No case where data from two rotor speeds was used was successful. It is apparent, from Figures 14-19, that the predicted changes in mode shape with rotor speed is quite small. Thus, the equations resulting from the same modes at different speeds will be nearly identical and result in a nearly singular matrix. In the simulation program, as used in this report, the same modes were used as generalized coordinates for all rotor speeds, thus accentuating this condition. Whether the use of actual test data will improve this situation is uncertain since it is well known that the mode shapes change only slightly with rotor speed.

It is also noted that any combination which included the third mode at $\Omega = 0$ yielded poor results. No particular reason is seen for this effect, except that the second and third modes contain highly coupled in and out-of-plane responses. Since the in-plane and first out-of-plane mode are quite similar, there may be some analytical problems in orthogonalizing those modes with the analytical model used.

As an illustration of the mode change analysis, keeping the mass matrix invariant, the three modes at $\Omega = 25$ rad/sec. were processed. The required changes are quite small and the results are shown in Table 11.

TABLE 11. MODE CHANGES REQUIRED FOR ORTHOGONALITY

Ω = 25 rad/sec

Percentage Changes

		Mode 2			Mode 3		
Sta	Mode 1	v	w	ϕ	v	w	ϕ
1	No change	0	-.15	0	.01	-2.53	0
2		-.01	-1.45	0	.01	-11.00	.01
3		-.02	-2.18	0	.20	-.52	0
4		-.04	-1.42	.01	.42	2.73	0
5		-.04	-.22	.01	.41	-.22	0
6		-.06	1.01	.01	.58	-1.00	0
7		-.09	3.01	.01	.87	3.75	.01
8		-.03	1.46	0	.31	4.21	0

CONCLUSIONS AND RECOMMENDATIONS

Two separate analytical methods have been developed. They both have been used as a basis for computer programs. The two programs are expected to be useful research tools for evaluating rotor dynamic analytical models in conjunction with the vacuum chamber whirl tests to be conducted at the Langley Research Center.

The first program allows the analyst to attempt to model these tests and to observe the agreement between analysis and experiment. The analytical model includes the important dynamic features of the test, such as hub degrees of freedom, non-uniform parameters, stiffness coupling between out-of-plane and in-plane motion, and the ability to simulate forcing frequency sweeps independent of rotor speed. The program has been designed to allow convenient changes in parameters, number of degrees of freedom, types of nonlinearities, periodic or transient solutions. The effects of parameters in blade responses, natural frequencies, and normal modes may be easily studied.

The second program, which is an adaptation of methods previously applied to nonrotating structures, makes use of observed blade normal modes to correct the mass and inertial coupling terms used in the analytical model. Other options allow the analyst to study the possibility of inaccurate modal measurements and combinations of modal and mass parameter changes. In addition, a feature which produces controlled random variations in the measured modes allows for a study of sensitivities of these results to inaccuracies in the observed data. The method also has the capability of making use of modes measured at more than one rotational speed.

Both programs have been extensively tested for validity and sample computations have been presented in this report. The second program which performs a class of system identification analyses, was tested using results obtained from the simulation program. The capability to handle more than a few modes or modes at more than one rotational frequency has not been demonstrated. The lack of adequate success is believed to be due to the relatively small number of generalized degrees of freedom used in the simulation program. Since other related applications of this technique have been significantly more successful, it is anticipated that the analysis of actual test data or the use of simulations having a larger number of participating modes will yield useful results.

The simulation program has the capability to use eleven blade generalized degrees of freedom. This limit is purely due to the dimensioning limitations and simple program modifications can increase this limit to any desired value. The simulation carried out used five modes as degrees of freedom. The lower frequency responses obtained are believed to be quite valid and this validity only becomes weaker as frequency ranges are reached which in reality include participation of modes which were not included in the analysis.

The following recommendations are made for useful continuation of this research.

(1) Develop an analytical model, which is a better intuitive representation of the actual rotor system to be tested.

(2) Simulate specific test conditions and make direct comparisons with actual test responses. If obvious apparent discrepancies exist, make rational intuitive changes in the analytical parameters whenever such changes can be justified by consideration of the physical characteristics of the rotor.

(3) Use actual measured normal modes in both the nonrotating and rotating conditions to correct the mass and inertial coupling parameters and to study the sensitivities to measurement errors. Use these results to evaluate the possibility of obtaining significant information from nonrotating tests alone. Evaluate the use of this method to improve the analyst's capability to derive a more satisfactory model from the physical characteristics of the blades prior to any testing.

(4) Use the simulation program for conditions and blades other than those tested to study the effects of blade and hub parameters on natural frequencies, blade and rotor responses and stability.

(5) Because the simulation program is a convenient, flexible and adaptable program, it is strongly recommended that further developments of this program to include aerodynamics,controls and a more comprehensive fuselage representation be considered.

REFERENCES

1. Piziali, R. A., "An Investigation of the Structural Dynamics of Helicopter Rotors", USAAVLABS Technical Report 70-24, U. S. Army Aviation Material Laboratories, Fort Eustis, VA, April 1970.

2. Houbolt, J. C. and Brooks, G. W., "Differential Equations of Motion for Combined Flapwise Bending, Chordwise Bending, and Torsion of Twisted Nonuniform Rotor Blades", Report 1346, National Advisory Committee for Aeronautics, Langley Aeronautical Laboratory, Langley Field, VA, 1958.

3. Hodges, D. H. and Dowell, E. H., "Nonlinear Equations of Motion for the Elastic Bending and Torsion of Twisted Nonuniform Rotor Blades", NASA TN D-7818, National Aeronautics and Space Administration, Washington, D. C., December 1974.

4. Berman, A. and Flannelly, W. G., "Theory of Incomplete Models of Dynamic Structures", AIAA Journal, Volume 9, No. 8, August 1971, pp 1481-1487.

5. Berman, A. and Flannelly, W. G., "Study of Incomplete Models of Dynamic Structures", NASA CR-109630, Goddard Space Flight Center, Greenbelt, Maryland, January 1970.

6. Gravitz, S. I., "An Analytical Procedure for Orthogonalization of Experimentally Measured Modes", Journal of the Aerospace Sciences, Volume 25, November 1958, pp 721-2.

7. McGrew, J., "Orthogonalization of Measured Modes and Calculation of Influence Coefficients", AIAA Journal, Volume 7, April 1969, pp 774-6.

8. Targoff, W. P., "Orthogonality Check and Correction of Measured Modes", AIAA Journal, Volume 14, February 1976, pp 164-7.

9. Lee, B. L., "Experimental Measurements of the Rotating Frequencies and Mode Shapes of a Full Scale Helocopter Rotor in a Vacuum and Correlations With Calculated Results", 35th Annual National Forum of the AHS, May 1979.

10. Hassal, C.J.W., "Development and Initial Application of a Technique to Measure Vibration Mode Shapes of a Rotating Blade", RAE Technical Report 77064, 1977.

APPENDIX A
DEFINITIONS OF INTEGRALS

Mass Integrals (Sta. No., Coefficient No.)

$$\int \equiv \int_{x}^{R} (\quad) dx$$

MI(I,1) = $\int m$

MI(I,2) = $\int mx$

MI(I,3) = $\int me$

MI(I,4) = $\int mex$

MI(I,5) = $\int me\theta$

MI(I,6) = $\int mex\theta$

MI(I,7) = $\int mK_{m_2}^{2}$

MI(I,8) = $\int mk_{m_2}\theta$

MI(I,9) = $\int m\Delta K\theta$

MI(I,10) = $\int K_A^{2}\tau\theta'$

MII(I,1) = $\int$MI(I,1)

MII(I,2) = $\int$MI(I,2)

MII(I,3) = $\int$MI(I,3)

MII(I,4) = $\int$MI(I,4)

MII(I,5) = $\int$MI(I,5)

MII(I,6) = $\int$MI(I,6)

MII(I,7) = $\int$MI(I,7)

MII(I,8) = $\int$MI(I,8)

MII(I,9) = $\int$MI(I,9)

I = 1 to number of blade stations

Y Integrals (Sta. No., Mode No., Coefficient No.)

$$\int \equiv \int_x^R (\quad) dx$$

$YI(I,J,1) = \int mY_J$

$YI(I,J,2) = \int meY_J$

$YI(I,J,3) = \int me\theta Y_J$

$YI(I,J,4) = \int mxY_J$

$YI(I,J,5) = \int meY'_J$

$YI(I,J,6) = \int mex\theta Y'_J$

$YI(I,J,7) = \int \tau Y''_J$

$YI(I,J,8) = \int e_A \tau\theta Y''_J$

$YI(I,J,9) = \int E_1 \theta' Y''_J$

$YI(I,J,10) = \int_o^x e_A Y''_J \, dx$

$YII(I,J,1) = \int YI(I,J,1)$

$YII(I,J,2) = \int YI(I,J,2)$

$YII(I,J,3) = \int YI(I,J,3)$

$YII(I,J,4) = \int YI(I,J,4)$

$YII(I,J,5) = \int YI(I,J,5)$

$YII(I,J,6) = \int YI(I,J,6)$

$YII(I,J,7) = \int YI(I,J,7)$

$YII(I,J,8) = \int YI(I,J,8)$

$YII(I,J,9) = \int YI(I,J,9)$

I = 1 to number of blade stations

J = 1 to number of in-plane modes

Z Integrals (Sta. No., Mode No., Coefficient No.)

$$\int \equiv \int_x^R (\quad)dx$$

$ZI(I,J,1) = \int mZ_J$

$ZI(I,J,2) = \int meZ_J$

$ZI(I,J,3) = \int mxZ'_J$

$ZI(I,J,4) = \int mexZ'_J$

$ZI(I,J,5) = \int me\theta Z'_J$

$ZI(I,J,6) = \int \tau Z''_J$

$ZI(I,J,7) = \int e_A \tau Z''_J$

$ZI(I,J,8) = \int E_1 \theta\theta' Z''_J$

$ZI(I,J,9) = \int_0^x e_A \theta Z_J$

$ZII(I,J,1) = \int ZI(I,J,1)$

$ZII(I,J,2) = \int ZI(I,J,2)$

$ZII(I,J,3) = \int ZI(I,J,3)$

$ZII(I,J,4) = \int ZI(I,J,4)$

$ZII(I,J,5) = \int ZI(I,J,5)$

$ZII(I,J,6) = \int ZI(I,J,6)$

$ZII(I,J,7) = \int ZI(I,J,7)$

$ZII(I,J,8) = \int ZI(I,J,8)$

I = 1 to number of blade stations

J = 1 to number of out-of-plane modes

ϕ Integrals (Sta. No., Mode No., Coefficient No.)

$$\int \equiv \int_x^R (\quad) dx$$

$PI(I,J,1) = \int me\Phi_J$

$PI(I,J,2) = \int mex\Phi_J$

$PI(I,J,3) = \int me\theta\Phi_J$

$PI(I,J,4) = \int mK_m^2\Phi_J$

$PI(I,J,5) = \int m\Delta K\Phi_J$

$PI(I,J,6) = \int E_\phi\Phi'_J$

$PI(I,J,7) = \int K_A^2\tau\Phi'_J$

$PI(I,J,8) = \int_0^x K_A^2\theta\Phi'_J dx$

$PII(I,J,1) = \int PI(I,J,1)$

$PII(I,J,2) = \int PI(I,J,2)$

$PII(I,J,3) = \int PI(I,J,3)$

$PII(I,J,4) = \int PI(I,J,4)$

$PII(I,J,5) = \int PI(I,J,5)$

I = 1 to number of blade stations

J = 1 to number of torsional modes

Special Integrals (Sta. No., Mode No., Coefficient No.)

$$\int \equiv \int_x^R \int_x^R (\quad)dxdx$$

$$SI(I,J,1) = \int m \int_o^x \frac{1}{EA} YI(I,J,1)dx$$

$$SI(I,J,2) = \int mYI(I,J,10)$$

$$SI(I,J,3) = \int mZI(I,J,9)$$

$$SI(I,J,4) = \int mPI(I,J,8)$$

$$SI(I,J,5) = \int_x^R K_A{}^2 \theta YI(I,J,1)dx$$

v Equation Integrals

$$\int \equiv \int_{0}^{R} (\quad) dx$$

$$DYYI(K,J,2) = \int Y_K \; YI(I,J,2)$$

$$DYYII(K,J,1) = \int Y_K \; YII(I,J,1)$$

$$DYYII(K,J,4) = \int Y_K \; YII(I,J,4)$$

$$DYYII(K,J,5) = \int Y_K \; YII(I,J,5)$$

$$DYYII(K,J,7) = \int Y_K \; YII(I,J,7)$$

$$DYZII(K,J,1) = \int Y_K \; ZII(I,J,1)$$

$$DYZII(K,J,5) = \int Y_K \; ZII(I,J,5)$$

$$DYPII(K,J,3) = \int Y_K \; PII(I,J,3)$$

$$DYMI(K,4) = \int Y_K \; MI(I,4)$$

$$DYMII(K,1) = \int Y_K \; MII(I,1)$$

$$DYMII(K,2) = \int Y_K \; MII(I,2)$$

$$DYMII(K,3) = \int Y_K \; MII(I,3)$$

$$DYMII(K,5) = \int Y_K\ MII(I,5)$$

$$DYSI(K,J,i) = \int Y_K\ SI(I,J,i) \qquad i = 1 \text{ to } 4$$

$$DYF(K,J,1) = \int Y_K\ (R - x)(meY_J)_R$$

$$DYF(K,J,2) = \int Y_K e_A YI(I,J,1)$$

$$DYF(K,J,3) = \int Y_K EvY''_J$$

$$DYF(K,J,4) = \int Y_K\ E\ Z''_J$$

$$DYF(K,J,5) = \int Y_K(EC_1{*}\theta P''_J + E_1\theta' P'_J)$$

$$DYF(K,1,6) = \int Y_K(e\tau + (me)_R R(R - x)$$

$$DYD(K,J) = gv\int_o^R Y_K \int_x^R \int_x^R Y_J$$

$$DYALII(K) = \int_o^R Y_K \int_x^R \int_x^R L_v$$

K, J = 1 to number of (1-P, 0-P or torsion) modes w Equation Integrals

$$\int \equiv \int_{o}^{R} (\quad)dx$$

$$DZYI(K,J,3) = \int Z_K \, YI(I,J,3)$$

$$DZPI(K,J,2) = \int Z_K \, PI(I,J,2)$$

$$DZZII(K,J,1) = \int Z_K \, ZII(I,J,1)$$

$$DZZII(K,J,3) = \int Z_K \, ZII(I,J,3)$$

$$DZZII(K,J,6) = \int Z_K \, ZII(I,J,6)$$

$$DZYII(K,J,1) = \int Z_K \, YII(I,J,1)$$

$$DZPII(K,J,1) = \int Z_K \, PII(I,J,1)$$

$$DZMI(K,6) = \int Z_K \, MI(I,6)$$

$$DZMII(K,i) = \int Z_K \, MII(I,i) \qquad i = 1 \text{ to } 3$$

$$DZI(K,J,1) = \int Z_K[(R - x)(me\theta Y_J)_R + e_A\theta YI(I,J,1)$$

$$DZF(K,J,2) = \int Z_K \Delta E\theta Y_J''$$

$$DZF(K,J,3) = \int Z_K EwZ_J''$$

$$DZF(K,J,4) = \int Z_K[EC_1{*}P_J'' + E_1\theta\theta' P_J']$$

$$DZF(K,1,5) = \int Z_K[R(R - x)(me\theta)_R - e_A\tau\theta]$$

$$DZF(K,J,6) = \int Z_K[e_A\tau P_J - R(R - x)(meP_J)_R$$

$$DZD(K,J) = g_w \int_{o}^{R} Z_K \int_{x}^{R} \int_{x}^{R} Z_J$$

$$DZALII(K) = \int_{o}^{R} Z_K \int_{x}^{R} \int_{x}^{R} L_w$$

K, J = 1 to number of corresponding modes

ϕ Equation Integrals

$$\int \equiv \int_0^R (\quad) dx$$

$DPYI(K,J,9) = \int \Phi_K \, YI(I,J,9)$

$DPYII(K,J,3) = \int \Phi_K \, YII(I,J,3)$

$DPYII(K,J,6) = \int \Phi_K \, YII(I,J,6)$

$DPYII(K,J,8) = \int \Phi_K \, YYII(I,J,8)$

$DPZI(K,J,8) = \int \Phi_K \, ZI(I,J,8)$

$DPZII(K,J,2) = \int \Phi_K \, ZII(I,J,2)$

$DPZII(K,J,4) = \int \Phi_K \, ZII(I,J,4)$

$DPZII(K,J,7) = \int \Phi_K \, ZII(I,J,7)$

$DPPI(K,J,6) = \int \Phi_K \, PI(I,J,6)$

$DPPI(K,J,7) = \int \Phi_K \, PI(I,J,7)$

$DPPII(K,J,4) = \int \Phi_K \, PII(K,J,4)$

$DPPII(K,J,5) = \int \Phi_K \, PII(I,J,5)$

$DPMII(K,i) = \int \Phi_K \, MI(I,i)$ $\qquad i = 3, 4; \;\; 6 \text{ to } 10$

$DPSI(K,J,1) = \int \Phi_K \, SI(I,J,5)$

$DPF(K,J,1) = \int \Phi_K \, EC_1 * Y''_J$

$DPF(K,J,2) = \int \Phi_K \, EC_1 * Y''_J$

$DPF(K,J,3) = \int \Phi_K \, EC_1 * Z''_J$

$DPD(K,J) = g_{\Phi} \int_0^R \Phi_{K} {}_{K} \int_x^R \int_x^R \Phi K_J$

$DPALII(K) = \int_0^R \Phi_K \int_x^R \int_x^R M\Phi$

K, J = 1 to number of appropriate modes

APPENDIX B

USERS GUIDE

V22

DYNAMIC ROTOR SIMULATION PROGRAM

First card of each case is HEADING CARD (see next page for description and exceptions).

All other data may be entered in any order (data blocks must maintain order within block). Data not entered (after 1st case) retains previous values (if any). All data is self identified by value of IO punched in col 1,2 of card on first card of block.

INPUT SUMMARY

IO	Type of Data	No. of Cards	Required?
01	Blade Properties	Block	Yes (Must precede IO = 3,4 or 5,13)
02	Blade Data	1	No (Default to 0's)
03	Modes: In-Plane (Y)	Block	No (At least one of 3,4,5 required)
04	Out-of-Plane (Z)	Block	No
05	Torsion (P)	Block	No
06	Frequencies (Ω, ω_f)	1	Yes
07	Hub Data, X,M,C,K,F	1	No
08	Y	1	No
09	Z	1	No
10			
11			
12			
13	Applied Forces, Blades	1	No
14			
15			
16			
17	Special Controls - Nonlin, Floquet	1	No (Default to Nonlinear)
18	Solution Controls	1	Yes
19			
20			
21	Special IO Cancel	1	No

HEADING CARD

Col 1	IC1	#0	Ends run (same as IEND = 3, see below)
2	IC2	#0	All input printed (else only new data printed)
3	IC3	#0	Prints definite integrals
4	IC4	#0	Prints coefficient matrices
5	IC5	#0	Writes data on tape (see below)
6-80	Arbitrary heading		

The heading card is the first card of the first case and the first card of each following case unless the preceding case ended with IEND = 2 (see below)

GENERAL INPUT

I0 in col 1,2 of 1st card only of each block.

IEND in col 80 of single card - see details of each block input.

IEND = 1 end of data, followed by HEADING and new data

= 2 same as 1 but omit HEADING card from next case

= 3 ends run at completion of case

No special ending required for block data input

All data has following format. Real and integer input may be mixed.

I2, F8.0, 6F10.0, F9.0, I1

Do not use col 1 or 2 except for I0 (on first card of block)

Do not use col 80 except to end case

TAPE DATA (IC5 #0)

Uses FORTRAN unit 9. Data records are as follows ψ (in degrees, not limited to 360), tip in-plane deflection, tip out-of-plane deflection, tip torsional deflection, x_H, y_H, z_H. Blade 1 only

IO = 1 BLADE PROPERTIES REQUIRED

Must precede IO = 3,4,5,13
IO on first card only, col 1,2 blank on all succeeding cards
2 cards per station (order 1,2,1,2...)
20 stations max
IEND (if used)on last card 1
Definitions consistent with TN D-7818

Word	Card 1	Card 2
1	X - sta (ascending sequence)	EOP - EI_y, (EI out of chord plane)
2	M - mass/unit length	EIP - EI_z, (EI for bending in chord plane)
3	E - e	GJ
4	SEA - e_A	EA - (if 0 then $\frac{1}{EA}$ is set to 0)
5	Km1 - k_{m_1}	EB1 - EB_1*
6	Km2 - k_{m_2}	EB2 - EB_2*
7	KA - k_A	EC - EC_1
8	THP - θ' built in pitch - rad/ unit length	ECS - EC_1*

IO = 2 BLADE DATA OPTIONAL (Default to 0)

Word	
1	NB - no of blades 4 max (Default to 1 if no hub DOF (Default to 2 if hub DOF included
2	THO - θ_o angle at x(1) - radians
3	BPC - β_{PC} - pre-cone - radians
4	GV - blade damping, 1-P appropriate units, viscous
5	GW - blade damping, 0-P appropriate units, viscous
6	GP - blade damping, torsion appropriate units, viscous

IO = 3 MODES IN-PLANE — Max 3 modes

IO = 4 MODES OUT-OF-PLANE — Max 5 modes

IO = 5 MODES TORSION — Max 3 modes

(At least one of IO = 3,4,5 reqd)

Each mode has one set of input - second derivative at each station followed by the first derivative at station 1 (slope and deflection are obtained by integration and normalized to unit deflection at tip)

Input - 8 elements per card - as many cards as necessary (3 max), all functions start on new card

IO on first ( )" card - all other col 1,2 blank
IEND (if used) on 1st ( )" card of last mode

Order of input:

1st mode: $(\ )''_{x_1}$ $(\ )''_{x_2}$ $(\ )''_{x_3}$ . . . .

$(\ )''_{x_9}$ . . .

new card $(\ )'_{x_1}$ word 1 only, slope at station 1 (normally = 0)

new card $(\ )_{x_1}$ word 1 only, deflection at station 1 (normally = 0)

next mode
new card $(\ )''_{x_1}$ $(\ )''_{x_2}$ . . . .

etc

IO = 6 FREQUENCIES REQUIRED

Word

1 OMEG - Ω - rotor speed, rad/sec

2 OMF - ω_f - forcing frequency, rad/sec

IO = 7 HUB DATA, X OPTIONAL

IO = 8 HUB DATA, Y OPTIONAL

IO = 9 HUB DATA, Z OPTIONAL

Impedance in each direction may be represented as spring-mass-damper at frequency ω_f. Data omitted implies infinite impedance. If any hub data is input - at least two blades required.

Word			
1	HM	x y z	Mass
2	HC	x y z	Damping Coeff
3	HK	x y z	Spring Rate
4	HF	(1) (2) (3)	Force - multiplied by sin $\omega_f t$ (or by 1 if $\omega_f = 0$)

IO = 13 APPLIED FORCES, BLADES OPTIONAL

Load may be applied at any one station, but in three directions. Amplitudes are multiplied by sin $\omega_f t$ (or by 1 if $\omega_f = 0$). Forces may be applied to one or all blades. $\omega_f t$ always refers to blade 1, however, producing "umbrella mode" forcing. (See IO = 7, 8, 9 for hub forcing).

Word		
1	NXF	Station index number (see IO = 1)
2	AFY	Amplitude in y direction
3	AFZ	z
4	AFP	ϕ
5	NBF	Blade number to which force is applied - 0 applies forces to all blades simultaneously. If >NB, NBF is set to 0.
6	PER	Period as fraction of 360° (1 - cos) force is applied from $\psi = 0$ to ψ = PER*2. OMF (IO = 6) is ignored. Integration interval must be selected with core (IO = 18).

Note: If in-plane hub degrees of freedom are used (IO = 7 or 8) AFY or NBF must = 0.

IO = 17 SPECIAL CONTROLS - NONLIN, FLOQUET OPTIONAL (Default to nonlinear, no floquet)

FLOQUET OPTION: Produces Floquet transition matrix using force cycle (ω_f) unless ω_f = 0 then rotor cycle is used. Note that if in-plane hub D-O-F are used equation contains terms periodic in Ωt. If a force is applied then the boundary conditions for a (linear) periodic solution are determined and solution is executed for number of cycles specified in IO = 18. This overrides any other initial condition(s).

A maximum at 15 degrees of freedom are allowed for this option (30 variables including velocities).

Word

1	NLIN =	0	All nonlinear terms included
	=	1	In-plane nonlinear terms only
	=	2	Linear terms only
2	NFLOQ =	1	Floquet option (see discussion just above)
	=	2	Same as 1, but steady effects of offsets and twists and precone are ignored.

IO = 18 SOLUTION CONTROLS REQUIRED

Errors and initial conditions are limited to one variable.

Word

1	CYCLES	Number of force* cycles for solution to run
2	HINIT	Number of integration intervals per cycle
3	ERROR	Error bound (appropriate units), see IYE
4	IYE	Index of variables tested for ERROR**
5	CIC	Initial condition (appropriate units), see IYIC
6	IYIC	Index of variable for initial condition
7	BERR	Upper limit (abs) of variable (IYE) which stops run. If = 0 no limit

* Force cycle is used unless ω_f = 0 (IO = 06), then rotor cycle is used.

** See section on variable numbers following.

IO = 21 SPECIAL IO CANCEL OPTIONAL

For cases after the first, IO's previously used may be cancelled. When this option is used all coefficients are recalculated and IC2 is set to 1 (see HEADING CARD) to insure data printout. There is no necessity to cancel when data is replaced.

Word

1-8 IO's to be cancelled (O's ignored)

VARIABLE NUMBERS

In IO18 the variables are referred to by numbers. These numbers are as follows:

I	Variable		
1	$\dot{x}_H$		
2	x_H		
3	$\dot{y}_H$		
4	y_H		
5	$\dot{z}_H$		
6	z_H		
11	$\dot{y}_1$	Blade 1	I = 9 + 2 NM(IB-1)
12	y_1		
13	$\dot{y}_2$	NM = no. of modes	
⋮	⋮		
	last y	IB = blade number	
	$\dot{z}_1$		
	z_1		
	⋮		
	last z		
	$\dot{\phi}_1$		
	ϕ_1		
	⋮		
	last ϕ		
	$\dot{y}_1$	Blade 2	
	y_1		
	etc.		

ERROR MESSAGES

Certain errors terminate the run. Others are warnings with correction as indicated below. All error numbers refer to a Fortran statement number in vicinity of error. (All are in INPU except for the 5000 series which occur in SOL).

NUMBER	REASON	TERMINATE
10	Inactive IO	Yes
11	"	Yes
14	"	Yes
15	"	Yes
16	"	Yes
19	"	Yes
20	"	Yes
200	Invalid IO	Yes
202	More than one input of same IO, last one used	No, IO*
203	I021, Attempt to cancel invalid Iϕ	Yes
215	I01, Stations out of seq	Yes
216	I01, Too many stations	Yes
262	I03, Too many Y modes	Yes
264	I04, Too many Z modes	Yes
266	I05, Too many P modes	Yes
500	No IO = 1	Yes
501	No IO = 3,4 or 5	Yes
502	No IO = 6	Yes
506	I02 NB > 4, set to 4	No, NB*
507	I02 NB < 1, set to 1 or 2 (2 if HUB DOF)	No, 1*
509	IO = 18 Missing	Yes
510	In-plane hub with AFY•OR•NBF•NE•0	No, NBF*

NUMBER	REASON	TERMINATE
510	I013, NYF < 0 CR >NX	Yes
511	I013, All forces 0	Yes
512	I013, NB < NBF < 0 Sets NBM to 6	No, NBF*
1100	I018, Error < 0	Yes
1105	I018, IYIC < 0	Yes
1106	I018, IYIC > NDIM	Yes
1107	I018, IYE < 0	Yes
1108	I018, IYE > NDIM	Yes
5010	Too many D-O-F for Floquet	Yes
5030	IHLF = 11	Yes
5031	IHLF = 12	Yes
5032	IHLF = 13	Yes

* This quantity is printed with warning.

```
***************************************************************************
    ROTSI              ROTSI              ROTSI              ROTSI
                 ROTOR SYSTEM IDENT -- INCOMPLETE MODEL
***************************************************************************

  INPUT
  -----                 COL

  (1) HEADING            1  IC1 .EQ 0  FIRST OR NORMAL RUN -- ALL INPUT
                                    1  REPLACE MODES  - INPUT 3,4,5
                                    2  ADD MODES - INPUT 4,5

                                    8  NEW OP CODE ONLY - INPUT 5
                                    9  END OF RUN - LAST CARD OF RUN

                         2  IC2 .EQ.1  PRINTS ORTHO CHECKS
                                    2      AND NORMALIZES MODES
                                           NOTE--MODES ARE REPLACED
                                           AFTER INPUT

                         3  IC3 .NE.0  PRINTS EQS FOR MASS IDENT

                         4  IC4 .NE.0  RESTORES INPUT MODES, IF IC1.EQ.8

                         5-80  ARBITRARY HEADING    HEAD(19)

  (2)  MASS DATA - ONE CARD PER BLADE STATION    20 MAX

                        1-10  X(I) STATION
                        11    * (SEE NOTE) WM
                        12-20 M - LUMPED MASS
                        21    * (SEE NOTE) WE
                        22-30 E - CG OFFSET FROM EA  + WHEN CG FORWARD
                        31    * (SEE NOTE) WT
                        32-40 TH - PITCH ANGLE -- RAD
                        41    * (SEE NOTE) WK
                        42-50 KM    RADIUS OF GYRATION IN TORSION

        *  1ST COL OF EACH WORD CONTAINS WEIGHTING FACTOR
           FROM 1-9 (0=1)  HIGHER VALUE INDICATES GREATER CONFIDENCE
           SEE IO1 = 3  WD1

    END WITH BLANK CARD

  (3)  CONTROL CARD - MODES

                        1-10  CALV   MULTIPLIES I-P MODE DEFL  (0=1)
                        11-20 CALW   MULTIPLIES O-P MODE DEFL  (0=1)
                        21-30 CALP   MULTIPLIES TOR MODE DEFL  (0=1)
                        31-40 THO    ROOT PITCH ANGLE -- RAD
                                     ADDS TO TH - (TH NOT CHANGED)
```

```
(4)  MODES  -  STATIONS CORRESPOND TO MASS DATA

     EACH MODE          1-10   FREQ    NATURAL ,   RAD/SEC
                       11-20   OMEG    ROTATIONAL, RAD/SEC
                       21-30   IF .NE. 0 TEMPORARILY REPLACES CALV
                       31-40   IF .NE. 0 TEMPORARILY REPLACES CALW
                       41-50   IF .NE. 0 TEMPORARILY REPLACES CALP

      NEXT CDS      V  I-P DISPLACEMENTS, 8F10.  UP TO 3 CARDS
      NEXT CDS      W  O-P                       START ON NEW CD
      NEXT CDS      P  TOR

      FOLLOW BY NEXT MODE   - 8 MODES MAX AT ONE OMEG
                            -16 MODES MAX AT ALL OMEG
          *** 30 EQS MAX (NOT INCL INVARIANCES) ***

   END WITH BLANK CARD

(5)  OPERATION CODES  COL  1,2    I01,I02

     COL 1   I01

         1   MODIFY MODES WITH RANDOM ERRORS  -  MODES REPLACED

             WD1  PERCENT RANDOM + OR - RECTANGULAR DIST
             WD2  PERCENT BIAS
             WD3  INTEGER SEED TO START RANDOM SEQUENCE

             *** FOLLOW BY NEXT OPERATION CARD (5) ***

         2   SOLVE FOR MINIMUM MODAL CHANGES - MASS MATRIX UNCHANGED

                  ALL MODES MUST BE AT SAME OMEGA     - 8 MAX
                  FIRST MODE UNCHANGED, LAST MODE WILL CHANGE MOST
                  MINIMUM SUM PERCENT CHANGES USED
                  WEIGHTING FACTORS NOT USED IN THIS OPTION

             WD1.EQ.0   NO LIMIT ON CHANGES
             WD1.EQ.1   LIMIT CHANGES - SCALE OPTION
             WD2-8   MAX PCT CHANGE ALLOWED IN EACH MODE.
                     CHANGES ARE SCALED SO MAX CHANGE .LE. MAXIMUM
                     0 INDICATES NO LIMIT.

             WD1.EQ.2   LIMIT CHANGES - TRUNCATE OPTION
             WD2-8   SAME AS FOR SCALE OPTIONEXCEPT THAT ONLY
                     CHANGES WHICH EXCEDE LIMITS ARE TRUNCATED.
                     OTHER CHANGES ARE NOT MODIFIED
```

```
      3   INCOMP MODEL MASS CHANGES

            WD1.EQ.1  WEIGHTING FACTORS ALL SET TO 1 (TEMP)
            WD1.EQ.2  STAS WITH INVATIANT PARAM.  READ-5(A)

                THE FOLLOWING CONTROLS CAUSE THE CORRESPONDING
                PROPERTIES TO REMAIN INVARIANT IF .NE. 0.

           COL 20  TOTAL MASS      M
               30  RADIAL CG       M*X
               40  CHORDWISE CG    M*E
               50  FLAPPING MOM OF INERT     M*X**2
               60  FEATHERING MOM OF INERT   M*KM**2

       COL 2  I 02

       0   ABOVE OPERATIONS DO NOT DISTURB ORIGINAL DATA

       1   ABOVE OPERATIONS REPLACE ORIGINAL DATA IN PREPERATION
           FOR SEQUENTIAL OPERATIONS

 (5A)    USED ONLY FOR INVAR STAS.    SEE 3,ABOVE, WD1 = 2

            COL1 = NO OF STATIONS (8 MAX)
            WD1,WD2,...STATION NUMBERS, NO ZEROES

 NEXT HEADING CARD
****************************************************************************************
****************************************************************************************
```

APPENDIX C

PROGRAM LISTINGS

```
C                    V22          V22               V22
C
      REAL M,KM1,KM2,KA
      LOGICAL LY
C                          COMMON FOR INPUT
      COMMON/INDAT/X(20),M(20),E(20),SEA(20),KM1(20),KM2(20),KA(20),
     1 THP(20),EOP(20),GJ(20),EA(20),EB1(20),EB2(20),ECS(20),EIP(20),
     2 THO,BPC,YPP(20,3),YP(20,3),ZPP(20,5),ZP(20,5),PPP(20,3),PP(20,3),
     3  OMEG,OMF,EC(20),NY,NZ,NP,NM,OMEGS,OMFS,IDIM,NMAX,NLIN
     4,NB,HMX,HMY,HMZ,HCX,HCY,HCZ,HKX,HKY,HKZ,NX,NFLOQ
     5 ,HINIT,ERROR,IYE,CIC,IYIC,BERR ,CYCLES ,NXF,AFY,AFZ,AFP,NBF
     6  ,R,GV,GW,GP,HE(3),PER
C                                COMMON COEFFICIENT MATRICES
      COMMON/COEF/COI(11,11),DCOI(11,11),COD(11,11),DCOD(11,11),
     1    CO(11,11),DCO(11,11),F(11),DF(11),FNL(11),COIR(11,12),
     2    CODR(11,11),COR(11,11),FR(11),RIOC(11,12),BF(11)
     3 ,BIN(3,11),BDAM(3,11),BSPR(3,11),COIH(11,3),CODH(11,3),BIRI(3,11)
     4,BIRID(3,11),BIRIO(3,11),BIRIDH(3,3 ),HF(3),TM(3,3),BIRIIH(3,3 )
     5 ,HC(3,3),HK(3,3)
C           COMMON FOR HEADING, CONTROL DATA
      COMMON/HED/IC1,IC2,IC3,IC4,HEAD(19),IPAGE,INPUT(20),IEND,LINE,IC5
C                          COMMON DIMENSION DATA
      COMMON/DIM/NINPUT,NSTA,NYMODE,NZMODE,NPMODE,NMODE,NM1,NDIM,NBLADE
C                        COMMON BASIC DERIVED DATA
      COMMON/DER/TH(20),EV(20),EW(20),EP(20),Y(20,3),Z(20,5),P(20,3)
C               COMMON VARIABLES AND SOLUTION CONTROLS
      COMMON/VAR/YVAR(98),DERY(98),PRMT(6),LY(98)
C                       DIMENSIONALIZATION

      NINPUT = 20
      NSTA = 20
      NYMODE = 3
      NZMODE = 5
      NPMODE = 3
      NMODE=11
      NM1 =  NMODE+1
      NBLADE = 4
      NDIM = 98
      DO 10 I=1,NINPUT
      F51=1.0E+51
   10 INPUT(I)=0
      ICASE=0
      IEND = 0
   20 CALL INPU (ICASE)

      LINE = 100
      CALL      SOL(PRMT,YVAR,DERY,IHLF,LY)
      IF(IC5.NE.0) WRITE(9) (F51,I=1,7)
  100 IF(IEND.EQ.3) CALL EXIT

      GO TO 20
      END
```

```
      FUNCTION DINT (DUMP,DUMPP,X,NX)
C       DUMP IS INTEGRAL OF DUMPP
      REAL DUMP(1),DUMPP(1),X(1)
      CALL INT (DUMP,DUMPP,0,X,NX,1)
      DINT=DUMP(NX)
      RETURN
      END
```

```
      FUNCTION DINT1 (A,B,I1    ,N,X,NA,NX,DUMP,DUMPP)
      REAL A(NA,1),B(NA,1),X(1),DUMP(1),DUMPP(1)
      DO 10  I=1,NX
   10 DUMPP(I)= A(I,I1)*B(I,N)
      CALL INT (DUMP,DUMPP,0,X,NX,1)
      DINT1=DUMP(NX)
       RETURN
      END
```

```
      FUNCTION DINT2 (A,B,I1,I2,N,NB,X,NA,NX,DUMP,DUMPP)
      REAL A(NA,1),B(NA,NB,1),X(1),DUMP(1),DUMPP(1)
      DO 10  I=1,NX
   10 DUMPP(I) = A(I,I1) * B(I,I2,N)
      CALL INT (DUMP,DUMPP,0,X,NX,1)
      DINT2=DUMP(NX)
       RETURN
      END
```

```
      SUBROUTINE ERR(N,I)
C    I = 0, TERMINATES RUN        I NE 0 WARNING ONLY, PRINTS I
      PRINT 10,N
   10 FORMAT(//10X,17H*** ERROR NUMBER  ,I5,5H  ***  )
      IF (I.NE.0) GOTO 20
      CALL EXIT
   20 PRINT 30,I
   30 FORMAT (20X,20H*** WARNING ONLY ***  ,I5//)
      RETURN
      END
```

```
      SUBROUTINE FCT(T,YVAR,DERY,LY,INDIM)
C       NOTE   INDIM NOT USED   INCLUDED FOR COMPATABILITY ONLY
C          MULTI BLADES,   3 DOF HUB,      NON-LIN CORIOLIS FORCES
      DIMENSION YVAR(1),DERY(1)
      LOGICAL  LY(1)
      REAL M,KM1,KM2,KA
      REAL DUMP(20),DUMPP(20),VDM(20)
      REAL VD(20),VDP(20),VP(20),VPP(20),WD(20),WDP(20),WP(20),WPP(20)
      COMMON/INDAT/X(20),M(20),E(20),SEA(20),KM1(20),KM2(20),KA(20),
     1 THP(20),EOP(20),GJ(20),EA(20),EB1(20),EB2(20),ECS(20),EIP(20),
     2 THO,BPC,YPP(20,3),YP(20,3),ZPP(20,5),ZP(20,5),PPP(20,3),PP(20,3),
     3   OMEG,OMF,EC(20),NY,NZ,NP,NM,CMEGS,OMFS,IDIM,NMAX,NLIN
     4,NB,HMX,HMY,HMZ,HCX,HCY,HCZ,HKX,HKY,HKZ,NX,NFLOQ
     5 ,HINIT,ERROR,IYE,CIC,IYIC,BERR ,CYCLES ,NXF,AFY,AFZ,AFP,NBF
     6   ,R,GV,GW,GP,HE(3),PER
      COMMON/COEF/COI(11,11),DCOI(11,11),COD(11,11),DCOD(11,11),
     1    CO(11,11),DCO(11,11),F(11),DF(11),FNL(11),COIR(11,12),
     2    CODR(11,11),CCR(11,11),FR(11),RIOC(11,12),BF(11)
     3 ,BIN(3,11),BDAM(3,11),BSPR(3,11),COIH(11,3),CODH(11,3),BIRI(3,11)
     4,BIRID(3,11),BIRIO(3,11),BIRICH(3,3 ),HF(3),TM(3,3),BIRIIH(3,3 )
     5 ,HC(3,3),HK(3,3)
      COMMON/HED/IC1,IC2,IC3,IC4,HEAD(19),IPAGE,INPUT(20),IEND,LINE,IC5
      COMMON/DIM/NINPUT,NSTA,NYMODE,NZMODE,NPMODE,NMODE,NM1,NDIM,NBLADE
      COMMON/DER/TH(20),EV(20),EW(20),EP(20),Y(20,3),Z(20,5),P(20,3)
C       NOTE    DO NOT, DO NOT USE CCMMON/VAR/         *******************
      LOGICAL LHUB
      INTEGER ICOL(4),IROW(4)
      REAL XHD(3),XH(3),XHDD(3),FIB(11,4)
      REAL YB(11),YDB(11),HUBI(3,4),HUBC(3,4),HUBBV(3,11),HUBBD(3,11),
     1   HUBBF(3,11),HUBB(3),SINB(4),COSB(4),PSI (4),RHS(11),FB(11),
     2   HINV(3,4),YDDB(11)
      LHUB=.FALSE.
      IF(LY(1).OR.LY(3).OR.LY(5))  LHUB=.TRUE.
      SOFT = SIN(OMF*T)
      IF(OMF.EQ.0) SCFT = 1.
      IF(.NOT.LHUB)GC TO 45
      PSI(1)= AMOD(T*OMEG,6.28319)
      DPSI=6.28319/FLOAT(NB)
      SINB(1)=SIN(PSI(1))
      COSB(1)=COS(PSI(1))
      DO 10 IB=2,NB
      PSI(IB)=PSI(IB-1)+DPSI
      IF(PSI(IB).GE.6.28319) PSI(IB)=PSI(IB)-6.28319
      SINB(IB)=SIN(PSI(IB))
   10 COSB(IB)=COS(PSI(IB))
      DO 20 I=1,3
      DO 20 J=1,3
      HUBI(I,J)=TM(I,J)
   20 HUBC(I,J)=HC(I,J)
      DO 30 IB=1,NB
      HUBI(1,1) = HUBI(1,1)-SINB(IB)**2*BIRIIH(1,1)
      HUBI(1,2) = HUBI(1,2)-SINB(IB)*COSB(IB)*BIRIIH(1,2)
```

```
      HUBI(2,1) = HUBI(2,1)-SINB(IB)*COSB(IB)*BIRIIH(2,1)
      HUBI(1,3) = HUBI(1,3)-SINB(IB)*BIRIIH(1,3)
      HUBI(3,1) = HUBI(3,1)-SINB(IB)*BIRIIH(3,1)
      HUBI(2,3)=HUBI(2,3)-COSB(IB)*BIRIIH(2,3)
      HUBI(3,2)=HUBI(3,2)-COSB(IB)*BIRIIH(3,2)
      HUBI(2,2)=HUBI(2,2)-COSB(IB)**2*BIRIIH(2,2)
      HUBI(3,3) = HUBI(3,3)-BIRIIH(3,3)
      HUBC(1,1) = HUBC(1,1)+SINB(IB)*COSB(IB)*BIRIDH(1,1)
      HUBC(1,2) = HUBC(1,2)+SINB(IB)*SINB(IB)*BIRIDH(1,2)
      HUBC(2,1)=HUBC(2,1)+COSB(IB)*COSB(IB)*BIRIDH(2,1)
      HUBC(1,3) = HUBC(1,3)+SINB(IB)*BIRIDH(1,3)
      HUBC(3,1) = HUBC(3,1)+COSB(IB)*BIRIDH(3,1)
      HUBC(2,3)=HUBC(2,3)+COSB(IB)*BIRIDH(2,3)
      HUBC(3,2)=HUBC(3,2)+SINB(IB)*BIRIDH(3,2)
      HUBC(2,2)=HUBC(2,2)+COSB(IB)*SINB(IB)*BIRIDH(2,2)
   30 HUBC(3,3) = HUBC(3,3)+BIRIDH(3,3)
      XHD(1)=YVAR(1)
      XH (1)=YVAR(2)
      XHD(2)=YVAR(3)
      XH (2)=YVAR(4)
      XHD(3)=YVAR(5)
      XH (3)=YVAR(6)
      DO 40 I=1,3
   40 RHS(I)=HF(I)*SOFT
      CALL MXV(RHS,HUBC,XHD,3,3,3,1)
      CALL MXV(RHS,HK  ,XH ,3,3,3,1)
   45 DO 200 IB=1,NB
      I=10+NM*(IB-1)*2
      DO 50 J=1,NM
      I=I+1
      YDB(J)=YVAR(I)
      I=I+1
   50 YB(J)=YVAR(I)
      IF(.NOT.LHUB) GO TO62
      DO 60 J=1,NM
      HUBBV(1,J)=SINB(IB)*BIRID(1,J)+COSB(IB)*BDAM(1,J)
      HUBBV(2,J)=COSB(IB)*BIRID(2,J)+SINB(IB)*BDAM(2,J)
      HUBBV(3,J)=BIRID(3,J)+BDAM(3,J)
      HUBBD(1,J)=SINB(IB)*BIRIO(1,J)
      HUBBD(2,J)=COSB(IB)*BIRIO(2,J)
      HUBBD(3,J)=BIRIO(3,J)
      HUBBF(1,J)=SINB(IB)*BIRI( 1,J)
      HUBBF(2,J)=COSB(IB)*BIRI( 2,J)
   60 HUBBF(3,J)=BIRI(3,J)
   62 DO 65 I=1,NM
   65 FNL(I)=0
C                  NON LINEAR TERMS     SUM MODES
      IF(NLIN.EQ.2) GO TO 160
      IF(NY.EQ.0) GO TO 160
      CALL SUMODE (VD, YDB,  Y,NSTA,NX,NY)
      CALL SUMODE (VDP,YDB,YP ,NSTA,NX,NY)
      CALL SUMODE (VPP,YB,YPP ,NSTA,NX,NY)
      CALL SUMODE (VP,YB,YP   ,NSTA,NX,NY)
      DO 70 I=1,NX
      WD (I)=0
```

```
      WP (I)=0
      WDP(I)=0
   70 WPP(I)=0
      IF(NZ.EQ.0) GO TO 85
      DO 80 I=1,NZ
      DUMP (I)= YB(NY+I)
   80 DUMPP(I)= YDB(NY+I)
      CALL SUMODE (WD ,DUMPP,Z  ,NSTA,NX,NZ)
      CALL SUMODE (WDP,DUMPP,ZP ,NSTA,NX,NZ)
      CALL SUMODE (WPP,DUMP ,ZPP,NSTA,NX,NZ)
      CALL SUMODE (WP ,DUMP ,ZP ,NSTA,NX,NZ)
   85 DO 90 I=1,NX
   90 DUMPP(I)=VDP(I)*VP(I)+WDP(I)*WP(I)
      CALL INT(DUMP,DUMPP,0,X,NX,1)
      DO 95 I=1,NX
   95 DUMPP(I)=M(I)*VD(I)
      CALLINT(VDM,DUMPP,0,X,NX,2)
      DO 100 I=1,NX
  100 DUMPP(I)=M(I)*(DUMP(I)-VD(I)*VP(I))+VPP(I)*VDM(I)
      CALL INT(DUMP,DUMPP,0,X,NX,2)
      CALL INT(DUMPP,DUMP,0,X,NX,2)
      DO 120 J=1,NY
      DO 110 I=1,NX
  110 DUMP(I)=Y(I,J)*DUMPP(I)
  120 FNL(J)=DINT(DUMPP,DUMP,X,NX)*2.*OMEG
      IF(NZ.EQ.0) GO TC 150
      IF(NLIN.EQ.1) GO TO 150
      DO 130 I=1,NX
  130 DUMPP(I)=WPP(I)*VDM(I)-M(I)*VD(I)*WP(I)
      CALL INT(DUMP,DUMPP,0,X,NX,2)
      CALL INT(DUMPP,DUMP,0,X,NX,2)
      DO 140 J=1,NZ
      DO 135 I=1,NX
  135 DUMP(I)=Z(I,J)*DUMPP(I)
  140 FNL(NY+J)=DINT(DUMPP,DUMP,X,NX)*2.*OMEG
  150 CONTINUE
  160 DO 170 I=1,NM
      FB(I)=FR(I)+FNL(I)
  170 FIB(I,IB)=FB(I)
C                      BLADE FCRCING
      IF(INPUT(13).EQ.0) GO TO 190
      IF(NBF.NE.0.AND.IB.NE.NBF) GO TO 190
      DO 180 I=1,NM
      IF(BF(I).EQ.0) GC TO 180
      IF(PER.NE.0) GO TO 175
      FB(I) = FB(I)+BF(I)*SOFT
      GO TO 180
  175 CONST=PSI(IB)/PER
      IF(CONST.GE.6.28319) GC TO 180
      FB(I)=FB(I)+BF(I)*(1.0-COS(CCNST))
  180 FIB(I,IB) = FB(I)
  190 IF(.NOT.LHUB) GO TO 200
      CALL MXV(RHS,HUBBV,YDB,3,NM,3,1)
      CALL MXV(RHS,HUBBD,YB ,3,NM,3,1)
      CALL MXV(RHS,HUBBF,FB ,3,NM,3,1)
```

```
  200 CONTINUE
      IF(.NOT.LHUB) GO TO 300
      CALL INVRS(HUBI,3,HINV,HUBC,IROW,ICOL,3,4)
      CALL MXV(XHDD,HINV,RHS,3,3,3,0)
C       NOTE THAT ALL 3 HUB MOTIONS COMPUTED,  THEY ARE IGNORED IF NOT
      IF(LY(1))
     1DERY(1) = XHDD(1)
      IF(LY(2))
     1DERY(2) = YVAR(1)
      IF(LY(3))
     1DERY(3) = XHDD(2)
      IF(LY(4))
     1DERY(4) = YVAR(3)
      IF(LY(5))
     1DERY(5) = XHDD(3)
      IF(LY(6))
     1DERY(6) = YVAR(5)
C                   BLADES
  300 DO 360 IB=1,NB
      I=10+NM*(IB-1)*2
      DO 310 J=1,NM
      I=I+1
      YDB(J)=YVAR(I)
      I=I+1
  310 YB(J)=YVAR(I)
      DO 320 I=1,NM
  320 RHS(I)=FIB(I,IB)
      CALL MXV(RHS,CODR,YDB,NM,NM,NMODE,1)
      CALL MXV(RHS,COR  ,YB ,NM,NM,NMODE,1)
      IF(.NOT.LHUB) GO TO 350
      DUMP(1)=SINB(IB)*DERY(1)
      DUMP(2)=COSB(IB)*DERY(3)
      DUMP(3)=DERY(5)
      CALL MXV(RHS,CCIH,DUMP,NM,3,NMODE,1)
      DUMP(1)=COSB(IB)*XHD(1)
      DUMP(2)=SINB(IB)*XHD(2)
      DUMP(3)=XHD(3)
      CALLMXV (RHS,CODH,DUMP,NM,3,NMODE,1)
  350 CALL MXV (YDDB,RIOC,RHS,NM,NM,NMODE,0)
      I=10+NM*(IB-1)*2
      DO 360 J=1,NM
      I=I+1
      DERY(I)=YDDB(J)
      I=I+1
  360 DERY(I)=YVAR(I-1)
      RETURN
      END
```

```
      SUBROUTINE HEADIN
      COMMON/HED/IC1,IC2,IC3,IC4,HEAD(19),IPAGE,INPUT(20),IEND,LINE,IC5
      IPAGE=IPAGE+1
      PRINT 100,IC1,IC2,IC3,IC4,IC5,HEAD,IPAGE,(I,I=1,20),INPUT
  100 FORMAT (1H1,9X,13HV22  11/12/76 /10X,15H--- --------- **,
     1               19(5H ****)/ 8X,5I2,14X,19A4,3X,4HPAGE,I5/
     2   10X,10(5H* ***),20I3/50X,10HINPUT =    ,20I3)
      RETURN
      END
```

```
      SUBROUTINE INPU (ICASE)
C                                   V-22
C
      REAL M,KM1,KM2,KA
      LOGICAL LY
      LOGICAL LCALC
      INTEGER IROW(12),ICOL(12)
C                       COMMON FOR INPUT
      COMMON/INDAT/X(20),M(20),E(20),SEA(20),KM1(20),KM2(20),KA(20),
     1 THP(20),EOP(20),GJ(20),EA(20),EB1(20),EB2(20),ECS(20),EIP(20),
     2 THO,BPC,YPP(20,3),YP(20,3),ZPP(20,5),ZP(20,5),PPP(20,3),PP(20,3),
     3  OMEG,OMF,EC(20),NY,NZ,NP,NM,OMEGS,OMFS,IDIM,NMAX,NLIN
     4,NB,HMX,HMY,HMZ,HCX,HCY,HCZ,HKX,HKY,HKZ,NX,NFLOQ
     5 ,HINIT,ERROR,IYE,CIC,IYIC,BERR ,CYCLES ,NXF,AFY,AFZ,AFP,NBF
     6  ,R,GV,GW,GP,HE(3),PER
C                            COMMON COEFFICIENT MATRICES
      COMMON/COEF/COI(11,11),DCOI(11,11),COD(11,11),DCOD(11,11),
     1   CO(11,11),DCO(11,11),F(11),DF(11),FNL(11),COIR(11,12),
     2   CODR(11,11),CCR(11,11),FR(11),RIOC(11,12),BF(11)
     3 ,BIN(3,11),BDAM(3,11),BSPR(3,11),COIH(11,3),CODH(11,3),BIRI(3,11)
     4,BIRID(3,11),BIRIO(3,11),BIRIDH(3,3 ),HF(3),TM(3,3),BIRIIH(3,3 )
     5 ,HC(3,3),HK(3,3)
C            COMMON FOR HEADING, CONTROL DATA
      COMMON/HED/IC1,IC2,IC3,IC4,HEAD(19),IPAGE,INPUT(20),IEND,LINE,IC5
C                       COMMON DIMENSION DATA
      COMMON/DIM/NINPUT,NSTA,NYMODE,NZMODE,NPMODE,NMODE,NM1,NDIM,NBLADE
C                     COMMON BASIC DERIVED DATA
      COMMON/DER/TH(20),EV(20),EW(20),EP(20),Y(20,3),Z(20,5),P(20,3)
C             COMMON VARIABLES AND SOLUTION CONTROLS
      COMMON/VAR/YVAR(98),DERY(98),PRMT(6),LY(98)
      REAL DUM(8),DUMPP(20),DUMP(20),WORK(11,12),DUMPPP(20)
      REAL DELE(20),EONE(20),DELK(20),KM(20)
      REAL MI(20,10),MII(20,9),YI(20 ,3,10),YII(20,3,9),ZI(20,5,9),
     1  ZII(20,5,8),PI(20,3,8),PII(20,3,7),SI(20,5,5)
      REAL DYYI(3,3,10),DYYII(3,3,9),DYZII(3,5,8),DYPII(3,3,7),
     1 DYSI(3,5,4),DYMI(3,10),DYMII(3,9),DZYI(5,3,10),DZYII(5,3,9),
     2 DZZII(5,5,8),DZPI(5,3,8),DZPII(5,3,7),DZMI(5,10),DZMII(5,9),
     3 DPYI(3,3,10),DPYII(3,3,9),DPZI(3,5,9),DPZII(3,5,8),DPPI(3,3,8),
     4DPPII(3,3,7),DPSI(3,3,1),DPMI(3,10),DPMII(3,9),DYF(3,5,6),
     5 DZF(5,5,6),DPF(3,5,3)
     6  ,ALII(20),DYALII(3),DZALII(5),DPALII(3)
      REAL YZPI(20),DYD(3,3),DZD(5,5),DPD(3,3)
      REAL D(2243)
      EQUIVALENCE (D(1),DYYI(1)),(D(91),DYYII(1)),(D(172),DYZII(1)),
     1 (D(292 ),DYPII(1)),(D(355 ),DYSI (1)),(D(415 ),DYMI(1)),
     2 (D(445 ),DYMII(1)),(D(472 ),DZYI (1)),(D(622 ),DZYII(1)),
     3 (D(757 ),DZZII(1)),(D(957 ),DZPI (1)),(D(1077),DZPII(1)),
     4 (D(1182),DZMI (1)),(D(1232),DZMII(1)),(D(1277),DPYI (1)),
     5 (D(1367),DPYII(1)),(D(1448),DPZI (1)),(D(1583),DPZII(1)),
     6 (D(1703),DPPI (1)),(D(1775),DPPII(1)),(D(1838),DPSI (1)),
     7 (D(1847),DPMI (1)),(D(1877),DPMII(1)),(D(1904),DYF  (1)),
     8 (D(1994),DZF  (1)),(D(2144),DPF  (1)),(D(2189),DYALII(1)),
```

```
    9 (D(2192),DZALII(1)),(C(2197),DPALII(1))
      EQUIVALENCE (D(2201),DYD(1)),(D(2210),DZD(1)),(D(2235),DPD(1))
C                     INITIALIZATICN
C                   HEADING
   50 IF(IEND.EQ.2) GO TO 52
      READ 9000,IC1,IC2,IC3,IC4,IC5,HEAD
 9000 FORMAT (5I1,18A4,A3)
      IF(IC1.NE.0) CALL EXIT
   52 ICASE = ICASE+1
      IPAGE = 0
C     INPUT(I) = 0, NEVER USED     = 1, USED     =2, MODIFIED OR NEW
      DO 100 I=1,NINPUT
      IF (INPUT(I).EQ.0) GO TO 100
      INPUT(I) = 1
  100 CONTINUE
C              CLEAR TO CLEAN UP OUTPUT OF INTEGRALS
      DO 90 I=1,2243
   90 D(I)=0.
      IF(INPUT(6).EQ.0) OLDOM = 1.
      IF(INPUT(6).NE.0) OLDOM = OMEG
      OLDOMS = OLDOM*OLDOM
      IF(IC5.EQ.0) GO TO 201
      I=7
      WRITE (9) I
      GO TO 201
C
C
C
C                   GENERAL INPUT
C
C
C
  200 IF (IEND.NE.0) GO TO 500
  201 READ 9010,IO,DUM,IEND
 9010 FORMAT (I2,F8.0,6F10.0,F9.0,I1)
      IF(IO.NE.21) GO TO202
      CALL HEADIN
      PRINT 9011
 9011 FORMAT(//20X,28HFOLLOWING IO'S ARE CANCELLED   /)
      DO 203 J=1,8
      I=DUM(J)
      IF(I.EQ.0) GO TO 203
      PRINT 9012,I
 9012 FORMAT(30X,I10)
      IF(I.LT.0.OR.I.GT.NINPUT) CALL ERR (203,0)
      INPUT(I)=0
  203 CONTINUE
C   NOTE  INPUT(1) SET TO 2 TO INSURE THAT ALL COEFS ARE RE CALCULATED
      INPUT(1)=2
      IC2=1
      GO TO 200
  202 IF(IO.GT.NINPUT.OR.IO.LT.1) CALL ERR(200,0)
      IF(INPUT(IO).EQ.2) CALL ERR (202,IO)
      INPUT (IO) = 2
      GO TO (210,220,230,230,230,270,320,330,340,10,11,12,
```

```
     1  350,14,15,16,280,300,19,20),IO
   10 CALL ERR(10,0)
   11 CALL ERR(11,0)
   12 CALL ERR (12,0)
   14 CALL ERR(14,0)
   15 CALL ERR(15,0)
   16 CALL ERR(16,0)
   19 CALL ERR(19,0)
   20 CALL ERR(20,0)
C           IO=1       BLADE PROPERTIES
  210 I = 1
  215 X(I) = DUM(1)
      M(I) = DUM(2)
      E(I) = DUM(3)
      SEA(I) = DUM(4)
      KM1(I) = DUM(5)
      KM2(I) = DUM(6)
      KA(I) = DUM(7)
      THP(I) = DUM(8)
      READ 9010,IO,DUM
      EOP(I) = DUM(1)
      EIP(I) ≐ DUM(2)
      GJ(I) = DUM(3)
      EA(I) = DUM(4)
      EB1(I) = DUM(5)
      EB2(I) = DUM(6)
      EC(I) = DUM(7)
      ECS(I) = DUM(8)
      R=X(I)
      IF(IEND.NE.0) GO TO 500
      READ 9010,IO,DUM,IEND
      IF(IO.NE.0) GC TC 202
      IF(DUM(1).LT.X(I)) CALL ERR(215,0)
      I = I+1
      NX = I
      IF(NX.GT.NSTA) CALL ERR (216,0)
      GO TO 215
C                IO=2             BLADE CATA
  220 NB =DUM(1)
      THO=DUM(2)
      BPC=DUM(3)
      GV =DUM(4)
      GW =DUM(5)
      GP =DUM(6)
      GO TO 200
C             IO = 3,4,5 MODES
  230 IF(INPUT(1).EQ.0) CALL ERR (230,0)
      J = 0
  235 J = J+1
      DO 240 I=1,8
  240 DUMPP(I) = DUM(I)
      IF (NX.LE.8) GC TO 250
      READ 9020, (DUMPP(I),I=9,NX)
 9020 FORMAT (7F10.0,F9.0)
  250 READ 9020, SC
```

```
C          INTEGRATE  AND  NORMALIZE  MODES
      CALL INT (DUMP,DUMPP,SC,X,NX,1)
      CALL INT(DUMPPP,DUMP,0,X,NX,1)
      CONST=DUMPPP(NX)
      IF(CONST.EQ.0) CONST=1.0
      DO 260 I=1,NX
      IF (IO-4) 252,254,256
  252 YPP(I,J) = DUMPP(I)/CONST
      YP(I,J) = DUMP(I)/CONST
      Y(I,J)=DUMPPP(I)/CONST
      GO TO 260
  254 ZPP(I,J) = DUMPP(I)/CONST
      ZP(I,J) = DUMP(I)/CONST
      Z(I,J)=DUMPPP(I)/CONST
      GO TO 260
  256 PPP(I,J) = DUMPP(I)/CONST
      PP(I,J) = DUMP(I)/CONST
      P(I,J)=DUMPPP(I)/CONST
  260 CONTINUE
      IF (IEND.NE.0) GO TO 261
      READ 9010,II,DUM,IENDT
      IF (II.EQ.0) GO TO 235
  261 IF (IO-4) 262,264,266
  262 NY = J
      IF (NY.GT.NYMODE) CALL ERR (262,0)
      GO TO 267
  264 NZ = J
      IF (NZ.GT.NZMODE) CALL ERR (264,0)
      GO TO 267
  266 NP = J
      IF (NP.GT.NPMODE) CALL ERR (266,0)
  267 IF (IEND.NE.0) GO TO 500
      IEND= IENDT
      IO = II
      GO TO 202
C                    IO = 6       FREQUENCIES
  270 OMEG = DUM(1)
      OMF = DUM(2)
      GO TO 200
C          IO = 17       NON LINEAR CONTROLS
  280 NLIN = DUM(1)
      NFLOQ=DUM(2)
      GO TO 200
C            IO = 18      SOLUTION CONTROLS
  300 CYCLES = DUM(1)
      HINIT  = DUM(2)
      ERROR  = DUM(3)
      IYE    = DUM(4)
      CIC    = DUM(5)
      IYIC   = DUM(6)
      BERR   = DUM(7)
      GO TO 200
C            IO = 7    HUBX
  320 HMX = DUM(1)
      HCX = DUM(2)
```

```
      HKX = DUM(3)
      HE(1)=DUM(4)
      GO TO 200
C        IO = 8    HUB Y
  330 HMY = DUM(1)
      HCY = DUM(2)
      HKY = DUM(3)
      HE(2)=DUM(4)
      GO TO 200
C          IO = 9     HUB Z
  340 HMZ = DUM(1)
      HCZ = DUM(2)
      HKZ = DUM(3)
      HE(3)=DUM(4)
      GO TO 200
C                             IO = 13    BLADE FORCE
  350 NXF = DUM(1)
      AFY = DUM(2)
      AFZ = DUM(3)
      AFP = DUM(4)
      NBF=DUM(5)
      PER=DUM(6)
      GO TO 200
C
C
C
C               PROCESS INPUT DATA
C       CHECKS, DEFAULTS    SEE ALSO 1100-1200
C
C
  500 IF(INPUT(1).EQ.0) CALL ERR(500,0)
      IF(INPUT(2).NE.0) GO TC 501
      NB=1
      THO=0
      BPC=0
      GV=0
      GW=0
      GP=0
  501 IF(INPUT(3).EQ.0) NY=0
      IF(INPUT(4).EQ.0) NZ=0
      IF(INPUT(5).EQ.0) NP=0
      NM=NY+NZ+NP
      IF(NM.EQ.0) CALL ERR(501,0)
      NMAX = NZ
      IF(NP.GT.NMAX) NMAX = NP
      IF(NY.GT.NMAX) NMAX = NY
      IF(INPUT(6).EQ.0) CALL ERR(502,0)
      IF(NB.EQ.1.AND.(INPUT(7).NE.0.OR.INPUT(8).NE.0.OR.INPUT(9).NE.0))
     1      NB=2
      IF(NB.GT.NBLADE) CALL ERR(506,NB)
      IF(NB.GT.NBLADE) NB=NBLADE
      IF(NB.LT.1) CALL ERR (507,1)
      IF(NB.LT.1) NB = 1
      IF(NB.EQ.1.AND.(INPUT(7).NE.0.OR.INPUT(8).NE.0.OR.INPUT(9).NE.0))
     1      NB=2
```

```
      IF(INPUT(7).NE.0) GO TC 502
      HMX=0
      HCX=0
      HKX=0
      HE(1)=0
  502 IF(INPUT(8).NE.0) GO TC 503
      HMY = 0
      HCY = 0
      HKY = 0
      HE(2)=0
  503 IF(INPUT(9).NE.0) GO TC 504
      HMZ=0
      HCZ=0
      HKZ=0
      HE(3)=0
  504 OMRAT = OMEG/OLDCM
      OMRATS=OMRAT*OMRAT
      IF(INPUT(13).NE.0.AND.(NXF.GT.NX.OR.NXF.LE.0)) CALL ERR(510,0)
      IF(INPUT(13).NE.0.AND.(AFY.EQ.0.AND.AFZ.EQ.0.AND.AFP.EQ.0))
     1   CALL ERR (511,0)
      IF(INPUT(13).NE.0.AND.NBF.GT.NB)  CALL ERR(512,NBF)
      IF(INPUT(13).NE.0.AND.NBF.GT.NB)  NBF = 0
      IF(INPUT(13).NE.0.AND.NBF.LT.0 )  CALL ERR(512,NBF)
      IF(INPUT(13).NE.0.AND.NBF.LT.0 )  NBF = 0
      IF(INPUT(17).EQ.0) NLIN=0
      IF(INPUT(17).EQ.0) NFLCQ=0
      IF (INPUT(18).EQ.0) CALL ERR(509,0)
C              ADD BLADE LOADS TO HUB
      DO 508 I=1,3
  508 HF(I)=HE(I)
      IF(INPUT(13).EQ.0) GO TC 509
      IF(AFZ.EQ.0) GO TO 506
      IF(INPUT(9).EQ.0) GO TC 506
      CONST=AFZ
      IF(NBF.EQ.0) CCNST=NB*CONST
      HF(3)=HF(3)+CONST
  506 IF(AFY.EQ.0)GO TC 509
      IF((INPUT(7).EQ.0.AND.INPUT(8).EQ.0).OR.NBF.EQ.0) GO TO 509
      CALL ERR(510,NBF)
      NBF=0
C
C              COMPUTE COEFFICIENTS, ETC.
C
  509 CALL INT(TH,THP,THO,X,NX,1)
      DO 510 I=1,NX
      DUMMY1 = SEA(I)**2*EA(I)
      DUMMY2 = EIP(I)-EOP(I)
      EV(I) = EIP(I)-DUMMY2*TH(I)**2-DUMMY1
      DELE(I) = DUMMY2-DUMMY1
      EONE(I) = SEA(I)*EA(I)*KA(I)**2-EB2(I)
      EW(I) = EOP(I)+DUMMY2*TH(I)**2-DUMMY1*TH(I)
      EP(I) = GJ(I)-(KA(I)**4*EA(I)-EB1(I))*THP(I)**2
      DELK(I) = KM2(I)**2-KM1(I)**2
  510 KM(I) = KM2(I)**2+KM1(I)**2
C              FORM MASS INTEGRALS
```

```
C RECOMPUTE ALL COEFS UNLESS ONLY IC = 6 OR .GE. 17 ARE CHANGED
      LCALC=.TRUE.
  600 DO 601 I=1,16
      IF(I.EQ.6) GO TO 601
      IF(INPUT(I).EQ.2) GO TO 602
  601 CONTINUE
      LCALC=.FALSE.
      IF(INPUT(6).EQ.2) GO TO 1075
      GO TO 1100
C                      FORM  INTEGRANDS
  602 DO 610 I = 1,NX
      MI(I,1) = M(I)
      MI(I,2) = M(I)*X(I)
      MI(I,3) = M(I)*E(I)
      MI(I,4) = MI(I,3)*X(I)
      MI(I,5) = MI(I,3)*TH(I)
      MI(I,6) = MI(I,5)*X(I)
      MI(I,7) = M(I)*KM2(I)**2
      MI(I,8) = MI(I,7)*TH(I)
  610 MI(I,9) = M(I)*DELK(I)*TH(I)
      DO 630 J = 1,9
      DO 620 I = 1,NX
  620 DUMPP(I) = MI(I,J)
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      CALL INT (DUMPP,DUMP,0,X,NX,2)
      DO 630 I = 1,NX
      MI(I,J) = DUMP(I)
  630 MII(I,J) = DUMPP(I)
C                                  MI(I,10)
      DO 635 I = 1,NX
  635 DUMPP(I) = MI(I,2)*KA(I)**2*THP(I)
      CALL INT(DUMP,DUMPP,0,X,NX,2)
      DO 640 I=1,NX
  640 MI(I,10) = DUMP(I)
C                         FORM Y INTEGRALS
  650 IF(INPUT(3).EQ.0) GO TO 700
C                           FORM INTEGRANDS
      DO 660 I = 1,NX
      DO 660 IM = 1,NY
      YI(I,IM,1) =  M(I)*Y(I,IM)
      YI(I,IM,2) = YI(I,IM,1)*E(I)
      YI(I,IM,3) = YI(I,IM,2)*TH(I)
      YI(I,IM,4) = M(I)*X(I)*YP(I,IM)
      YI(I,IM,5) = M(I)*E(I)*YP(I,IM)
      YI(I,IM,6) = YI(I,IM,5)*X(I)*TH(I)
      YI(I,IM,7) = MI(I,2)*YPP(I,IM)
      YI(I,IM,8) = YI(I,IM,7)*SEA(I)*TH(I)
      YI(I,IM,9) = YPP(I,IM)*EONE(I)*THP(I)
  660 YI(I,IM,10) = YPP(I,IM)*SEA(I)
      DO 670 J = 1,9
      DO 670 IM = 1,NY
      DO 665 I = 1,NX
  665 DUMPP(I) = YI(I,IM,J)
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      CALL INT (DUMPP,DUMP,0,X,NX,2)
```

```
      DO 670 I = 1,NX
      YI(I,IM,J) = DUMP(I)
  670 YII(I,IM,J) = DUMPP(I)
      DO 680 IM = 1,NY
      DO 675 I = 1,NX
  675 DUMPP(I) = YI(I,IM,10)
      CALL INT (DUMP,DUMPP,0,X,NX,1)
      DO 680 I = 1,NX
  680 YI(I,IM,10) = DUMP(I)
      DO 682 I = 1,NX
      DO 682 IM = 1,NY
      IF(EA(I).EQ.0) SI(I,IM,1) = 0
      IF(EA(I).NE.0) SI(I,IM,1) = YI(I,IM,1)/EA(I)
      SI(I,IM,2) = M(I)*YI(I,IM,10)
  682 SI(I,IM,5) = KA(I)**2*THP(I)*YI(I,IM,1)
      DO 685 IM = 1,NY
      DO 683 I = 1,NX
  683 DUMPP(I) = SI(I,IM,1)
      CALL INT(DUMP,DUMPP,0,X,NX,1)
      DO 684 I = 1,NX
  684 DUMP(I) = DUMP(I)*M(I)
      CALL INT(DUMPP,DUMP,0,X,NX,2)
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      DO 685 I = 1,NX
  685 SI(I,IM,1) = DUMP(I)
      DO 690 IM = 1,NY
      DO 686 I = 1,NX
  686 DUMPP(I) = SI(I,IM,2)
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      CALL INT (DUMPP,DUMP,0,X,NX,2)
      DO 690 I = 1,NX
  690 SI(I,IM,2) = DUMPP(I)
      DO 695 IM = 1,NY
      DO 692 I = 1,NX
  692 DUMPP(I) = SI(I,IM,5)
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      DO 695 I = 1,NX
  695 SI(I,IM,5) = DUMP(I)
C                          FORM Z INTEGRALS
  700 IF(INPUT(4).EQ.0) GO TO750
      DO 710 I = 1,NX
      DO 710 JM = 1,NZ
      ZI(I,JM,1) = M(I)*Z(I,JM)
      ZI(I,JM,2) = ZI(I,JM,1)*E(I)
      ZI(I,JM,3) = M(I)*X(I)*ZP(I,JM)
      ZI(I,JM,4) = ZI(I,JM,3)*E(I)
      ZI(I,JM,5) = M(I)*ZP(I,JM)*E(I)*TH(I)
      ZI(I,JM,6) = MI(I,2)*ZPP(I,JM)
      ZI(I,JM,7) = ZI(I,JM,6)*SEA(I)
      ZI(I,JM,8) = ZPP(I,JM)*EONE(I)*TH(I)*THP(I)
  710 ZI(I,JM,9) = ZPP(I,JM)*SEA(I)*TH(I)
      DO 720 J = 1,8
      DO 720 JM = 1,NZ
      DO 715 I = 1,NX
  715 DUMPP(I) = ZI(I,JM,J)
```

```
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      CALL INT (DUMPP,DUMP,0,X,NX,2)
      DO 720 I = 1,NX
      ZI(I,JM,J) = DUMP(I)
  720 ZII(I,JM,J) = DUMPP(I)
      DO 730 JM = 1,NZ
      DO 725 I = 1,NX
  725 DUMPP(I) = ZI(I,JM,9)
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      DO 730 I = 1,NX
  730 ZI(I,JM,9) = DUMP(I)
      DO 740 JM = 1,NZ
      DO 735 I = 1,NX
  735 DUMPP(I) = M(I)*ZI(I,JM,9)
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      CALL INT (DUMPP,DUMP,0,X,NX,2)
      DO 740 I = 1,NX
  740 SI(I,JM,3) = DUMPP(I)
C                                  FORM P INTEGRALS
  750 IF(INPUT(5).EQ.0) GO TO 800
      DO 760 I=1,NX
      DO 760 IM = 1,NP
      PI(I,IM,1) = M(I)*E(I)*P(I,IM)
      PI(I,IM,2) = PI(I,IM,1)*X(I)
      PI(I,IM,3) = PI(I,IM,1)*TH(I)
      PI(I,IM,4) = M(I)*KM(I)*P(I,IM)
      PI(I,IM,5) = M(I)*DELK(I)*P(I,IM)
      PI(I,IM,6) = EP(I)*PP(I,IM)
      PI(I,IM,7) = KA(I)**2*MI(I,2)*PP(I,IM)
  760 PI(I,IM,8) = KA(I)**2*THP(I)*PP(I,IM)
      DO 770 J = 1,7
      DO 770 IM = 1,NP
      DO 765 I = 1,NX
  765 DUMPP(I) = PI(I,IM,J)
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      IF( J.GT.5 ) GO TO 766
      CALL INT ( DUMPP,DUMP,0,X,NX,2)
  766 DO 770 I = 1,NX
      PI(I,IM,J) = DUMP(I)
      IF( J.GT.5 ) GO TO 770
      PII(I,IM,J) = DUMPP(I)
  770 CONTINUE
      DO 780 IM = 1,NP
      DO 775 I = 1,NX
  775 DUMPP(I) = PI(I,IM,8)
      CALL INT (DUMP,DUMPP,0,X,NX,1)
      DO 780 I = 1,NX
  780 PI(I,IM,8) = DUMP(I)
      DO 790 IM = 1,NP
      DO 785 I = 1,NX
  785 DUMPP(I) = M(I)*PI(I,IM,8)
      CALL INT (DUMP,DUMPP,0,X,NX,2)
      CALL INT (DUMPP,DUMP,0,X,NX,2)
      DO 790 I = 1,NX
  790 SI(I,IM,4) = DUMPP(I)
```

```
C                         DEFINITE INTEGRALS
C                 BLADE FORCE INTEGRALS
  800 IF(INPUT(13).EQ.0) GO TC 810
      DO 802 I=1,NX
  802 ALII(I)=AMAX1(0.0,X(NXF)-X(I))
  810 IF(NY.EQ.0) GO TC 851
      DO 850 I = 1,NY
      IF(INPUT(13).EQ.0.OR.AFY.EQ.0) GO TO 824
      DO 815 K=1,NX
  815 DUMPP(K)=AFY*Y(K,I)*ALII(K)
      DYALII(I)=DINT(DUMP,DUMPP,X,NX)
  824 DO 825 J = 1,NY
      DYSI(I,J,1) = DINT2(Y,SI,I,J,1,5,X,NSTA,NX,DUMP,DUMPP)
      DYSI(I,J,2) = DINT2(Y,SI,I,J,2,5,X,NSTA,NX,DUMP,DUMPP)
      DO 825 K = 1,9
      DYYI(I,J,K) = DINT2(Y,YI,I,J,K,NYMODE,X,NSTA,NX,DUMP,DUMPP)
  825 DYYII(I,J,K) = DINT2(Y,YII,I,J,K,NYMODE,X,NSTA,NX,DUMP,DUMPP)
      IF (NZ.EQ.0) GC TO 832
      DO 830 J = 1,NZ
      DYSI(I,J,3) = DINT2(Y,SI,I,J,3,5,X,NSTA,NX,DUMP,DUMPP)
      DO 830 K = 1,8
  830 DYZII(I,J,K) = DINT2(Y,ZII,I,J,K,NZMODE,X,NSTA,NX,DUMP,DUMPP)
  832 IF (NP.EQ.0) GO TO 836
      DO 835 J = 1,NP
      DYSI(I,J,4) = DINT2(Y,SI,I,J,4,5,X,NSTA,NX,DUMP,DUMPP)
  835 DYPII(I,J,3) = DINT2(Y,PII,I,J,3,NPMODE,X,NSTA,NX,DUMP,DUMPP)
  836 DO 845 K = 1,9
      DYMI(I,K) = DINT1(Y,MI,I,K,X,NSTA,NX,DUMP,DUMPP)
  845 DYMII(I,K) = DINT1(Y,MII,I,K,X,NSTA,NX,DUMP,DUMPP)
  850 DYMI(I,10) = DINT1(Y,MI,I,10,X,NSTA,NX,DUMP,DUMPP)
  851 IF (NZ.EQ.0) GC TO 881
      DO 880 I = 1,NZ
      IF(INPUT(13).EQ.0.OR.AFZ.EQ.0) GO TO 854
      DO 852 K=1,NX
  852 DUMPP(K)=AFZ*Z(K,I)*ALII(K)
      DZALII(I)=DINT(DUMP,DUMPP,X,NX)
  854 IF(NY.EQ.0) GO TC 856
      DO 855 J = 1,NY
      DO 855 K = 1,9
      DZYI(I,J,K) = DINT2(Z,YI,I,J,K,NYMODE,X,NSTA,NX,DUMP,DUMPP)
  855 DZYII(I,J,K) = DINT2(Z,YII,I,J,K,NYMODE,X,NSTA,NX,DUMP,DUMPP)
  856 DO 860 J = 1,NZ
      DO 860 K = 1,8
  860 DZZII(I,J,K) = DINT2(Z,ZII,I,J,K,NZMODE,X,NSTA,NX,DUMP,DUMPP)
      IF (NP.EQ.0) GO TO 866
      DO 865 J = 1,NP
      DZPI(I,J,2 )= DINT2(Z,PI,I,J,2,NPMODE,X,NSTA,NX,DUMP,DUMPP )
  865 DZPII(I,J,1)= DINT2(Z,PII,I,J,1,NPMODE,X,NSTA,NX,DUMP,DUMPP )
  866 DO 870 K = 1,9
      DZMI(I,K) = DINT1(Z,MI,I,K,X,NSTA,NX,DUMP,DUMPP)
  870 DZMII(I,K) = DINT1(Z,MII,I,K,X,NSTA,NX,DUMP,DUMPP)
  880 DZMI(I,10) = DINT1(Z,MI,I,10,X,NSTA,NX,DUMP,DUMPP)
  881 IF (NP.EQ.0) GC TO 901
      DO 900 I = 1,NP
      IF(INPUT(13).EC.0.OR.AFP.EQ.0) GO TO 884
```

```
      DO 882 K=1,NX
  882 DUMPP(K)=AFP*P(K,I)*ALII(K)
      DPALII(I)=DINT(DUMP,DUMPP,X,NX)
  884 IF(NY.EQ.0) GO TC 886
      DO 885 J = 1,NY
      DPSI(I,J,1) = DINT2(P,SI,I,J,5,5,X,NSTA,NX,DUMP,DUMPP)
      DO 885 K = 1,9
      DPYI(I,J,K) = DINT2(P,YI,I,J,K,NYMCDE,X,NSTA,NX,DUMP,DUMPP)
  885 DPYII(I,J,K) = DINT2(P,YII,I,J,K,NYMODE,X,NSTA,NX,DUMP,DUMPP)
  886 IF (NZ.EQ.0) GO TO 891
      DO 892 J = 1,NZ
      DO 890 K = 1,8
      DPZI(I,J,K) = DINT2(P,ZI ,I,J,K,NZMODE,X,NSTA,NX,DUMP,DUMPP)
  890 DPZII(I,J,K) = DINT2(P,ZII,I,J,K,NZMODE,X,NSTA,NX,DUMP,DUMPP)
  892 DPZI(I,J,9) = DINT2(P,ZI ,I,J,9,NZMODE,X,NSTA,NX,DUMP,DUMPP)
  891 DO 895 J = 1,NP
      DO 895 K = 1,7
      DPPI(I,J,K) = DINT2(P,PI,I,J,K,NPMODE,X,NSTA,NX,DUMP,DUMPP)
      IF ( K.GT.5 ) GO TO 895
      DPPII(I,J,K) = DINT2(P,PII,I,J,K,NPMODE,X,NSTA,NX,DUMP,DUMPP)
  895 CONTINUE
  896 DO 897 K = 1,9
      DPMI(I,K) = DINT1(P,MI,I,K,X,NSTA,NX,DUMP,DUMPP)
  897 DPMII(I,K) = DINT1(P,MII,I,K,X,NSTA,NX,DUMP,DUMPP)
  900 DPMI(I,10) = DINT1(P,MI,I,10,X,NSTA,NX,DUMP,DUMPP)
  901 IF (NY.EQ.0) GO TO 931
      DO 930 J = 1,NY
      DO 910 K = 1,NY
      DO 902 I = 1,NX
  902 DUMPP(I) = Y(I,J)*(R     -X(I))*M(NX)*E(NX)*Y(NX,K)
      DYF(J,K,1) = DINT(DUMP,DUMPP,X,NX)
      DO 904 I = 1,NX
  904 DUMPP(I) = Y(I,J)*SEA(I)*YI(I,K,1)
      DYF(J,K,2) = DINT (DUMP,DUMPP,X,NX)
      DO 906 I = 1,NX
  906 DUMPP(I) = Y(I,J)*EV(I)*YPP(I,K)
  910 DYF(J,K,3) = DINT (DUMP,DUMPP,X,NX)
      IF (NZ.EQ.0) GO TO 916
      DO 915 K = 1,NZ
      DO 912 I = 1,NX
  912 DUMPP(I) = Y(I,J)*DELE(I)*TH(I)*ZPP(I,K)
  915 DYF(J,K,4) = DINT (DUMP,DUMPP,X,NX)
  916 IF (NP.EQ.0) GO TO 925
      DO 920 K = 1,NP
      DO 917 I = 1,NX
  917 DUMPP(I) = Y(I,J)*(-ECS(I)*TH(I)*PPP(I,K)+EONE(I)*THP(I)*PP(I,K))
  920 DYF(J,K,5) = DINT (DUMP,DUMPP,X,NX)
  925 DO 927 I=1,NX
  927 DUMPP(I)=Y(I,J)*(SEA(I)*MI(I,2)+R*(R-X(I))*M(NX)*E(NX))
  930 DYF(J,1,6) = DINT (DUMP,DUMPP,X,NX)
  931 IF (NZ.EQ.0) GC TO 961
      DO 960 J = 1,NZ
      IF (NY.EQ.0) GC TO 936
      DO 935 K = 1,NY
      DO 932 I = 1,NX
```

```
932 DUMPP(I)=Z(I,J)*((R-X(I))*M(NX)*E(NX)*TH(NX)*Y(NX,K)
  1    +SEA(I)*TH(I)*YI(I,K,1))
    DZF(J,K,1) = DINT (DUMP,DUMPP,X,NX)
    DO 934 I = 1,NX
934 DUMPP(I) = Z(I,J)*DELE(I)*TH(I)*YPP(I,K)
935 DZF(J,K,2) = DINT (DUMP,DUMPP,X,NX)
936 DO 938 K=1,NZ
    DO 937 I = 1,NX
937 DUMPP(I) = Z(I,J)*EW(I)*ZPP(I,K)
938 DZF(J,K,3)=DINT(DUMP,DUMPP,X,NX)
    IF (NP.EQ.0) GO TO 946
    DO 945 K = 1,NP
    DO 940 I = 1,NX
940 DUMPP(I) = Z(I,J)*(ECS(I)*PPP(I,K)+EONE(I)*TH(I)*THP(I)*PP(I,K))
    DZF(J,K,4) = DINT (DUMP,DUMPP,X,NX)
    DO 942 I=1,NX
942 DUMPP(I) =-Z(I,J)*
  1 (SEA(I)*MI(I,2)*P(I,K)+X(NX)*(X(NX)-X(I))*M(NX)*E(NX)*P(NX,K))
945 DZF(J,K,6) = DINT (DUMP,DUMPP,X,NX)
946 DO 950 I = 1,NX
950 DUMPP(I) = Z(I,J)*( SEA(I)*MI(I,2)*TH(I)+X(NX)*M(NX)*E(NX)*TH(NX)
  1          *(R -X(I)))
960 DZF(J,1,5) = DINT (DUMP,DUMPP,X,NX)
961 IF (NP.EQ.0) GO TO 991
    DO 990 J = 1,NP
    IF (NY.EQ.0) GO TO 965
    DO 963 K = 1,NY
    DO 962 I = 1,NX
962 DUMPP(I) = P(I,J)*ECS(I)*TH(I)*YPP(I,K)
963 DPF(J,K,1) = DINT (DUMP,DUMPP,X,NX)
965 IF( NZ.EQ.0 ) GO TO 971
    DO 970 K=1,NZ
    DO 964 I = 1,NX
964 DUMPP(I) = P(I,J)*ECS(I)*ZPP(I,K)
970 DPF(J,K,2) = DINT (DUMP,DUMPP,X,NX)
971 IF (NZ.EQ.0) GO TO 990
    DO 980 K = 1,NP
    DO 975 I = 1,NX
975 DUMPP(I) = P(I,J)*ECS(I)*PPP(I,K)
980 DPF(J,K,3) = DINT (DUMP,DUMPP,X,NX)
990 CONTINUE
C          DAMPING DEFINITE INTEGRALS
991 IF(NY.EQ.0.OR.GV.EQ.0) GO TO 995
    DO 994 J=1,NY
    DO 992 K=1,NX
992 YZPI(K)= Y(K,J)
     CALL INT(DUMPP,YZPI,0,X,NX,2)
     CALL INT(YZPI,DUMPP,0,X,NX,2)
    DO 994 I=1,NY
    DO 993 K=1,NX
993 DUMPP(K)=YZPI(K)*Y(K,I)
994 DYD(I,J)=DINT(DUMP,DUMPP,X,NX)*GV
995 IF(NZ.EQ.0.OR.GW.EQ.0) GO TO 999
    DO 998 J=1,NZ
    DO 996 K=1,NX
```

```
  996 YZPI(K)=Z(K,J)
       CALL INT(DUMPP,YZPI,0,X,NX,2)
       CALL INT(YZPI,DUMPP,0,X,NX,2)
      DO 998 I=1,NZ
      DO 997 K=1,NX
  997 DUMPP(K)=YZPI(K)*Z(K,I)
  998 DZD(I,J)=DINT(DUMP,DUMPP,X,NX)*GW
  999 IF(NP.EQ.0.OR.GP.EQ.0) GO TO 1010
      DO 1002 J=1,NP
      DO 1000 K=1,NX
 1000 YZPI(K)=P(K,J)
       CALL INT(DUMPP,YZPI,0,X,NX,2)
       CALL INT(YZPI,DUMPP,0,X,NX,2)
      DO 1002 I=1,NP
      DO 1001 K=1,NX
 1001 DUMPP(K)=YZPI(K)*P(K,I)
 1002 DPD(I,J)=DINT(DUMP,DUMPP,X,NX)*GP
C                                 FORM BLADE COEFFICIENT MATRICES
 1010 II=0
      IF(NY.EQ.0) GO TO 1031
      DO 1030 I = 1,NY
      JJ = 0
      II = II+1
      DO 1015 J = 1,NY
      JJ = JJ+1
      COI (II,JJ) = DYYII(I,J,1)
      DCOI(II,JJ) = 4*DYSI(I,J,1)
      COD(II,JJ)=-DYD( I,J )
      DCOD(II,JJ) = -2*(DYSI(I,J,2)-DYYII(I,J,5)-DYF(I,J,2)+DYYI(I,J,2)
     1   -DYF(I,J,1))
      CO(II,JJ) = -DYF(I,J,3)
 1015 DCO(II,JJ) =    DYYII(I,J,7)-DYYII(I,J,4)+DYYII(I,J,1)
 1016 IF(NZ.EQ.0) GO TO 1021
      DO 1020 J = 1,NZ
      JJ = JJ+1
      COI(II,JJ) = 0
      DCOI(II,JJ) = 0
      COD(II,JJ) = 0
      DCOD(II,JJ) = -2*(DYSI(I,J,3)-DYZII(I,J,5)-BPC*DYZII(I,J,1))
      CO(II,JJ) = -DYF(I,J,4)
 1020 DCO(II,JJ) = 0
 1021 IF(NP.EQ.0) GO TO 1026
      DO 1025 J = 1,NP
      JJ = JJ+1
      COI(II,JJ) = -DYPII(I,J,3)
      DCOI(II,JJ) = 0
      COD(II,JJ) = 0
      DCOD(II,JJ) = 2*DYSI(I,J,4)
      CO(II,JJ) = -DYF(I,J,5)
 1025 DCO(II,JJ) = 0
 1026 DF(II) =    DYMII(I,3)-DYMI(I,4)+DYF(I,1,6)
      BF(II)=0
      IF(INPUT(13).NE.0.AND.AFY.NE.0) BF(II)=DYALII(I)
 1030 CONTINUE
 1031 IF(NZ.EQ.0) GO TO 1051
```

```
      DO 1050 I = 1,NZ
      JJ = 0
      II = II+1
      IF(NY.EQ.0) GO TC 1036
      DO 1035 J = 1,NY
      JJ = JJ+1
      COI(II,JJ) = 0
      DCOI(II,JJ) = 0
      COD(II,JJ) = 0
      DCOD(II,JJ) = -2*(DZYI(I,J,3)-DZF(I,J,1)+BPC*DZYII(I,J,1))
      CO (II,JJ) = -DZF(I,J,2)
 1035 DCO(II,JJ) = 0
 1036 DO 1040 J = 1,NZ
      JJ = JJ+1
      COI(II,JJ) = DZZII(I,J,1)
      DCOI(II,JJ) = 0
      COD(II,JJ)=-CZD( I,J )
      DCOD(II,JJ) = 0
      CO(II,JJ) = -DZF(I,J,3)
 1040 DCO(II,JJ) =    DZZII(I,J,6)-CZZII(I,J,3)
 1041 IF(NP.EQ.0) GO TO 1046
      DO 1045 J = 1,NP
      JJ = JJ+1
      COI(II,JJ) = DZPII(I,J,1)
      DCOI(II,JJ) = 0
      COD(II,JJ) = 0
      DCOD(II,JJ) = 0
      CO(II,JJ) = -DZF(I,J,4)
 1045 DCO(II,JJ) = -DZPI(I,J,2)+DZF(I,J,6)
 1046 DF(II) = -(DZMI(I,6)-DZF(I,1,5)+BPC*DZMII(I,2))
      BF(II)=0
      IF(INPUT(13).NE.0.AND.AFZ.NE.0) BF(II)=DZALII(I)
 1050 CONTINUE
 1051 IF(NP.EQ.0) GO TC 1075
      DO 1070 I = 1,NP
      JJ = 0
      II = II+1
      IF(NY.EQ.0) GC TC 1056
      DO 1055 J = 1,NY
      JJ = JJ+1
      COI(II,JJ) = -DPYII(I,J,3)
      DCOI(II,JJ) = 0
      COD(II,JJ) =0
      DCOD(II,JJ) = -2*DPSI(I,J,1)
      CO (II,JJ) = -DPYI(I,J,9)+DPF(I,J,1)
 1055 DCO(II,JJ) = -(DPYII(I,J,8)-CPYII(I,J,6)+DPYII(I,J,3))
 1056 IF(NZ.EQ.0) GO TC 1061
      DO 1060 J = 1,NZ
      JJ = JJ+1
      COI(II,JJ) = DPZII(I,J,2)
      DCOI(II,JJ) =0
      COD(II,JJ) = 0
      DCOD(II,JJ) = 0
      CO(II,JJ)=-(DPZI(I,J,8)+DPF(I,J,2))
 1060 DCO(II,JJ) = DPZII(I,J,7)-DPZII(I,J,4)
```

```
 1061 DO 1065 J = 1,NP
      JJ = JJ+1
      COI(II,JJ) = DPPII(I,J,4)
      DCOI(II,JJ) = 0
      COD(II,JJ)=-DPD( I,J )
      DCOD (II,JJ) = 0
      CO(II,JJ) = -DPF(I,J,3)-DPPI(I,J,6)
 1065 DCO(II,JJ) = -(DPPII(I,J,5)+DPPI(I,J,7))
      BF(II)=0
      IF(INPUT(13).NE.0.AND.AFP.NE.0) BF(II)=DPALII(I)
      DF(II) =-(DPMII(I,9)+BPC*DPMII(I,4))
 1070 CONTINUE
C                                 SUM WITH OMEGAS
 1075 OMEGS = OMEG*OMEG
      OMFS=CMF*OMF
      DO 1080 I = 1,NM
      DO 1076 J = 1,NM
      COIR(I,J) = COI(I,J)+OMEGS*DCCI(I,J)
      CODR(I,J) = COD(I,J)+OMEG *DCOD(I,J)
 1076 COR(I,J) = CO(I,J)+CMEGS*DCO(I,J)
C     NOTE F IS EVALUATED IF FCT
 1080 FR(I) = OMEGS*DF(I)
C                                 INVERT COIR
      CALL INVRS (COIR,NM,RICC,WORK,IROW,ICOL,NMODE,NM1)
C       HUB EFFECTS WITH OLD OMEG TO BE RATIOED LATER
      IF(INPUT(7).EQ.0.AND.INPUT(8).EQ.0.AND.INPUT(9).EQ.0) GO TO 1100
      IF(.NOT.LCALC)GO TO 1090
      JJ=0
      IF(NY.EQ.0) GO TC 1087
      DO 1081 J=1,NY
      JJ=JJ+1
      CONST = YI(1,J,1)
      BIN(1,JJ)= CCNST
      BIN(2,JJ)=-CCNST
      BIN(3,JJ)= 0
      BDAM(1,JJ) = CCNST *2.*CLDOM
      BDAM(2,JJ) = CCNST *2.*CLDOM
      BDAM(3,JJ) =  0
      BSPR(1,JJ) = -CONST*OLDCMS
      BSPR(2,JJ) =  CONST*OLDOMS
      BSPR(3,JJ) = 0
      COIH (JJ,1) =  DYMII(J,1)
      COIH (JJ,2) = -DYMII(J,1)
      COIH (JJ,3) = 0
      DO 1081 I=1,3
 1081 CODH(JJ,I) = 0
 1087 IF(NZ.EQ.0) GO TO 1083
      DO 1082 J=1,NZ
      JJ =JJ +1
      BIN(1,JJ)   =0
      BIN(2,JJ)   =0
      BIN(3,JJ)   =  -ZI(1,J,1)
      BDAM(1,JJ) = 0
      BDAM(2,JJ) = 0
      BDAM(3,JJ) = 0
```

```
      BSPR(1,JJ) = 0
      BSPR(2,JJ) = 0
      BSPR(3,JJ) = 0
      COIH(JJ,1) = 0
      COIH(JJ,2) = 0
      COIH(JJ,3) =    -DZMII(J,1)
      DO 1082 I=1,3
 1082 CODH(JJ,I) = 0
 1083 IF (NP.EQ.0) GO TO 1085
      DO 1084 J=1,NP
      JJ =JJ +1
      CONST = PI(1,J,3)
      BIN(1,JJ) = -CONST
      BIN(2,JJ) =  CONST
      BIN(3,JJ) =  -PI(1,J,1)
      BDAM(1,JJ) = -CONST*2.*CLDOM
      BDAM(2,JJ) = -CONST*2.*CLDOM
      BDAM(3,JJ) =  0
      BSPR(1,JJ) = 0
      BSPR(2,JJ) = 0
      BSPR(3,JJ) = 0
      CONST = DPMII(J,3)
      COIH(JJ,1) = -CONST
      COIH(JJ,2) =  CONST
      COIH(JJ,3) = -CONST
      CODH(JJ,1) = -DPMII(J,3)*CLDOM
      CODH(JJ,2) =  DPMII(J,3)*CLDOM
 1084 CODH(JJ,3) = 0
 1085 DO 1086 I=1,3
      DO 1086 J=1,3
      HC(I,J) = 0
      HK(I,J) = 0
 1086 TM(I,J) = 0
      TM(1,1) = HMX + NB*MI(1,1)
      TM(2,2) = HMY + NB*MI(1,1)
      TM(3,3) = HMZ + NB*MI(1,1)
      HC(1,1) = -HCX
      HC(2,2) = -HCY
      HC(3,3) = -HCZ
      HK(1,1) = -HKX
      HK(2,2) = -HKY
      HK(3,3) = -HKZ
C      INCLUDE OMEGA IN HUB EFFECTS      USES RATIOS
 1090 DO 1091 I=1,3
      DO 1091 J=1, NM
      BDAM(I,J)=BDAM(I,J)*OMRAT
      BSPR(I,J)=BSPR(I,J)*OMRATS
 1091 CODH(J,I)=CODH(J,I)*OMRAT
C        NOTE    NOTE   NOTE - - - SPECIFIC FOR 3 HUB DOF
      CALL MXM(BIRI  ,BIN  ,RIOC  ,3,NM,NM,3,3,NMODE )
      CALL MXM(BIRID ,BIRI ,CODR  ,3,NM,NM,3,3,NMODE )
      CALL MXM(BIRIC ,BIRI ,CCR   ,3,NM,NM,3,3,NMODE )
      DO 1092 I=1,3
      DO 1092 J=1,NM
 1092 BIRIO(I,J)=BIRIO(I,J)+BSPR(I,J)
```

```
      CALL MXM(BIRIDH,BIRI ,CCDH  ,3,NM, 3,3,3,NMODE )
      CALL MXM(BIRIIH,BIRI ,CCIH  ,3,NM, 3,3,3,NMODE )
C             SOLUTICN CONTROLS
 1100 PRMT(1) = 0
      OM=OMF
      IF(OM.EQ.0)OM=OMEG
      PRMT(2) =6.28319*CYCLES/OM
      PRMT(3) =6.28319/HINIT /OM
      PRMT(4) =ERROR
      IF(ERROR.LE.0) CALL ERR(1100,0)
      PRMT(6) = BERR
      DO 1105 I= 1,NDIM
      YVAR(I) = 0
      LY(I) = .FALSE.
 1105 DERY(I) = 0
      IF(IYIC.LE.0) CALL ERR(1105,0)
      IF(IYIC.GT.NDIM) CALL ERR (1106,0)
      YVAR(IYIC) = CIC
      IF(IYE.LE.0) CALL ERR (1107,0)
      IF(IYE.GT.NDIM) CALL ERR(1108,0)
      DERY(IYE) = 1.0
      IF (INPUT(7).NE.0) LY(1)= .TRUE.
      IF (INPUT(7).NE.0) LY(2)= .TRUE.
      IF (INPUT(8).NE.0) LY(3)= .TRUE.
      IF (INPUT(8).NE.0) LY(4)= .TRUE.
      IF (INPUT(9).NE.0) LY(5)= .TRUE.
      IF (INPUT(9).NE.0) LY(6)= .TRUE.
 1200 IDIM = 10+2*NM*NB
      DO 1205 I=11,IDIM
 1205 LY(I) = .TRUE.
C
C
C
C            OUTPUT            OUTPUT            OUTPUT
C
C
 2000 CALL HEADIN
      IF (INPUT(1).NE.2.AND.INPUT(2).NE.2.AND.IC2.EQ.0) GO TO 2050
C                             IO = 1,2
      PRINT 9060,NB,BPC,THO,GV,GW,GP
 9060 FORMAT (//30X,27HIO = 1,2   BLADE PROPERTIES//10X,I5,7H BLADES
     1  5X,9HPRECONE =   ,F6.3,5X,9HTHETA O = ,F6.3,5X,
     2  15HDAMPING (V,W,P)   ,1P3E11.3
     3                           //10X,98HX          M          E
     4SMALL EA        KM1        KM2        KA       THETA PRIME (C)THETA
     5  //)
      DO 2010 I = 1,NX
 2010 PRINT 9070,I,X(I),M(I),E(I),SEA(I),KM1(I),KM2(I),KA(I),THP(I)   ,
     1  TH(I)
 9070 FORMAT (1X,I3,1P10E12.3)
      PRINT 9080
 9080 FORMAT (// 7X, 89HEI CP        EI IP          GJ          EA
     1  EB1*   .    EB2*        EC1          EC1* //)
      DO 2020 I = 1,NX
 2020 PRINT 9070,I,ECP(I),EIP(I),GJ(I),EA(I),EB1(I),EB2(I),EC(I),ECS(I)
```

```
      CALL HEADIN
      PRINT 9090
 9090 FORMAT (//8X,29H(C)EW        (C)EV        (C)EP   //)
      DO 2030 I = 1,NX
 2030 PRINT 9070,I,EW(I),EV(I),EP(I)
C                             IO = 3,4,5
 2050 IF(INPUT(3).NE.2.AND.IC2.EQ.0.OR.INPUT(3).EQ.0) GO TO 2075
      CALL HEADIN
      PRINT 9100
 9100 FORMAT  (//20X,23HIO = 3   IN-PLANE MODES  //  20X,18HSECOND DERIV
     1ATIVES  //)
      DO 2055 I = 1,NX
 2055 PRINT 9070,I,(YPP(I,J),J=1,NY)
      PRINT 9110
 9110 FORMAT (//20X,28H(C) FIRST DERIV (NORMALIZED)  //)
      DO 2060 I=1,NX
 2060 PRINT 9070,I,(YP(I,J),J=1,NY)
      CALL HEADIN
      PRINT 9120
 9120 FORMAT (//20X,15H(C) MODE SHAPES//)
      DO 2065 I = 1,NX
 2065 PRINT 9070,I,(Y(I,J),J=1,NY)
 2075 IF(INPUT(4).NE.2.AND.IC2.EQ.0.OR.INPUT(4).EQ.0) GO TO 2100
      CALL HEADIN
      PRINT 9130
 9130 FORMAT(//20X,27HIO = 4   OUT-OF-PLANE MODES//20X,18HSECOND DERIVAT
     1IVES //)
      DO 2080 I = 1,NX
 2080 PRINT 9070,I,(ZPP(I,J),J=1,NZ)
      PRINT 9110
      DO 2085 I = 1,NX
 2085 PRINT 9070,I,(ZP(I,J),J=1,NZ)
      CALL HEADIN
      PRINT 9120
      DO 2090 I = 1,NX
 2090 PRINT 9070,I,(Z(I,J),J=1,NZ)
 2100 IF(INPUT(5).NE.2.AND.IC2.EQ.0.OR.INPUT(5).EQ.0) GO TO 2150
      CALL HEADIN
      PRINT 9140
 9140 FORMAT ( //20X,22HIO = 5   TORSION MODES //20X,18HSECOND DERIVATIV
     1ES //)
      DO 2105 I = 1,NX
 2105 PRINT 9070,I,(PPP(I,J),J=1,NP)
      PRINT 9110
      DO 2110 I = 1,NX
 2110 PRINT 9070,I,(PP(I,J),J=1,NP)
      CALL HEADIN
      PRINT 9120
      DO 2115 I = 1,NX
 2115 PRINT 9070,I,(P(I,J),J=1,NP)
 2150 IF(IC3 .EQ.0) GO TO 2500
C                         DEFINITE INTEGRALS
      IF(INPUT(3).EQ.0.OR.(INPUT(3).EQ.1.AND.IC2.EQ.0)) GO TO 2200
      CALL HEADIN
      PRINT 9150
```

```
9150 FORMAT (//20X,20H*** DYYI (I,J,N) *** //)
     PRINT 9160,(I,I=1,10)
9160 FORMAT (1X,3HI J,I7,9I12 /)
     DO 2160 I = 1,NY
     DO 2160 J = 1,NY
2160 PRINT 9170,I,J,(DYYI(I,J,N),N=1,10)
9170 FORMAT (1X,I1,I2,1P10E12.3)
     PRINT 9180
9180 FORMAT(//20X,21H*** DYYII (I,J,N) ***  //)
     PRINT 9160,(I,I=1,9)
     DO 2170 I = 1,NY
     DO 2170 J = 1,NY
2170 PRINT 9170,I,J,(DYYII(I,J,N),N=1,9)
     IF(INPUT(4).EQ.0) GO TO 2176
     PRINT 9190
9190 FORMAT (//20X,21H*** DYZII (I,J,N) *** //)
     PRINT 9160,(I,I=1,8)
     DO 2175 I = 1,NY
     DO 2175 J = 1,NZ
2175 PRINT 9170,I,J,(DYZII(I,J,N),N=1,8)
     CALL HEADIN
2176 IF(INPUT(5).EQ.0)  GO TO 2182
     PRINT 9200
9200 FORMAT (//20X,21H*** DYPII (I,J,N) *** //)
     PRINT 9160,(I,I=1,3)
     DO 2180 I = 1,NY
     DO 2180 J = 1,NP
2180 PRINT 9170,I,J,(DYPII(I,J,N),N=1,3)
2182 PRINT 9210
9210 FORMAT (//20X,20H*** DYSI (I,J,N) *** //)
     PRINT 9160,(I,I=1,4)
     DO 2185 I = 1,NY
     DO 2185 J = 1,NMAX
2185 PRINT 9170,I,J,(DYSI(I,J,N),N=1,4)
     PRINT 9220
9220 FORMAT (//20X,18H*** DYMI (I,N) ***  //)
     PRINT 9160,(I,I=1,10)
     DO 2190 I = 1,NY
2190 PRINT 9070,I,(DYMI(I,N),N=1,10)
     PRINT 9230
9230 FORMAT(// 20X,19H*** DYMII (I,N) *** //)
     PRINT 9160,(I,I=1,9)
     DO 2195 I = 1,NY
2195 PRINT 9070,I,(DYMII(I,N),N=1,9)
2200 IF(INPUT(4).EQ.0.OR.(INPUT(4).EQ.1.AND.IC2.EQ.0)) GO TO 2250
     CALL HEADIN
     IF(INPUT(3).EQ.0) GO TO 2211
     PRINT 9240
9240 FORMAT(//20X,20H*** DZYI (I,J,N) *** //)
     PRINT 9160,(I,I=1,10)
     DO 2205 I = 1,NZ
     DO 2205 J = 1,NY
2205 PRINT 9170,I,J,(DZYI(I,J,N),N=1,10)
     PRINT 9250
9250 FORMAT(//20X,21H*** DZYII (I,J,N) *** //)
```

```
      PRINT 9160,(I,I=1,9)
      DO 2210 I = 1,NZ
      DO 2210 J = 1,NY
 2210 PRINT 9170,I,J,(CZYII(I,J,N),N=1,9)
 2211 PRINT 9260
 9260 FORMAT(//20X,21H*** DZZII (I,J,N) ***  //)
      PRINT 9160,(I,I=1,8)
      DO 2215 I = 1,NZ
      DO 2215 J = 1,NZ
 2215 PRINT 9170,I,J,(DZZII(I,J,N),N=1,8)
      IF (INPUT(5).EC.0) GO TC 2226
      CALL HEADIN
      PRINT 9270
 9270 FORMAT(//20X,20H*** DZPI (I,J,N) *** //)
      PRINT 9160,(I,I=1,2)
      DO 2220 I = 1,NZ
      DO 2220 J = 1,NP
 2220 PRINT 9170,I,J,(DZPI(I,J,N),N=1,2)
      PRINT 9280
 9280 FORMAT(//20X,21H*** DZPII (I,J,N) *** //)
      PRINT 9160, (I,I=1,1)
      DO 2225 I = 1,NZ
      DO 2225 J = 1,NP
 2225 PRINT 9170,I,J, CZPII(I,J,1)
 2226 PRINT 9290
 9290 FORMAT(//20X,18H*** DZMI (I,N) *** //)
      PRINT 9160,(I,I=1,10)
      DO 2230 I = 1,NZ
 2230 PRINT 9070,I,(DZMI(I,N),N=1,10)
      PRINT 9300
 9300 FORMAT (//20X,19H*** DZMII (I,N) *** //)
      PRINT 9160,(I,I=1,9)
      DO 2235 I = 1,NZ
 2235 PRINT 9070,I,(DZMII(I,N),N=1,9)
 2250 IF(INPUT(5).EQ.0.OR.(INPUT(5).EC.1.AND.IC2 .EQ.0)) GO TO 2300
      CALL HEADIN
      IF(INPUT(3).EQ.0) GO TC 2261
      PRINT 9310
 9310 FORMAT (//20X,20H*** DPYI (I,J,N) *** //)
      PRINT 9160,(I,I=1,10)
      DO 2255 I = 1,NP
      DO 2255 I = 1,NY
 2255 PRINT 9170,I,J,(DPYI(I,J,N),N=1,10)
      PRINT 9320
 9320 FORMAT (//20X,21H*** CPYII (I,J,N) *** //)
      PRINT 9160,(I,I=1,9)
      DO 2260 I = 1,NP
      DO 2260 J = 1,NY
 2260 PRINT 9170,I,J,(CPYII(I,J,N),N=1,9)
 2261 IF(INPUT(4).EQ.0) GO TC 2271
      PRINT 9330
 9330 FORMAT (//20X,20H*** DPZI (I,J,N) *** //)
      PRINT 9160,(I,I=1,9)
      DO 2265 I = 1,NP
      DO 2265 J = 1,NZ
```

```
2265 PRINT 9170,I,J,(DPZI(I,J,N),N=1,9)
     CALL HEADIN
     PRINT 9340
9340 FORMAT(//20X,21H*** DPZII (I,J,N) *** //)
     PRINT 9160,(I,I=1,8)
     DO 2270 I = 1,NP
     DO 2270 J = 1,NZ
2270 PRINT 9170,I,J,(DPZII(I,J,N),N=1,8)
2271 PRINT 9350
9350 FORMAT(//20X,20H*** DPPI (I,J,N) *** //)
     PRINT 9160,(I,I=1,8)
     DO 2275 I = 1,NP
     DO 2275 J = 1,NP
2275 PRINT 9170,I,J,(DPPI(I,J,N),N=1,8)
     PRINT 9360
9360 FORMAT(//20X,21H*** DPPII (I,J,N) *** //)
     PRINT 9160,(I,I=1,7)
     DO 2280 I = 1,NP
     DO 2280 J = 1,NP
2280 PRINT 9170,I,J,(DPPII(I,J,N),N=1,7)
     CALL HEADIN
     PRINT 9370
9370 FORMAT (//20X,20H*** DPSI (I,J,1) *** //)
     PRINT 9160,(I,I=1,1)
     DO 2285 I = 1,NP
     DO 2285 J = 1,NY
2285 PRINT 9170,I,J, DPSI(I,J,1)
     PRINT 9380
9380 FORMAT (//20X,18H*** DPMI (I,N) *** //)
     PRINT 9160,(I,I=1,10)
     DO 2290 I = 1,NP
2290 PRINT 9070,I,(DPMI(I,N),N=1,10)
     PRINT 9390
9390 FORMAT (//20X,19H*** DPMII (I,N) *** //)
     PRINT 9160,(I,I=1,9)
     DO 2295 I = 1,NP
2295 PRINT 9070,I,(DPMII(I,N),N=1,9)
2300 CALL HEADIN
     IF(INPUT(3).EQ.0) GO TO 2310
     PRINT 9400
9400 FORMAT (//20X,19H*** DYF (I,J,N) *** //)
     PRINT 9160,(I,I=1,6)
     DO 2305 I = 1,NY
     DO 2305 J = 1,NMAX
2305 PRINT 9170,I,J,(DYF(I,J,N),N=1,6)
2310 IF(INPUT(4).EQ.0) GO TO 2320
     PRINT 9410
9410 FORMAT (//20X,19H*** DZF (I,J,N) *** //)
     PRINT 9160,(I,I=1,6)
     DO 2315 I = 1,NZ
     DO 2315 J = 1,NMAX
2315 PRINT 9170,I,J,(DZF(I,J,N),N=1,6)
2320 IF(INPUT(5).EQ.0) GO TO 2330
     PRINT 9420
9420 FORMAT (//20X,19H*** DPF (I,J,N) *** //)
```

```
      PRINT 9160,(I,I=1,3)
      DO 2325 I = 1,NP
      DO 2325 J = 1,NMAX
 2325 PRINT 9170,I,J,(CPF(I,J,N),N=1,3)
 2330 IF(GV.EQ.0.AND.GW.EQ.0.AND.GP.EQ.0) GO TO 2500
      CALL HEADIN
      PRINT 9421
 9421 FORMAT (//20X,21H*** DYD, DZD, DPD ***  //)
      PRINT 9160, (I,I=1,5)
      DO 2335 I=1,NY
 2335 PRINT 9070,I,(DYD(I,J),J=1,NY)
      PRINT 9470
      DO 2340 I=1,NZ
 2340 PRINT 9070,I,(DZD(I,J),J=1,NZ)
      PRINT 9470
      DO 2345 I=1,NP
 2345 PRINT 9070,I,(DPD(I,J),J=1,NP)
 2500 IF (INPUT(6).NE.2.AND.IC2.EQ.0) GO TO 2525
      PRINT 9430,OMEG,CMF
 9430 FORMAT (//20X,22HIC = 6   ROTOR SPEED = ,F6.2,17H   FORCING FREQ =
     1   ,F6.2,12H    (RAD/SEC)  //)
C                             COEFFICIENT MATRICES
 2525 IF(IC4.EQ.0) GO TO 2600
      CALL HEADIN
      PRINT 9450
 9450 FORMAT (//20X,31H*** COIR, CODR, COR, FR, BF *** //)
      DO 2530 I = 1,NM
 2530 PRINT 9460,(COIR(I,J),J=1,NM)
 9460 FORMAT(3X,1P11E11.3)
      PRINT 9470
 9470 FORMAT(//)
      DO 2540 I = 1,NM
 2540 PRINT 9460,(CODR(I,J),J=1,NM)
      PRINT 9470
      DO 2550 I = 1,NM
 2550 PRINT 9460,(COR(I,J),J=1,NM)
      PRINT 9470
      PRINT 9460,(FR(I),I=1,NM)
      PRINT 9470
      PRINT 9460,  (BF(I),I=1,NM)
      CALL HEADIN
      PRINT 9480
 9480 FORMAT (//20X,24H*** RICC = INV(COIR) ***  //)
      DO 2560 I=1,NM
 2560 PRINT 9460, (RIOC(I,J),J=1,NM)
      IF(INPUT(7).EQ.0.AND.INPUT(8).EQ.0.AND.INPUT(9).EQ.0)GO TO 2600
      PRINT 9500
 9500 FORMAT(//20X,20H*** BIRIIH,BIRID ***  //)
      DO 2565 I=1,3
 2565 PRINT 9460,(BIRIIH(I,J),J=1,3)
      PRINT 9470
      DO 2570 I=1,3
 2570 PRINT 9460,(BIRID(I,J),J=1,NM)
      CALL HEADIN
      PRINT 9510
```

```
 9510 FORMAT(//20X,37H*** BIRIO,BIRICH,BIRI,TM,HC,HK,HF ***     //)
      DO 2575 I=1,3
 2575 PRINT 9460,(BIRIC(I,J),J=1,NM)
      PRINT 9470
      DO 2580 I=1,3
 2580 PRINT 9460,(BIRICH(I,J),J=1,3)
      PRINT 9470
      DO 2585 I=1,3
 2585 PRINT 9460,(BIRI(I,J),J=1,NM)
      PRINT 9470
      PRINT 9460,(TM(I,I),I=1,3)
      PRINT9470
      PRINT 9460,(HC(I,I),I=1,3)
      PRINT 9470
      PRINT 9460,(HK(I,I),I=1,3)
      PRINT 9470
      PRINT 9460   ,HF
 2600 IF(INPUT(7).NE.0) PRINT 9600,HMX,HCX,HKX,HF(1)
 9600 FORMAT(//20X,19HIO = 7      HUB DATA  10X,16HHMX,HCX,HKX,HF =
     1  4F10.3)
      IF(INPUT(8).NE.0) PRINT 9601,HMY,HCY,HKY,HF(2)
 9601 FORMAT(//20X,19HIO = 8      HUB DATA  10X,16HHMY,HCY,HKY,HF =
     1  4F10.3)
      IF(INPUT(9).NE.0) PRINT 9602,HMZ,HCZ,HKZ,HF(3)
 9602 FORMAT(//20X,19HIO = 9      HUB DATA  10X,16HHMZ,HCZ,HKZ,HF =
     1  4F10.3)
 3900 IF(INPUT(13).NE.0) PRINT 9740,X(NXF),AFY,AFZ,AFP
 9740 FORMAT(//20X,27HIO = 13   STA, FY, FZ, FP =  ,4F10.3)
      IF(INPUT(13).NE.0.AND.PER.NE.0) PRINT 9743,PER
      IF(INPUT(13).NE.0.AND.PER.NE.0) NBF=0
      IF(INPUT(13).NE.0.AND.NB.GT.1.AND.NBF.EQ.0) PRINT 9741
      IF(INPUT(13).NE.0.AND.NB.GT.1.AND.NBF.NE.0) PRINT 9742,NBF
 9741 FORMAT (30X,10HALL BLADES  )
 9742 FORMAT (30X,9HBLADE NO.  I3)
 9743 FORMAT (/20X,16H1-COS FORCE FOR ,F5.3,24H OF ROTOR CYCLE (FROM 0))
      IF(INPUT(17).NE.2.AND.IC2.EQ.0) GO TO 4000
      PRINT 9750,NLIN
 9750 FORMAT(//20X,16HIO = 17    NLIN = ,I3)
      IF(NLIN.EQ.0) PRINT 9760
      IF(NLIN.EQ.2) PRINT 9770
      IF(NLIN.EQ.1) PRINT 9755
 9755 FORMAT(20X,27H*** I-P NON-LINEARITITS ***)
 9760 FORMAT (20X,27H*** ALL NON-LINEARITIES ***)
 9770 FORMAT (20X,25H*** NO CORIOLIS TERMS ***)
      IF(NFLOQ.NE.0) PRINT 9780
 9780 FORMAT(20X,43H*** AUTOMATIC FLOQUET TRANSITION MATRIX ***)
      IF(NFLOQ.EQ.2) PRINT 9785
 9785 FORMAT(20X,55H*** STEADY FORCES DUE TO STRUCTURAL EFFECTS IGNORED
     1***    )
C
 4000 IF(INPUT(18).NE.2.AND.IC2.EQ.0) GO TO 5000
      PRINT 9800,CYCLES,HINIT,ERROR,IYE,CIC,IYIC,BERR
 9800 FORMAT(//3X,20HIO = 18      CYCLES =,F5.1,4X,7HHINIT =,F5.1,
     1 4X,7HERROR =,F6.3,4X,5HIYE =,I4,4X,5HCIC =,F5.2,4X,6HIYIC =,I4,
     2 4X,6HBERR =,F6.2)

 5000 RETURN
      END
```

```
      SUBROUTINE INT(A,B,A0,X,NX,ICCNT)
C
C                    A(X) = INTEGRAL OF B(X) WITH BC = A0 AT X(1)
C                    X IS INDEPENDANT VARIABLE
C                    NX IS NUMBER OF STATIONS
C                    ICONT = 1    INTEGRAL FROM 0 TO X
C                            2    INTEGRAL FROMX TO R (LAST X)
C
C                    TRAPEZOIDAL INTEGRATION
C
      REAL A(1),B(1),X(1)
      A(1)=A0
      DO 10 I=2,NX
   10 A(I)=A(I-1)+(B(I-1)+B(I))*(X(I)-X(I-1))/2
      IF (ICONT.EQ.1) RETURN
      C=A(NX)
      DO 20 I=1,NX
   20 A(I)=C-A(I)
      RETURN
      END
```

```
      SUBROUTINE INVRS (B,N,A,D, IRCW,ICOL,NRW,NCL)
C     A = INVERSE CF B          B UNCISTURBED
C       VARIABLE DIMENSICNS     NCL MUST BE AT LEAST ONE GREATER THAN NRW
C       NRW MUST BE AT LEAST EQUAL TC N
C       IROW, ICCL ARE VECTCRS CF LENGTH NCL
      REAL A(NRW,NCL),B(NRW,NCL),D(NRW,NCL)
      INTEGER IROW(NCL),ICOL(NCL)
      DO 1 I=1,N
      DO 1 J=1,N
    1 A(I,J)=B(I,J)
      M=N+1
      DO 7  I=1,N
      IROW(I)=I
    7 ICOL(I)=I
      DO 20 K=1,N
      AMAX= A(K,K)
      DO 10 I=K,N
      DO 10 J=K,N
      IF(ABS( A(I,J))-ABS(AMAX))10,9,9
    9 AMAX= A(I,J)
      IC=I
      JC=J
   10 CONTINUE
      KI=ICOL(K)
      ICOL(K)=ICOL(IC)
      ICOL(IC)=KI
      KI=IROW(K)
      IROW(K)=IROW(JC)
      IROW(JC)=KI
      IF(AMAX) 11,12,11
   12 PRINT 13
   13 FORMAT(' SOLLTION OF MATRIX NCT POSSIBLE')
      GO TO 100
   11 DO 14 J=1,N
      E=A(K,J)
      A(K,J)=A(IC,J)
   14 A(IC,J)=E
      DO 15 I=1,N
      E=A(I,K)
      A(I,K)=A(I,JC)
   15 A(I,JC)=E
      DO 16 I=1,N
      IF(I-K) 18,17,18
   17 A(I,M)=1.
      GO TO 16
   18 A(I,M)=0.
   16 CONTINUE
      PVT=A(K,K)
      DO 8 J=1,M
    8 A(K,J)=A(K,J)/PVT
      DO 19 I=1,N
      IF(I-K)21,19,21
```

```
 21 AMULT=A(I,K)
    DO 22 J=1,M
 22 A(I,J)=A(I,J)-AMULT*A(K,J)
 19 CONTINUE
    DO 20 I=1,N
 20  A(I,K)=A(I,M)
    DO 25 I=1,N
    DO 24 L=1,N
    IF(IROW(I)-L)24,23,24
 24 CONTINUE
 23 DO 25 J=1,N
 25 D(L,J)=A(I,J)
    DO 26 J=1,N
    DO 28 L=1,N
    IF(ICOL(J)-L) 28,29,28
 28 CONTINUE
 29 DO 26 I=1,N
 26 A(I,L)=D(I,J)
100 RETURN
    END
```

```
      SUBROUTINE MXM(A,B,C,N,K,M,NA,NB,NC)
C
C     MATRIX MULT    A(NXM)=B(NXK)*C(KXM)
C
      DIMENSION A(NA,1),B(NB,1),C(NC,1)
      DO 20 I=1,N
      DO 20 J=1,M
      A(I,J)=0
      DO 20 L=1,K
   20 A(I,J)=A(I,J)+B(I,L)*C(L,J)
      RETURN
      END
```

```
      SUBROUTINE MXV(A,B,C,M,N,NDIM,ICONT)
C
C           MATRIX TIMES VECTOR    A(M)=B(M,N)*C(N)            FOR ICONT = 0
C                                                 +A(M)        FOR ICONT =1
C
      DIMENSION A(1),B(NDIM,1),C(1)
      DO 10 I=1,M
      IF(ICONT.EQ.0) A(I)=0
      DO 10 J=1,N
   10 A(I)=A(I)+B(I,J)*C(J)
      RETURN
      END
```

```
      SUBROUTINE OUTP(T,YV,DERY,IHLF,MDIM,PRMT,LY)
      REAL M,KM1,KM2,KA
      LOGICAL LY(1)
      REAL DATA (6),DATAT(3)
      DIMENSION YV(1),DERY(1),PRMT(1)
      COMMON/INDAT/X(20),M(20),E(20),SEA(20),KM1(20),KM2(20),KA(20),
     1 THP(20),EOP(20),GJ(20),EA(20),EB1(20),EB2(20),ECS(20),EIP(20),
     2 THO,BPC,YPP(20,3),YP(20,3),ZPP(20,5),ZP(20,5),PPP(20,3),PP(20,3),
     3  OMEG,OMF,EC(20),NY,NZ,NP,NM,CMEGS,OMFS,IDIM,NMAX,NLIN
     4,NB,HMX,HMY,HMZ,HCX,HCY,HCZ,HKX,HKY,HKZ,NX,NFLOQ
     5 ,HINIT,ERROR,IYE,CIC,IYIC,BERR ,CYCLES ,NXF,AFY,AFZ,AFP,NBF
     6  ,R,GV,GW,GP,HE(3),PER
      COMMON/HED/IC1,IC2,IC3,IC4,HEAD(19),IPAGE,INPUT(20),IEND,LINE,IC5
      IF(NFLOQ.NE.0)RETURN
      CYCF = T*OMF/6.28319
      CYCR = T*OMEG /6.28319
      CYCRP=CYCR*360.
      NCYCF=CYCF
      NCYCR=CYCR
      DEGF = (CYCF-FLOAT(NCYCF))*360.
      DEGR = (CYCR-FLOAT(NCYCR))*360.
      LINE=LINE+NMAX*NB+1
      IF(NB.GT.1)LINE=LINE+NB
      IF(NMAX.GT.1) LINE=LINE+NB
      IF(LY(1).OR.LY(3).OR.LY(5))LINE=LINE+2
      IF(LINE.GT.56) LINE=10
      IF(LINE.GT.10) GC TO 50
      CALL HEADIN
      PRINT 1000
 1000 FORMAT ( /119H  TIME      OMF      OMEGA     I     Y(I)DOT      Y(I)
     1                    Z(I)DOT        Z(I)                  PHI(I)DOT     PHI(I
     2) / 26H    SEC    CY  DEG    CY  DEG   )
   50 DO 110 IB=1,NB
      III=2*NM*(IB-1)+9
      DATAT(1)=0
      DATAT(2)=0
      DATAT(3)=0
      DO 100 I=1,NMAX
      DO 90 J=1,6
   90 DATA(J)=0
      IF(NY.LT.I)GC TO 91
      II=III+2*I
      DATA(1)=YV(II)
      DATA(2) =YV(II+1)
      DATAT(1)=DATAT(1)+DATA(2)
   91 IF(NZ.LT.I) GO TC 92
      II=III+2*(I+NY)
      DATA (3) =YV(II)
      DATA (4) =YV(II+1)
      DATAT(2)=DATAT(2)+DATA(4)
   92 IF(NP.LT.I) GO TC 93
      II=III+2*(I+NY+NZ)
```

```
      DATA (5) =YV(II)
      DATA (6) =YV(II+1)
      DATAT(3)=DATAT(3)+DATA(6)
   93 IF(I.GT.1) GC TO 95
      IF(NB.EQ.1) GO TC 94
      IF(IB.EQ.1) PRINT 1004,T,NCYCF,DEGF,NCYCR,DEGR
      IF(IB.GT.1 ) PRINT 1005,IB
      GO TO 95
 1004 FORMAT (/1X,F6.3,2(I4,F6.1),10H *BLADE 1*)
 1005 FORMAT (27X,7H *BLADE,I2,1H*)
   94 PRINT 1010, T,NCYCF,DEGF,NCYCR,DEGR,I,CATA
 1010 FORMAT (/1X,F6.3,2(I4,F6.1),I3,3(1PE12.3,E13.3,8X))
      GO TO 100
   95 PRINT 1020, I,DATA
 1020 FORMAT (20X,I10,3(1PE12.3,E13.3,8X))
  100 CONTINUE
      IF(NMAX.GT.1) PRINT 1021,DATAT
 1021 FORMAT (47X,1PE13.3,20X,E13.3,20X,E13.3)
      IF(IC5.NE.0.AND.IB.EQ.1)WRITE(9)  CYCRP,DATAT,YV(2),YV(4),YV(6)
  110 CONTINUE
      IF(LY(1).OR.LY(3).OR.LY(5)) PRINT 1025,(YV(L),L=1,6)
 1025 FORMAT(/4X,26HHUB XDOT,X, YDCT,Y, ZDOT,Z ,3(1PE12.3,E13.3,8X))
      IF (PRMT(6).EQ.0) GO TC 200
      IF ( ABS(YV(IYE)).LT.PRMT(6) ) GO TO 200
      PRINT 1030
 1030 FORMAT (//24H  *** LINIT EXCEEDED *** //)
      PRMT(5)=1
  200 RETURN
      END
```

```
      SUBROUTINE SCL(PRMT,YVAR,DERY,IHLF,LY)
      INTEGER IROW(31),ICOL(31)
      LOGICAL LY(1)
      REAL PRMT(1),YVAR(1),DERY(1)
      REAL AUX(8,98),BFTEMP(11),ERW(36),FLTM(30,31),FLTMI(30,31),
     1   WORK(30,31)
      REAL HFTEMP(3),FRTEMP(11)
      COMMON/INDAT/X(20),M(20),E(20),SEA(20),KM1(20),KM2(20),KA(20),
     1 THP(20),EOP(20),GJ(20),EA(20),EB1(20),EB2(20),ECS(20),EIP(20),
     2 THO,BPC,YPP(20,3),YP(20,3),ZPP(20,5),ZP(20,5),PPP(20,3),PP(20,3),
     3  OMEG,OMF,EC(20),NY,NZ,NP,NM,CMEGS,OMFS,IDIM,NMAX,NLIN
     4,NB,HMX,HMY,HMZ,HCX,HCY,HCZ,HKX,HKY,HKZ,NX,NFLOQ
     5 ,HINIT,ERROR,IYE,CIC,IYIC,BERR ,CYCLES ,NXF,AFY,AFZ,AFP,NBF
     6  ,R,GV,GW,GP,HE(3),PER
      COMMON/COEF/COI(11,11),DCOI(11,11),COD(11,11),DCOD(11,11),
     1   CO(11,11),DCO(11,11),F(11),DF(11),FNL(11),COIR(11,12),
     2   CODR(11,11),CCR(11,11),FR(11),RIOC(11,12),BF(11)
     3 ,BIN(3,11),BDAM(3,11),BSPR(3,11),COIH(11,3),CODH(11,3),BIRI(3,11)
     4,BIRID(3,11),BIRIO(3,11),BIRIDH(3,3 ),HF(3),TM(3,3),BIRIIH(3,3 )
     5 ,HC(3,3),HK(3,3)
      COMMON/HED/IC1,IC2,IC3,IC4,HEAD(19),IPAGE,INPUT(20),IEND,LINE,IC5
      COMMON/DIM/NINPUT,NSTA,NYMODE,NZMODE,NPMODE,NMODE,NM1,NDIM,NBLADE
      COMMON/DER/TH(20),EV(20),EW(20),EP(20),Y(20,3),Z(20,5),P(20,3)
      EQUIVALENCE(AUX(1),WORK(1))
      IF(NFLOQ.EQ.0)CALLRKGSV(PRMT,YVAR,DERY,IDIM,IHLF,AUX,LY)
      IF(NFLOQ.EQ.0) RETURN
      NVAR=0
      DO 10 I=1,IDIM
      IF(LY(I)) NVAR=NVAR+1
   10 ERW(I)=DERY(I)
      IF(NVAR.GT.30) CALL ERR (5010,0)
      DO 20 I=1,11
      BFTEMP(I)=BF(I)
      FRTEMP(I)=FR(I)
      FR(I)=0.
   20 BF(I)=0.
      DO 25 I=1,3
      HFTEMP(I)=HF(I)
   25 HF(I)=0.
      PRMT2=PRMT(2)
      PRMT(2)=PRMT2/CYCLES
      CALL HEADIN
      PRINT 1000,PRMT(2)
 1000 FORMAT(//30X,43HFLOQUET TRANSITION MATRIX      PERIOD(SEC) =
     1    ,F12.5//)
      II=0
C         NC REPITION CF SCLUTICNS FCR MULTIPLE BLADES
      NM2 = NM+NM
      IEB = 10+NM2
      DO 100 I=1,IDIM
      IF(.NOT.LY(I)) GC TO 100
      II=II+1
```

```
      IF(I.GT.IEB) GO TO 101
      DO 30 J=1,IDIM
      DERY(J)=ERW(J)
   30 YVAR(J)=0
      YVAR(I)=1.
      CALL RKGSV(PRMT,YVAR,DERY,IDIM,IHLF,AUX,LY)
      IF(IHLF.EQ.11) CALL ERR (5030,0)
      IF(IHLF.EQ.12) CALL ERR (5031,0)
      IF(IHLF.GT.12) CALL ERR (5032,0)
      JJ=0
      DO 50 J=1,IDIM
      IF(.NOT.LY(J)) GO TO 50
      JJ=JJ+1
      FLTM(JJ,II)=YVAR(J)
   50 CONTINUE
      PRINT 1010,II,(FLTM(JJ,II),JJ=1,NVAR)
 1010 FORMAT(1X,I3,1P10E12.3/(4X,10E12.3))
  100 CONTINUE
      GO TO 109
  101 ID1 = II-NM2
      ID11 = ID1-1
      IOD1 = II
      DO 108 JB = 2,NB
      DO 108 J = 1,NM2
      JJ = ID11+(JB-1)*NM2+J
      JREF = ID11+J
      IF(ID11.EQ.0) GO TO 103
      DO 102 I = 1,ID11
  102 FLTM(I,JJ) = FLTM(I,JJ-NM2)
  103 DO 107 IB = 1,NB
      IREF = IOD1-1
      IF(IB.EQ.JB) IREF = ID11
      DO 107 I = 1,NM2
      II = ID11+(IB-1)*NM2+I
  107 FLTM(II,JJ) = FLTM(IREF+I,JREF)
      PRINT 1010,JJ,(FLTM(II,JJ),II=1,NVAR)
  108 CONTINUE
  109 CONTINUE
      DO 110 J=1,IDIM
      DERY(J)=ERW(J)
  110 YVAR(J)=0
      DO 120 J=1,11
      IF(NFLOQ.EQ.2) GO TO 120
      FR(J)=FRTEMP(J)
  120 BF(J)=BFTEMP(J)
      DO 125 I=1,3
  125 HF(I)=HFTEMP(I)
      IF(INPUT(13).NE.0)  GO TO 115
      IF(LY(1)      .AND.HFTEMP(1).NE.0)  GO TO 115
      IF(LY(3)      .AND.HFTEMP(2).NE.0)  GO TO 115
      IF(LY(5)      .AND.HFTEMP(3).NE.0)  GO TO 115
      RETURN
  115 CALL RKGSV(PRMT,YVAR,DERY,IDIM,IHLF,AUX,LY)
      DO 130 I=1,NVAR
  130 FLTM(I,I)=FLTM(I,I)-1.
```

```
      CALL INVRS(FLTM,NVAR,FLTMI,WCRK,IROW,ICOL,30,31)
      II=0
      DO 140 I=1,IDIM
      IF(.NOT.LY(I)) GC TO 140
      II=II+1
      YVAR(II)=YVAR(I)
  140 CONTINUE
      PRINT 1020,(YVAR(I),I=1,NVAR)
 1020 FORMAT (//30X,19HPARTICULAR SCLUTION    /(4X,1P10E12.3))
      CALL MXV(DERY,FLTMI,YVAR,NVAR,NVAR,30,0)
      II=0
      DO 150 I=1,IDIM
      YVAR(I)=0.
      IF(.NOT.LY(I)) GC TO 150
      II=II+1
      YVAR(I)=-DERY(II)
  150 CONTINUE
      DO 160 I=1,IDIM
  160 DERY(I)=ERW(I)
      PRMT(2)=PRMT2
      NFLT=NFLOQ
      NFLOQ=0
      CALL RKGSV(PRMT,YVAR,DERY,IDIM,IHLF,AUX,LY)
      NFLOQ=NFLT
      IF(NFLOQ.NE.2) RETURN
      DO 170 I=1,11
  170 BF(I)=BFTEMP(I)
      RETURN
      END
```

```
      SUBROUTINE RKGSV(PRMT,Y,DERY,NDIM,IHLF,AUX,LY)
C
C        SUBROUTINE RKGSV
C           MODIFIED TO INCLUDE OPTIONAL COMPUTATION OF EACH Y(I)
C           FCT, OUTP REMOVED FROM ARG LIST, THUS NO EXTERNAL STMT REQD
C        PURPOSE
C           TO SOLVE A SYSTEM OF FIRST ORDER ORDINARY DIFFERENTIAL
C           EQUATIONS WITH GIVEN INITIAL VALUES.
C
C        USAGE
C           CALL RKGSV (PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX,LY)
C           PARAMETERS FCT AND OUTP REQUIRE AN EXTERNAL STATEMENT.
C
C        DESCRIPTION OF PARAMETERS
C           PRMT   - AN INPUT AND OUTPUT VECTOR WITH DIMENSION GREATER
C                    OR EQUAL TO 5, WHICH SPECIFIES THE PARAMETERS OF
C                    THE INTERVAL AND OF ACCURACY AND WHICH SERVES FOR
C                    COMMUNICATION BETWEEN OUTPUT SUBROUTINE (FURNISHED
C                    BY THE USER) AND SUBROUTINE RKGS. EXCEPT PRMT(5)
C                    THE COMPONENTS ARE NOT DESTROYED BY SUBROUTINE
C                    RKGS AND THEY ARE
C           PRMT(1)- LOWER BOUND OF THE INTERVAL (INPUT),
C           PRMT(2)- UPPER BOUND OF THE INTERVAL (INPUT),
C           PRMT(3)- INITIAL INCREMENT OF THE INDEPENDENT VARIABLE
C                    (INPUT),
C           PRMT(4)- UPPER ERROR BOUND (INPUT). IF ABSOLUTE ERROR IS
C                    GREATER THAN PRMT(4), INCREMENT GETS HALVED.
C                    IF INCREMENT IS LESS THAN PRMT(3) AND ABSOLUTE
C                    ERROR LESS THAN PRMT(4)/50, INCREMENT GETS DOUBLED.
C                    THE USER MAY CHANGE PRMT(4) BY MEANS OF HIS
C                    OUTPUT SUBROUTINE.
C           PRMT(5)- NO INPUT PARAMETER. SUBROUTINE RKGS INITIALIZES
C                    PRMT(5)=0. IF THE USER WANTS TO TERMINATE
C                    SUBROUTINE RKGS AT ANY OUTPUT POINT, HE HAS TO
C                    CHANGE PRMT(5) TO NON-ZERO BY MEANS OF SUBROUTINE
C                    OUTP. FURTHER COMPONENTS OF VECTOR PRMT ARE
C                    FEASIBLE IF ITS DIMENSION IS DEFINED GREATER
C                    THAN 5. HOWEVER SUBROUTINE RKGS DOES NOT REQUIRE
C                    AND CHANGE THEM. NEVERTHELESS THEY MAY BE USEFUL
C                    FOR HANDING RESULT VALUES TO THE MAIN PROGRAM
C                    (CALLING RKGS) WHICH ARE OBTAINED BY SPECIAL
C                    MANIPULATIONS WITH OUTPUT DATA IN SUBROUTINE OUTP.
C           Y      - INPUT VECTOR OF INITIAL VALUES.  (DESTROYED)
C                    LATERON Y IS THE RESULTING VECTOR OF DEPENDENT
C                    VARIABLES COMPUTED AT INTERMEDIATE POINTS X.
C           DERY   - INPUT VECTOR OF ERROR WEIGHTS.  (DESTROYED)
C                    THE SUM OF ITS COMPONENTS MUST BE EQUAL TO 1.
C                    LATERON DERY IS THE VECTOR OF DERIVATIVES, WHICH
C                    BELONG TO FUNCTION VALUES Y AT A POINT X.
C           NDIM   - AN INPUT VALUE, WHICH SPECIFIES THE NUMBER OF
C                    EQUATIONS IN THE SYSTEM.
C           IHLF   - AN OUTPUT VALUE, WHICH SPECIFIES THE NUMBER OF
```

```
C                 BISECTIONS OF THE INITIAL INCREMENT. IF IHLF GETS
C                 GREATER THAN 10, SUBROUTINE RKGS RETURNS WITH
C                 ERROR MESSAGE IHLF=11 INTO MAIN PROGRAM. ERROR
C                 MESSAGE IHLF=12 OR IHLF=13 APPEARS IN CASE
C                 PRMT(3)=0 OR IN CASE SIGN(PRMT(3)).NE.SIGN(PRMT(2)-
C                 PRMT(1)) RESPECTIVELY.
C        FCT    - THE NAME OF AN EXTERNAL SUBROUTINE USED. THIS
C                 SUBROUTINE COMPUTES THE RIGHT HAND SIDES DERY OF
C                 THE SYSTEM TO GIVEN VALUES X AND Y. ITS PARAMETER
C                 LIST MUST BE X,Y,DERY,LY SUBROUTINE FCT SHOULD
C                 NOT DESTROY X AND Y.
C        OUTP   - THE NAME OF AN EXTERNAL OUTPUT SUBROUTINE USED.
C                 ITS PARAMETER LIST MUST BE X,Y,DERY,IHLF,NDIM,PRMT,
C                 LY
C                 NONE OF THESE PARAMETERS (EXCEPT, IF NECESSARY,
C                 PRMT(4),PRMT(5),...) SHOULD BE CHANGED BY
C                 SUBROUTINE OUTP. IF PRMT(5) IS CHANGED TO NON-ZERO,
C                 SUBROUTINE RKGS IS TERMINATED.
C        AUX    - AN AUXILIARY STORAGE ARRAY WITH 8 ROWS AND NDIM
C                 COLUMNS.
C        LY       LOGICAL ARRAY,IF.TRUE. CORRESPONDING Y(I)
C                 IS CALCULATED
C
C     REMARKS
C        THE PROCEDURE TERMINATES AND RETURNS TO CALLING PROGRAM, IF
C        (1) MORE THAN 10 BISECTIONS OF THE INITIAL INCREMENT ARE
C            NECESSARY TO GET SATISFACTORY ACCURACY (ERROR MESSAGE
C            IHLF=11),
C        (2) INITIAL INCREMENT IS EQUAL TO 0 OR HAS WRONG SIGN
C            (ERROR MESSAGES IHLF=12 OR IHLF=13),
C        (3) THE WHOLE INTEGRATION INTERVAL IS WORKED THROUGH,
C        (4) SUBROUTINE OUTP HAS CHANGED PRMT(5) TO NON-ZERO.
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        THE EXTERNAL SUBROUTINES FCT(X,Y,DERY) AND
C        OUTP(X,Y,DERY,IHLF,NDIM,PRMT) MUST BE FURNISHED BY THE USER.
C
C     METHOD
C        EVALUATION IS DONE BY MEANS OF FOURTH ORDER RUNGE-KUTTA
C        FORMULAE IN THE MODIFICATION DUE TO GILL. ACCURACY IS
C        TESTED COMPARING THE RESULTS OF THE PROCEDURE WITH SINGLE
C        AND DOUBLE INCREMENT.
C        SUBROUTINE RKGS AUTOMATICALLY ADJUSTS THE INCREMENT DURING
C        THE WHOLE COMPUTATION BY HALVING OR DOUBLING. IF MORE THAN
C        10 BISECTIONS OF THE INCREMENT ARE NECESSARY TO GET
C        SATISFACTORY ACCURACY, THE SUBROUTINE RETURNS WITH
C        ERROR MESSAGE IHLF=11 INTO MAIN PROGRAM.
C        TO GET FULL FLEXIBILITY IN OUTPUT, AN OUTPUT SUBROUTINE
C        MUST BE FURNISHED BY THE USER.
C        FOR REFERENCE, SEE
C        RALSTON/WILF, MATHEMATICAL METHODS FOR DIGITAL COMPUTERS,
C        WILEY, NEW YORK/LONDON, 1960, PP.110-120.
C
C     ..............................................................
C
```

```
      SUBROUTINE RKGSV(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX,LY)
C
C
C
      DIMENSION Y(1),DERY(1),AUX(8,1),A(4),B(4),C(4),PRMT(1)
      LOGICAL LY(1)
      DO 100 I=1,NDIM
  100 AUX(8,I)=.06666667*DERY(I)
      X=PRMT(1)
      XEND=PRMT(2)
      H=PRMT(3)
      PRMT(5)=0.
      CALL FCT(X,Y,DERY,LY,NDIM)
C
C     ERROR TEST
      IF(H*(XEND-X))470,460,110
C
C     PREPARATIONS FOR RUNGE-KUTTA METHOD
  110 A(1)=.5
      A(2)=.2928932
       A(3)=1.707107
      A(4)=.1666667
      B(1)=2.
      B(2)=1.
      B(3)=1.
      B(4)=2.
      C(1)=.5
      C(2)=.2928932
      C(3)=1.707107
      C(4)=.5
C
C     PREPARATIONS OF FIRST RUNGE-KUTTA STEP
      DO 120 I=1,NDIM
      IF(.NOT.LY(I)) GO TO 120
      AUX(1,I)=Y(I)
      AUX(2,I)=DERY(I)
      AUX(3,I)=0.
      AUX(6,I)=0.
  120 CONTINUE
      IREC=0
      H=H+H
      IHLF=-1
      ISTEP=0
      IEND=0
C
C
C     START OF A RUNGE-KUTTA STEP
  130 IF((X+H-XEND)*H)160,150,140
  140 H=XEND-X
  150 IEND=1
C
C     RECORDING OF INITIAL VALUES OF THIS STEP
  160 CALL OUTP(X,Y,DERY,IREC,NDIM,PRMT,LY)
      IF(PRMT(5))490,170,490
  170 ITEST=0
  180 ISTEP=ISTEP+1
```

```
C
C     START OF INNERMOST RUNGE-KUTTA LOOP
      J=1
  190 AJ=A(J)
      BJ=B(J)
      CJ=C(J)
      DO 200 I=1,NDIM
      IF(.NOT.LY(I)) GC TO 200
      R1=H*DERY(I)
      R2=AJ*(R1-BJ*AUX(6,I))
      Y(I)=Y(I)+R2
      R2=R2+R2+R2
      AUX(6,I)=AUX(6,I)+R2-CJ*R1
  200 CONTINUE
      IF(J-4)210,240,240
  210 J=J+1
      IF(J-3)220,230,220
  220 X=X+.5*H
  230 CALL FCT(X,Y,DERY,LY,NDIM)
      GO TO 190
C     END OF INNERMOST RUNGE-KUTTA LOOP
C     TEST OF ACCURACY
  240 IF(ITEST)250,250,290
C     IN CASE ITEST=0 THERE IS NO POSSIBILITY FOR TESTING OF ACCURACY
  250 DO 260 I=1,NDIM
      IF(LY(I))AUX(4,I) = Y(I)
  260 CONTINUE
      ITEST=1
      ISTEP=ISTEP+ISTEP-2
  270 IHLF=IHLF+1
      X=X-H
      H=.5*H
      DO 280 I=1,NDIM
      IF(.NOT.LY(I)) GC TO 280
      Y(I)=AUX(1,I)
      DERY(I)=AUX(2,I)
      AUX(6,I)=AUX(3,I)
  280 CONTINUE
      GO TO 180
C
C     IN CASE ITEST=1 TESTING OF ACCURACY IS POSSIBLE
  290 IMOD=ISTEP/2
      IF(ISTEP-IMOD-IMOD)300,320,300
  300 CALL FCT(X,Y,DERY,LY,NDIM)
      DO 310 I=1,NDIM
      IF(.NOT.LY(I)) GC TO 310
      AUX(5,I)=Y(I)
      AUX(7,I)=DERY(I)
  310 CONTINUE
      GO TO 180
C
C     COMPUTATION OF TEST VALUE DELT
  320 DELT=0.
      DO 330 I=1,NDIM
      IF(.NOT.LY(I)) GC TO 330
```

```
      DELT=DELT+AUX(8,I)*ABS(AUX(4,I)-Y(I))
  330 CONTINUE
      IF(DELT-PRMT(4))370,370,340
C
C     ERROR IS TOO GREAT
  340 IF(IHLF-10)350,450,450
  350 DO 360 I=1,NDIM
      IF(LY(I))AUX(4,I)=AUX(5,I)
  360 CONTINUE
      ISTEP=ISTEP+ISTEP-4
      X=X-H
      IEND=0
      GO TO 270
C
C     RESULT VALUES ARE GOOD
  370 CALL FCT(X,Y,DERY,LY,NDIM)
      DO 380 I=1,NDIM
      IF(.NOT.LY(I)) GO TO 380
      AUX(1,I)=Y(I)
      AUX(2,I)=DERY(I)
      AUX(3,I)=AUX(6,I)
      Y(I)=AUX(5,I)
      DERY(I)=AUX(7,I)
  380 CONTINUE
      CALL OUTP(X-H,Y,DERY,IHLF,NDIM,PRMT,LY)
      IF(PRMT(5))490,390,490
  390 DO 400 I=1,NDIM
      IF(.NOT.LY(I)) GO TO 400
      Y(I)=AUX(1,I)
      DERY(I)=AUX(2,I)
  400 CONTINUE
      IREC=IHLF
      IF(IEND)410,410,480
C
C     INCREMENT GETS DOUBLED
  410 IHLF=IHLF-1
      ISTEP=ISTEP/2
      H=H+H
      IF(IHLF)130,420,420
  420 IMOD=ISTEP/2
      IF(ISTEP-IMOD-IMOD)130,430,130
  430 IF(DELT-.02*PRMT(4))440,440,130
  440 IHLF=IHLF-1
      ISTEP=ISTEP/2
      H=H+H
      GO TO 130
C
C     RETURNS TO CALLING PROGRAM
  450 IHLF=11
      CALL FCT(X,Y,DERY,LY,NDIM)
      GO TO 480
  460 IHLF=12
      GO TO 480
  470 IHLF=13
  480 CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT,LY)

  490 RETURN
      END
```

```
      SUBROUTINE SUMODE(A,Q,PHI,NSTA,NX,N)
C     MODAL SUMMATION      NSTA=DIMENSION    NX=NO OF STATIONS    N=NO OF MOD
C                                J=N
C                    A(I) =    SUM(Q(J)*PHI(I,J))
C                                J=1
      REAL A(1),Q(1),PHI(NSTA,1)
      DO 10 I=1,NX
      A(I)=0.
      DO 10 J=1,N
   10 A(I)=A(I)+Q(J)*PHI(I,J)
      RETURN
      END
```

```
C
C    *********************************************************************************
C         ROTSI                    ROTSI                    ROTSI                    ROTSI
C                        ROTOR SYSTEM IDENT  -  INCOMPLETE MODEL
C    *********************************************************************************
C
C       INPUT
C       -----                    COL
C
C       (1)  HEADING               1   IC1  .EQ.0  FIRST OR NORMAL RUN  -  ALL INPUT
C                                            1  REPLACE MODES   - INPUT 3,4,5
C                                            2  ADD MODES - INPUT 4,5
C
C                                            8  NEW OP CODE ONLY - INPUT 5
C                                            9  END OF RUN - LAST CARD OF RUN
C
C                                2   IC2  .EQ.1  PRINTS ORTHO CHECKS
C                                            2       AND NORMALIZES MODES
C                                                    NOTE--MODES ARE REPLACED
C                                                    AFTER INPUT AND AFTER
C                                                    RANDOM ERRORS
C
C                                3   IC3  .NE.0  PRINTS EQS FOR MASS IDENT
C
C                                4   IC4  .NE.0  RESTORES INPUT MODES, IF IC1.EQ.8
C
C                                5-80  ARBITRARY HEADING   HEAD(19)
C
C       (2)  MASS DATA - ONE CARD PER BLADE STATION     20 MAX
C
C                               1-10   X(I) STATION
C                               11     * (SEE NOTE) WM
C                               12-20  M - LUMPED MASS
C                               21     * (SEE NOTE) WE
C                               22-30  E - CG OFFSET FROM EA   + WHEN CG FORWARD
C                               31     * (SEE NOTE) WT
C                               32-40  TH - PITCH ANGLE - RAD
C                               41     * (SEE NOTE) WK
C                               42-50  KM    RADIUS OF GYRATION IN TORSION
C
C                *  1ST COL OF EACH WORD CONTAINS WEIGHTING FACTOR
C                  FROM 1-9 (0=1)  HIGHER VALUE INDICATES GREATER CONFIDENCE
C                  SEE IO1 = 3  WO1
C
C         END WITH BLANK CARD
C
C       (3)  CONTROL CARD - MODES
C
C                               1-10   CALV   MULTIPLIES I-P MODE DEFL   (0=1)
C                               11-20  CALW   MULTIPLIES O-P MODE DEFL   (0=1)
C                               21-30  CALP   MULTIPLIES TOR MODE DEFL   (0=1)
C                               31-40  THO    ROOT PITCH ANGLE - RAD
C                                                 ADDS TO TH - (TH NOT CHANGED)
C
```

```
C
C
C      (4)   MODES  -   STATIONS CORRESPOND TO MASS DATA
C
C            EACH MODE        1-10  FREQ   NATURAL ,  RAD/SEC
C                            11-20  OMEG   ROTATIONAL, RAD/SEC
C                            21-30  IF .NE. 0 TEMPORARILY REPLACES CALV
C                            31-40  IF .NE. 0 TEMPORARILY REPLACES CALW
C                            41-50  IF .NE. 0 TEMPORARILY REPLACES CALP
C
C             NEXT CDS      V  I-P DISPLACEMENTS, 8F10.  UP TO 3 CARDS
C             NEXT CDS      W  O-P                        START ON NEW CD
C             NEXT CDS      P  TOR
C
C             FOLLOW BY NEXT MODE  - 8 MODES MAX AT ONE OMEG
C                                  -16 MODES MAX AT ALL OMEG
C                  *** 30 EQS MAX (NOT INCL INVARIANCES) ***
C
C           END WITH BLANK CARD
C
C
C
C     (5)  OPERATION CODES  COL  1,2   IO1,IO2
C
C           COL 1  IO1
C
C
C             1   MODIFY MODES WITH RANDOM ERRORS  - MODES REPLACED
C
C                 WD1  PERCENT RANDOM + OR - RECTANGULAR DIST
C                 WD2  PERCENT BIAS
C                 WD3  INTEGER SEED TO START RANDOM SEQUENCE
C
C                 *** FOLLOW BY NEXT OPERATION CARD (5) ***
C
C             2   SOLVE FOR MINIMUM MODAL CHANGES - MASS MATRIX UNCHANGED
C
C                      ALL MODES MUST BE AT SAME OMEGA       - 8 MAX
C                      FIRST MODE UNCHANGED, LAST MODE WILL CHANGE MOST
C                      MINIMUM SUM PERCENT CHANGES USED
C                      WEIGHTING FACTORS NOT USED IN THIS OPTION
C
C                 WD1.EQ.0  NO LIMIT ON CHANGES
C                 WD1.EQ.1  LIMIT CHANGES - SCALE OPTION
C                 WD2-8  MAX PCT CHANGE ALLOWED IN EACH MODE.
C                        CHANGES ARE SCALED SO MAX CHANGE .LE. MAXIMUM
C                        0 INDICATES NO LIMIT.
C
C                 WD1.EQ.2  LIMIT CHANGES - TRUNCATE OPTION
C                 WD2-8  SAME AS FOR SCALE OPTIONEXCEPT THAT ONLY
C                        CHANGES WHICH EXCEDE LIMITS ARE TRUNCATED.
C                        OTHER CHANGES ARE NOT MODIFIED
C
C
```

```
C                3   INCOMP MODEL MASS CHANGES
C
C                      WD1.EQ.1  WEIGHTING FACTORS ALL SET TO 1 (TEMP)
C                      WD1.EQ.2  STAS WITH INVARIANT PARAM.  READ 5(A)
C
C                           THE FOLLOWING CONTROLS CAUSE THE CORRESPONDING
C                           PROPERTIES TO REMAIN INVARIANT IF .NE. 0.
C
C                     COL 20  TOTAL MASS     M
C                         30  RADIAL CG      M*X
C                         40  CHORDWISE CG     M*E
C                         50  FLAPPING MOM OF INERT    M*X**2
C                         60  FEATHERING MOM OF INERT    M*KM**2
C
C
C               COL 2   I02
C
C               0   ABOVE OPERATIONS DO NOT DISTURB ORIGINAL DATA
C
C               1   ABOVE OPERATIONS REPLACE ORIGINAL DATA IN PREPERATION
C                   FOR SEQUENTIAL OPERATIONS
C
C
C     (5A)      USED ONLY FOR INVAR STAS.    SEE 3,ABOVE, WD1 = 2
C
C                     COL1 = NO OF STATIONS (8 MAX)
C                     WD1,WD2,...STATION NUMBERS, NO ZEROES
C
C
C     NEXT HEADING CARD
C
C   ***************************************************************************
C   ***************************************************************************
      INTEGER HEAD(19),IROW(46),ICOL(46)
      INTEGER IJEQ(40,2)
      INTEGER NIN(8)
      REAL X(21),WM(20),M(21),WE(20),E(21),WT(20),TH(21),WK(20),KM(21),
     1     OMEG(16),FREQ(16),V(16,20),W(16,20),P(16,20)
      REAL DUM(8),V2(16,20),W2(16,20),P2(16,20),ME(20),MET(20),MK(20),
     1  A(60,7),PHI(60),B(60,7),C(7,8),D(7,8),WORK(7,8)
      REAL WOR(60),WO(7)
      REAL GMASS(16),OCHECK(16,16)
      REAL EQ(35,80),MA(80),WA(80)
      REAL              M2(20),E2(20),TH2(20),KM2(20),ME2(20),MET2(20),
     1  MK2(20),SM(5)
      REAL VSAV(16,20),WSAV(16,20),PSAV(16,20)
      REAL   WV(80),DM(80),         AWA(35,36),AWAI(35,36),DWA(35,36)
    1 READ 1000,IC1,IC2,IC3,IC4,HEAD
 1000 FORMAT(4I1,19A4)
      IF(IC1.EQ.9) CALL EXIT
      PRINT 1001,IC1,IC2,IC3,IC4,HEAD
 1001 FORMAT(1H1,10X,100(1H*)/
     1          /20X,58HROTOR SYSTEM IDENTIFICATION PROGRAM  ROTSI
     2  1/19/77   //10X,4I2,19A4//10X,100(1H*)//)
```

```
      IF(IC1.EQ.1)  GO TO 25
      IF(IC1.EQ.2) GO TO 29
      IF(IC1.EQ.8.AND.IC4.EQ.0) GO TO 100
      IF(IC1.EQ.0) GO TO 9
      DO 5 I=1,NX
      DO 5 J=1,NM
      V(J,I)=VSAV(J,I)
      W(J,I)=WSAV(J,I)
    5 P(J,I)=PSAV(J,I)
      PRINT 1006
 1006 FORMAT (//10X,31H*** ORIGINAL MODES RESTORED ***    //)
      GO TO 100
    9 NX=0
      DO 10 I=1,21
      READ 1005,X(I),IM,M(I),IE,E(I),IT,TH(I),IK,KM(I)
 1005 FORMAT(F10.0,4(I1,F9.0))
      IF(M(I).EQ.0) GO TO 20
      NX=NX+1
      WM(I)=AMAX0(1,IM)
      WE(I)=AMAX0(1,IE)
      WT(I)=AMAX0(1,IT)
   10 WK(I)=AMAX0(1,IK)
      CALL ERR(10,0)
   20 PRINT 1010,(I,X(I),WM(I),M(I),WE(I),E(I),WT(I),TH(I),WK(I),KM(I),
     1   I=1,NX)
 1010 FORMAT (10X,90HI          STA          W          M          W          E
     1           W          TH          W          KM    /(10X,0P I2,F12.3 ,
     2   4(0PF8.0,1PE12.3)))
   25 READ 1015, CALV,CALW,CALP,THO
 1015 FORMAT (8F10.0)
      N2=2*NX
      N3=3*NX
      N4=4*NX
      IF(CALV.EQ.0) CALV = 1
      IF(CALW.EQ.0) CALW = 1
      IF(CALP.EQ.0) CALP = 1
      PRINT 1016, THO
 1016 FORMAT (//10X,32HROOT PITCH ANGLE (ADDS TO TH) =   1PE12.3/ 1H1,
     1    10X,25HINPUT MODES (CARD IMAGES)    /)
      NM=0
      PRINT 1017,CALV,CALW,CALP,THO
 1017 FORMAT (/10X,8F12.5)
   29 IF(IC1.EQ.2) PRINT 1019
 1019 FORMAT (1H1,10X,27HADDED MODES   CARD IMAGES      //)
   30 READ 1015,F,O,CV,CW,CP
       PRINT 1017,F,O,CV,CW,CP
      IF(F.EQ.0.AND.O.EQ.0) GO TO 70
      NM=NM+1
      FREQ(NM)=F
      OMEG(NM)=O
      READ 1015,(V(NM,I),I=1,NX)
      PRINT 1018,(V(NM,I),I=1,NX)
 1018 FORMAT (10X,8F12.5)
      READ 1015,(W(NM,I),I=1,NX)
```

```
      PRINT 1018,(W(NM,I),I=1,NX)
      READ 1015,(P(NM,I),I=1,NX)
      PRINT 1018,(P(NM,I),I=1,NX)
   40 IF(NM.GT.16) CALL ERR(40,0)
C                                      APPLY CALIBRATION
      IF(CV.EQ.0) CV=CALV
      IF(CW.EQ.0) CW=CALW
      IF(CP.EQ.0) CP=CALP
      I=NM
      IF(CV   .EQ.1.)GO TO 50
      DO 45 J=1,NX
   45 V(I,J)=V(I ,J)*CV
   50 IF(CW   .EQ.1.)GO TO 60
      DO 55 J=1,NX
   55 W(I,J)=W(I ,J)*CW
   60 IF(CP   .EQ.1.)GO TO 30
      DO 65 J=1,NX
   65 P(I,J)=P(I ,J)*CP
      GO TO 30
   70 DO 41 I=1,NX
      DO 41 J=1,NM
      VSAV(J,I)=V(J,I)
      WSAV(J,I)=W(J,I)
   41 PSAV(J,I)=P(J,I)
C                                          PRINT MODES
      PRINT 1020
 1020 FORMAT (1H1//50X,31HINPUT MODE SHAPES (CAL APPLIED) )
      CALL PMODES (X,V,W,P,OMEG,FREQ,NM,NX,16)
   90 AM   =0
      AME  =0
      AMET=0
      AMK  =0
      SM(2)=0
      SM(3)=0
      SM(4)=0
      DO 95 I=1,NX
      ME(I) = M(I)*E(I)
      MET(I)= ME(I)*(TH(I)+THO)
      MK(I)  = M(I)*KM(I)**2
      AM = AM+M(I)
      SM(2)=SM(2)+M(I)*X(I)
      SM(3)=SM(3)+ME(I)
      SM(4)=SM(4)+M(I)*X(I)**2
      AME = AME +ABS(ME(I))
      AMET = AMET+ABS(MET(I))
   95 AMK = AMK+MK(I)
      SM(1)=AM
      SM(5)=AMK
      AM   =  AM/NX
      AME = AME/NX
      AMET = AMET/NX
      AMK = AMK/NX
      IF(AM .EQ.0) CALL ERR(95,0)
      IF(AMK.EQ.0) CALL ERR(96,0)
```

```
      IF(IC2.EQ.0) GO TO100
      PRINT 1031
 1031 FORMAT (1H1//30X,25HINPUT ORTHOGONALITY CHECK  //)
      CALL ORTH( V,W,P,M,ME,MET,MK,NM,NX,16,GMASS,OCHECK,16,IC2)
      IF (IC2.EQ.2) PRINT 1032
 1032 FORMAT(40X,42H*** MODES REPLACED BY NORMALIZED MODES ***  //)
      IF(IC2.EQ.2) CALL PMODES (X,V,W,P,OMEG,FREQ,NM,NX,16)
      IF(IC2.EQ.2.AND.IO1.NE.1) IC2=1
C                            READ PROGRAM OPTIONS
  100 READ 1035,IO1,IO2,DUM
 1035 FORMAT(2I1,F8.0,7F10.0)
      GO TO (110,200,500,130),IO1
C
C     FOR IO1=1
C
C     WD1=UNIFORMLY DISTRIBUTED RANDOM ERROR HAVING A
C     +/- MAXIMUM OF PCT         ON AMPLITJDE
C     WD2=BIAS ERROR OF PCTB         ON AMPLITUDE
C     IZ IS USED IN CALCULATING AN INTEGER RANDCM NUMBER
C     USED IN SUBROUTINE RANDU
C
  110 WD1=DUM(1)/100.
      WD2=DUM(2)/100.
      IZ=DUM(3)
      IX=IZ*2+1
      CALL ERRA( V,WD1,WD2,NX,NM,IX,16 )
      CALL ERRA( W,WD1,WD2,NX,NM,IX,16 )
      CALL ERRA( P,WD1,WD2,NX,NM,IX,16 )
      PRINT 2050, DUM(1),DUM(2),IZ
 2050 FORMAT (///30X,27H*** RANDOM ERROR OPTION ***
     1   /   13X,10HPCT ERROR=,F7.3,5X,11HBIAS ERROR=,F7.2,5X,
     215HRANDOM NO SEED=,I10/)
      IF(IC2.NE.0)CALL ORTH(V,W,P,M,ME,MET,MK,NM,NX,16,GMASS,OCHECK,16,
     1    IC2)
      PRINT 1036
 1036 FORMAT (//40X, 43H*** MODES REPLACED BY MODES WITH ERRORS ***  /)
      IF(IC2.EQ.2) PRINT 1032
      IF(IC2.EQ.2) IC2=1
      CALL PMODES( X,V,W,P,OMEG,FREQ,NM,NX,16 )
      GO TO 100
  130 CALL ERR(130,0)
C
C                   CORRECT MODES ONLY        IO1 = 2
C
C                     ORIGINAL MODES UNDISTURBED
C                     CORRECTED MODES IN V2,W2,P2
C                             CHECK FREQUENCIES, MODES
  200 PRINT 1040
 1040 FORMAT(1H1,30X,18HMODE CHANGE OPTION //30X,26HPERCENTAGE CHANGES (
     1V,W,P)  //20X,16HMODE 1 UNCHANGED  )
      IF(DUM(1).EQ.1) PRINT1043
      IF(DUM(1).EQ.2) PRINT1044
 1043 FORMAT (20X,21HLIMIT OPTION - SCALED  )
 1044 FORMAT (20X,24HLIMIT OPTION - TRUNCATED  )
```

```
      IF (NM.GT. 8) CALL ERR(200,0)
      OM=OMEG(1)
      DO 210 I=2,NM
      IF(OMEG(I).NE.OM) CALL ERR(211,0)
  210 CONTINUE
C                                  CHANGED MODE IN V2,W2,P2
C                                  FIRST MODE UNCHANGED
      DO 220 I=1,NX
      V2(1,I)=V(1,I)
      W2(1,I)=W(1,I)
  220 P2(1,I)=P(1,I)
      N=1
C                                  FORM  A  M1 TH  COLUMN      A IS COMPRESSED
  250 M1=N
      N=N+1
      DO 260 I=1,NX
      A(I,M1) = M(I)*V2(M1,I)-MET(I)*P2(M1,I)
      A(NX+I,M1) = M(I)*W2(M1,I)+ME(I)*P2(M1,I)
  260 A(N2+I    ,M1) = -MET(I)*V2(M1,I)+ME(I)*W2(M1,I)+MK(I)*P2(M1,I)
C                                  FORM  COMPRESSED M TH MODE
      DO 270 I=1,NX
      PHI(I) = V(N,I)
      PHI(NX+I) = W(N,I)
  270 PHI(N2+I)    = P(N,I)
      DO 280 I = 1,N3
      DO 280 J = 1,M1
  280 B(I,J) = PHI(I)*A(I,J)
C                                  C = B(TRAN) * B     (M1XM1)
      DO 290 I = 1,M1
      DO 290 J = 1,M1
      C(I,J) = 0
      DO 290 L = 1,N3
  290 C(I,J) = C(I,J)+B(L,I)*B(L,J)
C                                  INVERT C INTO D
      IF(M1.NE.1) GO TO 300
      D(1,1) = 1.0/C(1,1)
      GO TO 310
  300 CALL INVRS (C,M1,D,WORK,IROW,ICOL,7,8)
C                                  A(TRAN) * PHI
  310 DO 320 I = 1,M1
      WOR(I)=0
      DO 320 J = 1,N3
  320 WOR(I) = WOR(I)+A(J,I)*PHI(J)
      CALL MXV (WO,D,WOR,M1,M1,7,0)
      CALL MXV (WOR,B,WO,N3,M1,60,0)
C                                  WOR = - FRACTIONAL CHANGE IN EACH ELEMENT
C                                  PRINT PERCENT CHANGES
      EMAX = 0
      DO 330 I = 1,N3
      WOR(I) = -WOR(I)*100.
  330 EMAX = AMAX1 (EMAX,ABS(WOR(I)))
      PRINT 1050,N,EMAX
 1050 FORMAT (/20X,5HMODE  I2,10HMAX CHANGE  F8.1)
      IF(DUM(1).EQ.0) GO TO331
```

```
      IF(DUM(N).NE.0) PRINT 1051,DUM(N)
 1051 FORMAT (45X,18HMAX ALLOWED CHANGE   F6.2)
      PRINT 1055,(WOR(I),I=1,NX)
      I1 = NX+1
      I3 = N2+1
      PRINT 1055,(WOR(I),I=I1,N2)
 1055 FORMAT(/(20X,10F10.3))
      PRINT 1055,(WOR(I),I=I3,N3)
  331 TEMP = .01
      IF(DUM(1).EQ.0) GO TO 335
      IF(DUM(N).EQ.0.OR.EMAX.LE.DUM(N)) GO TO 335
      IF(DUM(1).EQ.2.)GO TO 342
      TEMP = .01*DUM(N)/EMAX
  335 DO 340 I = 1 ,N3
  340 PHI(I) = PHI(I)*(1.+TEMP*WOR(I))
      GO TO 349
  342 DO 345 I=1,N3
      IF(WOR(I).GT.0)  WOR(I)=AMIN1(WOR(I),DUM(N))
      IF(WOR(I).LT.0)  WOR(I)=AMAX1(WOR(I),-DUM(N))
  345 PHI(I) =PHI(I)*(1.+TEMP*WOR(I))
  349 DO350 I= 1,NX
      V2(N,I) = PHI(I)
      W2(N,I) = PHI(NX+I)
  350 P2(N,I) = PHI(N2+I)
      IF(N.LT.NM) GO TO 250
  355 PRINT 1060
 1060 FORMAT (1H1 // 30X,15HCORRECTED MODES /)
      CALL PMODES (X,V2,W2,P2,OMEG,FREQ,NM,NX,16)
  370 IF (IC2.EQ.0) GO TO 1
      PRINT 1061
 1061 FORMAT (1H1//30X,30HCORRECTED ORTHOGONALITY CHECK     //)
      CALL ORTH (V2,W2,P2,M,ME,MET,MK,NM,NX,16,GMASS,OCHECK, 16,IC2)
      IF (IC2.EQ.2) CALL PMODES (X,V2,W2,P2,OMEG,FREQ,NM,NX,16)
      IF (IO2.EQ.0) GO TO 1
      DO 380 I=1,NX
      DO 380 J=1,NM
      V(J,I)=V2(J,I)
      W(J,I)=W2(J,I)
  380 P(J,I)=P2(J,I)
      PRINT 1065
 1065 FORMAT (//10X,47H*** ORIGINAL DATA REPLACED BY MODIFIED DATA ***
     1    //)
      GO TO 1
C                                  MASS ONLY SI        IO1 = 3
C                           ORIGINAL MASS PARAMETERS UNDISTURBED
C                           CORRECTED VALUES IN M2,E2,TH2,KM2
C                                  SET UP EQUATION PAIRS
  500 NEQ = 0
      NSI=0
      NM1 = NM-1
      DO 510 I = 1,NM1
      I1 = I+1
      DO 510 J = I1,NM
      IF (OMEG(J).NE.OMEG(I)) GO TO 510
```

```
      NEQ = NEQ+1
      IJEQ(NEQ,1) = I
      IJEQ(NEQ,2) = J
  510 CONTINUE
      IF(NEQ.GT.30) CALL ERR(510,0)
      PRINT 2000,(IJEQ(I,1),IJEQ(I,2),I=1,NEQ)
 2000 FORMAT (1H1,30X,18HMASS CHANGE OPTION  //30X,25HEQUATION PAIRS (M3
     1DE NOS)  //(10X,10(I7,I4)))
      IF(DUM(1).EQ.1.) PRINT 1999
 1999 FORMAT (/30X,37HALL WEIGHTING FACTORS SET TO 1 (TEMP)     )
      IF(NEQ.GT.N4) CALL ERR (511,0)
      IF(DUM(1).NE.2.)GO TO 520
      READ 1997,NSI,(NIN(J),J=1,NSI)
 1997 FORMAT (I1,I9,7I10)
      PRINT 1998,(NIN(J),J=1,NSI)
 1998 FORMAT (30X,28HNO CHANGES AT FOLLOWING STAS   /30X,8(2X,I3))
C                                SET UP EQUATION COEFFICIENTS
  520 DO 550 I = 1,NEQ
      II = IJEQ(I,1)
      JJ = IJEQ(I,2)
      DO 550 J = 1,NX
      EQ(I,J) = V(II,J)*V(JJ,J)+W(II,J)*W(JJ,J)
      EQ(I,NX+J) = W(II,J)*P(JJ,J)+W(JJ,J)*P(II,J)
      EQ(I,N2+J)   = -V(II,J)*P(JJ,J)-V(JJ,J)*P(II,J)
  550 EQ(I,N3+J)     = P(II,J)*P(JJ,J)
      DO 551 I=1,NEQ
  551 WV(I)=0.
      IF(DUM(2).EQ.0) GO TO 553
      PRINT 2001,SM(1)
 2001 FORMAT (30X,36HTOTAL MASS INVARIANT AT                   F10.3 )
      NEQ=NEQ+1
      WV(NEQ)=-SM(1)
      DO 552 I=1,NX
      EQ(NEQ,I) = 1.0
      EQ(NEQ,NX+I) = 0.0
      EQ(NEQ,N2+I) = 0.0
  552 EQ(NEQ,N3+I) = 0.0
  553 IF(DUM(3).EQ.0) GO TO 555
      TEMP=SM(2)/SM(1)
      PRINT 2002, TEMP
 2002 FORMAT (30X,36HRADIAL CG INVARIANT AT                  F10.2  )
      NEQ=NEQ+1
      WV(NEQ)=-SM(2)
      DO 554 I=1,NX
      EQ(NEQ,I) = X(I)
      EQ(NEQ,NX+I) = 0.0
      EQ(NEQ,N2+I) = 0.0
  554 EQ(NEQ,N3+I) = 0.0
  555 IF (DUM(4).EQ.0) GO TO 557
      TEMP = SM(3)/SM(1)
      PRINT 2003,TEMP
 2003 FORMAT (30X,36HCHORDWISE CG INVARIANT AT                F10.4  )
      NEQ=NEQ+1
      WV(NEQ)=-SM(3)
```

```
      DO 556 I=1,NX
      EQ(NEQ,I) = 0.0
      EQ(NEQ,NX+I) = 1.0
      EQ(NEQ,N2+I) = 0.0
  556 EQ(NEQ,N3+I) = 0.0
  557 IF(DUM(5).EQ.0) GO TO 559
      PRINT 2004, SM(4)
 2004 FORMAT (30X,34HFLAPPING MOM OF INERT INVARIANT AT    F12.2)
      NEQ=NEQ+1
      WV(NEQ)=-SM(4)
      DO 558 I=1,NX
      EQ(NEQ,I) = X(I)**2
      EQ(NEQ,NX+I) = 0.0
      EQ(NEQ,N2+I) = 0.0
  558 EQ(NEQ,N3+I) = 0.0
  559 IF(DUM(6).EQ.0) GO TO 565
      PRINT 2005,SM(5)
 2005 FORMAT (30X,36HFEATHERING MOM OF INERT INVARIANT AT    F10.4)
      NEQ=NEQ+1
      WV(NEQ)= -SM(5)
      DO 560 I=1,NX
      EQ(NEQ,I) = 0.0
      EQ(NEQ,NX+I) =0.0
      EQ(NEQ,N2+I) =0.0
  560 EQ(NEQ,N3+I) =1.0
  565 N41=N4
      IF(DUM(1).EQ.2.) N41=N4-NSI
      PRINT 2006,NEQ,N41
 2006 FORMAT(/30X,17HTOTAL EQUATIONS =  I5,18H, NO OF UNKNOWNS =  I4/)
      IF(IC3.EQ.0) GO TO 580
      PRINT 2010
 2010 FORMAT (//30X,35HEQUATION COEFFICIENTS FOR MASS SI     /)
      DO 570 I=1,NEQ
  570 PRINT 2020,(EQ(I,J),J=1,N4)
 2020 FORMAT (/(10X,1P10E12.3))
C
C                         FORM COMPRESSED MA MATRIX
C
  580 DO 590 I = 1,NX
      MA(I) = M(I)
      MA(NX+I) = ME(I)
      MA(N2+I)    = MET(I)
  590 MA(N3+I)    = MK(I)
C
C     FORM INVERSE, COMPRESSED PERCENTAGE WEIGHTED WEIGHTING FUNCTION
C
      IF(DUM(1).EQ.1.)GO TO 602
      DO 600 I = 1,NX
      WA(I)=M(I)/WM(I)
      WA(NX+I)=ME(I)/WE(I)
      IF(ME(I).EQ.0) WA(NX+I)=AME/WE(I)
      WA(N2+I)=MET(I)/WT(I)
      IF(MET(I).EQ.0)   WA(N2+I)=AMET/WT(I)
      WA(N3+I)=MK(I)/WK(I)
```

```
      IF(MK(I).EQ.0) WA(N3+I)=AMK/WK(I)
  600 CONTINUE
      IF(NSI.EQ.0)GO TO 609
      DO 601 I=1,NSI
      J=NIN(I)
      IF(J.LE.0.OR.J.GT.NX) CALL ERR(601,0)
      WA(J)=0
      WA(NX+J)=0
      WA(N2+J)=0
  601 WA(N3+J)=0
      GO TO 609
  602 DO 605 I = 1,NX
      WA(I)=M(I)
      WA(NX+I)=ME(I)
      IF(ME(I).EQ.0) WA(NX+I)=AME
      WA(N2+I)=MET(I)
      IF(MET(I).EQ.0)   WA(N2+I)=AMET
      WA(N3+I)=MK(I)
      IF(MK(I).EQ.0) WA(N3+I)=AMK
  605 CONTINUE
C                                AWA = EQ*WA**(-2)*EQ(T)      (NEQ*NEQ)
  609 DO 610 I = 1,NEQ
      DO 610 J = 1,NEQ
      AWA(I,J) = 0
      DO 610 L = 1,N4
  610 AWA(I,J) = AWA(I,J)+EQ(I,L)*EQ(J,L)*WA(L)*WA(L)
C                               NOTE DWA IS DUMMY ONLY,AWA IS FREE
      IF(IC3.EQ.0) GO TO 612
      PRINT 2021
 2021 FORMAT (1H1 // 30X,21HMATRIX TO BE INVERTED   //)
      DO 611 I=1,NEQ
  611 PRINT 2020,(AWA(I,J),J=1,NEQ)
  612 CALL INVRS (AWA,NEQ,AWAI,DWA,IROW,ICOL,35,36)
      IF(IC3.EQ.0) GO TO 615
      PRINT 2022
 2022 FORMAT (// 30X,14H        INVERSE   //)
      DO 614 I=1,NEQ
  614 PRINT 2020,(AWAI(I,J),J=1,NEQ)
  615 DO 618 I=1,NEQ
      DO 618 J=1,N4
  618 WV(I)=EQ(I,J)*MA(J)+WV(I)
      IF(IC3.EQ.0) GO TO 619
      PRINT 2023
 2023 FORMAT (//30X,12HEQ*MA (TRAN)    //)
      PRINT 2020,(WV(I),I=1,NEQ)
  619 DO 625 I=1,NEQ
      DM(I)=0
      DO 625 J=1,NEQ
  625 DM(I)=DM(I)+AWAI(I,J)*WV(J)
C          FORM      WV = EQ(T)*DM     THEN  DM = DELTA MASS
      DO 620 I = 1,N4
      WV(I) = 0
      DO 620 J = 1,NEQ
  620 WV(I) = WV(I)+EQ(J,I)*DM(J)
```

```
      DO 630 I = 1,N4
  630 DM(I) = -WV(I)*WA(I)**2
C                                     FORM CORRECTED CHARACTERISTICS
      DO 640 I = 1,NX
      M2(I) = M(I)+DM(I)
      ME2(I) = ME(I)+DM(NX+I)
      E2(I) = ME2(I)/M2(I)
      MET2(I) = MET(I)+DM(N2+I)
      IF(ME2(I).EQ.0) GO TO 635
      TH2(I) = MET2(I)/ME2(I)-THO
      GO TO 636
  635 TH2(I)=TH(I)
  636 MK2(I) = MK(I)+DM(N3+I)
      TEMP = MK2(I)/M2(I)
      IF(TEMP.GE.0) GO TO 639
      KM2(I) = -SQRT(-TEMP)
      GO TO 640
  639 KM2(I) = SQRT(TEMP)
  640 CONTINUE
C                               COMPUTE PCT CHANGES IN AWA
      DO 650 I = 1,NX
      AWA(I,1) = DM(I)/M(I)*100.
      AWA(I,4) = (KM2(I)-KM(I))/KM(I)*100.
      IF(TH(I).EQ.0) GO TO647
      AWA(I,3) = (TH2(I)-TH(I))/TH(I)*100.
      GO TO 648
  647 AWA(I,3)=100.
      IF(TH2(I).EQ.0) AWA(I,3)=0
  648 IF(E(I).EQ.0) GO TO 649
      AWA(I,2)= (E2(I)-E(I))/E(I)*100.
      GO TO 650
  649 AWA(I,2) = 100.
      IF(E2(I).EQ.0)  AWA(I,2)=0
  650 CONTINUE
C                                     PRINT CHANGED VALUES
      PRINT 2030
 2030 FORMAT (1H1//130H  I     ORIG M        NEW  M     PCT          ORIG E
     1  NEW  E       PCT          ORIG TH       NEW  TH     PCT         ORIG KM
     2NEW  KM     PCT      //)
      DO 655 I=1,NX
  655 PRINT 2040, I,M(I),M2(I),AWA(I,1),E(I),E2(I),AWA(I,2),
     1 TH(I),TH2(I),AWA(I,3),KM(I),KM2(I),AWA(I,4)
 2040 FORMAT (I3,4(1PE13.3,E12.3,0PF7.1))
C                                     ORTH CHECK
      IF (IC2.EQ.0) GO TO 1
      PRINT 1061
      CALL ORTH (V,W,P,M2,ME2,MET2,MK2,NM,NX,16,GMASS,OCHECK,16,IC2)
      IF(IC2.EQ.2) PRINT 1032
      IF(IC2.EQ.2) CALL PMODES (X,V,W,P,OMEG,FREQ,NM,NX,16)
      IF(IO2.EQ.0) GO TO 1
      DO 660 I=1,NX
      M(I)=M2(I)
      E(I)=E2(I)
      TH(I)=TH2(I)
```

```
      KM(I)=KM2(I)
      ME(I)=ME2(I)
      MET(I)=MET2(I)
  660 MK(I)=MK2(I)
      PRINT 1065
      GO TO 1
      END
```

```
      SUBROUTINE RMODES (X,V,W,P,OMEG,FREQ,NM,NX,NDIM)
      REAL X(1),V(NDIM,1),W(NDIM,1),P(NDIM,1),OMEG(1),FREQ(1)
      IMO=1
      IM1 = MINO(NM,3)
   75 PRINT 1025,(OMEG(I),I=IMO,IM1)
 1025 FORMAT (//13X,8HOMEGA = , F18.3,2F39.3)
      PRINT 1026,(FREQ(I),I=IMO,IM1)
 1026 FORMAT(/ 13X,7HFREQ  = , F18.3,2F39.3)
      PRINT 1027
 1027 FORMAT (/2X, 11HI     STA    ,3(39H           V             W
     1P      )/)
      DO 80 I = 1,NX
   80 PRINT 1030, I,X(I),(V(J,I),W(J,I),P(J,I),J=IMO,IM1)
 1030 FORMAT (1X,I2,0P F10.3,3(3X,1P 3E12.3))
      IF(IM1.GE.NM) GO TO 90
      IMO=IMO+3
      IM1= MINO(NM,IM1+3)
      IF(IMO.EQ.4.OR.IMO.EQ.10.OR.IMO.EQ.16) GO TO 75
      PRINT 1020
 1020 FORMAT (1H1  50X, 11HMODE SHAPES  //)
      GO TO 75
   90 RETURN
      END
```

```
      SUBROUTINE ERR(N,I)
C     I = 0, TERMINATES RUN        I NE 0 WARNING ONLY, PRINTS I
      PRINT 10,N
   10 FORMAT(//10X,17H*** ERROR NUMBER   ,I5,5H  ***  )
      IF (I.NE.0) GOTO 20
      CALL EXIT
   20 PRINT 30,I
   30 FORMAT (20X,20H*** WARNING ONLY ***   ,I5//)
      RETURN
      END
```

```
      SUBROUTINE ORTH(V,W,P,M,ME,MET,MK,NM,NX,MDIM,GMASS,OCHECK,MCDIM,IP)
C
C                    PERFORMS ORTHOGONOLITY CHECK
C                    GMASS ARE DIAGONAL ELEMENTS
C                    OCHECK IS NORMALIZED BY DIVIDING ROW,COL BY SQRT
C                      OF DIAGONAL
C
C          IP.NE.0      GMASS,OCHECK ARE PRINTED
C          IP.EQ.2      MODES ARE NORMALIZED  (GEN MASS = 1.0)
C
      REAL V(MDIM,1),W(MDIM,1),P(MDIM,1),ME(1),MET(1),MK(1),GMASS(1),
     1  OCHECK(MCDIM,1),M(1)
      DO 20 I = 1,NM
      DO 20 J = 1,NM
      OCHECK(I,J) = 0
      DO 20 L = 1,NX
   20 OCHECK(I,J) = OCHECK(I,J)+V(I,L)*M(L)*V(J,L)-P(I,L)*MET(L)*V(J,L)
     1  +W(I,L)*M(L)*W(J,L)+P(I,L)*ME(L)*W(J,L)-V( I,L )*MET(L)*P(J,L)
     2  +W(I,L)*ME(L)*P(J,L)+P(I,L)*MK(L)*P(J,L)
      DO 30 I=1,NM
      GMASS(I) = OCHECK(I,I)
      SQ = SQRT(GMASS(I))
      IF(IP.NE.2) GO TO 29
      DO 25 L=1,NX
      V(I,L)=V(I,L)/SQ
      W(I,L)=W(I,L)/SQ
   25 P(I,L)=P(I,L)/SQ
   29 DO 30 J = 1,NM
      OCHECK(I,J) = OCHECK(I,J)/SQ
   30 OCHECK(J,I) = OCHECK(J,I)/SQ
      IF(IP.EQ.0) RETURN
      PRINT 100,(GMASS(I),I=1,NM)
  100 FORMAT (20X,40HDIAGONAL ELEMENTS OF ORTHO CHECK MATRIX   /
     1  (10X,1P8E14.3))
      PRINT 200
  200 FORMAT (//20X,30HNORMALIZED ORTHO CHECK MATRIX  /)
      DO 40 I = 1,NM
   40 PRINT 300, (OCHECK(I,J),J=1,NM)
  300 FORMAT (/2X,16F8.3)
      IF(IP.EQ.2) PRINT 350
  350 FORMAT (1H1,30X,16HNORMALIZED MODES  //)
      RETURN
      END
```

```
      SUBROUTINE INVRS (B,N,A,D, IROW, ICOL,NRW,NCL)
C     A = INVERSE OF B          B UNDISTURBED
C       VARIABLE DIMENSIONS    NCL MUST BE AT LEAST ONE GREATER THAN NRW
C       NRW MUST BE AT LEAST EQUAL TO N
C       IROW, ICCL ARE VECTORS OF LENGTH NCL
      REAL A(NRW,NCL),B(NRW,NCL),D(NRW,NCL)
      INTEGER IROW(NCL),ICOL(NCL)
      DO 1 I=1,N
      DO 1 J=1,N
    1 A(I,J)=B(I,J)
      M=N+1
      DO 7  I=1,N
      IROW(I)=I
    7 ICOL(I)=I
      DO 20 K=1,N
      AMAX= A(K,K)
      DO 10 I=K,N
      DO 10 J=K,N
      IF(ABS( A(I,J))-ABS(AMAX))10,9,9
    9 AMAX= A(I,J)
      IC=I
      JC=J
   10 CONTINUE
      KI=ICOL(K)
      ICOL(K)=ICOL(IC)
      ICOL(IC)=KI
      KI=IROW(K)
      IROW(K)=IROW(JC)
      IROW(JC)=KI
      IF(AMAX) 11,12,11
   12 PRINT 13
   13 FORMAT(' SOLUTION OF MATRIX NOT POSSIBLE')
      GO TO 100
   11 DO 14 J=1,N
      E=A(K,J)
      A(K,J)=A(IC,J)
   14 A(IC,J)=E
      DO 15 I=1,N
      E=A(I,K)
      A(I,K)=A(I,JC)
   15 A(I,JC)=E
      DO 16 I=1,N
      IF(I-K) 18,17,18
   17 A(I,M)=1.
      GO TO 16
   18 A(I,M)=0.
   16 CONTINUE
      PVT=A(K,K)
      DO 8 J=1,M
    8 A(K,J)=A(K,J)/PVT
      DO 19 I=1,N
      IF(I-K)21,19,21
   21 AMULT=A(I,K)
      DO 22 J=1,M
```

```
   22 A(I,J)=A(I,J)-AMULT*A(K,J)
   19 CONTINUE
      DO 20 I=1,N
   20  A(I,K)=A(I,M)
      DO 25 I=1,N
      DO 24 L=1,N
      IF(IROW(I)-L)24,23,24
   24 CONTINUE
   23 DO 25 J=1,N
   25 D(L,J)=A(I,J)
      DO 26 J=1,N
      DO 28 L=1,N
      IF(ICOL(J)-L) 28,29,28
   28 CONTINUE
   29 DO 26 I=1,N
   26 A(I,L)=D(I,J)
  100 RETURN
      END
```

```
      SUBROUTINE MXV(A,B,C,M,N,NDIM,ICONT)
C
C         MATRIX TIMES VECTOR   A(M)=B(M,N)*C(N)          FOR ICONT = 0
C                                               +A(M)     FOR ICONT =1
C
      DIMENSION A(1),B(NDIM,1),C(1)
      DO 10 I=1,M
      IF(ICONT.EQ.0) A(I)=0
      DO 10 J=1,N
   10 A(I)=A(I)+B(I,J)*C(J)
      RETURN
      END
```

```
      SUBROUTINE ERRA( A,PCT,PCTB,NJ,NM,IX,NDIM )
C     A BIAS ERROR PCTB(RATIO) ON AMPLITUDE AND A UNIFORMLY DISTRIBUTED
C     RANDOM ERROR HAVING A +/- MAXIMUM OF PCT(RATIO) ON AMPLITUDE
C
      DIMENSION A(NDIM,1)
      IF(PCT.NE.0 ) GO TO 110
  100 IF( PCTB.EQ.0 ) GO TO 130
  110 DO 120 K=1,NM
      DO 120 I=1,NJ
      CALL RANDU( IX,IY,YFL )
      IX=IY
      E=1.+2.*PCT*(YFL-.5)+PCTB
  120 A(K,I)=A(K,I)*E
  130 RETURN
      END
```

```
      SUBROUTINE RANDU ( IX,IY,YFL )
C     USAGE
C     CALL RANDU ( IX,IY,YFL )
C
C
C     COMPUTES UNIFORMLY DISTRIBUTED RANDOM REAL NUMBERS BETWEEN
C     0 AND 1.0 AND RANDOM REAL INTEGERS BETWEEN 0 AND 2**31.
C
C
C     EACH ENTRY USES AS INPUT AN INTEGER RANDOM NUMBER AND
C     PRODUCES A NEW INTEGER AND REAL RANDOM NUMBER.
C
C     VARIABLES
C     IX= FOR THE FIRST ENTRY THIS MUST CONTAIN ANY ODD INTEGER NUMBER
C     WITH NINE OR LESS DIGITS.  AFTER THE FIRST ENTRY IX SHOULD BE
C     THE PREVIOUS VALUE OF IY COMPUTED BY THIS SUBROUTINE
C
C     IY= A RESULTANT INTEGER RANDOM NUMBER REQUIRED FOR THE NEXT ENTRY
C     TO THIS SUBROUTINE. THE RANGE OF THIS NUMBER IS BETWEEN 0 AND 2**31
C
C     YFL= THE RESULTANT UNIFORMLY DISTRIBUTED ,FLOATING POINT,RANDOM
C     NUMBER IN THE RANGE 0 TO 1.0
C
C
      IY=IX*65539
      IF(IY) 100,110,110
  100 IY=IY+2147483647+1
  110 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
```

APPENDIX D

NORMAL MODES AND NATURAL FREQUENCIES OBTAINED FROM VACUUM WHIRL DATA*

INTRODUCTION

The rotor was forced vertically along the axis of rotation with no other external forces. The natural frequencies of the symetric flapping modes with infinite hub impedance are the driving point antiresonant frequencies along the rotational axis. These frequencies were identified and a modal analysis done to determine the mode shapes using strain/hub acceleration transmissibility in the following manner.

Strain readings, calibrated in terms of bending moments, and hub vertical accelerations were recorded simultaneously on analog tape at the selected rotational speeds of 0, 5.24, 10.47 and 15.71 rod/sec. (0, 50, 100 and 150 RPM). The time domain hub acceleration signal was fed from the tape reader to the force input of a Fast Fourier Transform Digital Signal Analyzer, type Hewlett Packard 5420, while the time domain strain signal from the j^{th} station along the blade was fed to the response input of the Digital Signal Analyzer for stations j = 1 to j = 12 at each of the rotor RPM settings. Over a narrow band of frequency covering each hub antiresonant frequency, determined approximately from broad band analysis in which the hub driving point antiresonant frequencies appear in the Fourier Transform in the form of natural frequencies, a Fourier Transform of 2^8 frequency line was obtained for each strain/hub acceleration transfer function. The narrow band data were then analyzed for global properties.

The transmissibility residues for the 12 blade stations in a given mode were found to be complex, due to the nature of the transfer function, but complex normalization showed the bending moment modes to be real (classical). The deflection modes were obtained from the bending moment modes by simple double trapezoidal integration of the curvature from the root to the tip.

---

*The tests from which this data were obtained are described in Ref. 9

The Antiresonant Method. - It is obviously impossible to achieve infinite terminating impedance in practice but the modal effects of infinite terminating impedance along a single motion coordinate can be obtained quite accurately through antiresonance theory even though the terminating coordinate never reaches absolutely zero motion. It never reaches absolute zero because, and only because, in this case, the rotor dissipates energy to a sink. The nature of this energy dissipation, called "damping", is not known. If the rotor were undamped the vertical motion along the axis of rotation, the coordinate of sole external excitation, would be absolutely zero at the natural frequencies of the symmetric flapping modes of infinite hub impedance regardless of the actual hub impedance. The sum of the inertial forces of the undamped rotor acting vertically on the hub would, at these frequencies, be exactly equal to the sole excitation force acting vertically at the hub, regardless of its magnitude (within the linear range) or phasing to any base, in the steady state. This is the principle of the undamped vibration absorber of 1909; its notable early 19th century predecessor, the una corda or "soft" pedal on aftersound of the concert grand piano; the Thearle invention of the 1930 on which shaft and turbine balancing machines are based; the 1947 method of stabilization by Thor which made spin dry home washing machines practical and the many obvious helicopter applications along with the less obvious one recently in which a military helicopter initially had little pilot seat vibration at the expense of intolerable tail fatigue.

Mathematically, a damped antiresonance is merely a zero of zero magnitude. In the case at hand the single excitation along the axis of rotation is unknown (because the measured applied force in the rig is below the hub with an intervening unknown impedance) but as it is the same for hub vertical acceleration and blade bending moment the quotient of blade bending mobility and hub acceleration mobility involves cancellation of the pole roots leaving the denominator a polynomial whose roots are hub driving point zeros the undamped parts of which are the desired antiresonances. These can be determined from the Fourier Transform of the transfer function as will be shown below.

From elementary considerations of complex variable theory it is easily seen that the residues are without physical significance in themselves because the polynomial quotient has an arbitrary factor. For this reason one cannot use this procedure to obtain physically meaningful orthonormal modes. However, in normalizing on a station on the blade the arbitrary factor of the multiplying factor cancels, being the same for each station, and a valid bending-moment mode shape can be readily obtained. That is, the validity of the quotient of residues is maintained. This is precisely the same as ratioing the vectorial chords of the Nyquist plots of each blade station between given frequencies in the zero root range of the hub mobility to that of any given blade station.

Because the complex chordal vectors between given frequencies are parallel to the modal diameter of any transmissibility having the hub driving point product of roots of the zeros in the denominator and because the length of such chords are necessarily proportional to their associated diameters each it follows that the ratio of the complex chordal vectors is the same as that of the complex diametral vectors. In other words, if one were to transfer the Nyquist axes to an origin corresponding to the antiresonant frequency, do a bilinear transformation and ratio the distances of the resulting lines to the origin for any station to a given blade station one would find a canonical invariance of the polynomial in the poles and the frequency invariant factor for any given pole.

Finding the Natural Frequency. - Most often one will find three peaks in mobility associated with a mode, two in the real and one in the imaginary or vice versa. If the angle of a complex mode is near 45°, 135°, 225° or 315° there will be only two sharp peaks, one in the real and one in the imaginary.

The following is done for acceleration mobility. q refers to a frequency in the imaginary and p to a frequency in the real. The subscript x refers to an acceleration mobility maximum and m to an acceleration mobility minimum.

If the modal angle is in the range from about -40° to about +40° or narrower there will be a maximum in the acceleration imaginary and a minimum and maximum in the real.

$$2\, q_x^2 - \frac{p_x^2 + p_m^2}{2} = \Omega^2 \, [1 + g(2 \tan \phi/2 - \tan \phi)] \tag{D-1}$$

Let the natural frequency be approximated by

$$\Omega^2 \cong 2\, q_x^2 - \frac{p_x^2 + p_m^2}{2} \tag{D-2}$$

TABLE D-I. ERROR IN Ω^2 BY EQUATION (D-2)

ϕ	g = .02	g = .05	g = .10	g = .20
40°	0.22%	0.56%	1.11%	2.22%
30°	0.08%	0.21%	0.41%	0.83%
20°	0.02%	0.06%	0.11%	0.23%
10°	0.003%	0.006%	0.01%	0.03%
0°	0%	0%	0%	0%
-10°	0.003%	0.006%	0.01%	0.03%
-20°	0.02%	0.06%	0.11%	0.23%
-30°	0.08%	0.21%	0.41%	0.83%
-40°	0.22%	0.56%	1.11%	2.22%

If the modal angle is in the range from 50° to 130° one will observe a p_m, q_x and q_m with the identical errors over the range as given in Table D-I by adding 90° to the angle. Similarly for the other cases.

$$\Omega^2 \cong 2\, p_m^2 - \frac{q_x^2 + q_m^2}{2} \qquad \text{(D-3)}$$

$$\Omega^2 \cong 2\, q_m^2 - \frac{p_x^2 + p_m^2}{2} \qquad 140° \text{ to } 220° \qquad \text{(D-4)}$$

$$\Omega^2 \cong 2\, p_x^2 - \frac{q_x^2 + q_m^2}{2} \qquad 230° \text{ to } 310° \qquad \text{(D-5)}$$

Equation D-2, D-3, D-4 and D-5 involve frequencies merely as twice the square of the single peak frequency less half the sum of the squares of the double peak frequencies.

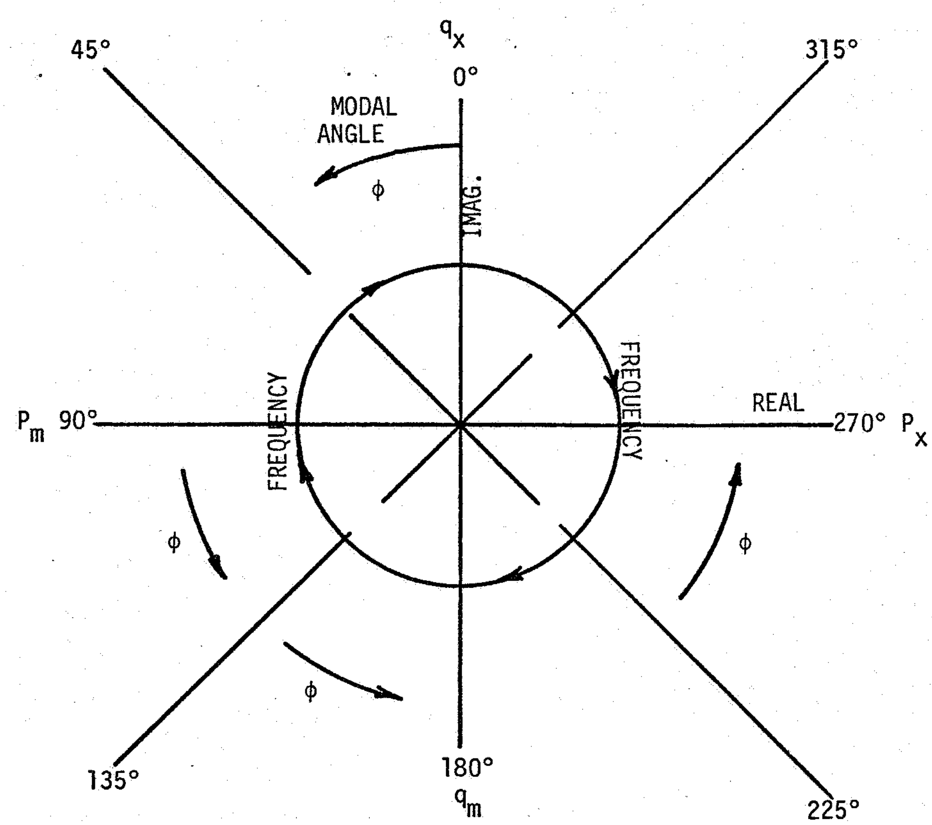


Figure D-1. A diagram of acceleration mobility peak frequencies.

Two Peaks Only - Natural Frequency. - If there is only one real and one imaginary peak associated with a mode, the modal angle must be near 45° + n*90° for n = 0, 1, 2, 3 as seen in Figure D-1.

For n = 0

$$q_x^2 + p_m^2 = \Omega^2 \left[2 + g \left(\tan \phi/2 - \cot \frac{\phi+\pi/2}{2}\right)\right] \tag{D-6}$$

Let the natural frequency be approximated by

$$\Omega^2 = \frac{q_x^2 + p_m^2}{2} \tag{D-7}$$

TABLE D-II. INHERENT ERROR IN EQUATION 7. $\frac{\Omega^2 - \Omega^2}{2}$

ϕ	g = .02	g = .05	g = .10	g = .20
nx90° + 35°	0.206%	0.516%	1.04%	2.096%
nx90° + 40°	0.102%	0.257%	0.514%	1.034%
nx90° + 45°	0%	0%	0%	0%
nx90° + 50°	0.102%	0.257%	0.514%	1.034%
nx90° + 55°	0.206%	0.516°	1.04%	20.096%

The actual inherent error in natural frequency is about half those in Table D-II.

Local Spectrum Analysis of a Complex Mode Given the Natural Frequency

This procedure may be used over any portion of the modal arc. In an acceleration mobility Kennedy-Pancu plot let N be the natural frequency and f_1 be any frequency on the modal arc selected by the operator. The chord from frequency f_1 at $N\sqrt{1 - b}$ to frequency $f_2 = N\sqrt{1 + b}$ over an arc of 180° or less is perpendicular to a diameter through the natural frequency, b is an arbitrary number less than unity. See Figure D-2.

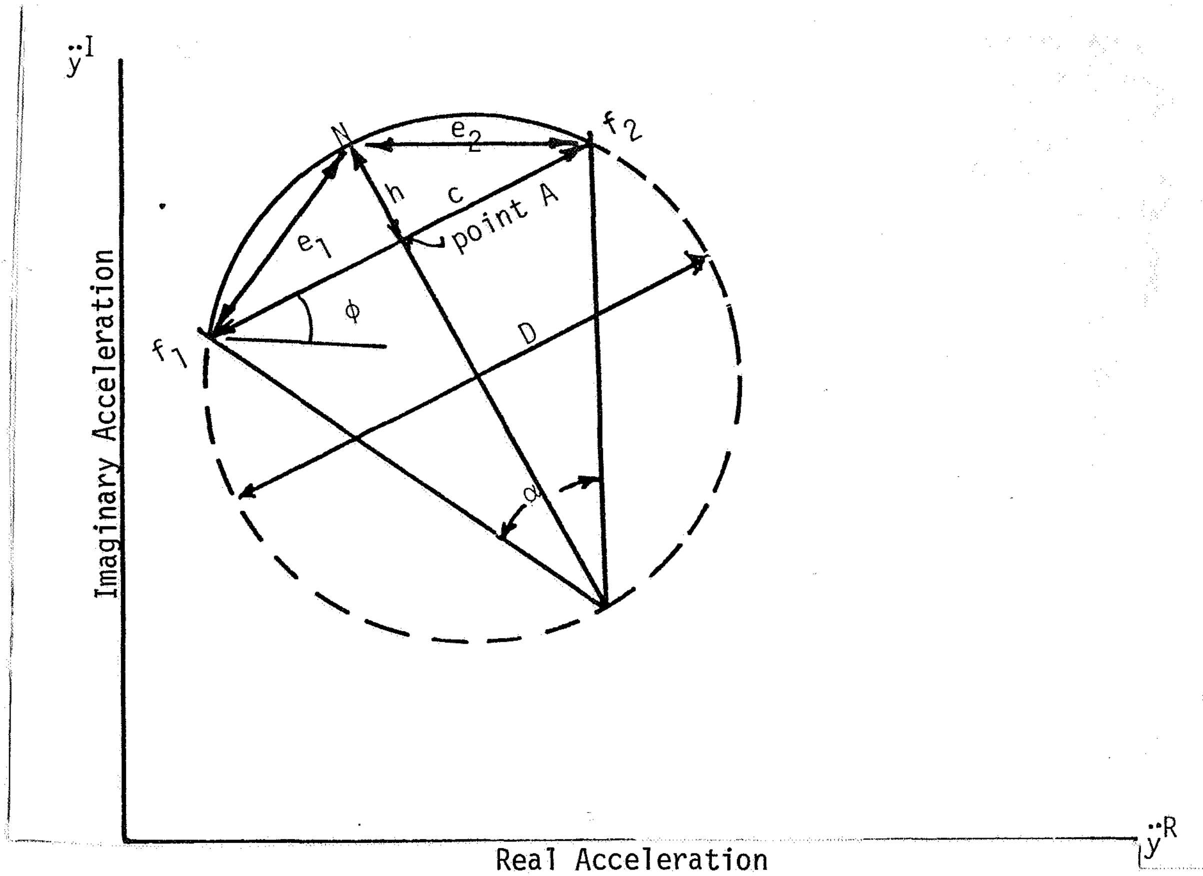


Figure D-2. Nyquist Plot

The modal angle is ϕ.

$$\frac{c/2}{D-h} = \tan\frac{\alpha}{2} \tag{D-8}$$

For practical purposes (see the mensuration section of any standard engineering handbook)

$$\frac{c/2}{D-h} = \frac{2h}{c} = \tan\frac{\alpha}{2} \tag{D-9}$$

and

$$c = D\sin\alpha. \tag{D-10}$$

$$c = \sqrt{\left(Y_2^R - Y_1^R\right)^2 + \left(Y_2^I - Y_1^I\right)^2} \tag{D-11}$$

$$Y_A^R = \left(Y_2^R + Y_1^R\right)/2, \left(Y_A^I = Y_2^I + Y_1^I\right)/2 \quad \text{(D-12)}$$

$$h = \sqrt{\left(Y_N^R - Y_A^R\right)^2 + \left(Y_N^I - Y_A^I\right)^2} \quad \text{(D-13)}$$

$$e_1 = \sqrt{\left(Y_N^R - Y_1^R\right)^2 + \left(Y_N^I - Y_1^I\right)^2} \quad \text{(D-14)}$$

$$e_2 = \sqrt{\left(Y_N^R - Y_2^R\right)^2 + \left(Y_N^I - Y_2^I\right)^2} \quad \text{(D-15)}$$

If $e_2/e_1 \cong 1.0$ then N is not the natural frequency for points 1 and 2 on the modal arc. If $e_2/e_1 < 1.0$ then the natural frequency is less than N, if $e_2/e_1 > 1.0$ then the natural frequency is greater than N.

$$\frac{f_2^2}{N^2} = 1 + g \tan \frac{\alpha}{2} \quad \text{(D-16)}$$

$$\frac{f_1^2}{N^2} = 1 - g \tan \frac{\alpha}{2}$$

$$\frac{f_2^2 - f_1^2}{N^2} = 2 g \tan \frac{\alpha}{2}$$

$$g = \frac{1}{2} \frac{f_2^2 - f_1^2}{N^2 \tan \alpha/2} \quad \text{(D-17)}$$

The natural frequencies determined from HP 5420 data using Equations D-2 through D-5 are shown in Table D-III in comparison to the natural frequencies found by NASA. The strain data for 100 RPM was quite noisy and was therefore not analyzed. Figures D-3 through D-11 show the bending moment normal modes and Figures D-12 through D-20 show the normalized deflection mode shapes.

TABLE D-III. NATURAL FREQUENCIES Hz (cassette number, record number)

	0 rad/s (0 RPM)	5.24 rad/s (50 RPM)	15.71 rad/s (150 RPM)
2nd Flapping	8.2 NASA 8.16 (1,1) 8.18 (2,41)	8.7 NASA 8.46 (1,37) 8.47 (3,19)	10.8 NASA 10.78 (2,23) 10.80 (3,47)
3rd Flapping	21.8 NASA 21.71 (1,10) 21.82 (3,1) 21.81 (2,48)	22.2 NASA 21.93 (2,1) 21.97 (3,26) 21.93 (1,46)	24.4 NASA 24.24 (2,23)
4th Flapping	41.2 NASA 41.66 (1,19) 41.73 (3,5)	42.0 NASA 41.92 (2,5) 41.99 (3,33)	44.1 NASA 44.18 (5,41) 44.19 (5,44) 44.20 (5,47)
1st Torsion	26.6 NASA 26.41 (1,28)	27.4 NASA 27.02 (2,14) 27.02 (3,40)	28.3 NASA 28.36 (12,19) 28.36 (12,20)

RECOMMENDATIONS

If this test were to be repeated it would be useful to measure strain on the hub near the center of rotation to provide the initial condition for integration of strains and it would be practical to calibrate in terms of the differential strains of the bending bridges, instead of bending moment, to eliminate the need for theoretical EI values in the integration.

In the photographic method of obtaining mode shapes the assumption is that the modes are uncoupled, that is, that the shaking excites only one mode. With that assumption, a promising method of obtaining rotating mode shapes is that pioneered by Hassal[2] of the Royal Aircraft Establishment:

$$\left\{q^{(R)}\right\} = \left[\Phi\right]\left[\Phi^{(\varepsilon)}\right]^{+}\left\{\varepsilon^{(R)}\right\}$$

where $\varepsilon^{(R)}$ is the vector of blade strains measured in rotation

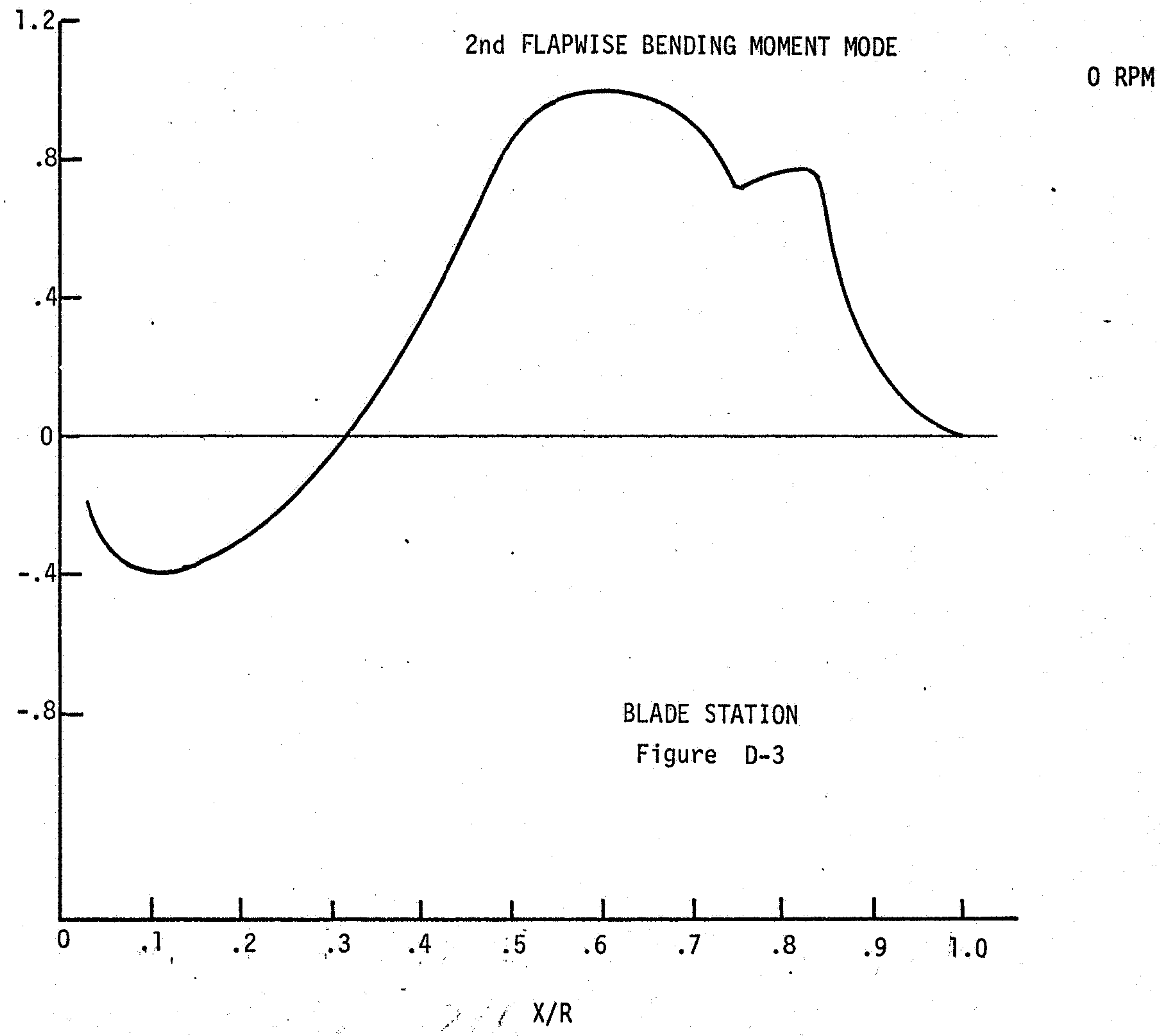
Φ is the matrix of nonrotating normalized normal translational modes

$\Phi^{(\varepsilon)}$ is the matrix of nonrotating normalized normal strain modes

Normalization of the Left Hand Side at a natural frequency, given very light damping and widely separated natural frequencies, would be the rotating normal mode. Φ and $\Phi^{(\varepsilon)}$ are obtained in a nonrotating shake test after which the accelerometers are removed from the blade and have the same number of columns but not necessarily the same number of rows. The strains used need not be directly related to bending moments.

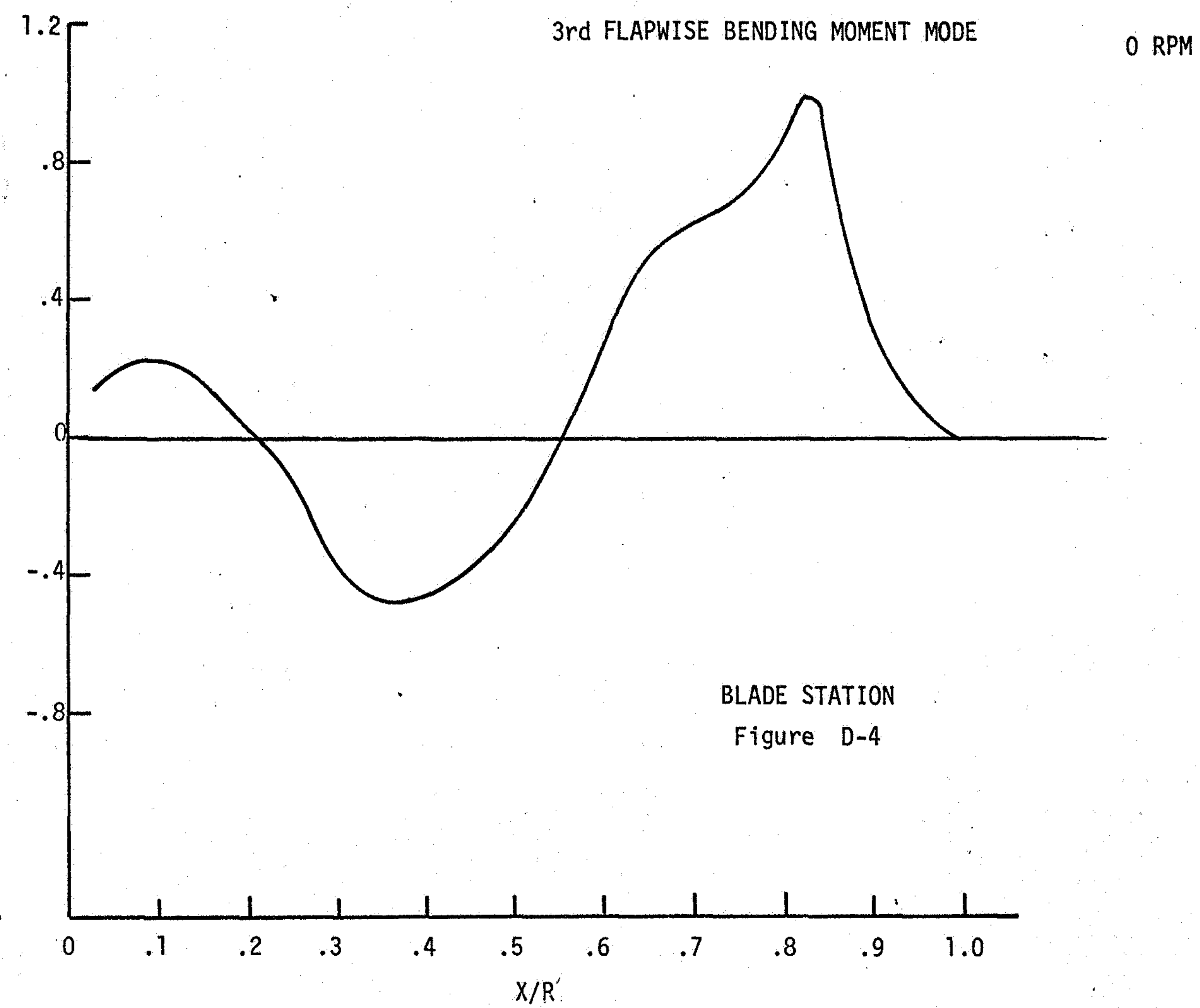
CONCLUSIONS

The rotating and nonrotating modes in flatwise bending for the cantilever condition were found to be real. The natural frequencies found in bending moment modal analysis agreed closely with those found by other methods.



2nd FLAPWISE BENDING MOMENT MODE
0 RPM
1.2
.8
.4
0
-.4
-.8
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
X/R
BLADE STATION

Figure D-3



BLADE STATION
Figure D-4

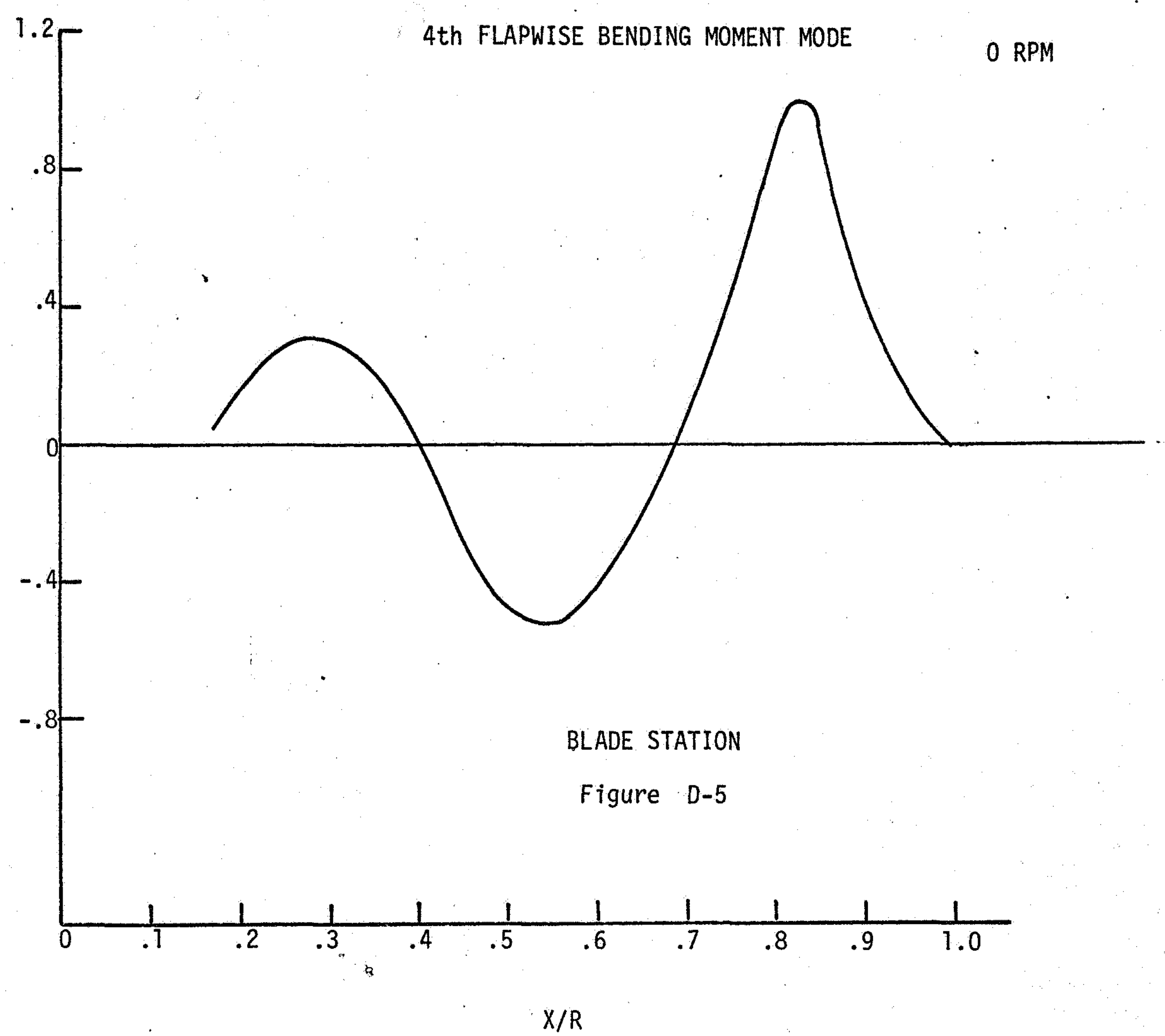

4th FLAPWISE BENDING MOMENT MODE
0 RPM
1.2
.8
.4
0
-.4
-.8
BLADE STATION
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
X/R

Figure D-5

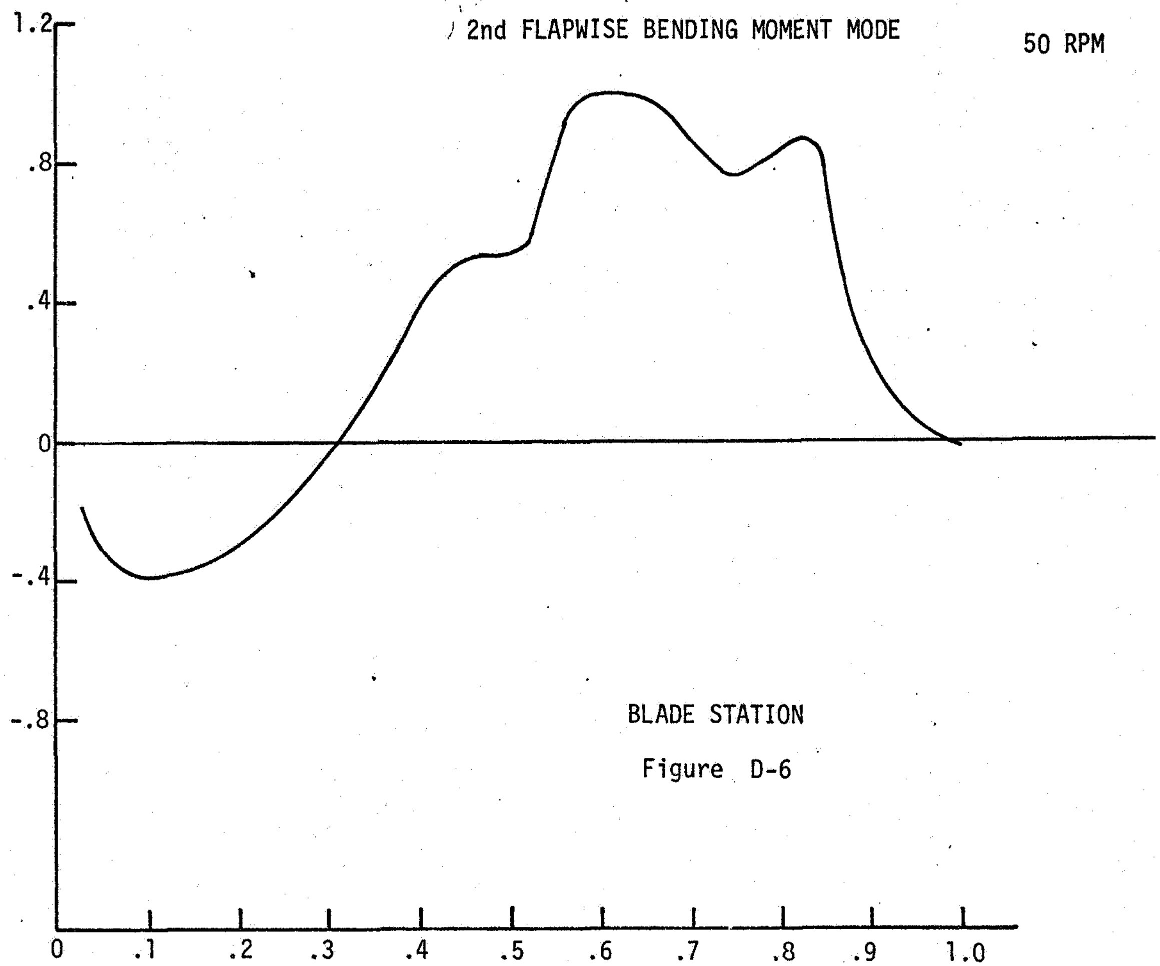

2nd FLAPWISE BENDING MOMENT MODE
50 RPM
1.2
.8
.4
0
-.4
-.8
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
BLADE STATION

Figure D-6

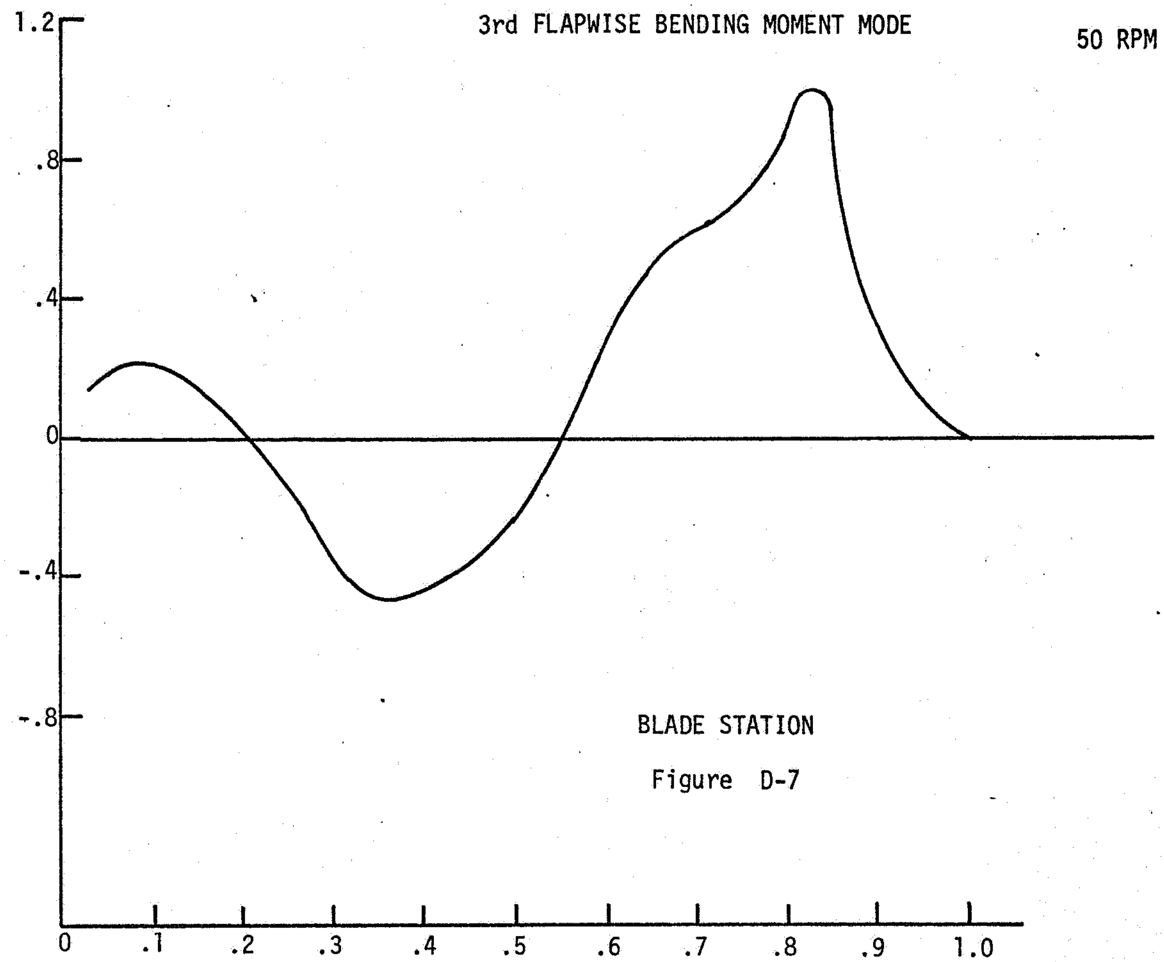

3rd FLAPWISE BENDING MOMENT MODE
50 RPM
1.2
.8
.4
0
-.4
-.8
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
BLADE STATION

Figure D-7

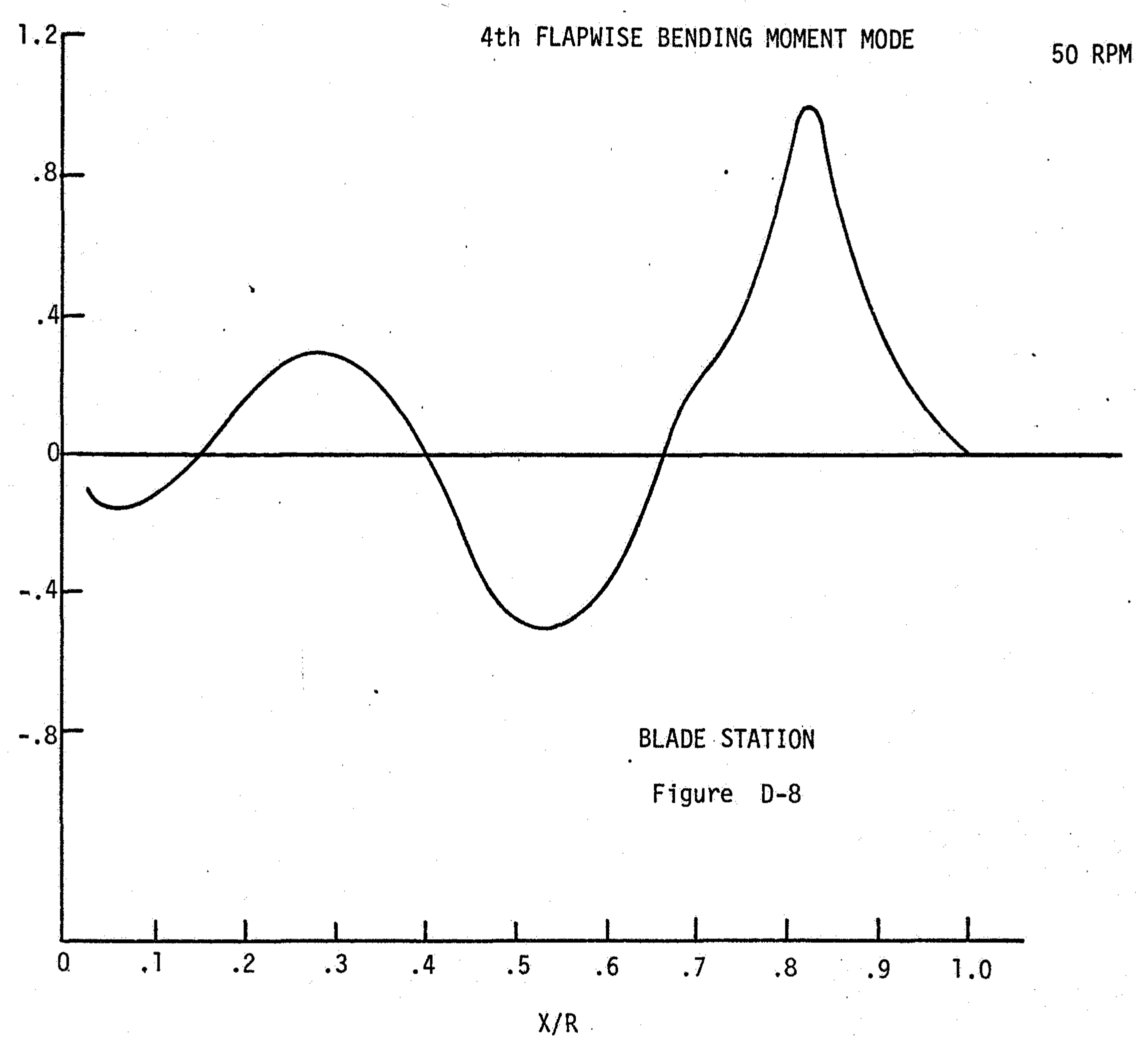

4th FLAPWISE BENDING MOMENT MODE
50 RPM
1.2
.8
.4
0
-.4
-.8
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
BLADE STATION
X/R

Figure D-8

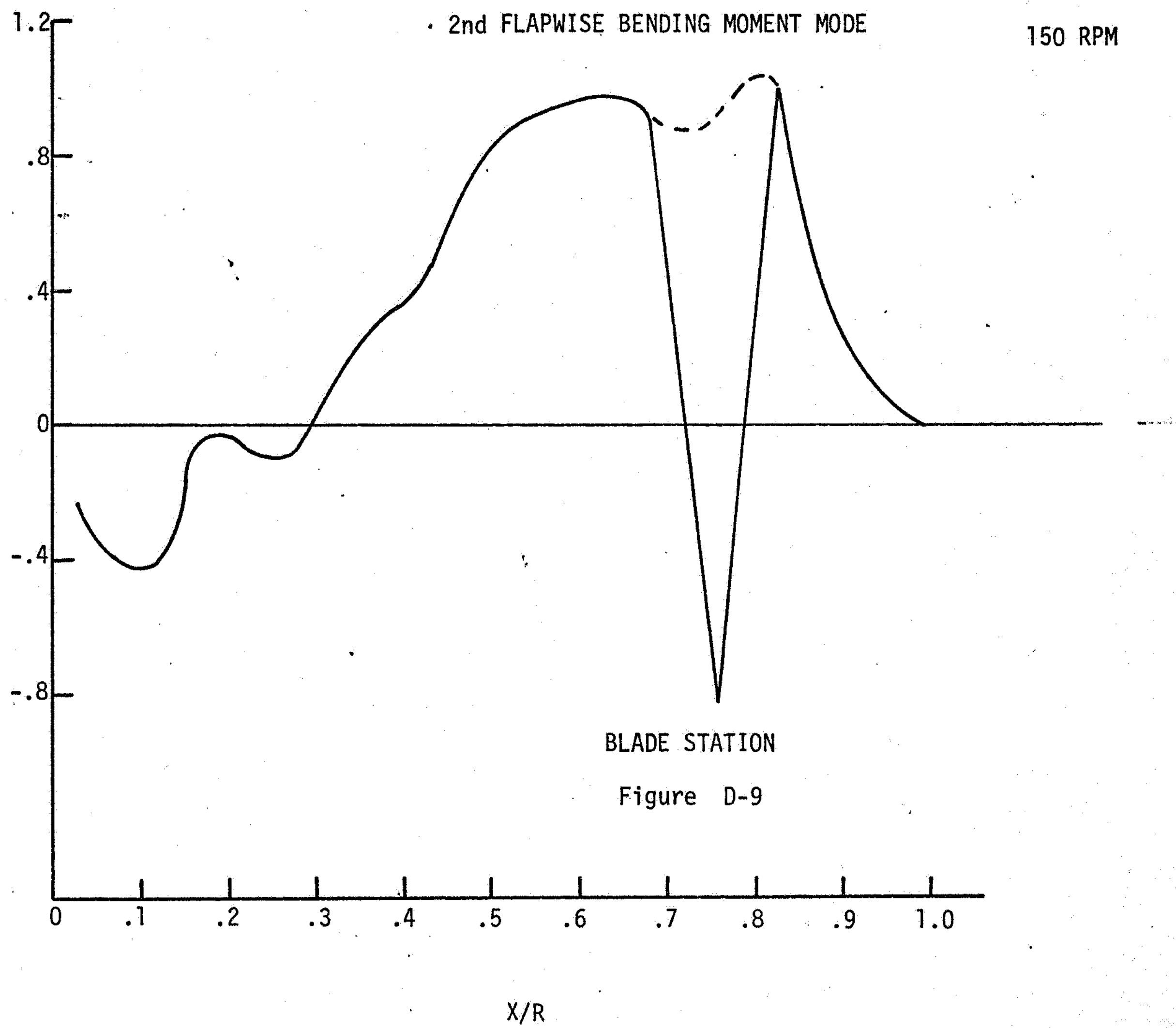
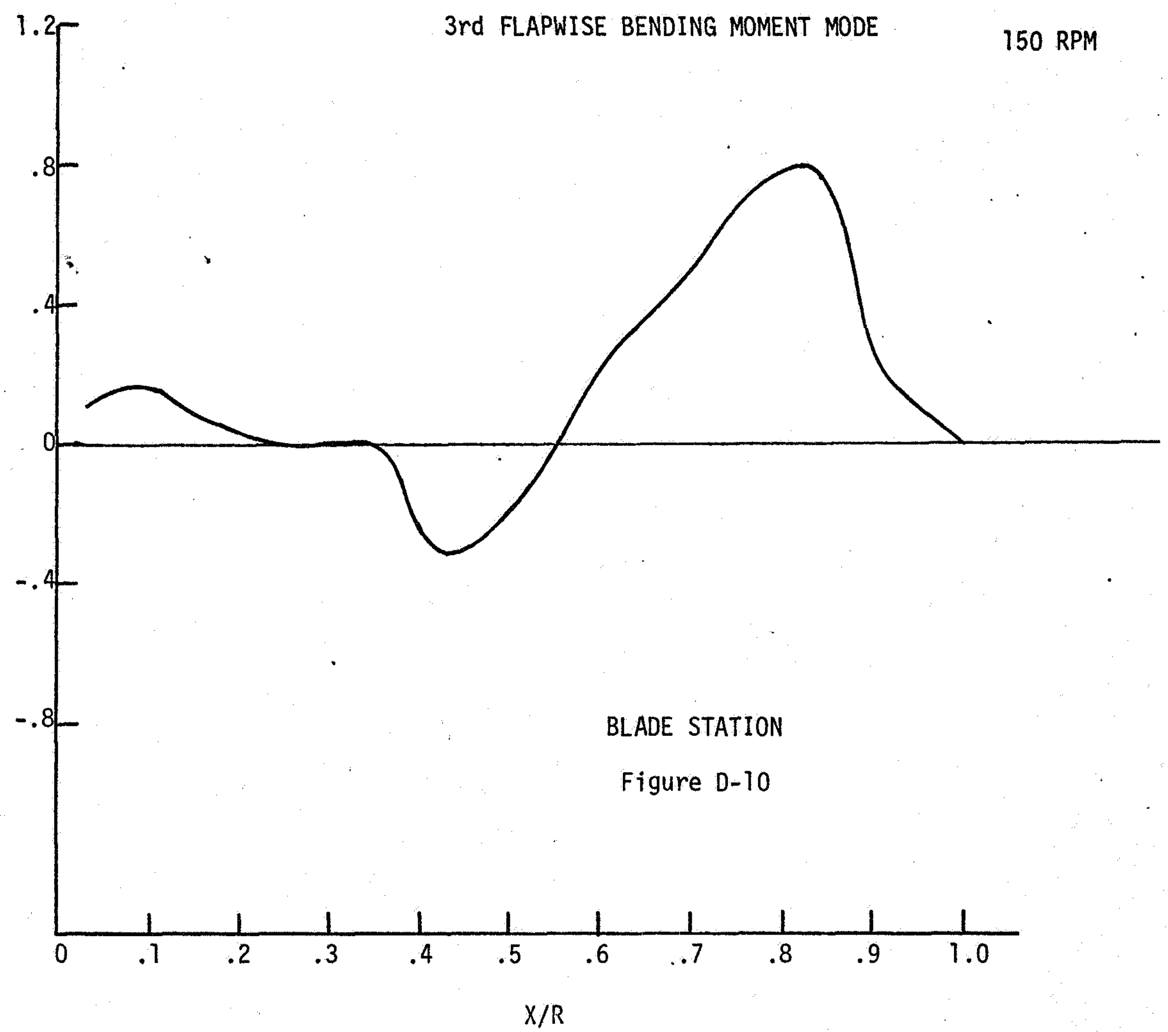
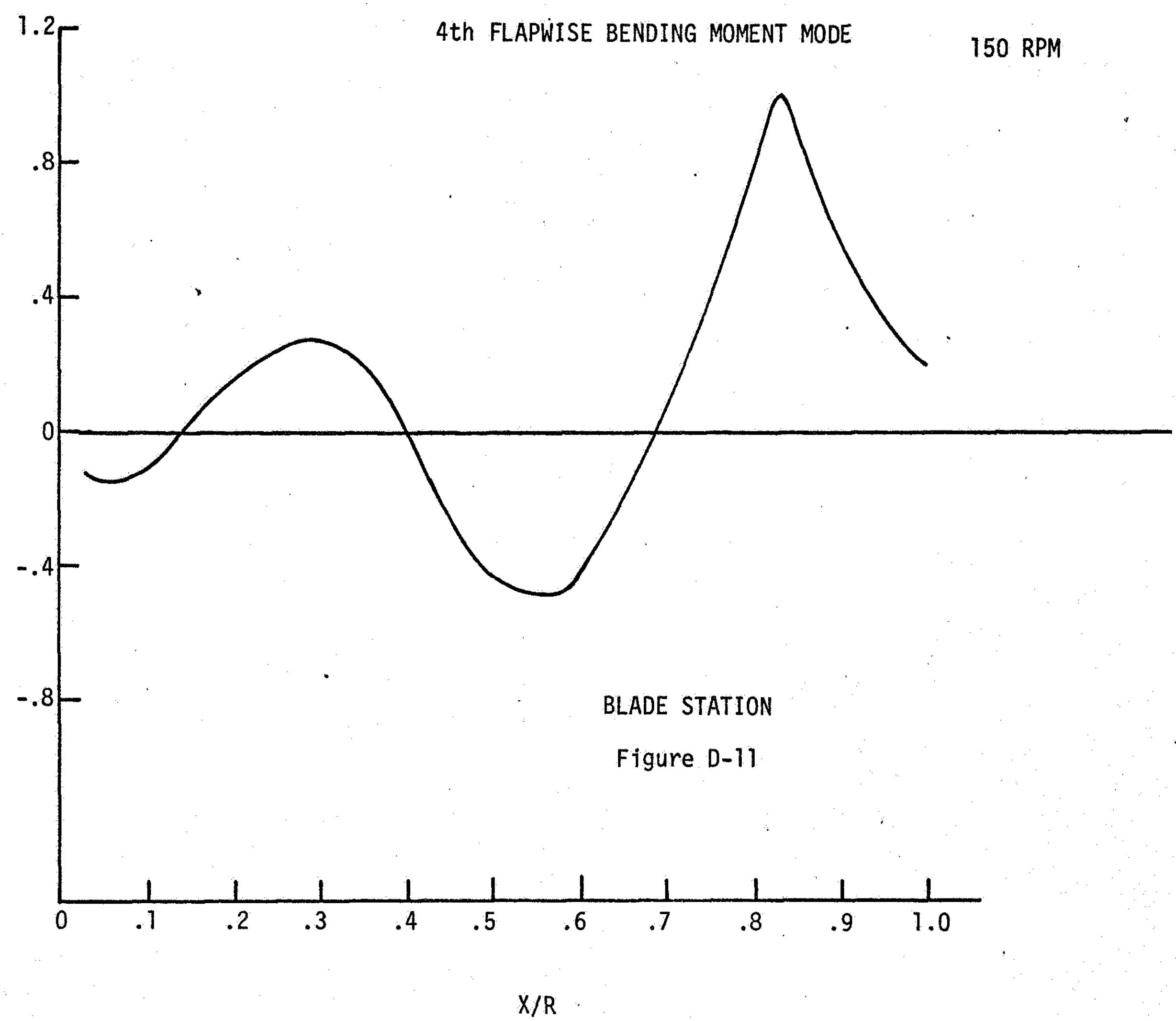

2nd FLAPWISE BENDING MOMENT MODE
150 RPM
1.2
.8
.4
0
-.4
-.8
BLADE STATION
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
X/R

Figure D-9



BLADE STATION

Figure D-10



BLADE STATION

Figure D-11

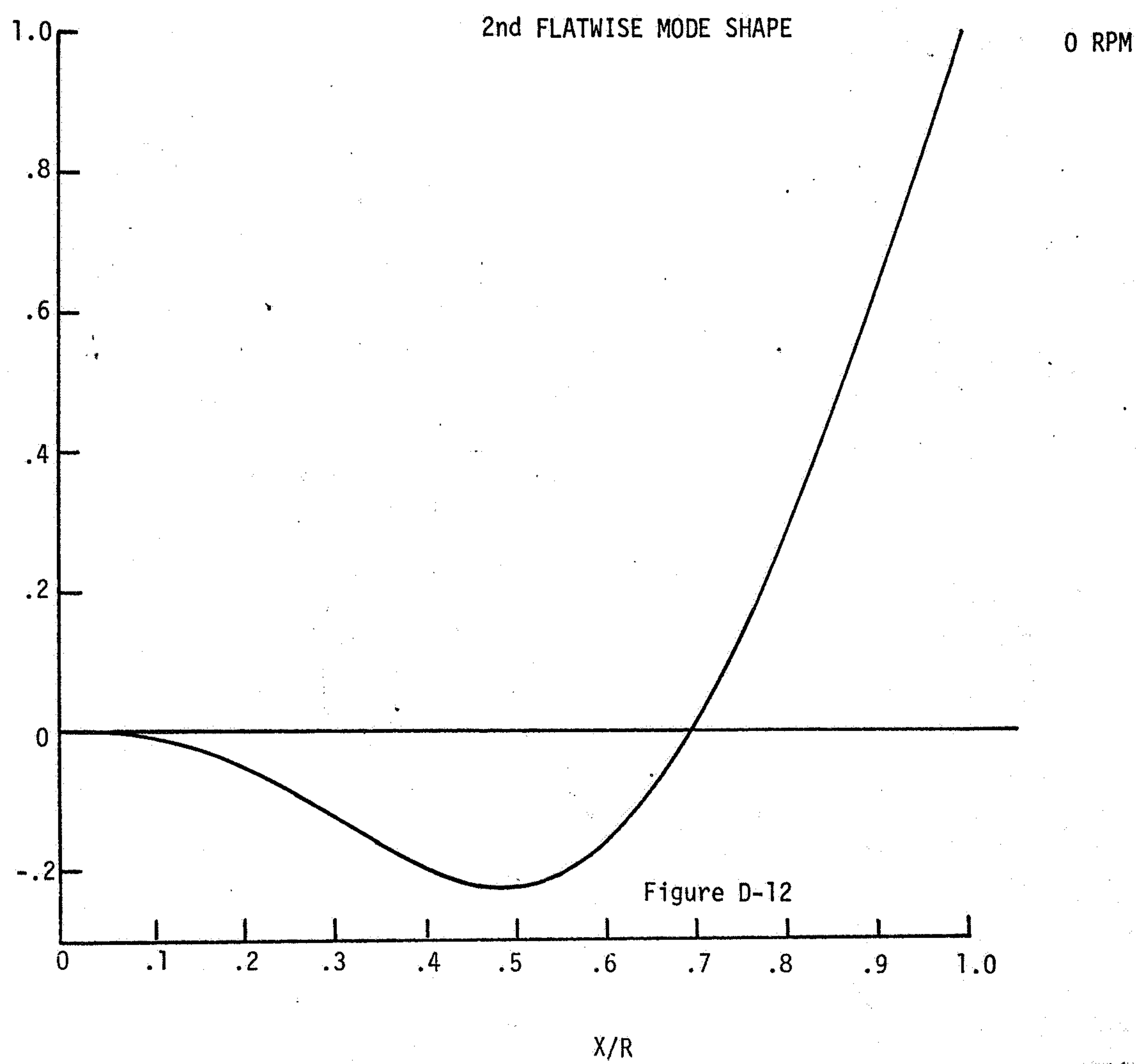


Figure D-12

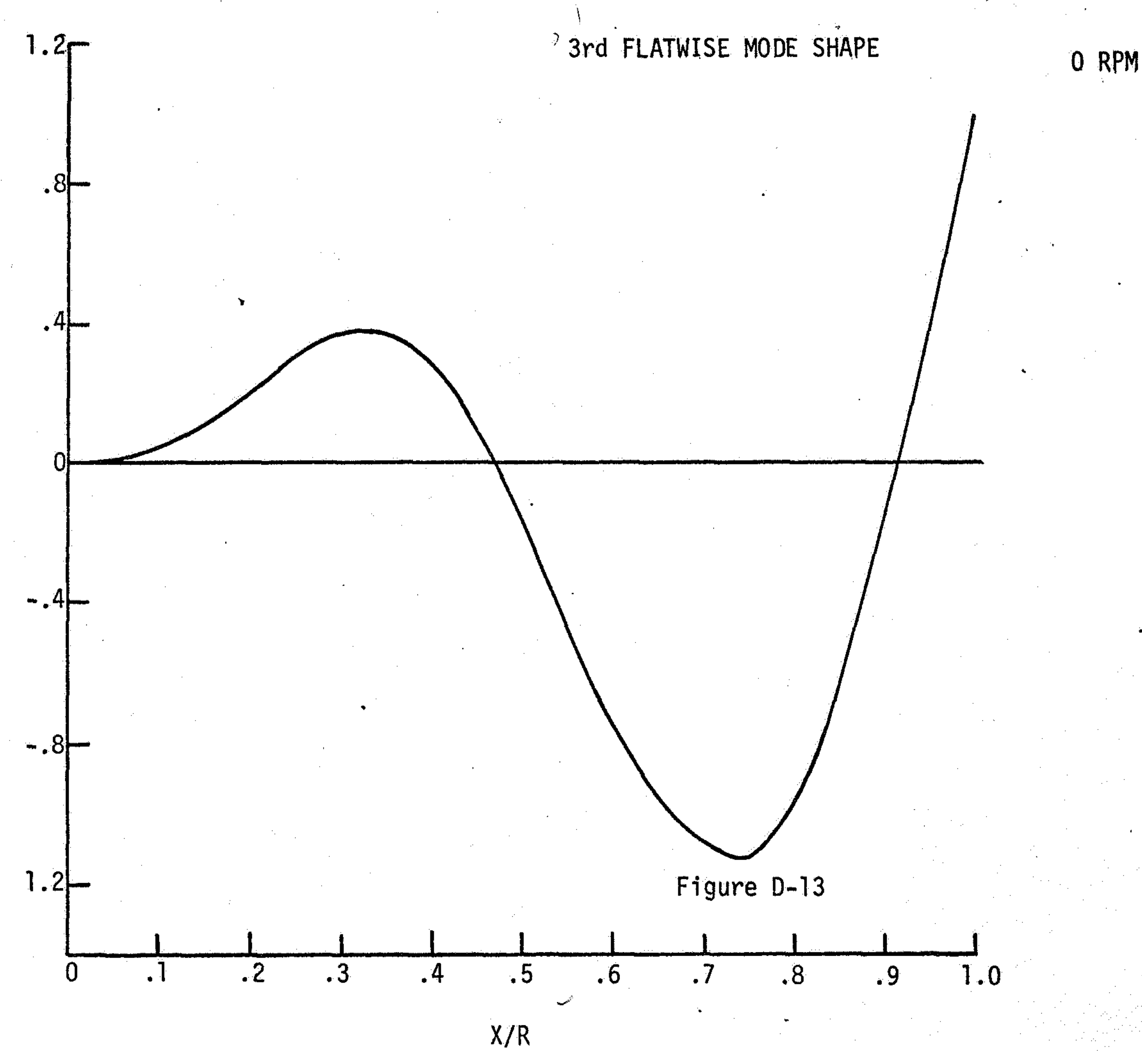


Figure D-13

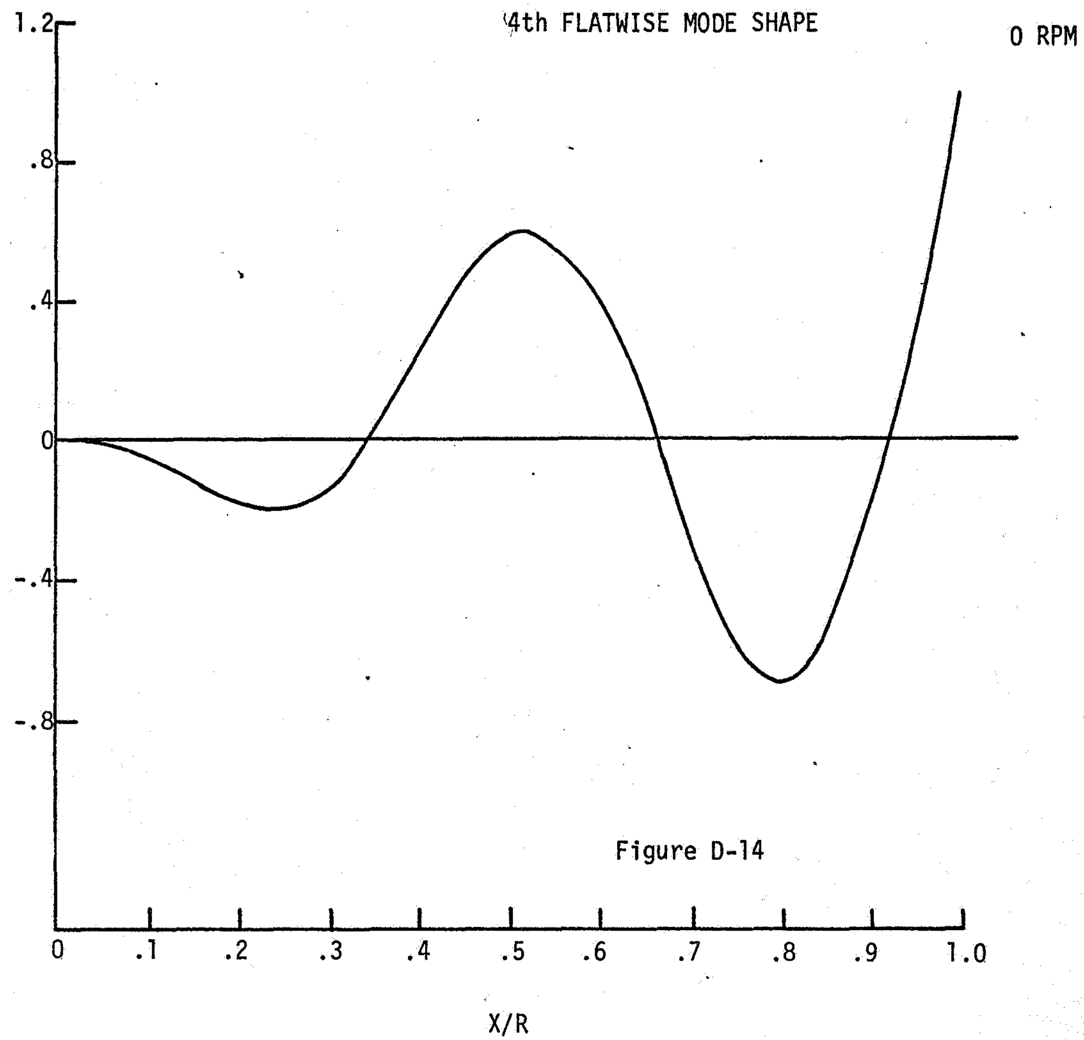


Figure D-14

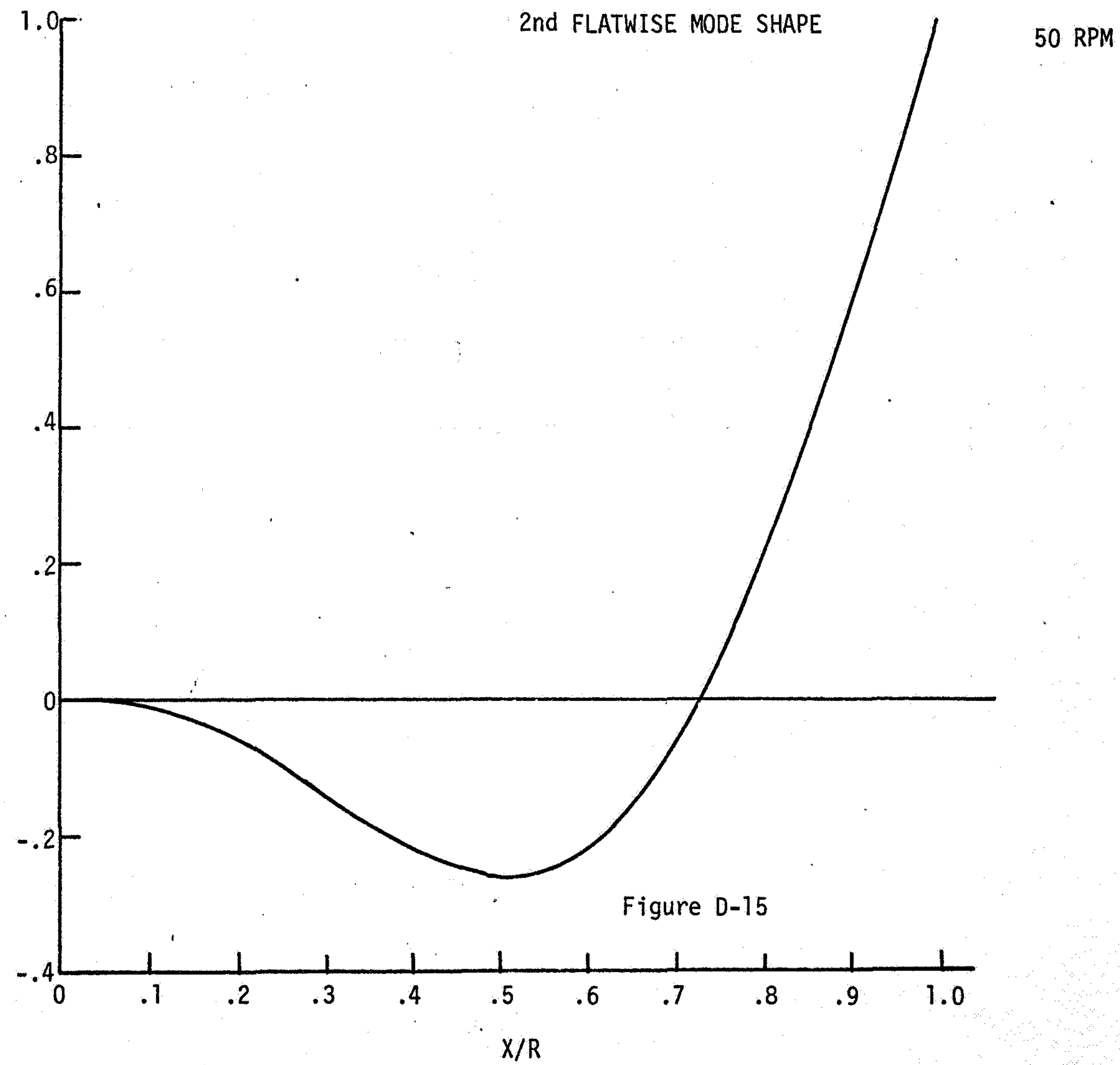


Figure D-15

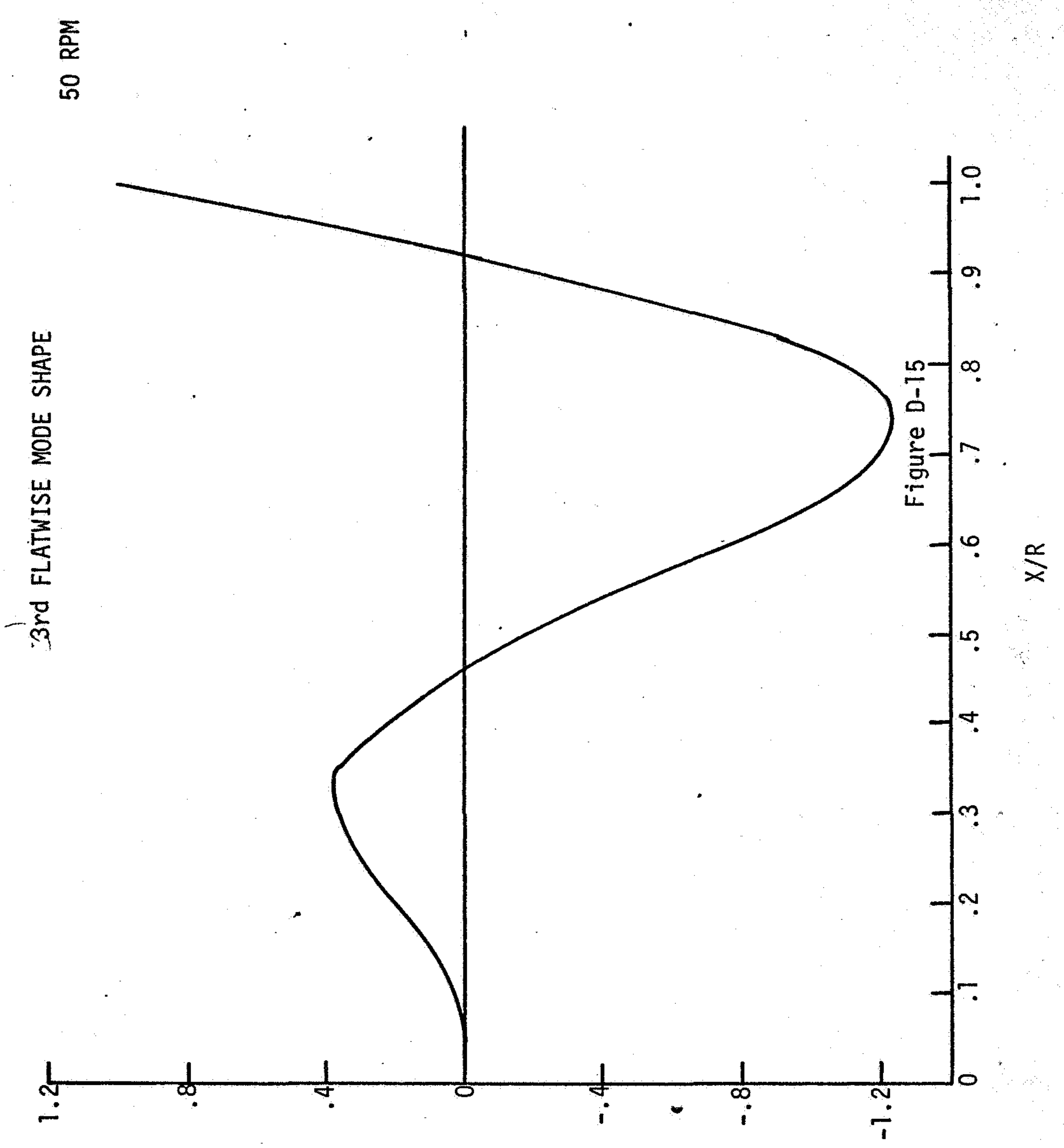


Figure D-15

4th FLATWISE MODE SHAPE

50 RPM

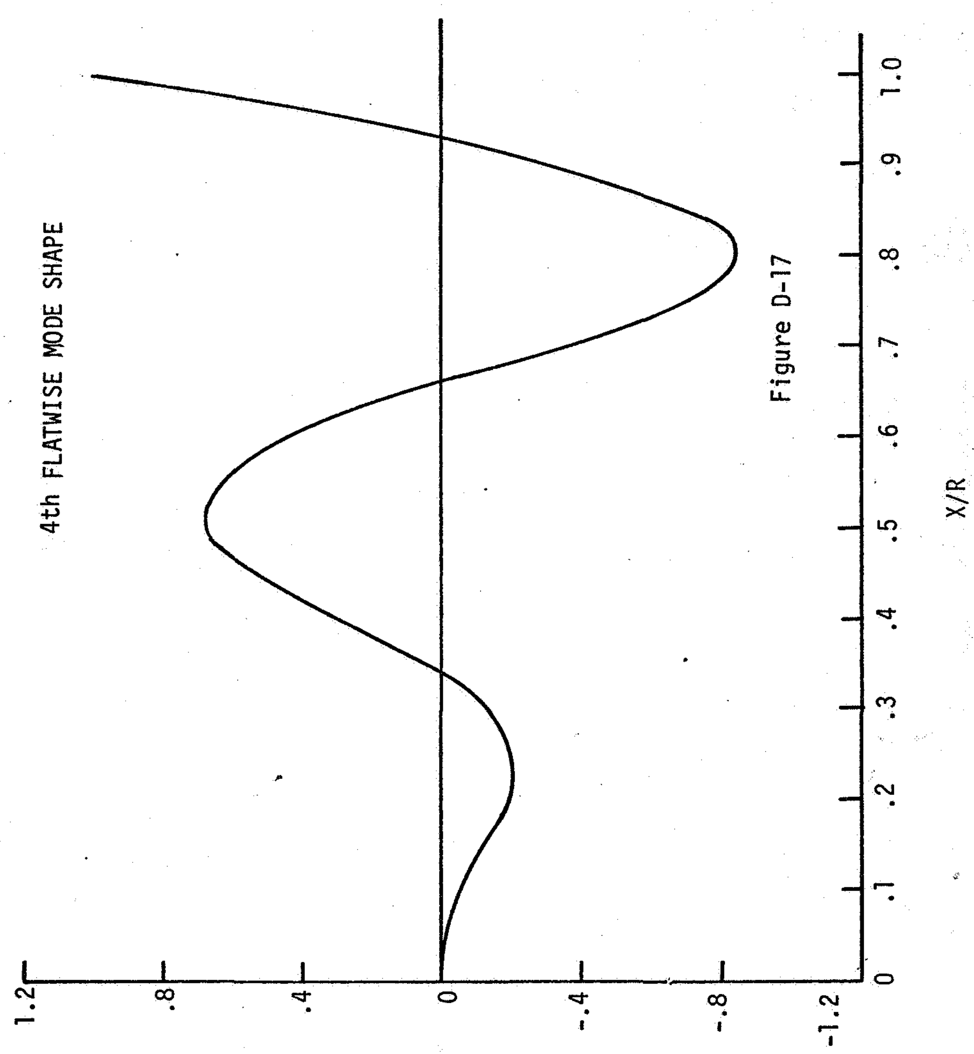


Figure D-17

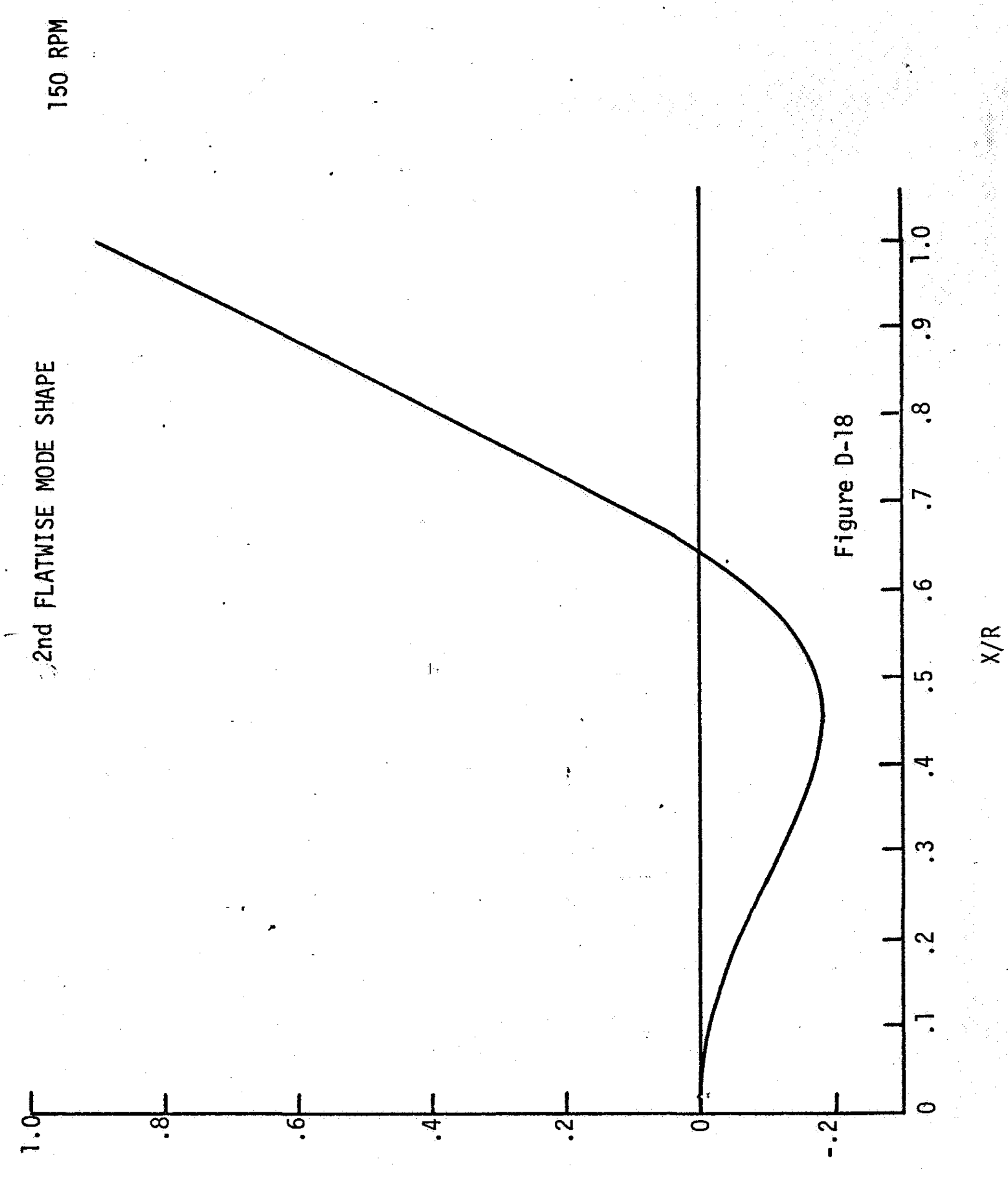


Figure D-18

3rd FLATWISE MODE SHAPE 150 RPM

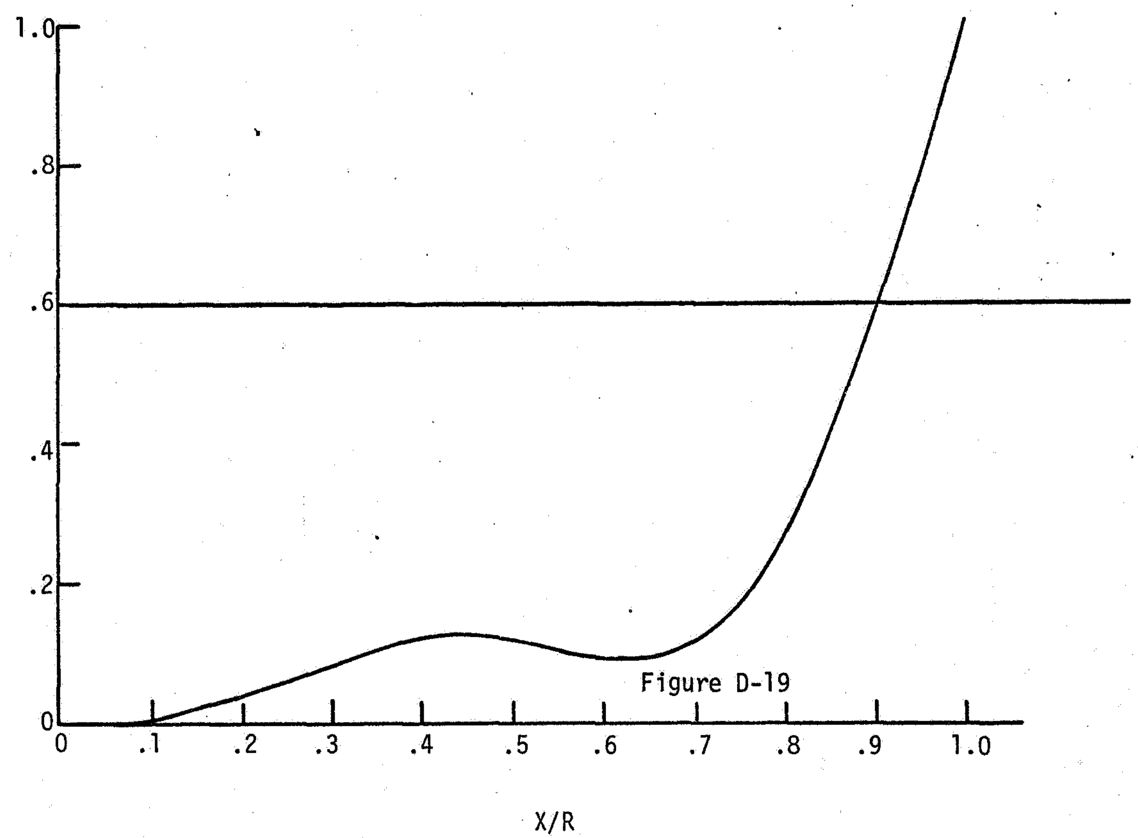


Figure D-19

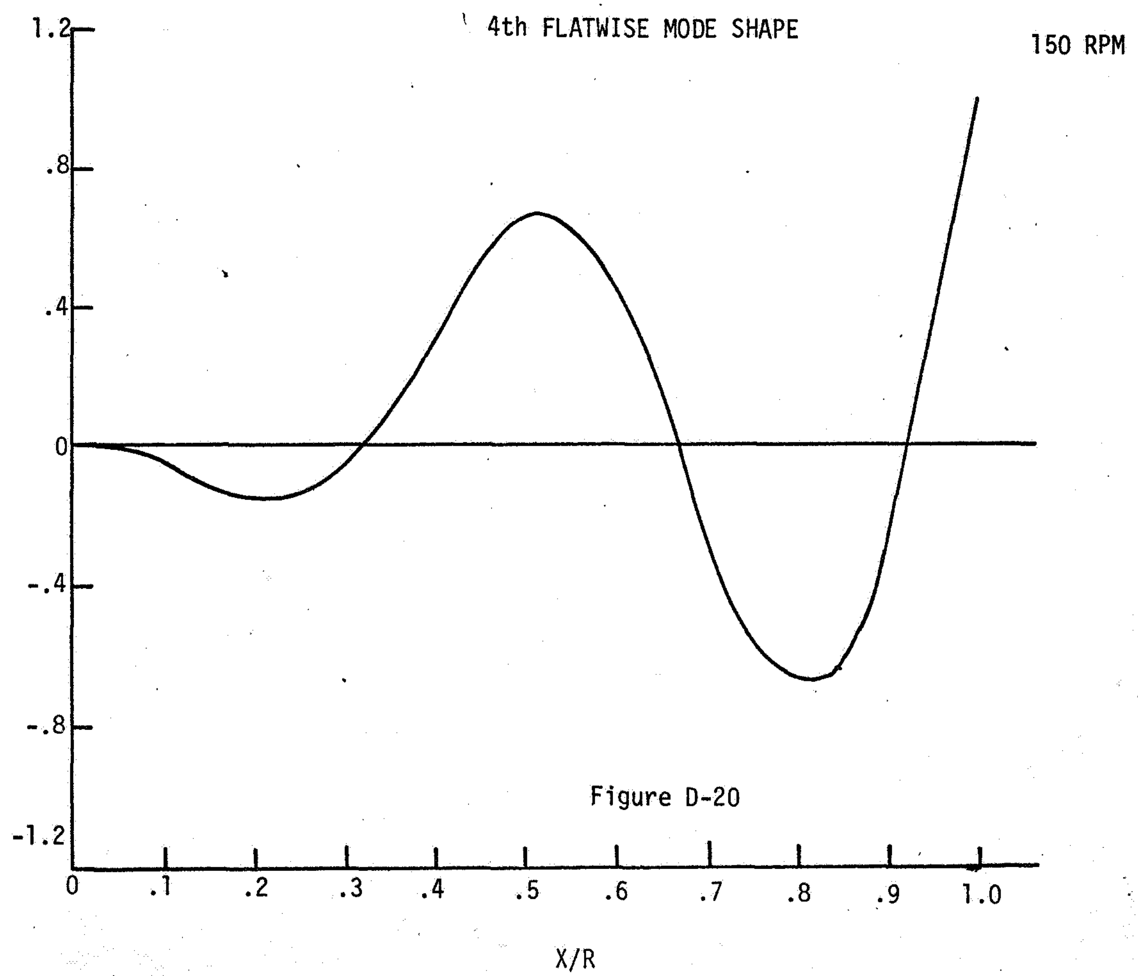


Figure D-20

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR. 159356		

4. Title and Subtitle
ROTOR DYNAMIC SIMULATION AND SYSTEM IDENTIFICATION METHODS FOR APPLICATION TO VACUUM WHIRL DATA

5. Report Date
Sept. 30, 1980

6. Performing Organization Code

7. Author(s)
A. Berman, N. Giansante, W.G. Flannelly

8. Performing Organization Report No.
R - 1496

9. Performing Organization Name and Address
KAMAN AEROSPACE CORPORATION
Old Windsor Road
Bloomfield, CT

10. Work Unit No.

11. Contract or Grant No.
NAS1-13710

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665

13. Type of Report and Period Covered
Contractor report

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The work presented in this report was performed in order to develop methods of using rotor vacuum whirl data to improve the ability to model helicopter rotors. The work consisted of the following: (1) formulation of the equations of motion of elastic blades on a hub using a Galerkin method; (2) development of a general computer program for simulation of these equations; (3) study and implementation of a procedure for determining physical parameters based on measured data; (4) application of a method for computing the normal modes and natural frequencies based on test data.

17. Key Words (Suggested by Author(s))
Rotor Hub Equations
System Identification
Normal Modes
Vacuum Whirl

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified		

N-305 [redacted]