Symbolic Generation of Elastic Rotor Blade Equations Using a FORTRAN Processor and Numerical Study on Dynamic Inflow Effects on the Stability of Helicopter Rotors

T. S. R. Reddy, Ames Research Center, Moffett Field, California

June 1986

SYMBOLS

a	lift-curve slope, 2π/radian
$A_{1j}, A_{2j}, A_{3j}, G_{1i}$	row matrices, equation (16c)
b	number of blades
$B_{1j}, B_{2j}, B_{3j}, G_{2i}$	row matrices, equation (16c)
c	chord, m
C_{d0}	profile drag coefficient
C_{d0}^*	equivalent drag coefficient, equation (14)
$C_{1j}, C_{2j}, C_{3j}, G_{3i}$	row matrices, equation (16c)
C_T, C_M, C_L	harmonic perturbation coefficients of thrust pitching moment, and rolling moment, equation (11)
$\bar{C}_T, C_H, C_W$	rotor steady thrust, drag force and weight coefficient, equation (32)
C_{mx}, C_{my}	rotor steady pitch and roll moments, equation (32)
F	forcing function, equation (27)
f	flat plate area, m^2
J	number of points used in harmonic analysis, equation (28)
K_A	blade cross-section polar radius of gyration, m
K_{m1}, K_{m2}	principal mass radii of gyration, m
K_m	blade cross-section mass radius of gyration, m
L	number of harmonics used in the harmonic analysis
$\bar{L}_{wp}$	perturbed aerodynamic force in flap direction
M	number of coupled rotating modes
$[M], [C], [K]$	constant mass, damping and stiffness matrices, equation (27)
$[m], [\ell]$	dynamic inflow matrices

PRECEDING PAGE BLANK NOT FILMED

N	rotal number of blade modes used
n	number of the harmonic in harmonic analysis, equation (30)
q_0, q_c, q_s	vectors of collective and cyclic modes, respectively
R	rotor radius, m
U_T, U_P	tangential and perpendicular velocity components, m/sec
v_i	induced velocity
w, v, ϕ	flap, lag and torsion deflections
$w_{oj}, v_{oj}, \phi_{oj}$	steady flap, lag and torsion deflections, equation (8)
X_{jk}	generalized coordinate of jth mode of kth blade
X_{0j}, X_{cj}, X_{sj}	multiblade coordinates of jth mode, equations (17) and (38)
x	blade coordinate along the radius
α	rotor shaft plane angle of attack, equation (32)
α_R	wake skew angle, equation (25)
β_{pc}	precone angle, radian
γ	Lock number
γ^*	equivalent Lock number, equation (14)
$\Delta w_j, \Delta v_j, \Delta \phi_j$	perturbation flap, lag and torsion deflections, equation (8)
ζ_k, ω_k	real and imaginary parts of the characteristic exponent
$\eta_i, \zeta_i, \theta_i$	mode shapes for flap, lag and torsion
θ	pitch angle, radian
θ_j	blade torsion mode shape
$\bar{\lambda}$	steady inflow
μ	advance ratio
ν	inflow parameter, equations (12) and (26)

ν_0, ν_c, ν_s	uniform, longitudinal, and lateral inflow components
ρ	air density, kg/m^3
σ	solidity ratio, $= bc/\pi R$
ψ	azimuthal angle, nondimensional time
Ω	blade rotational speed, radian/sec
$\omega_w, \omega_v, \omega_\phi$	nondimensional rotating flap, lag, and torsional frequencies
$(\bar{})$	nondimensionalized quantity, equilibrium deflection
$()'$	space derivative
$(\dot{})^*$	time derivative
$()_k$	quantity associated with kth blade

SUMMARY

The process of performing an automated aeroelastic stability analysis for an elastic-bladed helicopter rotor is discussed. A symbolic manipulation program, written in FORTRAN, is used to aid in the derivation of the governing equations of motion for the elastic-bladed rotor. The blades undergo coupled bending and torsional deformations. Two-dimensional quasi-steady aerodynamics below stall are used. Although reversed flow effects are neglected, unsteady effects, modeled as dynamic inflow are included. Using a Lagrangian approach, the governing equations are derived in generalized coordinates using the symbolic program. The symbolic program generates the steady and perturbed equations and writes into subroutines to be called by numerical routines. The symbolic program can operate both on expressions and on matrices. For the case of hovering flight, the blade and dynamic inflow equations are converted to equations in a multiblade coordinate system by rearranging the coefficients of the equations. For the case of forward flight, the multiblade equations are obtained through the symbolic program. The final multiblade equations are capable of accommodating any number of elastic blade modes. The computer implementation of this procedure consists of three stages: (1) the symbolic derivation of equations; (2) the coding of the equations into subroutines; and (3) the numerical study after identifying mass, damping, and stiffness coefficients for each equation. Damping results are presented in hover and in forward flight with, and without, dynamic inflow effects for various rotor blade models, including rigid blade lag-flap, elastic flap-lag, flap-lag-torsion, and quasi-static torsion. Results from dynamic inflow effects which are obtained from a lift deficiency function for a quasi-static inflow model in hover are also presented.

The numerical results for hovering flight indicate that dynamic inflow increases the lead-lag regressing-mode damping for torsionally rigid blades. For torsionally flexible blades, the dynamic inflow effect depends on the elastic coupling parameter. For zero elastic coupling, dynamic inflow increases the modal damping. For full elastic coupling, it decreases the damping. This implies that there exists an elastic coupling parameter value for which dynamic inflow effects are negligible. The study for forward flight indicates that for a large number of degrees of freedom and nonlinear models, the amount of input data to the symbolic program increases exponentially, making it inconvenient to explicitly consider the harmonic and multiblade equations. However, a combination of symbolic and numerical programs at the proper stage in the derivation process makes it effective and beneficial to obtain the stability results from this approach. The numerical study indicates that dynamic inflow does change the magnitude of the predicted damping, yet its influence on damping trends is generally small with varying advance ratio or elastic-coupling parameter for torsionally flexible blades. In this report, Part I describes the symbolic program concepts. Part II presents the numerical results obtained using this process.

1

INTRODUCTION

It is a general experience that derivation of governing equations of motion for complex structures represents a task of significant magnitude and is subject to error when performed by hand. When consideration is given to helicopter aeroelastic problems in particular, even for a rigid flap-lag model, the derivation of the governing equations of motion and the multiblade coordinate transformation requires much time and determination of accuracy by independent means. Experience with elastic hingeless rotor blade analyses indicates that for a given ordering scheme, the final equations differ in small nonlinear terms in the process of derivation depending on the stage at which the ordering scheme is applied. When the ordering scheme is consistently applied at a later stage in the derivation process, in general, this process requires more time and more lengthy independent checking.

In this situation, a symbolic manipulator allows the analyst to share the algebra with the computer. Use of symbolic programs has been reported in several branches of science and engineering for over 30 years. There are many general symbolic processors available, for example, MACSYMA (ref. 1), REDUCE, etc., written in LISP. MACSYMA has been used in many structural applications (ref. 2), and REDUCE has been used in helicopter applications (ref. 3). When applied to a particular problem, the numerous capabilities available from such general purpose programs tend to slow the program execution and puts restrictions on computer memory.

To circumvent these shortcomings, programs have been developed which are specially tailored to given tasks, as for example, in celestial mechanics (refs. 4-9). These programs take advantage of the special form of the expressions to be manipulated. They deal with the powers of the fixed number of variables forming the expressions and they manipulate the algebraic operations. These programs were written partly in FORTRAN and partly in machine language. In reference 10, concepts have been presented to manipulate matrices and series (expressions) of general form using FORTRAN. Using these basic concepts, a program called HESL has been written in FORTRAN (ref. 11) and was used in deriving rigid blade helicopter rotor equations for aeroelastic analysis.

In the present study, HESL has been extended and used in the derivation of elastic blade equations in generalized coordinates using a Lagrangian formulation. The equations can be derived for any given ordering scheme. These equations are coded into FORTRAN subroutines. The statements such as the subroutine names COMMON, DIMENSION, etc., which are required in making these subroutines are read as data during the coding. The program writes two subroutines, one for nonlinear equations to calculate the rotor trim position and the other for use with linearized perturbation equations to analyze stability. These subroutines are subsequently called by numerical routines which identify the mass, damping, stiffness, and load terms for each equation to form the required matrices which govern the blade behavior. Harmonic balance equations and transformed multiblade coordinate equations are obtained from the symbolic program. A schematic of the entire operation is shown in figure 1(a). The figure shows three segments of the analysis process; segment 1 for

2

computer derivation of the equations, segment 2 for incorporating the matrix elements into FORTRAN subroutines, and segment 3 for numerical study.

In this report, Part I describes the symbolic program aspects and Part II presents numerical results on the aeroelastic stability of an elastic rotor with dynamic inflow both in hover and in forward flight.

I: SYMBOLIC PROCESSOR PROGRAM PRINCIPLES

In this section, the concepts used to develop a symbolic (algebraic) manipulation program using FORTRAN are described. The basic manipulation of multiplying two expressions symbolically is then explained. Using this manipulation, the algebraic operations required in the aeroelastic stability and in response analysis of helicopter rotors are developed and presented. The matrix operations are performed by assuming that the elements of the matrix are expressions. The remaining sections explain the mode of input to the symbolic program. A sample output from the computer program is given in appendix A. All the symbols used in this section are given in table A1.

Program HESL

Reading and storing information- The basic principle (refs. 10 and 11) is to associate numbers with the variables (blade deflections, rotations, and time derivatives) that are to be manipulated. This allows numbers instead of variables to be manipulated. The program automatically assigns a number to the variable as soon as it reads it for the first time. Initially the program assumes that an expression or a relation is composed of a number of individual terms. Each term within an expression or a relation consists of (1) a single numerical coefficient, and (2) a pattern which consists of the numbers associated with each variable in the term. For example, tension T is given by

$$T = EA\ u' + \frac{1}{2}\ v'^2 + \frac{1}{2}\ w'^2 \tag{1a}$$

where E is Young's modulus, A is the area of cross section, u' is the derivative of the extensional displacement, and v' and w' are bending slopes in two planes. The equation for T could be defined as an expression given by

$$T \qquad\qquad = E*A*(1.0*US + 0.5*VS*VS + 0.5*WS*WS) \tag{1b}$$
$$(\text{expression}) = \qquad (\text{term}) + (\text{term}) \quad + \quad (\text{term})$$

which contains three terms. Each term has a numerical coefficient and is made up of variables US, VS, WS, E, and A. The data for each term are read in alphanumeric format (e.g., F10.0, A1, A4), and numbers are assigned by the program to variables

E, A, US, VS, and WS. Let 1, 2, 3, 4, and 5 respectively, be the numbers assigned to these variables. Then the first term (E A US) has a numerical coefficient of 1.0 and a pattern of 1 2 3. The second term (0.5 E A VS VS) has a numerical coefficient of 0.5 and a pattern of 1 2 4 4; the third term has a numerical coefficient of 0.5 and a pattern of 1 2 5 5. In the program, the expression T is (where brackets are used for convenience)

$$T = 1.0(1)(2)(3) + 0.5(1)(2)(4)(4) + 0.5(1)(2)(5)(5) \tag{2}$$

As another example, a relation given in a table which is used for subsequent substitution later in the derivation process might be

$$
\begin{array}{lcccc}
\theta & = & \theta_0 & + & \theta_c \cos \psi & + & \theta_s \sin \psi \\
\text{THTA} & = & 1.0*\text{THTO} & + & 1.0*\text{THTC}*\text{CSCY} & + & 1.0*\text{THTS}*\text{SNCY} \\
\text{(relation)} & = & \text{(term)} & + & \text{(term)} & + & \text{(term)}
\end{array}
\tag{3}
$$

which shows the total pitch angle as given by collective and cyclic pitch angles and ψ is nondimensionalized time. By letting 6, 7, 8, 9, 10, 11 represent THETA, THTO, THIC, CSCY, THTS, and SNCY, equation (3) in the program is

$$(6) = 1.0(7) + 1.0(8)(9) + 1.0(10)(11) \tag{4}$$

The numerical coefficient and the pattern of each term are stored in a coefficient-storage stack (or array), and in a pattern storage stack. These stacks are common to all expressions and to all relations given for integration, differentiation, perturbation, trigonometric relations for multiblade coordinate transformation, etc. The definition of expressions and relations and their storing is schematically shown in figure 1(b). A separate array (expression array) is provided to store the number of terms of the expression, such as T, its identification number, and the starting position in both the storage stacks. This information is required for subsequent expression retrieval. Similarly, the relations (right-hand side of THTA, in eq. 3) required for integration, differentiation, perturbation, etc., are read as tables of relations and are stored in stacks. The starting position and number of terms of the relations (THTA) for all the tables (distinguished by different table names), table identification number and left-hand side of the relation (THTA), are stored in a table array for retrieval. If the relations are not available in closed form, numerical schemes are used. The elements of a matrix are assumed to be formed by expressions. A matrix array stores the number of rows and columns of the matrices and the position of the corresponding numerical coefficient and pattern of the elements of the matrix in the storage stacks.

Special symbols are provided to identify a variable, symbol ' ' (blank), an expression, symbol '%', a matrix, symbol '#', a table containing known relations, symbol '@'. For example, expression T (eq. (1)) is identified as an expression in the program by placing the symbol % in front of the expression. Similarly, variables WS, US, VS, THTO, THTC, THTS, CSCY, and SNCY are recognized as variables by placing a blank before them. Let the table containing relation (eq. (3)) be named

4

TRIG. It is identified in the program by placing '@' in front of the table name as
@TRIG.

   Algebraic manipulation- The basic manipulation is the multiplication of two
expressions, say X and Y, to get expression Z. The details of X and Y are
obtained from the expression array (number of terms, starting location in common
stacks), and from the common stacks (numerical coefficient and pattern of each
term). The multiplication is then performed term by term. The product of two terms
yields a new term. The numerical coefficient of the new term is the product of the
numerical coefficients of the two terms. The pattern of the new term is the conca-
tenation of the patterns of the two terms in the product. For example, if term
E A US and term 0.5E A VS VS are multiplied, we get the new term
0.5 E A US E A VS VS, which in symbolic notation becomes

$$0.5(1)(2)(3)(1)(2)(4)(4)$$

   After all the terms are multiplied, the resulting terms are stored in the
common stacks. The location of these terms in the common stacks, and the number of
terms of the expression Z, and identification number for Z are stored in the
expression array for retrieval.

   The second manipulation is substituting relations, given in a table within the
computer code, into an expression. This is conceived as substituting the pattern
for the relation (such as the pattern for THTA) into an expression containing THTA
and carrying out the required multiplication, as explained above. The position of
THTA in the common stacks is obtained from the table array. It can be seen that
integration, perturbation, multiblade coordinate transformation, etc., can be easily
performed with the known relations in this way. For differentiation, the same
substitution technique is used; however, care is taken to store the dependent and
independent variables separately. It can be realized that substitution of known
relations for integration and differentiation is similar to the table look up proce-
dure used in many aerodynamic calculations.

   A compacting capability is provided for the addition of similar terms within an
expression. This is done by comparing two terms within an expression for patterns,
and, if they are identical, numerical coefficients are added and the number of terms
in the expression is reduced by one.

   The ordering scheme is applied in the following manner. The order of each
variable and the overall ordering scheme are read as user-supplied data. Once all
the terms in each expression are determined, the order of each variable is added on
a term-by-term basis. If the order sum for a particular term is greater than the
specified order to be retained, the term is neglected.

   In order to collect all the coefficients that multiply a state variable in the
final equations, the pattern of each term is checked to see whether the number
assigned to that state variable is present. If it is present, the term is stored
separately.

To perform a multiblade transformation, the relevant trigonometric relations are supplied as a table of relations. The multiblade functions 1, cos ψ, sin ψ, etc., are read as data depending on the number of blades. For example, these functions for a five-bladed rotor are 1, cos ψ, sin ψ, cos 2ψ, and sin 2ψ. The multiblade expansion of each generalized coordinate is given as data. Then the required multiplications and substitutions are performed. The terms are then checked for cos Nψ and sin Nψ (N = number of blades). Terms which do not have integer multiples of number of blades are deleted. Harmonic balance equations are obtained by substituting the trigonometric relations and by collecting all like harmonics.

Using the above concepts, the program HESL was written in FORTRAN and used to generate the governing equations of motion of a coupled rigid fuselage and rigid rotor blade analysis (ref. 11). HESL can perform all the algebraic manipulations required in the equation derivation. It can perform these operations at an expression level or matrix level. Matrix multiplication, transpose multiplication, addition, subtraction, and multiplication of matrix elements with a constant or expression can be done by using HESL. The matrix operations are done by assuming that the elements of the matrix are expressions. The matrix elements can be changed to expressions. As explained above, integration, differentiation, perturbation, and multiblade coordinate transformations are done by substituting known relations provided as data to the program. Numerical schemes are used if necessary. It should be noted that for the harmonic balance equations and multiblade coordinate transformation, trigonometric relations (products of sines and cosines expressed as sums of sines and cosines) are given as a table of relations and substitution is done. Equations can be derived for any ordering scheme.

The program has 40 subroutines excluding numerical routines, and are called through a main program by reading the manipulations to be performed as commands. For example, the command FORM EXPRESSION is used to call the necessary subroutines to multiply two or more expressions. The command FORM MATRIX is used to call the subroutines to multiply two matrices. The command SUBSTITUTE TABLE INTO EXPRESSION will substitute known relations into an expression, etc. The HESL code has 22 recognized commands. Table 1 lists each of the various commands and their functions.

The basic input to the program are the position vector, transformation relations, order of each variable, and the ordering scheme. If the relations are known, the relations for integration, differentiation, perturbation, etc., are given as tables of relations. The nonlinear equations are obtained by calculating strain energy, kinetic energy, and work done in generalized coordinates, then a Lagrangian formulation is used to obtain the governing equations of motion. The equations are linearized using the perturbation relations. When using trigonometric relations, harmonic balance equations and multiblade equations are obtained. The multiblade summation rules are embedded in the program. It should be noted that apart from the basic relations, the data to generate a required function has to be given as input. For example, one has to define what functions are to be multiplied to get strain energy. For a different blade model only the basic relations that define the change in the model need to be defined.

Data and program aspects- The main program initializes the program data, reads the commands, and calls the appropriate subroutines. When it encounters the command, END OF DATA, execution is stopped. In order to keep the available core space to a maximum, all the expressions which are no longer needed are erased from the common stacks by a RESET COUNTER.

The first input line in the program deck defines the identifiers to be used for a blank variable 'bbb', a variable 'b', an expression '%', a table '@', and a matrix '#' in that order.

This first input line is immediately followed by the input for the algebraic manipulations:

$$12345678 \qquad \text{column numbers (not an input)}$$
$$\text{bbbbb\%@\#} \qquad \text{(b \quad denotes a blank space)}$$

In general the input is the command name, followed by the name of the expression (table, matrix) and number of terms (relations, size) of the expression (table, matrix) followed by the term (relation, term) details. These expressions are in fixed format so as to provide consistency and to avoid confusion.

The names of variables, expressions, matrices, and tables are restricted to four alphanumeric characters and are read in A4 format. The identifiers are read in A1 format. For example, variables WS and VS are recognized by reading them as bbbWS and bbbVS. Expressions T and VEL are read as %bbbT and %bVEL, matrix LAFP is read as #LAFP. Table TRIG which contains given relations is read as @TRIG. Other symbols are used to separate them from this predefined set. For example, symbol '*' in *E2D1 is used for identifying an ordering scheme, and symbol "ε' in εPECF is used for identifying a group having certain variables.

The terms of an expression or relation are read as input with only one term for each input line. The first ten columns of the line are reserved for the numerical constant and the remaining columns are used for the variables (or expressions) forming the term. The format is

$$\text{F10.0,14(A1,A4)} \qquad\qquad\qquad (5)$$

For example, the term 0.5 WS WS VS VS is read as

$$123456789012345678901234567890 \qquad \text{column numbers (not an input)}$$
$$\text{0.5bbbbbbbbbbWSbbbWSbbbVSbbbVS}$$

The input and output of the program can be best explained for an example by deriving equations for the flap motion of a rotor blade model. Appendix A gives the complete input and output for the problem chosen. The problem definition starts with a flap-lag transformation definition, but is subsequently reduced to a flap-only model. This simplified model is used to clarify the program aspects rather than clarifying the modeling aspects. It should be noted that other forms of achieving the required objective may exist in addition to those defined here which

7

use the basic algebraic manipulations. Table A1 shows all the symbols used in the program.

The following explains how commands and data are read to the program. The data can be considered as consisting of two parts. One is directly read by the read commands (READ EXPRESSION, READ TABLE, etc.). For the second, data operations for the manipulations, i.e., FORM EXPRESSION, FORM MATRIX, SUBSTITUTE TABLE INTO EXPRESSION, etc., are used. The commands can be divided into three parts: (1) Input Commands; (b) Algebraic Manipulation Commands; and (3) Application Commands (see table 1).

General Rules

1. Commands names are entered from the first column. Only the first eight characters of the command are important for the operation.

2. The names of variables, expressions, matrices, and tables are always associated with their identifiers, ' ', '%', '#', and '@', respectively (read in the first input line).

3. Other identifiers can be used to distinguish them from variables, expressions, matrices, and tables, for example, *,ε,1, to be used in specifying ordering schemes, group names for collection of coefficients.

Input commands- These commands are used to read the input data, such as details of the terms forming the expressions, matrices, and tables of relations.

Command "READ EXPRESSION". This command is used to input an expression consisting of variables only. The input sequence is:

1. Command name;

2. Expression identifier (%), expression name, and the number of terms in the expression in format (A1,A4,I2);

3. For each term, input details of the term in format (F10.0,14(A1,A4)).

Example to input the strain energy of the blade given by

$$S_{eb\ell} = \frac{1}{2} K_p \beta^2$$

where K_β is the flapping stiffness and β is the flapping angle and to read expression T given by

$$T = EA \ u' + \frac{1}{2} v'^2 + \frac{1}{2} w'^2$$

Input	Explanation
123456789012345678901234567890	(column numbers, not an input)
READ EXPRESSION	command name.
%SEBL01	identifier, expression name, number of terms.
0.5 KBT BT BT	numerical coeff., variables KBT,BT forming the term.
READ EXPRESSION	command name.
% T03	identifier, expression name, number of terms.
1.0 E A US	numerical coeffi., variables E,A, US.
0.5 E A WS WS	numerical coeffi., variables E,A, WS,WS.
0.5 E A VS VS	numerical coeffi., variables E,A, VS,VS.

Command "READ MATRIX". This command is used to input the elements of a matrix. The term details of the matrix A are read as (A(I,J),J=1,N),I=1,M). The terms of the matrix element can be variables or already defined expressions or a combination of these expressions. The input sequence is:

1. Command name;

2. Matrix identifier (#), matrix name, size of the matrix (rows by columns), and ordering scheme if required, in the format (A1,A4,2I2,A1,A4);

3. Then for each element of the matrix, read the number of terms and for each term read the term details in format (F10.0,14(A1,A4)). Example: To input the flap transformation matrix, given as

$$T_{flp} = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix}$$

where β is the flapping angle.

Input	Explanation
1234567890123456789012345678901234567890	(column numbers, not an input)
READ MATRIX	command name.
#TFLP0303	matrix identifier, matrix name, number of rows, number of columns.
01	number of terms in element T(1,1).
1.0 COSB	numerical coefficient, variable COSB (COSB is cos β).
01	number of terms in element T(1,2).

9

Input (cont'd)	Explanation

```
0.0                        numerical coefficient.
01                         number of terms in element T(1,3).
-1.0 SINB                  numerical coefficient, variable SINB
                             (SINB is sin β).
01                         number of terms in element T(2,1).
0.0
01
1.0
01
0.0
01
1.0      SINB
01
0.0
01
1.0      COSB
```

Command "READ TABLE FOR SUBSTITUTION". This command is used for reading a table of relations which is used in subsequent substitutions. Each table of relations is identified by a table name. HESL identifies two types of tables of relations. Type 1 is a substitution of relations of the type $v = \bar{v} + \delta v$, and type 2 is a relations table which has powers (>1) and a product of the variables such as $\sin^2 \psi = 0.5-0.5*\cos 2\psi$. In type 2, the relations can be handled in the general form of $A^\ell B^m C^n =$ terms, where A, B, and C are variables, and ℓ, m, and n are their powers. For example, $\sin^3 \psi \cos^2 \theta \sin^2 \beta =$ terms. The input sequence is:

1. Command name;

2. Table identifier (#), table name, number of relations in the table, and type of the table used in format (A1,A4,I2). A zero value indicates type 1 relations, a nonzero value indicates type 2 relations;

3. Then, for each relation, input:

    (a) the number of terms on the right-hand side of the relation, the left-hand side variables and their power in format (I2,4(A4,I2));

    (b) the details of the terms (of the right-hand side of the relation) in format (F10.0,14(A1,A4)).

Example: To input two tables of relations. Table 1, named SUPR contains powers of unity of the variables in the relation. Table 2 contains powers greater than the unity of the variables and the product of variables.

Table 1   (1) $\sin \zeta = 0.0$; (2) $\cos \zeta = 1.0$; (3) $\theta = \theta_o + \theta_c \cos \psi + \theta_s \sin \psi$

10

Table 2

(1) $\sin^2 \beta = 1.0 - 1.0 \cos \beta \cos \beta$; (2) $\sin \psi \cos^2 \psi = 0.25 \sin \psi + 0.25 \sin 3\psi$
where ζ is the lagging angle and β is the flapping angle, and ψ is nondimensional time.

Input	Explanation
12345678901234567890123456789 0	(column numbers, not an input)
READ TABLE FOR SUBSTITUTION	command name.
@SUPRO300	table identifier, table name, number of relations, table type.
01 SINZO1	number of terms on the right hand side, relation name, its power.
0.0	numerical coeff. (term details).
01 COSZO1	same for second relation.
1.0	
03 THTAO1	
1.0 THTO	
1.0 THTC CSCY	
1.0 THTS SNCY	
READ TABLE FOR SUBSTITUTION	command name.
@DUMYO201	identifier, table name, number of terms, table type.
02 SINBO2	number of terms on the right hand side, relation name, its power.
1.0	numerical coefficient.
1.0 COSB COSB	numerical coefficient, variable COSB.
02 SNCYO1 CSCYO2	number of terms, left hand side variables with their powers.
0.25 SNCY	numerical coefficient, variable SNCY.
0.25 S3CY	numerical coefficient, variable S3CY.

Command "READ DIFFERENTIATION TABLE". This command is used for reading the rules of differentiation of variables. The input sequence is:

1. Command name;

2. Table identifier, differentiation relations table name and the number of relations in the table in format (A1,A4,I2).

Then for each relation provide:

(a) The independent variable for differentiation, the dependent variable to be differentiated, and the number of terms in the differentiation relation in format (2{A1,A4}I2);

(b) Each term detail in format (F10.0,14(A1,A4));

11

Example: To read a table of relations containing the differentiation rules, given by

$$\frac{\partial \beta}{\partial \beta} = 1.0; \quad \frac{\partial \dot{\beta}}{\partial \dot{\beta}} = 1.0; \quad \frac{\partial \cos \beta}{\partial \beta} = -\sin \beta; \quad \frac{\partial \sin \beta}{\partial \beta} = \cos \beta; \quad \frac{\partial \beta}{\partial \tau} = \dot{\beta}$$

$$\frac{\partial \dot{\beta}}{\partial \tau} = \ddot{\beta}; \quad \frac{\partial \cos \beta}{\partial \tau} = -\sin \beta \cdot \dot{\beta}; \quad \frac{\partial \sin \beta}{\partial \tau} = \cos \beta \cdot \dot{\beta}; \quad \frac{\partial \cos \psi}{\partial \tau} = -\sin \psi; \quad \frac{\partial}{\partial \tau} \sin \psi = \cos \psi$$

where β is the flapping angle, and ψ, τ denote time.

Input	Explanation
123456789012345678901234567890	(column numbers, not an input)
READ DIFFERENTIATION TABLE	command name.
@DERV10	identifier, name of the table, number of relations.
BT BTO1	independent variable, dependent variable, its power.
1.0	numerical coefficient.
BTD BTDO1	
1.0	
BT COSBO1	
-1.0 SINB	
BT SINBO1	
1.0 COSB	
TAU BTO1	
1.0 BTD	
TAU BTDO1	
1.0 BTDD	
TAU COSBO1	
-1.0 SINB BTD	
TAU SINBO1	
1.0 COSB BTD	
TAU CSCYO1	
-1.0 SNCY	
TAU SNCYO1	
1.0 CSCY	

Command "READ GROUP AND ORDER OF THE VARIABLES". This command is used to input the variables group to which they belong and their order of magnitude. Assigning group numbers and order of magnitude to the variables shows their importance in the analysis. For this example, original variables and steady quantities, $\beta, \dot{\beta}, \ddot{\beta}, C_d{}_o/a$ are considered to belong to group 1 and perturbed quantities $\delta\ddot{\beta}, \delta\dot{\beta}, \delta\beta{}_o$ belong to group 2. This allows retention of only linear terms in perturbation quantities. The input sequence is:

1. Command name;

2. The total number of variables to which a group and order of magnitude is assigned (highest order implies lowest importance) in format I3;

3. For each variable, provide the identifier for variable, variable name, a blank and its group number and order of magnitude in format (A1,A4,1X,2I2). Eight variables can be typed per line.

Example: To input the following variables showing their group and order of magnitude (gr = group, O = order).

$$\ddot{\beta} \; gr(1) \; O(1); \; \dot{\beta} \; gr(1) \; O(1); \; \beta \; gr(1) \; O(1);$$

$$\delta\ddot{\beta} \; gr(2) \; O(1); \; \delta\dot{\beta} \; gr(1) \; O(1); \; \delta\beta \; gr(2) \; O(1);$$

$$C_{d_o}/a \; gr(1) \; O(2); \; \theta \; gr(1) \; O(1)$$

Input	Explanation
READ GROUP AND ORDER OF THE VARIABLES	command name.
b08	total number of variables assigned group and order.

BTDD 0101 BTD 0101   BT 0101 DBDD 0201   DBD 0201   DB 0201 CDOA 0102 THTA 010

Variables BTDD, BTD, BT are assigned group 1(01) and order 1(01), variable CDOA is assigned group 1(01) and order 2(02), variables DBDD, DBD, DB are assigned group 2(02) and order 1(01), since they are considered as perturbed quantities. Note that all the variables have their identifier (a blank) in front of them.

Note: If the order and group of any variable is not defined, the variable is assigned to have an order of zero (highest importance) and belong to group 1.

Command "READ ORDERING SCHEME". This command is used for specifying the ordering scheme. It specifies up to what order group (1) variables and group (2) variables should be retained. The input is:

1. Command name;

2. Identifier for ordering scheme (*), the ordering scheme name and number of groups of variables considered in format (A1,A4,I2);

3. For each group, provide in sequence the group number and the highest order of magnitude to be retained, in format (4I2).

Example: To specify two groups of variables and retain group 1 variables up to total order of magnitude 2 and group 2 variables up to total order of magnitude 1

13

Retain $Gr(1) \to O(2)$ and $Gr(2) \to O(1)$ in a term.

Input	Explanation
READ ORDERING SCHEME	command name.
*E2D102	identifier for ordering scheme, name of the ordering scheme, number of groups.
01020201	group 1(01), highest order to be retained (02), group 2(02), highest order to be retained (01).

Command "READ VARIABLES FOR COLLECTION OF COEFFICIENTS". This command identi-
fies the name of a group of variables for which the terms containing these variables
have to be collected. The input is:

1. Command name;

2. Group identifier (ϵ), the name of the group and number of variables in
format (A1,A4,I2);

3. The variables in format (16(A1,A4)).

Example: (1) To input a group name containing variables $\ddot{\beta}$, $\dot{\beta}$, β, θ for
collecting terms containing these variables subsequently. (2) To input a group
name containing variables θ_o, θ_1, θ_2, β_o, β_1, β_2, $\dot{\beta}_o$, $\dot{\beta}_1$, $\dot{\beta}_2$, β_o, β_1, β_2 for col-
lecting terms containing these variables subsequently.

Input	Comment
READ VARIABLES FOR COLLECTION OF COEFFICIENTS	command name.
εPECF04	group identifier, its name, number of variables in the in the group.
BTDD BTD BT THTA	variables.
READ VARIABLES FOR COLLECTION OF COEFFICIENTS	
εMUCF12	
THO TH1 TH2 BDD0 BDD1 BDD2 BD0 BD1 BD2 B0 B1 B2	

Note that the terms containing these variables will be subsequently collected. Also
note that variables are typed along with their identifier (a blank).

Command "READ TAPE". This command is used to read and separate steady and
perturbed quantities and to write on the disk to be used by the symbolic program.
The input is:

1. Command name;

2. Expression name to be read from tape with its identifier, the number of the tape (disk) from which data has to be read; a nonzero value if perturbed and steady terms have to be separated, number of perturbation terms, number of the disk on which steady and perturbed terms have to be written, number of equations to be read in format (A1,A4,10I2).

Example: To write the details of expression LAEQ on tape number 52.

Input	Explanation
READ TAPE	command name.
%LAEQ520010000001	expression identifier, expression name, number of the tape and other options.

Algebraic manipulation commands- These commands are used to perform a single algebraic manipulation such as multiplication or substitution of relations for expressions and matrices. These commands mostly operate on the data read by input commands.

Command "FORM EXPRESSION". This command is used to form another expression by addition, subtraction, and multiplication of variables and previously read expressions (by READ EXPRESSION command). The input is similar to that for READ EXPRESSION command. The input sequence is:

1. Command name;

2. Expression identifier (%), the name of the new expression to be formed, the number of terms forming the new expression, and if required, the name of the ordering scheme, the name of the group containing the variables for which the coefficients have to be collected, and a nonzero value to write on the disk (disk number), in format (A1,A4,I2,2(A1,A4),2I2);

3. The details of each term in format (F10.0,14(A1,A4)).

Example: (1) To form a new expression ζ, from three already defined or read expressions, $\zeta_{xt}, \zeta_y, \Omega_z$, with ordering scheme E2D1, and collect the terms containing variables in group PECF, to write on tape number 8. (2) To form a new expression M_β, with variables RAC2, θ, $\bar{r}$ and already defined expressions U_T and U_P.

Input	
FORM EXPRESSION	command name.
% RXD02*E2D1εPECF08	expression identifier, new expression name, number of terms, name of the ordering scheme, name of the group containing variables for collection, number to write on tape.

Input (cont'd)

1.0 % RXT   numerical coefficient, expression name.
1.0 % RY%OMEZ   numerical coefficient, expressions RY and OMEZ.

FORM EXPRESSION   command name
% MB02   expression MB with its identifier, number of terms
-0.5 % UT% UT THTA RAC2 RB   numerical coefficient, expression UT, variables
                                      THTA, RAC2, RB.
0.5 % UT% UP   RAC2 RB   similarly second term details.

Command "FORM MATRIX". This command is used to multiply two matrices
(C = A*B), including transpose multiplication (C = A transpose * B). The input
sequence is:

1. Command name;

2. Names of the two matrices to be multiplied, (A,B), the name of the result-
ing matrix (C) with matrix identifiers, name of the ordering scheme, and a nonzero
value for transpose multiplication in format (4(A1,A4)I2).

Example: To multiply two matrices T_{lag} and T_{flp} to give T_{flg} with an
ordering scheme specified by E2D1 (if required transpose multiplication)

$$[T_{flg}] = [T_{lag}][T_{flp}] \quad \text{or} \quad [T_{flg}] = [T_{lag}]^T[T_{flp}]$$

Input   Explanation

FORM MATRIX   command name.
#TLAG#TFLP#TFLG*E2D100   matrix identifier, matrix name, matrix identifier,
                                      the resulting matrix name, matrix identifier, the
                                      resulting matrix name, ordering scheme, nonzero
                                      for transpose multiplication.

Command "ADD MATRICES". This command is used to add or subtract two
matrices. The input sequence is:

1. Command name;

2. Names of two matrices to be added or subtracted and the name of the
resulting matrix with their identifiers and a zero for addition and nonzero for
subtraction, in format (3(A1,A4),I2).

Example: To add matrices A and B to give matrix C

$$[C] = [A]\pm[B]$$

16

Input	Explanation
ADD MATRICES	command name.
# A# B# COO	matrix identifier, matrix name, matrix identifier, matrix name, matrix identifier, matrix name, zero for addition (nonzero for subtraction).

Command "CONSTANT MATRIX". This command is used for multiplying a matrix with a constant or an expression having only one term (both of which are read earlier in the program).

After multiplication the resultant matrix is written in the same location. For multiplying with expressions having more terms, define a diagonal matrix with this expression and perform matrix multiplication. The constant has to be specified as an expression before multiplication. The input sequence is:

1. Command name;

2. The matrix name, and the expression name with their corresponding identifiers in format 2(A1,A4).

Example: To perform [A] = const * [A].

Input	Explanation
CONSTANT MATRIX	command name.
# A%CONS	matrix identifier, matrix name, expression identifier, expression name.

Command "MATRIX EXPRESSION". This command is used to convert matrix elements (usually a column vector) to expressions. The input sequence is:

1. Command name;

2. The number of expressions to be formed, the name of the matrix with its identifier;

3. For each expression, name of the expression with its identifier and its location in the matrix.

Example: To redefine U_T as given by element (2,1) of matrix VX2Y, and U_p as element (3,1) of matrix VX2Y.

Input	Explanation
MATRIX TO EXPRESSION	command name.
02#VX2Y	number of expressions, matrix identifier and matrix name.

% UTO201% UPO301   expression identifier, name, element (2,1) and
                                    expression identifier, name, element (3,1).

   Command "SUBSTITUTE TABLE INTO EXPRESSION". This command is used to substitute
relations given in a table (read earlier by READ TABLE FOR SUBSTITUTION command)
into an expression with, or without, an ordering scheme. After substitution, coef-
ficients for certain variables can be collected, and if required, can be written on
a disk. The input sequence is:

   1. Command name;

   2. Name of the table containing the relations, the name of the expression,
name of the resulting expression, ordering scheme, group for collecting coefficients
with their respective identifiers, a nonzero value to write on the disk, in format
(5(A1,A4),I2).

   Example: To substitute relations given in table SUPR in expression R1, result-
ing in expression RX with an ordering scheme E2D1, group name for collecting PECF
and no writing on the tape.

           Input   Explanation

SUBSTITUTE TABLE INTO   command name.
  EXPRESSION
@SUPR% R1% RX*E2D1εPECF00   table name, old expression R1, resulting expression
                                      RX, ordering scheme E2D1, group for collection
                                      PECF with their respective identifiers, no
                                      writing on disk.

Note: If E2D1 and PECF are not used, then leave blank spaces.

   Command "DIFFERENTIATE EXPRESSION". This command is used to differentiate an
expression with respect to a specified variable using the differentiation rules
previously read in the READ DIFFERENTIATION TABLE command (e.g., table DERV). The
input sequence is:

   1. Command name;

   2. The variable name with which the expression has to be differentiated, the
name of the expression to be differentiated, and the name of the resulting expres-
sion in format (3(A1,A4)).

   Example: To differentiate expression r_x(RX) with respect to item (TAU) result-
ing in expression r_{xt}(RXT).

Input	Explanation
DIFFERENTIATE EXPRESSION	command name.
TAU% RX% RXT	variable identifier, dependent variable, original expression and resulting expression with their identifiers.

Application commands- Application commands are those which combine all, or some of the functions of the manipulation commands defined above to meet specific derivation requirements. Application of Lagrangian, multiblade coordinate transformation, etc., require a number of operations to be done simultaneously. These commands are written to satisfy these requirements.

Command "FORM LAGRANGIAN". By a careful consideration of the governing equation deviation process, the required algebraic process can be suitably divided into a number of computational steps to be calculated without intermediate expression swell. For example if I is given by

$$I = \int \frac{\partial}{\partial \tau} [A*B] * C \, dr \qquad (6a)$$

It can be divided into

$$I = \int \frac{\partial}{\partial \tau} [A_1*B_1] *C_1 \, dr + \int \frac{\partial}{\partial \tau}[A_2*B_2] * C_2 \, dr \qquad (6b)$$

where A, B, and C, etc., are expressions and τ is the independent variable. Here the evaluation of I is performed in two computational steps. The contribution from each computational step to the final expression is added to form the final expression. Each computational step requires a definition of expressions A_i, B_i, and C_i. It also requires that there be relevant integral tables for integration, a differentiation table for differentiation, an independent variable, and perturbation relations for linearization. The evaluation of the total expression may also require an ordering scheme, or a group name for collecting terms. Accordingly, the input consists of: (1) the number of computational steps into which the problem is divided, the maximum number of tables of relations used in the evaluation, identifiers (1 or 0) for suppressing terms according to $1 + \epsilon^2 = 1$ are required, printing before collection of terms, and for writing FORTRAN subroutines on disk; (2) the name of the resulting expression from this command, the name of the ordering scheme, and the name of the group having variables for which the coefficients have to be collected; (3) then for each computational step provide the names of the three expressions involved in the computational step, the name of the variable with which the product of the first and second expression is to be differentiated (leave blank if differentiation is not needed), and the names of integration and perturbation tables.

Example: To evaluate

$$S_{beq} = \int \frac{\partial}{\partial \tau} [\dot{r}_x r_{xq}] \, c \, dx + \int cM_\beta c \, dx$$

This is equal to two computational steps. The tables of required relations are INRL for integration, PERT for perturbation, DYNM for integration in aerodynamic terms, the independent variable τ (TAU), the final equation name, S_{beq} (SBEQ), the ordering scheme, E2D1, and the group name for the collection of terms, PECF.

Input	Explanation

FORM LAGRANGIAN — command name.

0202000100 — two computational steps, using two tables of relations and other options.

%SBEQεPECF*E2D1 — final expression name, group for collection, ordering scheme.

% RXD%RXDQ%CONS TAU@INRL@PERT — expressions, variable, tables

%CONS% MB%CONS   @DYNM@PERT — expressions, variables, tables

Note: When comparing the above inputs to equation (6b), A is RXD, B is RXDQ, C is CONS, τ is TAU, with integration, perturbation relation in INRL and PERT. A similar process is repeated for the second term.

Command "INITIALIZE MULTIBLADE". In performing the multiblade transformation, the single-blade equation has to be converted from a rotating frame of reference to a nonrotating frame of reference. This requires that the single-blade equations be multiplied and multiblade functions 1, cos ψ, sin ψ, (-1), etc., depending on the number of blades. This command is used to read a number of blades and to generate these expressions. This has to be called before performing a multiblade coordinate transformation. The input is:

1. Command name;

2. The number of blades, and (-1), cos ψ, sin ψ, $\cos^2 \psi$, $\sin^2 \psi$, $\cos^3 \psi$, $\sin^3\psi$, $\cos^4 \psi$, $\sin^4 \psi$, $\cos^5 \psi$, $\sin^5 \psi$, $\cos^6 \psi$, $\sin^6 \psi$, $\cos^7 \psi$, $\sin^7 \psi$, in format (I2,3X,15(A1,A4)).

Example: To read multiblade functions for a three-bladed rotor also cos(3ψ) and $\sin^2 \psi$.

Input	Comments

INITIALIZE MULTIBLADE — command name.

03   MOPK CSCY SNCY C2CY — number of blades, multiblade functions.

S2SY C3CY S3CY — cos(3ψ), sin(3ψ).

Command "PERFORM MULTIBLADE TRANSFORMATION". This command is used to obtain equations in multiblade coordinates from the single blade equations. The input sequence is:

1. Command name;

2. Number of resulting multiblade equations (equal to number of blades). Note: For dynamic inflow equations, the number of equations is one;

3. The names of the tables containing multiblade relations, trigonometric relations, the name of the group containing multiblade variables for collection of coefficients;

4. The name of the expression to be transformed, the names of the resulting multiblade expressions.

Example: To convert equation S_{beq} into multiblade equations for a three-bladed rotor. Multiblade expansion of each degree of freedom is given in table MULB, trigonometric relations are given in table TRIG, the variables for collection of coefficients is given in MUCF. The final multiblade equations are named AEM1, AEM2, and AEM3.

Input	Explanation
PERFORM MULTIBLADE TRANSFORMATION	command name.
03	number of resulting multiblade equations.
@MULB@TRIG&MUCF	relation tables (MULB,TRIG), group name for collection (MUCF).
%SBEQ%AEM1%AEM2%AEM3	single blade expression name and resulting multiblade expressions names with their identifiers.

Command "WRITE TAPE". This command is used to write on disk or tape, the steady and perturbed equations. The input is:

1. Command name;

2. Name of the expression to be written on disk, and the number of the disk in format (A1,A4,I2).

Example: To write expression ABCD on disk number 3.

Input	Explanation
WRITE TAPE	command name.
%ABCD03	expression name, disk number.

21

Command "NEGATE EXPRESSION". This command is used to get the negative of an expression when required. The input is command name and expression name.

Example: To rewrite U_T (UT) = $-U_T$

Input	Explanation
NEGATE EXPRESSION	command name.
% UT	expression name with identifier.

Command "RESET COUNTER". This command is used to reset the counter (starting number of expressions, see reading and storing section), and to save specified expressions that are required for further manipulation and calculations, so that further computations can over-write the unneeded expressions and thus save memory. The input is:

1. Command name;

2. The name of the expression whose starting number will become the new starting number (see reading and storing), and number of expressions to be saved.

3. The names of expressions to be saved.

Example: To reset the counter from expression EXP5 and to save expressions EX25 and EX26, which are generated after EXP5 is generated.

Input	Explanation
RESET COUNTER	command name.
%EXP502	identifier, name and number of expressions to be saved.
%EX25%EX26	identifier, expression name(s) to be saved.

Command "END OF DATA". When this command is read, execution is stopped. This command can be placed at any location in the derivation process to check the program output. The input is:

1. Command name.

Input	Explanation
END OF DATA	command name.

Appendix A shows the output for the equation derivation for a rigid rotor blade model. The problem initially formulated with flap-lag degrees of freedom is reduced to flap degree of freedom (by SUBSTITUTE TABLE INTO EXPRESSION command). The output includes both the single-blade equation SBEQ, and the complete multiblade equations, AEM1, AEM2, and AEM3 for a three-bladed rotor. During the equation generation, the required quantities are written on a separate disk (tape) area for subsequent FORTRAN translation. The program TRANS (see table 1) is executed to read this data

and code it into subroutines. The elements of each equation, and each coefficient of the state variables, defined in PECF or MUCF (for multiblade equations) are written as elements of a matrix. The coefficients are separated into mass, damping, and stiffness matrices by calling this subroutine through a numerical program.

Comments

An algebraic manipulation program using FORTRAN has been described. The program can operate on both expressions and on matrices. Because of the modularity of the program, other operations can be easily added. The following limitations are observed with regard to data input and the size of the individual terms. The program requires data as input for differentiation, integration (because of the table look-up procedure), and for other operations required in helicopter rotor problems such as perturbation and multiblade coordinate transformation. The amount of these data increase with the complexity of the problem. For example, the input data increase with the specified number of harmonics for a trim calculation, the number of blades, and the number of degrees of freedom describing the model. The program at present assumes that each term of an expression is made up of no more than 20 variables. This assumption simplified the program development. If more variables per term are expected, the dimensions need to be increased.

II: NUMERICAL STUDY

Introduction

The hingeless rotor blade configuration reduces mechanical complexity and increases rotor control power and damping relative to rotors with articulated blades. However, it also introduces complex aeroelastic behavior that requires a rigorous analysis for effective design. The derivation of the equations governing this behavior is tedious and error-prone because of the nonlinear nature of the problem. This has led to the use of symbolic derivation and to automation of the entire process, from derivation to numerical calculation with only limited user interface required. In the present report, this automation of the aeroelastic analysis is applied to the flap, lag, and torsion dynamics of an elastic rotor blade in hover and in forward flight with dynamic inflow.

A complete dynamic analysis of a hingeless rotor blade consists of all three elastic degrees of freedom--flap, lag, and torsion. Initial analyses focused on the investigation of flap-lag stability of torsionally rigid blades. The models consisted of a rigid blade with spring-restrained hinges at the hub to simulate bending flexibility. The stability of this type of model was analyzed for both hover (ref. 12) and forward flight (ref. 13). Flap-lag stability of elastic blades with uniform properties was studied by Ormiston and Hodges (ref. 12), based on a derivation of nonlinear partial differential equations suitable for elastic hingeless blades. Similar equations were studied by Friedmann and Tong (ref. 14).

Concurrently with the flap-lag stability analyses, efforts were also made to investigate the complete blade problem by including blade torsional deflections. Friedmann and Tong (ref. 14) approximated the torsional deflection by rigid body pitching motion (root torsion); they found that torsion motion was important and that the stability characteristics were sensitive to the number and type of assumed bending mode shapes used. Flap-lag structural coupling was not included. Hodges and Ormiston (ref. 15) presented extensive numerical results for the stability characteristics of elastic hingeless blades with flap-lag-torsion motion in hover. They found that torsional deflections of hingeless rotor blades are strongly influenced by the nonlinear structural moments caused by flap and lead-lag bending. This bending-torsion structural coupling is proportional to the product of the flap and lead-lag bending curvatures and to the difference between the two bending flexibilities. This study also showed the effect of precone, structural coupling, and torsional rigidity on the isolated blade stability boundaries. Friedmann and Kottapalli (ref. 16) analyzed the coupled flap-lag-torsional dynamics of hingeless rotor blades in forward flight. They noted that nonlinearities are important in an aeroelastic stability analysis, and that forward flight is strongly coupled with the trim state. Only flapping motion was used in calculating the rotor trim condition. It was observed that forward flight is stabilizing for soft in-plane rotors and destabilizing for stiff in-plane rotors. In all these studies, the aerodynamic forces were obtained from strip theory based on a quasi-steady approximation of two-dimensional unsteady airfoil theory.

Simultaneously, efforts have been made to improve the aerodynamic model used in these analyses by including unsteady airflow effects. One approach to include these effects is to model the induced velocity as a time-dependent, three degree-of-freedom system. This dynamic inflow theory has been applied to rigid blade flap-lag analyses both in hover and in forward flight (refs. 17-19), and to a coupled rotor-fuselage problem in hover (refs. 20 and 21). It was observed that the dynamic inflow increased the lag-regressing mode damping and reduced the body pitch and roll damping for the parameters considered. These analytical results correlated well with experimental results (ref. 20). However, the conclusions presented in references 17 to 21 were based on several restrictive assumptions: zero elastic coupling, fixed-solidity ratio, and rigid flap-lag blade models with no torsional flexibility.

It was observed in reference 15 for hover and in reference 16 for forward flight that when compared with a coupled flap-lag-torsion analysis, flap-lag analyses underpredict the in-plane (lead-lag damping) damping. As a result, it was pointed out in reference 22 that for torsionally soft blades to correctly assess the influence of dynamic inflow, it is necessary to formulate a model with both elastic torsion dynamics and dynamic inflow effects.

General nonlinear differential equations for an elastic rotor blade used in the above analyses have been developed by Hodges and Dowell (ref. 23), Kaza and Kvaternik (ref. 24), and Rosen and Friedmann (ref. 25). The models have elastic flap, lag, and torsion degrees of freedom. Nonlinearities owing to moderate deflections are also included. In these studies it was observed that for a given ordering

scheme, the final equations differ in small nonlinear terms, depending on the stage in the derivation process at which the ordering scheme is applied. The application of the ordering scheme at a later stage in the derivation process requires greater time in deriving and in independently checking the final equations. This has led to attempts to share the algebra with computers through symbolic processors. With such a processor to derive the governing equations, it will also be possible to study the effect of different ordering schemes on the calculated results in a minimum amount of time. Both general purpose and special purpose programs have been developed and are available as mentioned in Part I of this report. The HESL program, which is appropriate to rotory-wing aeroelastic investigation, has been developed in reference 11. The principles and method of organization of the program are explained in Part I of this report.

In the present report, the governing equations of motion are derived using HESL. The program generates the steady state and linearized perturbation equations in symbolic form and then codes them into FORTRAN subroutines. The coefficients for each equation and for each mode are identified through a numerical program. A Lagrangian formulation is used to obtain equations in generalized coordinates. The coupled flap-lag-torsion-dynamic inflow equations are converted to equations in a multiblade coordinate system by rearranging the coefficients in the case of hovering flight. The explicit multiblade equations are symbolically derived using HESL in the case of forward flight. The multiblade equations are capable of accommodating any number of elastic blade modes. The whole process, from derivation to numerical calculation, is automated with minimum user interface.

The hingeless rotor aeroelastic stability results presented in this report using the symbolic program reflect the combined effect of an improved structural model (by including torsion) and an improved aerodynamic model (by including dynamic inflow). Data are presented for several analytical models, a rigid blade lag-flap analysis, an elastic blade flap-lag analysis, an elastic blade flap-lag-torsion analysis, and a modified flap-lag analysis in which torsional inertia and damping are neglected (quasi-torsion). Damping data with and without dynamic inflow and from a quasi-static inflow model, in which dynamic inflow effects can be included as a lift deficiency function without increasing the system dimension, are included.

The following sections present the numerical study performed using both the symbolic program described in Part I and the numerical programs developed to analyze the resulting equations. The first part presents the procedure, solution methods, and numerical results obtained for hovering flight with and without dynamic inflow. The second part presents the same for forward flight.

Hovering Flight

Formulation- Figure 2 shows an elastic blade with the coordinate system used in this study. The blade has uniform mass and stiffness, no twist, and no chordwise offsets of the elastic axis, tension axis, or center of mass. The elastic axis is coincident with the x-axis of the x,y,z coordinate system which rotates with a constant angular velocity (Ω) about a fixed point at the origin. The y-axis lies in the plane of rotation, and the x-axis is rotated through a small angle (β_{pc})

25

from the plane of rotation. The deflections of the beam are u (axial), v (lagwise bending), and w (flapwise bending) of the elastic axis parallel to the x,y,z coordinates, respectively. A second coordinate system, x', y', and z' fixed to the blade, with y' and z' axes parallel to the beam cross section principal axes, moves with the blade as it undergoes bending displacements, torsional displacements, and pitch angle (θ) rotation. Before deformation, the principal axes of the blade are rotated with respect to the undeformed coordinates by the pitch angle. After deformation, the elastic axis is displaced by u,v,w, and the blade is twisted through the angle ϕ. The inflow dynamics couple with the blade dynamics as a feedback loop, as shown in figure 3. The total inflow (v_i) is assumed to consist of a steady value ($\bar{\lambda}$), and dynamic inflow components (v_0, v_c, and v_s) which vary with time.

In this study the entire problem formulation is performed by the computer with minimum user interface other than a specification of blade geometry and desired blade model representation. In general, the formulation of the rotary-wing aero-elastic problem consists of the following steps (see refs. 22-25 for greater detail): writing the transformation matrices between the coordinate systems before and after deformation; calculating the position vector of a mass point of the deformed blade section; forming the strain displacement relations; and calculating stresses and air velocity components in the flap, lag, and torsion, directions. These expressions include geometric nonlinearities owing to the assumption of small strains and moderate deflections which give rise to numerous higher-order nonlinear terms. So an ordering scheme, based on assigning orders of magnitude to the various physical parameters is used to reduce the number of terms. The governing equations of motion are then obtained using Hamilton's principle. These equations are non-linear, partial differential equations in u,v,w, and in torsion deflection. These are converted to ordinary differential equations by using Galerkin's method by expressing the bending and torsion deflections in terms of generalized coordinates and mode shape functions,

$$v = \sum_{j=1}^{N_L} RV_j(\psi)\psi_j(\bar{x})$$

$$w = \sum_{j=1}^{N_F} RW_j(\psi)\psi_j(\bar{x}) \tag{7a}$$

$$\phi = \sum_{j=1}^{N_T} \phi_j(\psi)\theta_j(\bar{x})$$

and by expressing the induced velocity as

$$v_i = \bar{\lambda} + v_0 + v_c\bar{x}\cos\psi + v_x\bar{x}\sin\psi \tag{7b}$$

26

where $\psi = \Omega t$, $\bar{x} = x/R$, and ψ_j, θ_j are mode shapes, R is the blade radius; and N_F, N_L, and N_T are the numbers of flap, lag, and torsion modes, respectively, used in the analysis. This yields N nonlinear, nonhomogeneous ordinary differential equations in terms of modal generalized coordinates V_j, W_j, and ϕ_j, where N is the total number of flap, lag, and torsion modes used in the analysis. The equations have constant coefficients for mass, damping, and stiffness in the case of hovering flight (and periodic coefficients in the case of forward flight). These equations are then linearized for small perturbation motions about the equilibrium position by expressing the generalized coordinates in terms of equilibrium quantities plus small perturbation quantities.

$$V_j(\psi) = V_{0j} + \Delta V_j(\psi)$$

$$W_j(\psi) = W_{0j} + \Delta W_j(\psi) \qquad (8)$$

$$\phi_j(\psi) = \phi_{0j} + \Delta \phi_j(\psi)$$

Two sets of equations are obtained from this operation: a set of N nonlinear algebraic equations in V_{0j}, W_{0j}, and ϕ_{0j} which define the equilibrium position and a set of N equations obtained by subtracting the equilibrium equations and discarding all nonlinear products of perturbation quantities.

Three more equations are obtained for the dynamic inflow components from rotor perturbations in aerodynamic thrust (C_T) and in pitch (C_M) and roll (C_L) moments (see dynamic inflow section). The coefficients of these equations are also functions of equilibrium solution. The quasi-static torsion model equations are obtained by dropping torsional inertia and damping terms in the torsion equation. The torsion equation is then solved for $\Delta \phi_j$, and the result is substituted into the flap-lag equations. This procedure from reference 26 is explained in Appendix B.

Dynamic Inflow- The total inflow is assumed to consist of a steady trim value and a dynamic inflow component as

$$V_i = \bar{\lambda} + \nu \qquad (9)$$

At any point on the rotor disk (x,ψ), the dynamic inflow is assumed to have a first harmonic representation in terms of inflow distribution (uniform, fore to aft, side to side) as

$$\nu = \nu_0 + \nu_c \bar{x} \cos \psi + \nu_s \bar{x} \sin \psi \qquad (10)$$

These velocity components assume the role of degrees of freedom and are linearly related to the unsteady thrust, pitching, and rolling moments by first-order differential equations:

$$[m] \begin{Bmatrix} \dot{\nu}_0 \\ \dot{\nu}_c \\ \dot{\nu}_s \end{Bmatrix} + [\ell]^{-1} \begin{Bmatrix} \nu_0 \\ \nu_c \\ \nu_s \end{Bmatrix} = \begin{Bmatrix} C_T \\ C_M \\ C_L \end{Bmatrix}_{aero} \qquad (11)$$

27

where the subscript aero indicates that C_T, C_M, and C_L are aerodynamic contributions only. The nonzero elements of $\{m\}$ and $\{\ell\}$ are (ref. 17)

$$m_{11} = 8/3\pi \; ; \quad m_{22} = m_{33} = -16/45\pi$$

$$\ell_{11} = 1/2\nu \; ; \quad \ell_{22} = \ell_{33} = -2/\nu \tag{12}$$

where $\nu = 2\bar{\lambda}$ and

$$\bar{\lambda} = \frac{a\sigma}{16} \left(\sqrt{1 + \frac{24}{a\sigma} |\theta - \phi|_{0.75\,R}} - 1 \right)$$

For rotors with a finite number of blades, however, blade loading rather than disk loading must be defined. Therefore, the disk loading is approximated by

$$\bar{C}_T = \frac{a\sigma}{\gamma b} \sum_{k=1}^{b} \left[\int_0^1 (\bar{L}_{wp})_k \, d\bar{x} \right]$$

$$C_M = \frac{-a\sigma}{\gamma b} \sum_{k=1}^{b} \left[\int_0^1 (\bar{L}_{wp})_k \bar{x} \, d\bar{x} \cos \psi_k \right] \tag{13}$$

$$C_L = \frac{-a\sigma}{\gamma b} \sum_{k=1}^{b} \left[\int_0^1 (\bar{L}_{wp})_k \bar{x} \, d\bar{x} \sin \psi_k \right]$$

where $\bar{L}_{wp}$ is the perturbed aerodynamic force in the flapping direction.

The quasi-static inflow formulation, in which the effects of apparent inflow mass are neglected, provides a means of approximating the unsteady inflow effects without increasing the system dimension. With momentum theory as a basis, quasi-static inflow theory leads to the method of an equivalent Lock number γ^* and drag coefficient C_d^* (refs. 17-20)

$$\gamma^* = \left[\gamma \frac{1}{1 + a\sigma/8\nu} \right] \tag{14}$$

$$\left(\frac{C_{d0}}{a} \right)^* = \left(\frac{C_{d0}}{a} \right) \left(1 + \frac{a\sigma}{8\nu} \right) + \left(\frac{a\sigma}{8\nu} \right) \left(\frac{6C_T^2}{a\sigma} \right)$$

This implies that inflow decreases the expected lift and increases the expected drag. It should be noted that γ^*, $(C_{d0}/a)^*$ are only in the perturbation equations and true γ, C_{d0}/a are used in calculating the time-independent hovering equilibrium displacements.

Equations through HESL- In this study, a Lagrangian formulation is used to obtain the governing nonlinear ordinary differential equations through symbolic manipulation. The input to the program are the relations, in alphanumeric format, for the position vector, strain expressions, air velocity components, and transformation matrices as given by Kaza and Kvaternik (ref. 24). The integration relations (if known), differentiation relations, the order of the variables, the ordering scheme to be used, and the variables for which coefficients are to be collected, are also given as data. The program calculates the strain energy, kinetic energy, and the generalized force for the given ordering scheme in generalized coordinates using assumed modes as in equation (7a). The order of the variables and the ordering scheme as prescribed in reference 15 is used in the present derivation. The perturbation relations as given in equation (8) are substituted to obtain the steady and perturbed terms.

The program generates both the steady-state and linearized perturbation equations and the loading terms necessary for calculating aeroelastic stability and response. The thrust, pitch-, and roll-moment equations required in the dynamic inflow equations are also generated using the perturbed aerodynamic forces in the flap direction, which are generated by the program. Contributions from the equilibrium displacements to the perturbation equations from each mode are included. The equations generated are written automatically into FORTRAN subroutines for subsequent numerical calculations. A numerical program subsequently identifies the mass, damping, stiffness, and load coefficients for each equation and for each mode. It should be noted that for a hingeless rotor, the axial displacement can be solved for a priori as a function of flap bending and lag bending. In the present study, expressions for axial displacement and axial velocity are taken from reference 15 and are supplied as data to the program.

For the results presented here, the program was run on an IBM 360. It took approximately 450 sec to derive structural terms and 165 sec for aerodynamic terms with inflow (60 sec without inflow) for an elastic flap-lag-torsion analysis. It then took 25 sec to write the individual matrix elements into the subroutines for subsequent analysis.

Sample input and output- A brief description of the program input and output follows. Table 2 shows the FORTRAN symbol definition used for the original variables. Table 3 shows the input to calculate tangential and perpendicular velocities using the transformation matrix LAFP(READ MATRIX) and air velocity vector VEL(READ MATRIX). By multiplying the two matrices (FOTM MATRIX) with ordering scheme *E2D1, the vector AVEL is obtained which gives the components of the velocities in radial, tangential, and perpendicular directions. The vector components are redefined as expressions by command MATRIX EXPRESSIONS, and since the actual components are a negative form of the original expressions, the expressions are negated by calling the NEGATE command, giving the actual velocity expressions.

The following points are to be noted in giving the data to the symbolic program. (1) In hover, the nonlinear rate terms do not contribute to the steady solution or the linearized stability solution. Hence, the number of nonlinear terms and perturbation relations may be smaller. (2) By redefining the orders of variables at proper stages of the input, higher order terms can be retained in lag and torsion equations.

Method of solution- The bending and torsion mode shapes used in equation (7) are taken as the uncoupled nonrotating mode shapes of a cantilever beam (ref. 26). A total of 15 (N = 15) uncoupled nonrotating modes are used in calculating the steady state deflection. The integrals in the problem are evaluated numerically with a 16-point Gaussian integration scheme. The steady state equations obtained using equation (8) are time-independent in hovering flight. They are nonlinear, algebraic equations in V_{0j}, W_{0j}, and ϕ_{0j}. These equations are solved iteratively using Brown's algorithm (ref. 27). This algorithm is a modified form of the Newton-Raphson method, and the solution is estimated such that the sum of the squares of the errors is a minimum. The solution to the linear equations is used as the initial estimate in the solution procedure.

The aeroelastic stability of the blade's motion about the equilibrium position is determined by the eigenvalues of the perturbation equations (ΔV_j, ΔW_j, $\Delta \phi_j$, ν_0, ν_c, and ν_s). This results in (($2N+3$)x($2N+3$)) matrix for the eigenvalue analysis. As explained in reference 15, this can be reduced to a (($M+3$)x($M+3$)) matrix by using the lowest M eigenvectors of the free vibration problem. In the present study, six (M = 6) eigenvectors, based on the free vibration analysis using 15 (N = 15) uncoupled nonrotating modes, are used. The reduced matrices are analogous to stiffness and damping matrices generated from M-coupled, rotating-blade modes. These equations (with M generalized coordinates, say x_1, x_2, ... x_M) are then transformed to multiblade coordinate equations. The coupled blade and dynamic inflow equations are then solved using an eigenanalysis for stability of the rotor.

Multiblade coordinate transformation- To avoid periodicity, and to have a better understanding of the rotor behavior, it is necessary to express the final equations in a multiblade coordinate system (a nonrotating frame of reference). As pointed out in reference 28, for a constant-coefficient system (as in hover), the multiblade coordinate transformation acts only on the degrees of freedom. Let (X_j) represent the M-generalized coordinates (obtained above) in the rotating frame,

$$\{X_j\} = \{X_1, X_2, X_3, X_4 \ldots X_M\} \qquad (15)$$

Then the equations for kth blade including dynamic inflow are

$$[M]\{\ddot{X}_{jk}\} + [C]\{\dot{X}_{jk}\} + [k]\{X_{jk}\} + [T]\begin{Bmatrix} \nu_0 \\ \nu_c \cos \psi_k \\ \nu_s \sin \psi_k \end{Bmatrix} = 0 \qquad (16a)$$

30

$$[m]\left\{\begin{array}{c}\dot{\nu}_0\\ \nu_c\\ \nu_s\end{array}\right\} + [\ell]^{-1}\left\{\begin{array}{c}\nu_0\\ \nu_c\\ \nu_s\end{array}\right\} = \left\{\begin{array}{c}\bar{C}_T\\ C_M\\ C_L\end{array}\right\} \qquad (16b)$$

$\bar{C}_T$, C_M, and C_L can be expressed in terms of the blade degrees of freedom and inflow degrees of freedom as

$$\bar{C}_T = \frac{a\sigma}{\gamma b}\sum_{k=1}^{b}(A_{1j}\ddot{X}_{jk} + B_{1j}\dot{X}_{jk} + C_{1j}X_{jk} + G_{1i}u_i)$$

$$C_M = \frac{-a\sigma}{\gamma b}\sum_{k=1}^{b}(A_{2j}\ddot{X}_{jk} + B_{2j}\dot{X}_{jk} + C_{2j}X_{jk} + G_{2i}u_i)\cos\psi_k \qquad (16c)$$

$$C_L = \frac{-a\sigma}{\gamma b}\sum_{k=1}^{b}(A_{3j}\ddot{X}_{jk} + B_{3j}\dot{X}_{jk} + C_{3j}X_{jk} + G_{3i}u_c)\sin\psi_k$$

where X_{jk} are the generalized coordinates of kth blade; j runs from 1 to M, the number of coupled rotating-blade modes used; i is 1 to 3, the inflow degrees of freedom, and $U_1 = \nu_0$, $U_2 = \nu_c \cos\psi_k$; and $U_3 = \nu_s \sin\psi_k$. It should be noted that matrices [M], [C], [K], [T], [A], etc., are already reduced to MxM matrices using the eigenvectors of a free vibration of coupled rotating beam, as explained in the previous section and in reference 15.

Each generalized coordinate X_{jk} is written in multiblade coordinates as

$$X_{jk} = X_{0j} + X_{cj}\cos\psi_k + X_{sj}\sin\psi_k$$

$$\dot{X}_{jk} = \dot{X}_{0j} + (\dot{X}_{cj} + X_{sj})\cos\psi_k + (\dot{X}_{sj} - X_{cj})\sin\psi_k \qquad (17)$$

$$\ddot{X}_{jk} = \ddot{X}_{0j} + (\ddot{X}_{cj} + 2\dot{X}_{sj} - X_{cj})\cos\psi_k + (\ddot{X}_{sj} - 2\dot{X}_{cj} - X_{sj})\sin\psi_k$$

where X_{0j} represents collective mode, X_{cj} and X_{sj} represent cyclic modes in the nonrotating frame.

31

Let q_0 be the vector of the collective modes of all modes; let q_c be the vector of the longitudinal cyclic modes of all modes; and let q_s be the vector of the lateral cyclic modes of all modes. Then the multiblade coordinate transformation, for a three-bladed rotor is given by

$$\begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{bmatrix} \begin{Bmatrix} \ddot{q}_0 \\ \ddot{q}_c \\ \ddot{q}_s \end{Bmatrix} + \begin{bmatrix} C & 0 & 0 \\ 0 & C & 2M \\ 0 & -2M & C \end{bmatrix} \begin{Bmatrix} \dot{q}_0 \\ \dot{q}_c \\ \dot{q}_s \end{Bmatrix}$$

$$+ \begin{bmatrix} K & 0 & 0 \\ 0 & k-M & C \\ 0 & -C & K-M \end{bmatrix} \begin{Bmatrix} q_0 \\ q_c \\ q_s \end{Bmatrix} + \begin{bmatrix} T_{j1} & 0 & 0 \\ 0 & T_{j2} & 0 \\ 0 & 0 & T_{j3} \end{bmatrix} \begin{Bmatrix} \nu_0 \\ \nu_c \\ \nu_s \end{Bmatrix} = 0 \qquad (18a)$$

$$[m] \begin{Bmatrix} \dot{\nu}_0 \\ \dot{\nu}_c \\ \dot{\nu}_s \end{Bmatrix} + [\ell]^{-1} \begin{Bmatrix} \nu_0 \\ \nu_c \\ \nu_s \end{Bmatrix} = \frac{-a\sigma}{2\gamma} * \left\{ \begin{bmatrix} -2A_{ij} & 0 & 0 \\ 0 & A_{2j} & 0 \\ 0 & 0 & A_{3j} \end{bmatrix} \begin{Bmatrix} \ddot{q}_0 \\ \ddot{q}_c \\ \ddot{q}_s \end{Bmatrix} \right.$$

$$+ \begin{bmatrix} -2\beta_{ij} & 0 & 0 \\ 0 & \beta_{2j} & 2A_{2j} \\ 0 & -2A_{3j} & \beta_{3j} \end{bmatrix} \begin{Bmatrix} \dot{q}_0 \\ \dot{q}_c \\ \dot{q}_s \end{Bmatrix}$$

$$+ \begin{bmatrix} -2C_{ij} & 0 & 0 \\ 0 & C_{2j}-A_{2j} & B_{2j} \\ 0 & -B_{3j} & C_{3j}-A_{3j} \end{bmatrix} \begin{Bmatrix} q_0 \\ q_c \\ q_s \end{Bmatrix}$$

$$\left. + \begin{bmatrix} -2G_{11} & 0 & 0 \\ 0 & G_{22} & 0 \\ 0 & 0 & G_{33} \end{bmatrix} \begin{Bmatrix} \nu_0 \\ \nu_c \\ \nu_s \end{Bmatrix} \right\} \qquad (18b)$$

For a four-bladed rotor, the equation for the differential collective mode is not coupled to the equations for the other modes (in hover only). Consequently, the same equations apply for four-bladed rotors also, and to rotors with any number of blades greater than, or equal to, 3.

The equations are written for clarity as

$$[P]\{\ddot{Y}\} + [Q]\{\dot{Y}\} + [R]\{Y\} + [T]\{u\} = 0 \qquad (19a)$$

32

$$[A]\{\ddot{Y}\} + [B]\{\dot{Y}\} + [C]\{Y\} + [G]\{u\} = [m]\{\dot{u}\} + [\ell]^{-1}\{u\} \qquad (19b)$$

$$Y = q_0, q_c, q_s \quad , \qquad u = \nu_0, \nu_c, \nu_s$$

In state vector form, this becomes

$$\left\{\begin{array}{c} \dot{Y} \\ \dot{Y} \\ u \end{array}\right\} = \left[\begin{array}{ccc} 0 & 0 & 0 \\ -P^{-1}R & -P^{-1}Q & -P^{-1}T \\ m^{-1}(C - AP^{-1}R) & m^{-1}(B - AP^{-1}Q) & m^{-1}(G - \ell^{-1} - AP^{-1}T) \end{array}\right] \left\{\begin{array}{c} Y \\ \dot{Y} \\ u \end{array}\right\} \qquad (20)$$

or

$$\dot{Z} = [A_s]Z$$

where $[A_s]$ is a constant-coefficient matrix. The real part of the eigenvalues of $[A_s]$ gives the modal damping information.

It can be observed from equations (18a) and (18b) that the collective mode decouples from the progressing and regressing modes. This fact can be used for efficient programming in the calculation of eigenvalues.

Results and Discussion

To demonstrate the capability of the symbolic processor program discussed, the aeroelastic stability of a hingeless rotor system in hover is investigated. As schematically shown in figure 1(a), the numerical results presented here are obtained directly from the computer-generated state equations. Results are compared with previously published damping data to validate the program. Next, the influence of the structural model with and without dynamic inflow is studied. In the derivation of equations, the orders of magnitude assigned for each parameter is the same as that followed in reference 15. The parameters used for numerical study are

$$\omega_v = 0.7, 1.5 \ , \qquad \omega_w = 1.15 \ , \qquad \omega_\phi = 3, 5 \ , \qquad K_{m1}/K_{m2} = 0 \ ,$$

$$K_m/R = 0.025 \ , \qquad c/R = 0.07854 \ , \qquad \gamma = 5.0 \ , \qquad \sigma = 0.05, 0.1 \ ,$$

$$a = 2\pi \ , \qquad C_{do} = 0.01 \ , \qquad \beta_{pc} = 0.0 \ , \qquad (K_A/K_m)^2 = 1.15$$

Figures 4 and 5 compare the lead-lag damping values obtained in reference 5 from the equations of Hodges and Dowell (ref. 12), and from the computer-generated equations. The results are presented for two torsional frequencies ($\omega_\phi = 2.5, 5.0$) for both soft in-plane ($\omega_v = 0.7$, fig. 4) and stiff in-plane ($\omega_v = 1.5$, fig. 5) configurations. The other parameters are $\omega_w = 1.15$, $R = 1.0$, and $\sigma = 0.1$. It can

33

be seen that each set of equations gives almost identical results for the first mode lead-lag damping in the operating range of $\theta = 0$ to 0.4 rad. The slight differences are a result of the following. In deriving the computer-generated equations, basic expressions have been taken from Kaza and Kvaternik (ref. 24). The equations of references 23 and 24 differ in the structural terms $(\phi'w'GJ)''$ in the lag equation of motion, $-(\phi'v''GJ)'$ in the flap equation of motion, and $(v''w'GJ)'$ in the torsion equation of motion. The aerodynamic $(\rho ac/2)(c/2)[\Omega x(\theta + \phi) + v_i]\dot\phi$ term in the lag equation and $(\rho ac/2)\Omega^2 x^2 v'w'$ in the flap equation differ between the two aerodynamic models. The term

$$\left(\frac{\rho ac}{2}\right)\Omega^2 x^2 \int_0^x v'w'' \, dx$$

does not appear in the present flap equation which is included in reference 23. The kinematics of these two sets of equations are discussed in references 29 and 30.

The results from these two comparisons provide a validation of the scheme for symbolic generation of the equations. Results are now presented for more complex rotor modeling problems using the computer-generated equations.

Elastic blade versus rigid blade (flap-lag model)- The elastic flap-lag-torsion equations are reduced to flap-lag equations by dropping all the torsion terms. Three modes each for flap and lag are used in the numerical calculations. By carefully studying the rigid blade equations (ref. 13) and the elastic blade flap-lag equations (ref. 15), it can be seen that the elastic blade equations have additional stiffness contributions in the aerodynamic expressions. This coupling of aerostiffness is a function of both flap and lag trim values. Figures 6 and 7 show a comparison of rigid blade and elastic blade results for soft in-plane and stiff in-plane rotors, respectively. These figures show the lead-lag mode damping plotted versus pitch angle for a rigid blade model, for an elastic blade model, and for an elastic blade model with the aerostiffness terms dropped in the elastic blade equations. Also shown is lead-lag mode damping plotted for a rigid blade model with no elastic coupling effects. It can be observed for both configurations that the elastic blade model without the aerostiffness terms shows good correlation with the rigid blade equation results, since in the rigid blade model, there is no coupling aerostiffness term. The aerostiffness terms seem to have a destabilizing effect on the stability. As noted in reference 12, the rigid blade model with no elastic couplings (R = 0.0) yields poor correlation with the elastic blade model for both rotors. It should be noted that the damping curves from the present study for the stiff in-plane blade in figure 7 are identical to those given in reference 12, again validating the symbolic derivation and numerical results of this study.

The following paragraphs and figures study different blade structural models with and without dynamic inflow. The convention used for the figures is one which uses solid lines to refer to results without dynamic inflow and dashed lines to show results with dynamic inflow.

Rigid blade versus elastic blade with dynamic inflow (flap-lag model)- Figure 8 shows the lead-lag regressing mode damping for the flap-lag model (all torsion terms dropped) with dynamic inflow for a rigid blade model, elastic blade model, and an elastic blade model with the aerostiffness terms dropped. Again, it can be seen that the elastic blade flap-lag equations without aerostiffness terms give good agreement with rigid blade results. However, the elastic blade is always less stable than is the rigid blade model. It may be noted that the rigid blade results with dynamic inflow are identical to those presented in reference 18 for zero elastic coupling (R = 0.0).

Elastic flap-lag-torsion with and without dynamic inflow- Figures 9 and 10 show the lead-lag regressing mode damping plotted for varying lag frequency for two torsional frequencies (ω_ϕ = 3.0, 5.0). Results for an elastic flap-lag model are also shown. By comparing the no dynamic inflow curves with those given in reference 15 (figs. 42 and 44), it can be seen that for the results of this study the solid-line curves do not cross at the lag frequency that is equal to the matched stiffness value (ω_v = 0.57). This is due to the following reason. In the present study, the kinematic pitch-flap and pitch-lag couplings, as mentioned in reference 15, are not at zero even for the matched stiffness case. There is a contribution to these couplings that is equal to the torsional rigidity times the bending trim values owing to the terms mentioned in the second paragraph.

Figure 9 shows the lead-lag regressing mode damping plotted for zero elastic coupling for different torsional frequencies (ω_ϕ = 3.0, 5.0), and for a flap-lag model, with and without dynamic inflow. It can be observed for the flap-lag model that the dynamic inflow effect reduces with increasing lag frequency and is always stabilizing. For a torsion frequency of 5.0, the dynamic inflow effects are stabilizing, and the effect increases with increasing lag frequency. For a torsion frequency of 3.0, the dynamic inflow effects are zero or slightly negative, but are stabilizing with increasing lag frequency (ω_v > 0.9).

Figure 10 shows the lead-lag regressing mode damping plotted for the same parameters but for full elastic coupling. It is observed that dynamic inflow again increases damping for torsionally rigid blades, and this effect is reduced with increased lag frequency as in the previous case. But for torsionally flexible blades, dynamic inflow reduced damping for both the torsion frequencies considered, except for high values of lag frequency.

Figure 11 shows the lead-lag regressing mode damping plotted for varying lag frequency for three values of the elastic coupling parameter for a blade with torsion frequency, ω_ϕ = 5.0. It can be seen that for an elastic coupling value of 0.4, the dynamic inflow effects are almost negligible, except for 0.9 < ω_v < 1.2. This effect of the elastic coupling parameter R on damping with dynamic inflow is further investigated in figures 12 and 13. Here the lag regressing and flap regressing mode damping are plotted with varying elastic coupling for a flap-lag model and for a flap-lag-torsion model with torsional frequencies of 3.0 and 5.0. The lead-lag frequency is ω_v = 1.4. Figure 12 shows the lead-lag regressing mode damping. It can be seen that for torsionally flexible blades, depending on the value of the coupling parameter R, dynamic inflow can be stabilizing (R < 0.3) or

destabilizing (R > 0.4). Consequently, there exists an elastic coupling value for which dynamic inflow effects on lead-lag damping are negligible. This can also be interpreted as showing that for torsionally soft blades, the dynamic inflow reduces the magnitude of the lag mode damping regardless of sign. The value of R for which dynamic inflow effects are negligible is where the lag damping is approximately zero. Noting that the points at which lag damping with and without dynamic inflow are zero and almost identical, it may be stated that dynamic inflow does not have a significant impact on the stability boundary. These results are different from those for the elastic flap-lag blade for which dynamic inflow effects are always stabilizing for the lead-lag mode. In figure 13, the flap regressing mode is reduced with dynamic inflow for all values of elastic coupling parameter (0.0 < R < 1.0), as expected.

The influence of the dynamic inflow model is further investigated in figure 14. The lead-lag regressing mode damping is plotted for varying elastic coupling values for a dynamic inflow model and for a quasi-static inflow model. Also shown are the damping values obtained by taking $\theta = \bar{\lambda}$ (eq. (12)), as in reference 21. The dynamic inflow models of references 17 and 21 give similar results, particularly at high elastic coupling parameters. The quasi-static model is a reasonable approximation for this case.

Figure 15 shows the lead-lag regressing mode damping plotted from a quasi-static torsion model with and without dynamic inflow ($\omega_\phi = 5.0$). It also shows a comparison with a coupled flap-lag-torsion analysis. The quasi-static torsion model has been obtained by dropping torsional inertia and by damping the terms in the torsion equation. The torsion equation is solved for the perturbed torsion deflection ($\Delta\phi_j$), and this is substituted in flap-lag equations. It can be seen that the quasi-static torsion model exhibits the same qualitative behavior as the flap-lag-torsion model does, although the in-plane damping is uniformly reduced for all values of R. It may also be observed that dynamic inflow again reduces damping for elastic parameter values greater than 0.4.

As was explained in reference 15, the quasi-static torsion model attempts to represent the torsion effect as equivalent kinematic pitch-flap and pitch-lag couplings. These couplings depend on the blade's torsion frequency, the elastic coupling parameter, equilibrium bending deflections, and the difference in lead-lag and flap bending stiffness. In the present study it has been shown earlier that these couplings also depend on torsional rigidity. For clarity, a rigid blade model with pitch-flap and pitch-lag couplings is studied. Results are presented with and without dynamic inflow for a varying elastic coupling parameter. Figure 16 shows the lead-lag regressing mode damping versus the elastic coupling parameter from a rigid blade flap-lag model having pitch-flap and pitch-lag couplings with and without dynamic inflow. Results from the quasi-static torsion model are also plotted for comparison. For a given flap, lag, and torsion frequency, the values of pitch-flap and pitch-lag couplings are calculated from equation (40) of reference 15. For comparison, the trim-bending values obtained from the quasi-static torsion model are used to calculate these couplings. For $\omega_v = 1.4$, $\omega_w = 1.15$, $\omega_\phi = 5.0$, and $\theta = 0.3$, it is observed that the value of pitch-flap coupling varies from +0.125

36

for $R = 0.0$ to -0.414 for $R = 1.0$. The pitch-lag coupling varies from -1.4602 for $R = 0.0$ to -1.497 for $R = 1.0$. Comparing the curves for both the models with no dynamic inflow, it can be seen that the rigid blade model yields conservative results. With dynamic inflow, both models qualitatively exhibit the same trend, namely, that a reduction in damping results with increasing elastic coupling for $R > 0.5$. It may be concluded from figures 15 and 16 that a rigid blade flap-lag model with pitch-flap and pitch-lag couplings, rather than a model with coupled flap-lag-quasi-static torsion for full flap-lag-torsion, may in some cases be sufficient to yield qualitative trends.

Summary of Hover Study

The combined effect of blade torsion and dynamic inflow on the aeroelastic stability of an elastic rotor blade in hover has been studied. The governing equations of motion of the elastic blade with flap-lag-torsion degrees of freedom and the dynamic inflow equations are derived using a symbolic processor written in FORTRAN. The blade and the dynamic inflow equations are converted to equations in a multiblade coordinate system by rearranging the coefficients of the equations. Conclusions drawn from this study are presented in the last section of this report.

Forward Flight

In forward flight the rotor does not have an axisymmetry flow environment as it does in hover. The velocities of the rotor blade not only depend on the blade motion and on rotational speed but also on the on-coming air speed. These additional components of air velocity are shown in figure 17. The advancing blade has a velocity relative to the air higher than the rotational velocity, whereas the retreating blade has a lower velocity relative to the air. The aerodynamic forces vary periodically as the blade rotates. This causes asymmetry of the loads. The rotor blade loading and motion are periodic with a fundamental frequency equal to the rotor speed Ω. The analysis is more complicated because of the dependence of the loads and motion on the azimuth angle ψ. A phenomenon which can be introduced by forward flight is the reverse-flow region, an area on the retreating side of the rotor disk where the velocity relative to the blade is directed from the trailing edge to the leading edge. In general, the reverse-flow region is defined as the area of the disk where the tangential velocity is less than zero, which has the boundary $\bar{x} + \mu \sin \psi = 0$. This yields a circular region with a diameter μ, centered at $x = \mu/2$, and $\psi = 270$ deg on the retreating side. Aeroelastic studies in forward flight are consequently more complex than are hover studies. The governing equations are lengthy and have periodic coefficients. However, as in hover, reliable solutions can be obtained by linearizing the nonlinear equations about a steady-state equilibrium position. In forward flight this position is a time-dependent periodic solution. Calculation of this time-dependent equilibrium position is inherently coupled with the trim state of the complete helicopter in forward flight.

Formulation of forward flight stability analysis- The blade model remains the same as described for hover and shown in figure 2. The blades undergo axial (u),

lagwise benging (v), and flapwise bending (w) and torsion deflection (φ). The aerodynamic forces will have additional contributions from the airspeed. These contributions have been shown in figure 17. The general formulation of the problem follows the same procedure as the one for hover. However, in forward flight the implementation of Galerkin's method requires the rotating modes to be calculated at every instant of time for a given pitch. It is proposed here to derive the governing equations of motion using the uncoupled rotating modes evaluated at zero pitch. The elastic degrees of freedom v, w, and φ are expressed in terms of the generalized coordinates and mode shape functions as

$$w = \sum_{i=1}^{N_F} Rw_i(\psi)\eta_i(\bar{x})$$

$$v = \sum_{i=1}^{N_L} Rv_i(\psi)\zeta_i(\bar{x})$$

(23)

$$\phi = \sum_{i=1}^{N_T} \phi_i(\psi)\theta_i(\bar{x})$$

where $\psi = \Omega t$, $\bar{x} = x/R$, η_i, ζ_i, and θ_i are rotating mode shapes; N_F, N_L, and N_T are the number of flap, lag, and torsion modes used in the analysis. These mode shapes are obtained from five nonrotating modes for each degree of freedom. In the actual derivation of the equations through HESL, the integral quantities are evaluated by the numerical program using these modes. Two rotating modes are used in this study for each flap, lag, and torsion degree of freedom.

Equations through HESL- The derivation of the governing equations of motion for a single blade using HESL is the same as described for hovering flight. The only change in data input is the air velocity components which are caused by the air speed. The data for the position vector, strain expressions, air velocity components, and transformation matrices from reference 24 are given as the basic data. The displacements as given in terms of the generalized coordinates (eq. (23)) are substituted and the equations are derived using a Lagrangian approach. The perturbed equations are obtained by substituting the relations given by equation (8). The nonlinear single blade equations are written to a separate file, to be used in the time-dependent equilibrium position calculation. The perturbed equations are similarly written to a file to be used in a multiblade coordinate transformation. The relations required in dynamic inflow calculations, thrust (C_T), pitching moment (C_M), and rolling moment (C_L) are derived using the perturbed aerodynamic force in the flap direction. The equations are lengthy because of the contributions from airspeed. The program generates the steady and perturbed equations in a single

38

operation and outputs them individually. This is convenient in the case of forward flight not only because of the large number of terms present in each equation, but also for the different analysis processes that are required for the steady and perturbed sets of equations.

In the present study, the order of the variables and the ordering scheme used are same as those followed in reference 15. All the $o(\epsilon^2)$ terms, when compared to $o(1)$, except those that contribute to lead-lag and torsion damping are neglected. Nonlinear rate products ($v\dot{w}$, $\dot{v}^2$, etc.) are retained since they contribute to the linearized stability analysis. Although any general ordering scheme could have been used to obtain the final equations of motion, this ordering scheme is considered representative and adequate for demonstrating the capability of the symbolic analysis process. For the results presented here, it took approximately 300 sec to symbolically derive both the structural and aerodynamic equations on a VAX 11/780 computer. It should be noted that for a hingeless rotor, the axial displacement can be solved for a priori as a function of flap and lag bending. In this report, expressions for axial displacement and axial velocity are taken from reference 15 and supplied as data to the program.

Dynamic inflow modeling- The dynamic inflow equations are related to the blade degrees of freedom through the variations in thrust, pitching and rolling moments

$$[m]\left\{\begin{array}{c} \dot{\nu}_0 \\ \dot{\nu}_c \\ \dot{\nu}_s \end{array}\right\} + [\ell]^{-1}\left\{\begin{array}{c} \nu_0 \\ \nu_c \\ \nu_s \end{array}\right\} = \left\{\begin{array}{c} \bar{C}_T \\ C_M \\ C_L \end{array}\right\} \tag{24}$$

The elements of (m) and (ℓ) define the various dynamic inflow models that can be included in an analysis. Reference 19 presents a hierarchy of models, having different elements for (m) and (ℓ) from actuator disk theory in forward flight. The elements of (ℓ) depend on the wake skew angle at the rotor

$$\alpha_R = \tan^{-1}\frac{\bar{\lambda}}{\mu} \tag{25}$$

where $\bar{\lambda}$ is the steady inflow. Of the 13 models presented in reference 19, the partially constrained model gave good results. In the present report, this partially constrained theory is used to obtain the dynamic inflow results. The elements of (m) and (ℓ) are given by

$$m_{11} = \frac{128}{75}\pi \ , \quad m_{22} = m_{33} = \frac{-16}{45}\pi \ , \quad m_{ij} = 0 \ , \quad i \neq j$$

$$\ell_{11} = \frac{1}{2} \ , \quad \ell_{13} = \frac{15\pi}{64}\sqrt{\frac{1 - \sin\alpha_R}{1 + \sin\alpha_R}} \ , \quad \ell_{22} = \frac{-4}{1 + \sin\alpha_R} \tag{26}$$

$$\ell_{31} = \ell_{13} \ , \quad \ell_{33} = \frac{-4 \sin \alpha_R}{1 + \sin \alpha_R} \ , \quad \ell_{12} = \ell_{21} = \ell_{23} = \ell_{32} = 0$$

$$\ell_{ij} = \frac{1}{\nu} \ell_{ij} \ , \quad \nu = \frac{[\mu^2 + \bar{\lambda} \ (2\bar{\lambda} - \mu \tan \alpha)]}{(\sqrt{\lambda^2 + \mu^2})}$$

Trim and periodic equilibrium solution- The nonlinear periodic coefficient equations obtained earlier can be solved for the periodic response in the time domain using a Floquet method or in the frequency domain using a harmonic balance method. Either will yield the time-dependent equilibrium position about which the nonlinear equations can be linearized for an eigensolution. In forward flight, this equilibrium position is coupled with the entire trim state of the helicopter. The trim state is the steady state condition achieved by the system as time increases without bound with the controls fixed and no external output present. Calculation of a trim position requires establishing the control settings for a given flight condition. The control settings are collective pitch, longitudinal and lateral cyclic pitch, and the rotor shaft angle of attack. The induced velocity which depends on the generated thrust and advance ratio is also calculated. In this report, the harmonic analysis method coupled with an iteration on the trim state is employed to calculate the equilibrium position and the trim settings. This numerical procedure, taken from reference 26 is as follows. An iterative inner loop in which the solution for the periodic motion is obtained with fixed values of the trim variables is nested within an outer loop in which the solution for the trim variables is obtained. The rotorcraft motion is solved for the periodic motion by the harmonic analysis method which directly calculates the harmonics of a Fourier series representation of the motion. The procedure advances the rotor around the azimuth calculating the forcing functions in the time domain and then updating the harmonics of the response. The forces and moments acting on the rotor are calculated from this response, and the controls are adjusted until the equilibrium of forces and moments that are required for the specified operating state is achieved.

For the harmonic analysis method, the governing equations of motion are written with all the time-dependent and nonlinear terms as a forcing function, as

$$[M]\{\ddot{X}\} + [C]\{\dot{X}\} + [K]\{X\} = F(\dot{X}, X, \psi) \tag{27}$$

where matrices M, C, and K are the constant parts of the mass, damping and stiffness matrices, and where X is the vector of degrees of freedom. The function F is evaluated at J points around the rotor azimuth

$$F_j = F(\psi_j) \tag{28}$$

and the harmonics of a complex Fourier series representation of F are given by

$$F_n = \frac{1}{J} \sum_{j=1}^{J} F_j \, e^{-in\psi_j} \qquad (29)$$

Then the nth harmonic of the motion is given by

$$X_n = H_n^{-1} F_n \qquad (30)$$

where $H = K - n^2 M + iCn$. The iterative solution proceeds as follows. At a given azimuth ψ_j, the blade motion is calculated using the current estimates of the harmonics

$$X = \sum_{n=-L}^{L} X_n \, e^{in\psi_j} \qquad (31)$$

$$\dot{X} = \sum_{n=-L}^{L} X_n \, e^{in\psi_j}$$

where L is the number of terms used in the expansion of X. The forcing function is evaluated with this motion. Then the change in the harmonics caused by the change in the forcing function is calculated and added to the harmonics calculated in the last revolution. After every revolution the old and new harmonics are checked until convergence is achieved.

After obtaining the harmonics of the blade motion, the equilibrium of the rotor forces and moments is checked. If overall equilibrium is not satisfied, the trim settings are changed and the procedure is repeated until equilibrium is obtained. The following assumptions are made in arriving at the equilibrium of the forces and moments. The helicopter is in straight and level steady flight, the rotor hub moments are trimmed to zero with fuselage and other aerodynamic contributions neglected. Then the equilibrium of forces and moments are given by

$$\left. \begin{array}{c} \bar{C}_T \cos \alpha + C_H \sin \alpha = C_W \\[2mm] -\bar{C}_T \sin \alpha + C_H \cos \alpha = -C_X \\[2mm] C_{mx} = C_{my} = 0.0 \end{array} \right\} \qquad (32)$$

where $C_X = (1/2)\bar{f}\mu^2$ and α is the shaft angle of attack.

In calculating thrust, horizontal force, and the hub moments, the steady inflow appears as a parameter which in turn depends on the rotor thrust and shaft angle of attack. In this study, the study inflow is taken as an equation of constraint and solved along with the four equations above.

$$\bar{\lambda} - \mu \tan \alpha - \left[C_T \Big/ 2 \sqrt{(\bar{x})^2 + \mu^2} \right] = 0.0 \qquad (33)$$

The increments in the trim settings are calculated using a modified Newton-Raphson method (ref. 30). If v is the control variable and M is the target to be achieved, then a first-order approximation of $M(v)$ is

$$M_{target} = M_{n+1} = M_n + (v_{n+1} - v_n) \quad \text{or} \quad v_{n+1} = v + D^{-1}(M_{target} - M_n)f \qquad (34)$$

where v_n and v_{n+1} indicate the current and new estimates of v respectively and $f < 1$ is included to avoid overshoot oscillations in the trim iteration by reducing the step size. The partial derivative matrix D is

$$D = \frac{\partial M}{\partial v} = \left[\cdots \frac{\partial M}{\partial v_i} \cdots \right] = \left[\cdots \frac{M(v_i) - M(v_i - \Delta v_i)}{\Delta v_i} \cdots \right] \qquad (35)$$

where v_i is the ith control variable and Δv_i is its increment. The convergence is checked when the tolerence level as specified by the parameter ε is met. For more details on this procedure see reference 30. In the present study, all the degrees of freedom, blade flap, lead-lag and torsion degrees of freedom are used in calculating the trim state and periodic response.

The periodic response can also be obtained using Floquet methods (ref. 16). A review of these methods to obtain the initial conditions, forced response and stability data is given in reference 31. Explicit harmonic balance equations can also be derived symbolically using the program HESL. This requires the definition of the harmonics for each degree of freedom and a table of trigonometric relations giving the products of sines and cosines as sums of sines and cosines. Then the SUBSTITUTE TABLE INTO RELATIONS command substitutes these relations into the governing equations. A numerical program collects the coefficients of sines and cosines and forms the harmonic balance equations. However, this procedure requires modification of the data for each degree of freedom when a larger number of harmonics is involved. Hence, HESL is not convenient to use to derive explicit harmonic balance equations and numerical method as explained above is employed in this study.

Aeroelastic stability solution (single blade)- Once the time dependent equilibrium position is determined, the nonlinear equations are perturbed about this equilibrium position, equation (8). As mentioned earlier, the symbolic program generates the perturbed equations and writes them into subroutines. It should be noted that squares of the perturbation quantities are typically neglected by the symbolic

program when employing a realistic ordering scheme. The final system of equations for stability is

$$[P]\{\Delta\ddot{q}\} + [Q]\{\Delta\dot{q}\} + [R]\{\Delta q\} = 0 \qquad (36)$$

or

$$\dot{Z} = [A_s]Z$$

The stability of this linearized system is determined from Floquet theory by evaluating the characteristic exponents of $[A_s]$

$$\lambda_k = \zeta_k + i\omega_k \qquad (37)$$

The mass, damping and stiffness terms are identified by a numerical program for each time step. The linearized system is stable when $\zeta_k < 0$.

Multiblade coordinate transformation- It is necessary to convert the equations into a fixed coordinate system to provide a better understanding and to include dynamic inflow effects which are referenced to the fixed system. The multiblade (Fourier) coordinate transformation is a linear transformation of the degrees of freedom from the rotating to nonrotating frame. In the case of hovering flight, this transformation has been easily performed by rearranging the coefficients of the equations, since the coefficients are constant. For forward flight the coefficients are periodic. Hence the multiblade coordinate transformation is more complicated. Let X be a generalized degree of freedom (dimensionless) in the rotating frame for the ith blade. Then for a three-bladed rotor, the relations

$$X_i = X_0 + X_c \cos \psi_i + X_s \sin \psi_i$$

$$\dot{X}_i = \dot{X}_0 + (\dot{X}_c + X_s)\cos \psi_i + (\dot{X}_s - X_c)\sin \psi_i \qquad (38)$$

$$\ddot{X}_i = \ddot{X}_0 + (\ddot{X}_c + (\dot{X}_c + 2\dot{X}_s - X_c)\cos \psi_i + (\ddot{X}_s - 2\dot{X}_c - X_s)\sin \psi_i$$

give the ith generalized blade degree of freedom using multiblade coordinates in the nonrotating frame. The variables X_0, X_c, and X_s are the rotor degrees of freedom, and they describe the motion of the rotor as a whole in the nonrotating frame.

The Multiblade Coordinate Transformation involves the following steps (ref. 29): (1) expansion of each degree of freedom into multiblade coordinates; (2) multiplication of the resulting expression with multiblade functions 1, cos ψ, sin ψ, cos 2ψ, sin 2ψ, etc., depending on the number of blades; (3) replacement of products of sines and cosines as sums of sines and cosines using trigonometric relations; and (4) deletion of terms which are not periodic multiples of the number of blades (summation rules). Using the symbolic program, this is achieved as follows. The multiblade expansion of each degree of freedom and its time derivative is given as a table of relations (eq. (38)). The multiblade functions 1, cos ψ,

43

sin ψ, cos 2ψ, sin 2ψ, etc., required in transforming the equations, are read as data (or can be generated within the program). The trigonometric relations giving the product of sines and cosines as sums of sines and cosines are given as a table of relations. This requires that the equation derivation consist of several runs until all the required relations are included in the table of relations without exception. Then the command PERFORM MULTIBLADE TRANSFORMATION multiplies the equation with each of the multiblade functions, substitutes the multiblade expansion for each degree of freedom, substitutes the trigonometric relations (from the given tables of relations), and checks for the multiples of the blade harmonics. Only terms which contain a periodic multiple of number of blades are retained. The interface routines separate the terms into constant and periodic parts (coefficient of each harmonic separately), and writes them into subroutines. In this manner, the constant coefficient approximation is easily done. In this study the perturbed governing equations of motion and perturbations in thrust and in moment equations are converted into multiblade coordinate equations. The multiblade solution was subsequently checked for accuracy by comparing results with a single blade solution without dynamic inflow. It should be noted that the trim value harmonics entering as nonlinear contributions should be defined as symbolic data. The input data increase as more nonlinear terms are taken into account. These data are in addition to the data given for the multiblade expansion of the degrees of freedom. However, the output may be smaller since only terms which are multiples of the number of blades are retained. Since this is a feasibility study to obtain explicit multiblade equations using a symbolic program in FORTRAN, the nonlinear quantities are assumed to provide only first harmonic forcing contributions. For the results presented here, the program was run on a VAX 11/780 computer. For each equation, it took approximately 250 sec to derive the multiblade equations and 120 sec to write these into subroutines for numerical analysis.

From the amount of input required to perform a symbolic formulation of the harmonic balance and multiblade equations, the program HESL is convenient to explicitly consider the symbolically derived equations only if the number of degrees of freedom are small. As pointed out in reference 29, numerical schemes are better suited for general models so that the harmonic balance equations and multiblade equations can be efficiently obtained after obtaining the steady and perturbed equations from the symbolic program.

The symbolic program separates the terms containing the periodic variable cos Nt and sin Nt and writes the equation as

$$A(t) = A_0 + A_N \cos Nt + B_N \sin Nt \tag{39}$$

In subroutine form they are referred to as $A(1)$, $A(2)$, and $A(3)$. This allows for direct elimination of the matrices $A(2)$ and $A(3)$ for a simple constant coefficient analysis.

Aeroelastic stability solution (multiblade and dynamic inflow)- The final governing equations of motion can be written as

44

$$[P]\{\ddot{q}\} + [Q]\{\dot{q}\} + [R]\{q\} + [T]\{u\} = 0 \qquad (40a)$$

for the blade equations, and

$$[A]\{\ddot{q}\} + [B]\{\dot{q}\} + [C]\{q\} + [G]\{u\} = m\{\dot{u}\} + [\ell]^{-1}\{u\} \qquad (40b)$$

for the dynamic inflow equations, where $\{q\}$ is $\{q_0, q_c, q_s\}$, q_0 is vector of all collective modes, q_c is vector of all lateral cyclic modes, q_s is vector of all longitudinal cyclic modes, and $\{u\}$ is (ν_0, ν_c, ν_s).

Equations (40a) and (40b) can be combined as

$$\begin{bmatrix} P & 0 \\ A & -m \end{bmatrix}\{\ddot{X}\} + \begin{bmatrix} Q & T \\ B & (G - \ell^{-1}) \end{bmatrix}\{\dot{X}\} + \begin{bmatrix} R & 0 \\ C & 0 \end{bmatrix}\{X\} = 0 \qquad (41)$$

where

$$\{X\} = \begin{Bmatrix} q \\ u \end{Bmatrix}$$

The final stability equations in state vector form are

$$\{\dot{Y}\} = [A_s]\{Y\} \qquad (42)$$

where

$$\{Y\} = \begin{Bmatrix} X \\ \dot{X} \end{Bmatrix}$$

The stability results are obtained by calculating the eigenvalues of $[A_s]$, in a similar manner to the single blade case.

The size of the state matrix depends on the number of modes and blades. For the flap-lag-torsion model with two modes each, the size of the state matrix is 36 × 36 without dynamic inflow and 39 × 39 with dynamic inflow. The corresponding values for a flap-lag model are 24 × 24 and 27 × 27, respectively.

Results and discussion- Results are presented for a uniform blade with zero built-in twist, zero blade offsets, and zero precone angle. Reversed flow effects are neglected. A three-bladed rotor is considered. Two rotating modes for each flap, lag, and torsion degrees of freedom are used in the calculations. These modes are calculated at zero pitch and are obtained from five nonrotating modes. Results for both a single blade solution and a multiblade solution are presented for different blade structural models. All results are for a propulsive trim condition, calculated for a specified weight coefficient of $C_w = 0.007$ and an equivalent drag

area $D/q = f = 0.012(\pi R^2)$, where D is the drag force and q is the dynamic pressure.

In the derivation of the equations, the orders of magnitude assigned for each parameter are the same as that which followed in reference 15. The other parameters used for the numerical study are

$$\omega_v = 0.7, 1.4 , \quad \omega_w = 1.15 , \quad \omega_\phi = 3.0 , \quad c/R = 0.07854 , \quad \gamma = 5.0 ,$$

$$\sigma = 0.1 , \quad a = 2\pi , \quad C_{d_o} = 0.01 , \quad \beta_{pc} = 0.0 , \quad K_{m1}/K_{m2} = 0.0 ,$$

$$K_m/R = 0.025, \quad (K_A/K_m)^2 = 1.5$$

Lead-lag damping values (real part of the characteristic exponent) are presented for a soft in-plane and stiff in-plane rotor with and without dynamic inflow. The results present an investigation of: (1) the effect of degrees of freedom used in the trim analysis on the lead-lag damping, (2) the effect of using only one torsion mode, (3) the inclusion of a dynamic inflow model, and (4) the difference between periodic and a constant coefficient approximation (where all time-dependent coefficients are neglected).

Influence of trim analysis. The effect of the number of degrees of freedom used in the trim analysis on the lead-lag damping is investigated in figures 18 and 19. Figure 18 shows the lead-lag damping plotted versus advance ratio for a soft in-plane rotor ($\omega_v = 0.7$). It can be seen that a flap-lag-torsion stability analysis from a flap trim analysis underpredicts the lead-lag damping. The second mode shows the same trend with the difference in predicted damping increasing with advance ratio. Figure 19 shows the lead-lag damping plotted for a stiff in-plane rotor ($\omega_v = 1.4$) as a function of advance ratio. The results also show an increase in damping when a flap-lag-torsion trim analysis is used. It is also noted that at an advance ratio of $0.37 < \mu < 0.41$, the roots separate, and one root becomes less stable whereas the other becomes more stable. The damping does reduce as the advance ratio is increased beyond $\mu = 0.37$. The second mode remains stable at all advance ratios considered.

The increase in damping observed above for both soft in-plane and stiff in-plane rotors is due to the different time-dependent equilibrium positions that are used to linearize the full nonlinear equations. A full flap-lag-torsion trim analysis is consistent in that the blade model has the same degree of complexity in both the trim and in the stability analysis. It should be noted qualitatively that the same type of trend is observed in reference 16, for both soft in-plane and stiff in-plane rotors. This verifies the symbolic and numerical programs for the single blade results and forms the basis to check the symbolically derived explicit multi-blade equations (and numerical results) presented in the next section.

Influence of elastic torsion model. Figures 20 and 21 present the lead-lag damping plotted versus advance ratio from a flap-lag model, flap-lag-torsion model

46

(two modes each), and flap-lag-torsion model with only one torsion mode, for a soft in-plane rotor (ω_v = 0.7) using a single blade analysis. Figure 20 presents the damping results for full elastic coupling (R = 1.0). It can be seen that the flap-lag model underpredicts the lead-lag damping. The model with only one torsion mode increases the damping above the model with two modes each. Figure 21 shows the lead-lag damping value plotted for zero elastic coupling (R = 0.0). The damping levels are much reduced when they are compared with the full elastic coupling case. However, the flap-lag model is again lowest damped.

Lead-lag damping is plotted for a stiff in-plane rotor (ω_v = 1.4) with varying advance ratio in figures 22 and 23. Figure 22 presents the damping results for full elastic coupling. The same trend that was observed for the case of a soft in-plane rotor (fig. 6) exists. Note that root splitting for high-advance ratios occurs even when only one torsion mode is used.

Figure 23 presents the lead-lag damping for increasing advance ratio for a stiff in-plane rotor for a zero elastic coupling parameter and as a result, the rotor is unstable. However, a flap-lag model predicts a stable system. This demonstrates the importance of elastic blade torsion in the analysis.

Lead-lag damping is plotted in figure 24 for an advance ratio of μ = 0.25 while varying the elastic coupling parameter for a stiff in-plane rotor. Here a flap-lag model predicts positive damping for all values of R, whereas for a flap-lag-torsion model, the damping varies with elastic coupling parameter, increasing with elastic coupling parameter.

Multiblade equation results. The following figures present the lead-lag regressing mode damping results obtained using multiblade equations. The multiblade equations are explicitly derived using the symbolic program. This required explicit definition of all nonlinear contributions and degrees of freedom in terms of their harmonics. This significantly increased the amount of data that is required by the symbolic program. Since this is a feasibility study on the use of symbolic programs in FORTRAN, only first harmonics were considered in the nonlinear contributions. Consequently, damping data determined from the multiblade equations may slightly differ from the single-blade solution. Additionally, the multiblade results are obtained by retaining only one torsion mode; however, the nonlinear contribution from both torsion modes is used. This reduces the time step and time for integration in performing the Floquet-analysis.

The damping values are first checked with those obtained from a single-blade solution obtained earlier to validate the multiblade equation derivation process. It was found that the approximation $\cos \theta$ = 1.0, used in deriving the explicit multiblade equations, will predict slightly higher (yet less than 2%) damping for stiff in-plane rotors with a full elastic coupling parameter, since this approximation most greatly affects the coupling elements. For all other values of the elastic coupling parameter, this approximation does not affect the damping value. Where required for comparison, the single-blade damping values are recalculated using this approximation, to avoid the rederivation of the multiblade equations.

47

Figures 25 and 26 show the lead-lag regressing mode damping plotted for varying advance ratio with and without dynamic inflow from a flap-lag-torsion and flap-lag model for a soft in-plane rotor. Figure 25 shows the damping for full elastic coupling (R = 1.0). For the flap-lag-torsion model, the dynamic inflow reduces the damping at all advance ratios. Its effect is almost negligible at advance ratios 0.15 to 0.25. For the flap-lag model, the dynamic inflow increases the damping up to an advance ratio of 0.33, it then reduces the damping. Figure 26 presents the damping results for the zero elastic coupling (R = 0.0). It is seen that for the flap-lag-torsion model, the dynamic inflow again reduces the damping in hover. Yet at intermediate advance ratios, dynamic inflow increases the damping, and at higher advance ratios, it once again reduces the damping. For the flap-lag model, the dynamic inflow increases lead-lag damping for all the advance ratios that are considered. This is consistent with previous studies, for example, as in reference 18.

The lead-lag regressing mode damping is plotted for a stiff in-plane rotor for varying advance ratio in figures 27 and 28. Figure 27 is for a rotor with full elastic coupling. For a flap-lag-torsion model, the dynamic inflow reduces the damping up to an advance ratio of 0.41. For $\mu > 0.41$, this model shows a slightly increased damping value. The flap-lag model with dynamic inflow shows a small decrease in damping, yet this damping increment gets smaller with higher advance ratios. Figure 28 is for a rotor with zero elastic coupling. For this configuration, the dynamic inflow increases damping for all advance ratios. Consequently, both flap-lag-torsion and flap-lag models show the same trend.

Figure 29 shows the lead-lag regressing mode damping as it is plotted for a stiff in-plane rotor at an advance ratio of 0.25 while varying the elastic coupling parameter. For the flap-lag-torsion model, dynamic inflow reduces the damping for R > 0.3; however, it increases the available damping for R < 0.3. This increase is not sufficient to stabilize the in-plane mode. With the flap-lag model, dynamic inflow shows an increase in damping for all values of elastic coupling.

Constant coefficient approximation. The effect of a constant coefficient approximation is studied in figure 30, where the real part of the exponent is plotted for a stiff in-plane rotor with full elastic coupling which showed a splitting of the roots when analyzed with a full periodic coefficient. The constant coefficient approximation does not show this splitting, since the frequencies are away from the real axis. For this analysis, the regressing and collective modes do predict the same damping trend with advance ratio as shown by a full Floquet analysis. However, the progressing mode shows poor agreement between a constant coefficient approximation analysis and a Floquet theory analysis. This is because the constant coefficient approximation can only be good for low frequency modes.

III. CONCLUSIONS

A symbolic manipulation program written in FORTRAN is used to derive the aeroelastic analysis equations of an elastic blade with flap-lag-torsion degrees of

freedom in hover and in forward flight. A study is made of the feasibility of using the program to obtain explicit equations in a harmonic balance method and to obtain multiblade equations. Numerical results are presented for a flap-lag-torsion model and for a flap-lag model with and without dynamic inflow for a propulsive-trimmed rotor. Both soft in-plane and stiff in-plane rotors are considered.

The following conclusions are drawn from using a symbolic program:

1. The program HESL, written in FORTRAN, can be conveniently used to derive elastic blade equations.

2. The process of aeroelastic stability analysis starting from equation derivation to numerical calculation can be automated.

3. With the present program capability, the amount of data to the symbolic program increases rapidly with degrees of freedom and the number of harmonics that have been analyzed.

4. In deriving the explicit harmonic balance equations and multiblade equations, the following should be noted:

(a) To obtain the harmonic balance equations, a numerical method is suggested since an arbitrary number of harmonics can be used without increasing the input data to the symbolic program.

(b) To obtain the multiblade equations, the perturbed equations in their Fourier series form are derived using the symbolic program. Then the multiblade equations themselves are obtained numerically.

It is recommended that a selective combination of the symbolic and numerical programs is required for the derivation and numerical study to be efficiently conducted.

The following conclusions are drawn from the numerical study performed for a hovering rotor.

1. The unsteady aerodynamic effects on lag regressing mode damping, modeled as dynamic inflow, are dependent on the elastic coupling parameter for torsionally flexible blades. For zero elastic coupling, dynamic inflow increased the lag regressing mode damping. For other elastic coupling parameter values, dynamic inflow either increased or decreased lag regressing mode damping.

2. Use of a quasi-static torsion blade model is conservative, and yields the same trend with varying elastic coupling parameter as flap-lag-torsion model.

3. When compared with a flap-lag model, it is seen that the relationship between dynamic inflow and lag regressing mode damping can be a result of the effect of pitch-flap and pitch-lag couplings, which depend on the elastic coupling parameter.

The following conclusions are drawn from the numerical study of a single blade solution in forward flight.

1. A flap-lag-torsion stability analysis from a trim procedure where only the flap degree of freedom is used underpredicts the lead-lag damping.

2. In the case of stiff in-plane rotors, high-forward flight speed is destabilizing. At high advance ratios, a splitting of the roots is encountered which yields two real values at the same frequency.

3. Using only one torsion mode increases the damping value from the flap-lag structural model.

4. The damping values for a stiff in-plane rotor are sensitive to elastic coupling parameter.

The following conclusions are drawn from the numerical study of a multiblade solution with dynamic inflow in forward flight.

1. For a flap-lag model, for both soft in-plane and stiff in-plane rotors, with zero elastic coupling, the dynamic inflow increased damping at all advance ratios considered; with full elastic coupling the dynamic inflow increased the damping at low-advance ratios, but reduced damping at high-advance ratios.

2. For a flap-lag-torsion model, dynamic inflow slightly reduced lead-lag regressing mode damping for all advance ratios. The same trend is observed for both soft in-plane and stiff in-plane rotors.

3. For a given advance ratio, the variation of damping with an elastic coupling parameter for a stiff in-plane rotor showed the same trend as it did for the hover case.

4. The constant coefficient approximation for the stiff in-plane rotor does not show the splitting of the rotors, since the frequency of the lag mode is away from the real axis.

50

This appendix shows the input data (Table A1) and the output data (Table A2) from the symbolic program to derive flap equations for a rigid rotor blade. The problem definition starts with a flap-lag transformation but is subsequently reduced to flap model. This simplified model is used to clarify the program aspects rather than the modeling aspects. Even though the data is given at the matrix level to show that the symbolic program is capable of handling matrices, in this example it is simple to use the definition of the position vector directly as input. Both the single blade and multiblade equations for a three-bladed rotor are obtained from the symbolic program.

Table A3 gives the FORTRAN symbols used to define the original variables.

TABLE A1.- INPUT TO THE PROGRAM HESL TO OBTAIN LINEARIZED FLAP EQUATION AND
MULTIBLADE EQUATIONS FOR THREE-BLADED ROTORS

%@#
READ MATRIX
#TFLP0303
01
1.0   COSB
01
0.0
01
-1.0   SINB
01
0.0
01
1.0
01
0.0
01
1.0   SINB
01
0.0
01
1.0   COSB
READ MATRIX
#TLAG0303
01
1.0   COSZ
01
-1.0   SINZ
01
0.0
01
1.0   SINZ
01
1.0   COSZ
01
0.0
01
0.0
01
0.0
01
1.0
READ MATRIX
#POVE0301
01
1.0   RB
01
0.0
01

```
0.0
READ DIFERENTIATION TABLE
@DERV10
   BT   BTO1
1.0
  BTD  BTDO1
1.0
  BT COSBO1
-1.0         SINB
   BT SINBO1
1.0          COSB
  TAU   BTO1
1.0           BTD
  TAU  BTDO1
1.0          BTDD
  TAU COSBO1
-1.0         SINB   BTD
  TAU SINBO1
1.0          COSB   BTD
  TAU CSCYO1
-1.0         SNCY
  TAU SNCYO1
1.0          CSCY
READ TABLE FOR SUBSTITUTION
&SUPRO2
01 SINZO1
0.0
01 COSZO1
1.0
READ TABLE FOR SUBSTITUTION
@DUMYO101
02 SINBO2
1.0
-1.0         COSB COSB
READ TABLE FOR SUBSTITUTION
@DUM101
01 ABCDO1
1.0
READ TABLE FOR SUBSTITUTION
@INRLO3O1
01   RBO2
1.0           IB
01   RBO1
1.5           IB
01   RBOO
3.0           IB
READ TABLE FOR SUBSTITUTION
@PERTO5
02 BTDDO1
1.0          BBDD
1.0          DBDD
```

```
02  BTDO1
1.0           BBD
1.0           DBD
02   BTO1
1.0            BB
1.0            DB
02 SINBO1
1.0            BB
1.0            DB
01 COSBO1
1.0
READ VARIABLES FOR COLLECTION OF COEFFICIENTS
@MUCF12
  THO   TH1   TH2 BDDO BDD1 BDD2  BDO  BD1  BD2   BO   B1   B2
READ TABLE FOR SUBSTITUTION
@MULBO4
07 DBDDO1
1.0           BDDO
1.0           BDD1 CSCY
-2.0           BD1 SNCY
-1.0            B1 CSCY
1.0           BDD2 SNCY
2.0            BD2 CSCY
-1.0            B2 SNCY
05   DBDO1
1.0            BDO
1.0            BD1 CSCY
-1.0            B1 SNCY
1.0            BD2 SNCY
1.0             B2 CSCY
03    DBO1
1.0            BO
1.0            B1 CSCY
1.0            B2 SNCY
03 THTAO1
1.0           THO
1.0           TH1 CSCY
1.0           TH2 SNCY
READ TABLE FOR SUBSTITUTION
@TRIG1201
02 SNCYO2
0.5
-.5           C2CY
01 SNCYO1 CSCYO1
0.5           S2CY
02 CSCYO2
0.5
0.5           C2CY
02 SNCYO3
0.75          SNCY
-.25          S3CY
```

```
02 SNCY02 CSCY01
0.25      CSCY
-.25      C3CY
02 SNCY01 CSCY02
0.25      SNCY
0.25      S3CY
02 CSCY03
0.75      CSCY
0.25      C3CY
03 SNCY04
0.375
-0.5      C2CY
0.125     C4CY
02 SNCY03 CSDY01
0.25      S2CY
-.125     S4CY
02 SNCY02 CSCY02
0.125
-.125     C4CY
02 SNCY01 CSCY03
0.25      S2CY
0.125     S4CY
03 CSCY04
0.375
0.5       C2CY
0.125     C4CY
FORM MATRIX
#TLAG#TFLP#TFLG
FORM MATRIX
#TFLG#POVE#PVEC
MATRIX EXPRESSION
03#PVEC
%   R10101%  R20201%  R30301
SUBSTITUTE TABLE INTO EXPRESSION
@SUPR%  R1%  RX
SUBSTITUTE TABLE INTO EXPRESSION
@SUPR%  R2%  RY
SUBSTITUTE TABLE INTO EXPRESSION
@SUPR5  R3%  RZ
READ EXPRESSION
%OMEZ01
1.0
DIFERENTIATE EXPRESSION
  TAU%  RX%  RXT
DIFERENTIATE EXPRESSION
  TAU%  RY%  RYT
DIFERENTIATE EXPRESSION
  TAU%  RZ%  RZT
FORM EXPRESSION
%  RXD02
1.0       %  RXT
```

```
-1.0        %  RY%OMEZ
FORM EXPRESSION
%  RYD02
1.0         %  RYT
1.0         %OMEZ%  RX
FORM EXPRESSION
%  RZD01
1.0         %  RZT
READ EXPRESSION
%SEBL01
0.5            KBT    BT    BT
READ EXPRESSION
%XVEL01
1.0            MU CSCY
READ EXPRESSION
%YVEL01
-1.0           MU SNCY
READ EXPRESSION
%ZVEL01
-1.0           LAMD
FORM EXPRESSION
%AVL102
1.0         %XVEL
-1.0        %  RXD
FORM EXPRESSION
%AVL202
1.0         %YVEL
-1.0        %  RYD
FORM EXPRESSION
%AVL302
1.0         %ZVEL
-1.0        %  RZD
READ MATRIX
#VXYZ0301
01
1.0         %AVL1
01
1.0         %AVL2
01
1.0         %AVL3
FORM MATRIX
#TFLP#VXYZ#VX2Y        01
MATRIX EXPRESSION
02#VX2Y
%  UT0201%  U1P0301
NEGATE EXPRESSION
%  UT
NEGATE EXPRESSION
```

```
% U1P
SUBSTITUTE TABLE INTO EXPRESSION
@DUMY% U1P%  UP
FORM EXPRESSION
%  MB02
-0.5       %  UT%  UT THTA RAC2    RB
0.5        %  UT%  UP RAC2    RB
READ TABLE FOR SUBSTITUTION
@DYNM0401
01 RAC201    RB01
0.5          GAMI
01 RAC201    RB02
0.33333      GAMI
01 RAC201    RB03
0.25         GAMI
01 RAC201
1.0          GAMI
DIFERENTIATE EXPRESSION
   BTD% RXD%RXDQ
DIFERENTIATE EXPRESSION
    BT% RXD% RXQ
DIFERENTIATE EXPRESSION
BTD% RYD%RYDQ
DIFERENTIATE EXPRESSION
    BT% RYD% RYQ
DIFERENTIATE EXPRESSION
   BTD% RZD%RZDQ
DIFERENTIATE EXPRESSION
    BT% RZD% RZQ
READ EXPRESSION
%CONS01
1.0
READ EXPRESSION
%CON101
-1.0
READ VARIABLES FOR COLLECTION OF COEFFICIENTS
@PECF06
 DBDD   DBD    DB BBDD   BBD    BB
FORM LAGRANGIAN
0803000100
%SBEQ$PECF*E2D1
%  RXD% RXQ%CON1     @INRL@DUMY@PERT
%  RYD% RYQ%CON1     @INRL@DUMY@PERT
%  RZD% RZQ%CON1     @INRL@DUMY@PERT
%  RXD%RXDQ%CONS  TAU@INRL@DUMY@PERT
%  RYD%RYDQ%CONS  TAU@INRL@DUMY@PERT
%  RZD%RZDQ%CONS  TAU@INRL@DUMY@PERT
%CONS%SEBL%CONS    BT@DUM1@DUM1@PERT
%CONS%  MB%CONS     @DYNM@DUMY@PERT
INITIALIZE MULTIBLADE
03    MOPK CSCY SNCY C2CY S2CY C3CY S3CY C4CY S4CY C5CY S5CY C6CY S6CY C7CY S7CY
```

57

```
PERFORM MULTIBLADE TRANSFORMATION
03
@MULTB@TRIG@MUCF
%SBEQ%AEM1%AEM2%AEM3
END OF DATA
```

TABLE A2.- SAMPLE OUTPUT

```
READ VARIABLES FOR COLLECTION OF COEFICIENTS

RDCCVR   DBDD    DBD     DB   BBDD    BBD    BB

FORM LAGRANGIAN

    ***********************************************
    *                                             *
    *      DETAILS OF THE EXPRESSION    SBEQ       *
    *                 NUMBER OF TERMS     19       *
    *                                             *
    ***********************************************


    ---------------------------------------------
    -  GROUP OF TERMS WITH VARIABLE    DBDD   -
    ---------------------------------------------


    1    1.000*  IB*


    ---------------------------------------------
    -  GROUP OF TERMS WITH VARIABLE    DBD   -
    ---------------------------------------------


    1    0.167*SNCY*  MU*GAMI*
    2    0.125*GAMI*


    ---------------------------------------------
    -  GROUP OF TERMS WITH VARIABLE    DB   -
    ---------------------------------------------


    1    1.000*  KBT*
    2    0.250*CSCY*SNCY*  MU*  MU*GAMI*
    3    0.167*CSCY*  MU*GAMI*
    4    1.000*  IB*


    ---------------------------------------------
    -  GROUP OF TERMS WITH VARIABLE    BBDD   -
    ---------------------------------------------


    1    1.000*  IB*


    ---------------------------------------------
    -  GROUP OF TERMS WITH VARIABLE    BBD   -
    ---------------------------------------------


    1    0.167*SNCY*  MU*GAMI*
    2    0.125*GAMI*
```

```
------------------------------------------------
-  GROUP OF TERMS WITH VARIABLE      BB    -
------------------------------------------------


1    1.000* KBT*
2    0.167*CSCY*  MU*GAMI*
3    1.000*  IB*
4    0.250*CSCY*SNCY*  MU*  MU*GAMI*


------------------------------------------------
-       REMAINING TERMS IN EQUATION        -
------------------------------------------------


1   -0.250*SNCY*SNCY*THTA*  MU*  MU*GAMI*
2   -0.125*THTA*GAMI*
3   -0.333*SNCY*THTA*  MU*GAMI*
4    0.250*SNCY*  MU*LAMD*GAMI*
5    0.167*LAMD*GAMI*
```

PERFORM MULTIBLADE TRANSFORMATION

MULTIBLADE EQUATION   1

```
*********************************************
*                                           *
*    DETAILS OF THE EXPRESSION     AEM1     *
*            NUMBER OF TERMS         17     *
*                                           *
*********************************************


------------------------------------------------
-  GROUP OF TERMS WITH VARIABLE     THO   -
------------------------------------------------


1   -0.375*  MU*  MU*GAMI*
2   -0.375*GAMI*


------------------------------------------------
-  GROUP OF TERMS WITH VARIABLE     TH1   -
------------------------------------------------


1    0.188*C3CY*  MU*  MU*GAMI*


------------------------------------------------
-  GROUP OF TERMS WITH VARIABLE     TH2   -
------------------------------------------------


1    0.188*S3CY*  MU*  MU*GAMI*
2   -0.500*  MU*GAMI*
```

```
----------------------------------------------
- GROUP OF TERMS WITH VARIABLE    BDDO   -
----------------------------------------------

1    3.000*  IB*

----------------------------------------------
- GROUP OF TERMS WITH VARIABLE    BDO    -
----------------------------------------------

1    0.375*GAMI*

----------------------------------------------
- GROUP OF TERMS WITH VARIABLE    BD2    -
----------------------------------------------

1    0.250*  MU*GAMI*

----------------------------------------------
- GROUP OF TERMS WITH VARIABLE    BO     -
----------------------------------------------

1    3.000*  IB*
2    3.000*  KBT*

----------------------------------------------
- GROUP OF TERMS WITH VARIABLE    B1     -
----------------------------------------------

1    0.188*S3CY*  MU*  MU*GAMI*

----------------------------------------------
- GROUP OF TERMS WITH VARIABLE    B2     -
----------------------------------------------

1   -0.188*C3CY*  MU*  MU*GAMI*

----------------------------------------------
-        REMAINING TERMS IN EQUATION        -
----------------------------------------------

1    0.375*  BBD*GAMI*
2    3.000*  BB*  KBT*
3    3.000*  IB*BBDD*
4    3.000*  IB*  BB*
5    0.500*LAMD*GAMI*
```

MULTIBLADE EQUATION   2

```
*******************************************
*                                         *
*    DETAILS OF THE EXPRESSION    AEM2    *
*              NUMBER OF TERMS      19    *
*                                         *
*******************************************
```

- GROUP OF TERMS WITH VARIABLE   THO   -
----

1   0.188*C3CY* MU* MU*GAMI*

- GROUP OF TERMS WITH VARIABLE   TH1   -
----

1   -0.094* MU* MU*GAMI*
2   -0.188*GAMI*
3   -0.250*S3CY* MU*GAMI*

- GROUP OF TERMS WITH VARIABLE   TH2   -
----

1   0.250*C3CY* MU*GAMI*

- GROUP OF TERMS WITH VARIABLE   BDD1   -
----

1   1.500* IB*

- GROUP OF TERMS WITH VARIABLE   BD1   -
----

1   0.188*GAMI*
2   0.125*S3CY* MU*GAMI*

- GROUP OF TERMS WITH VARIABLE   BD2   -
----

1   3.000* IB*
2   -0.125*C3CY* MU*GAMI*

```
--------------------------------------------
-   GROUP OF TERMS WITH VARIABLE      BO    -
--------------------------------------------


1    0.250*  MU*GAMI*
2    0.188*S3CY*  MU*   MU*GAMI*


--------------------------------------------
-   GROUP OF TERMS WITH VARIABLE      B1    -
--------------------------------------------


1    1.500*  KBT*
2    0.250*C3CY*  MU*GAMI*


--------------------------------------------
-   GROUP OF TERMS WITH VARIABLE      B2    -
--------------------------------------------


1    0.250*S3CY*   MU*GAMI*
2    0.188*GAMI*
3    0.094*  MU*   MU*GAMI*


--------------------------------------------
-      REMAINING TERMS IN EQUATION          -
--------------------------------------------


1    0.188*  BB*S3CY*  MU*   MU*GAMI*
2    0.250*  BB*  MU*GAMI*

MULTIBLADE EQUATION    3

    *********************************************
    *                                           *
    *    DETAILS OF THE EXPRESSION     AEM3    *
    *                NUMBER OF TERMS       21    *
    *                                           *
    *********************************************


--------------------------------------------
-   GROUP OF TERMS WITH VARIABLE      THO   -
--------------------------------------------


1    0.188*S3CY*  MU*   MU*GAMI*
2   -0.500*  MU*GAMI*


--------------------------------------------
-   GROUP OF TERMS WITH VARIABLE      TH1   -
--------------------------------------------


1    0.250*C3CY*  MU*GAMI*
```

TABLE A2.- CONTINUED

```
--------------------------------------------
- GROUP OF TERMS WITH VARIABLE    TH2    -
--------------------------------------------

1  -0.281*  MU*  MU*GAMI*
2  -0.188*GAMI*
3   0.250*S3CY*  MU*GAMI*

--------------------------------------------
- GROUP OF TERMS WITH VARIABLE    BDD2   -
--------------------------------------------

1   1.500*  IB*

--------------------------------------------
- GROUP OF TERMS WITH VARIABLE    BDO    -
--------------------------------------------

1   0.250*  MU*GAMI*

--------------------------------------------
- GROUP OF TERMS WITH VARIABLE    BD1    -
--------------------------------------------

1  -3.000*  IB*
2  -0.125*C3CY*  MU*GAMI*

--------------------------------------------
- GROUP OF TERMS WITH VARIABLE    BD2    -
--------------------------------------------

1   0.188*GAMI*
2  -0.125*S3CY*  MU*GAMI*

--------------------------------------------
- GROUP OF TERMS WITH VARIABLE    BO     -
--------------------------------------------

1  -0.188*C3CY*  MU*  MU*GAMI*

--------------------------------------------
- GROUP OF TERMS WITH VARIABLE    B1     -
--------------------------------------------

1   0.250*S3CY*  MU*GAMI*
2  -0.188*GAMI*
3   0.094*  MU*  MU*GAMI*
```

```
----------------------------------------------
-   GROUP OF TERMS WITH VARIABLE      B2    -
----------------------------------------------


1   -0.250*C3CY*  MU*GAMI*
2    1.500* KBT*


----------------------------------------------
-       REMAINING TERMS IN EQUATION         -
----------------------------------------------


1    0.250* BBD*  MU*GAMI*
2   -0.188*  BB*C3CY*  MU*  MU*GAMI*
3    0.375*  MU*LAMD*GAMI*

END OF DATA
```

Symbol	FORTRAN Symbol	Description
A	A	area of cross section, matrix name
	B,C	matrix names
cos ψ, cos 2ψ cos 3ψ, cos 4ψ	CSCY,C2CY C3CY,C4CY	
cos β, sin β	COSB,SINB	
cos ζ, sin ζ	COSZ,SINZ	
C_{d0}/a	CDOA	drag coefficient to lift curve slope
E	E	Young's modulus
I_β	IB	flap moment of inertia
K_β	KB	flap spring stiffness
$\bar{r}$	RB	rotor radius (nondimensional)
sin ψ, sin 2ψ sin 3ψ, sin 4ψ	SNCY,S2CY S3CY,S3CY	
U_T, U_P	UT,UP	tangential and perpendicular velocity components of flow over the airfoil
u'	US	derivative of extensional displacement
v',w'	VS,WS	bending slopes
$\bar{\beta}, \dot{\bar{\beta}}, \ddot{\bar{\beta}}$	BB,BBD,BBDD	steady flap angles and derivatives
$\delta\beta, \delta\dot{\beta}, \delta\ddot{\beta}$	DB,DBD,DBDD	perturbed flap angle and derivative
$\beta_0, \beta_1, \beta_2$	BO,B1,B2	flap collective and cyclic angles
$\dot{\beta}_0, \dot{\beta}_1, \dot{\beta}_2$	BDO,BD1,BD2	1st derivatives
$\ddot{\beta}_0, \ddot{\beta}_1, \ddot{\beta}_2$	BDDO,BDD1,BDD2	2nd derivatives
μ	MU	advance ratio
θ	THTA	pitch angle
$\theta_0, \theta_c, \theta_s$	THO,TH1,TH2 (THTO,THTC,THTS)	collective and cyclic pitch angles

66

Symbol	FORTRAN Symbol	Description
λ	LAMD	uniform inflow
$\Omega_x, \Omega_y, \Omega_z$	OMEX,OMEY,OMEZ	expressions defining rotation vector
γ_I	GAMI	Lock number
$\rho ac/2$	RAC2	air density
ψ		azimuth angle
τ	TAU	dimensionless time
M_β	MB	flap moment expression name
ζ	ZETA	lag angle
$\beta, \dot{\beta}, \ddot{\beta}$	BT,BTD,BTDD	flap angle and time derivatives
S_{ebl}, S_{ebq}	SEBL,SEBQ	strain energy and single blade expression na
	PECF,MUCF	group names for collection of coefficients in perturbed equations and multiblade equations
	MULB	table containing multiblade relations
	TRIG	table containing trigonometric relations
	PERT	table containing perturbation relations
	DERV	table containing differentiation rules
	INRL,DYNM	tables containing integration rules
	SUPR	table containing rules to suppress lag motion
	AEM1,AEM2,AEM3	names of multiblade equations (expressions)
	TFLP,TLAG,POVE	matrices giving transformation relations
r_x, r_y, r_z	RX,RY,RZ	expressions defining position vector
$\dot{r}_x, \dot{r}_y, \dot{r}_z$	RXT,RYT,RZT	velocity expressions
	RXD,RYD,RZD	$\partial/\partial\tau \{RXT,RYT,RZT\} + \Omega \times \{RX,RY,RZ\}$
	RXDQ,RYDQ,RZDQ	$\partial/\partial\dot{\beta} \{RXD,RYD,RZD\}$
	RXQ,RYQ,RZQ	$\partial/\partial\beta \{RXD,RYD,RZD\}$

TABLE A3.- CONCLUDED

Symbol	FORTRAN Symbol	Description
	DUMY,DUM1	tables containing dummy relations
	R1,LAEQ	intermediate expression names
	E2D1	name of the ordering scheme
	VX2Y	intermediate matrix name

APPENDIX B

QUASI-STATIC APPROXIMATION

It is frequently possible to reduce the order of the system of equations describing the rotorcraft dynamics by considering a quasi-static approximation for certain degrees of freedom. Assume that the equations of motion have been reordered so that the quasi-static variables (X_0) appear last in the state vector.

$$X = \begin{pmatrix} X_1 \\ X_0 \end{pmatrix} \tag{B1}$$

The quasi-static approximation consists of neglecting the acceleration and velocity terms of the quasi-static variables. Thus the equations of motion take the form

$$\begin{bmatrix} A_2^{11} & 0 \\ A_2^{01} & 0 \end{bmatrix} \begin{pmatrix} \ddot{X}_1 \\ X_0 \end{pmatrix} + \begin{bmatrix} A_1^{11} & 0 \\ A_1^{01} & 0 \end{bmatrix} \begin{pmatrix} \dot{X}_1 \\ X_0 \end{pmatrix} + \begin{bmatrix} A_0^{11} & A_0^{10} \\ A_0^{01} & A_0^{00} \end{bmatrix} \begin{pmatrix} X_1 \\ X_0 \end{pmatrix} = \begin{pmatrix} B^1 \\ B^0 \end{pmatrix} V \tag{B2}$$

The quasi-static variables are not described by differential equations but rather by linear algebraic equations. The solution for (X_0) then is simply

$$X_0 = \left(A_0^{00}\right)^{-1} [-A_2^{01}\ddot{X}_1 - A_1^{01}\dot{X}_1 - A_0^{01}X_1] + B^0 V \tag{B3}$$

Substituting for X_0 in the X_1 equation of motion gives then the reduced order equations for the quasi-static approximation:

$$[A_2^{11} - A_0^{10}(A_0^{00})^{-1}A_2^{01}]\ddot{X}_1 + [A_1^{11} - A_0^{10}(A_0^{00})^{-1}A_1^{01}]\dot{X}_1 + [A_0^{11} - A_0^{10}(A_0^{00})^{-1}A_0^{01}]X_1$$

$$= [B^1 - A_0^{10}(A_0^{00})^{-1}B^0]V \tag{B4}$$

In the present analysis, the quasi-static approximation is applied to the inflow dynamics and torsion degree of freedom of the blade.

REFERENCES

1. Proceedings of the 1977 MACSYMA User's Conference, Berkeley, Calif., July 1977,
    NASA CP-2012.

2. Noor, Ahmed K.; and Anderson, C. M.: Computerized Symbolic Manipulation in
    Structural Mechanics-Progress and Potential. Computers and Structures,
    vol. 10, 1979, pp. 95-118.

3. Kiessling, F.: Computer Aided Derivation of Equations of Motion for Rotarywing
    Aeroelastic Problems. ICAS-82-2.1.3, AIAA Aircraft Systems and Technology
    Conference, Seattle, WA, Aug. 1982.

4. Arnold, R. M.: Rom, Manipulation of Algebraic Expressions. Comunications of
    ACM, vol. 4, 1961, pp. 396-397.

5. Arnold, R. M.: Rom, Mechanized Algebraic Operations (MAO). Celestial Mechan-
    ics, vol. 1, 1970, pp. 301-319.

6. Broucke, R.; and Garthwaite, K.: A Programming System for Analytic Series
    Expressions on a Computer. Celestial Mechanics, vol. 1, 1969, pp. 271-284.

7. Jefferys, W. H.: A FORTRAN Based List Processor for Poisson Series. Celestial
    Mechanics, vol. 2, no. 4, Nov. 1970, pp. 474-480.

8. Broucke, R.: A FORTRAN-IV System for the Manipulation of Symbolic Poisson
    Series with Application to Celestial Mechanics. Univ. of Texas, IASOM
    TR 80-1, Jan. 1980.

9. Dasenbrock, R. R.: A FORTRAN Based Program for Computerized Algebraic Manipu-
    lation. National Research Laboratories Report 8611, Sept. 1982.

10. Hale, A. L.; and Meirovitch, L.: A Special Purpose Symbolic Manipulation
    Program for the Derivation of the Equations of Motion for Large Flexible
    Structures. Presented at Conference on Dynamics and Control of Large Flexi-
    ble Spacecraft, June 13-15, 1977, Ed. L. Meirovitch, Blacksburg, Virginia.

11. Nagabhushanam, J.; Gaonkar, G. H.; and Reddy, T. S. R.: Automatic Generation
    of Equations for Rotor-Body Systems with Dynamic Inflow for a Priori Order-
    ing Schemes. 7th European Rotorcraft Forum, Paper No. 37, Sept. 8-11, 1981,
    Garmish-Partenkirchen, Germany.

12. Ormiston, R. A.; and Hodges, D. H.: Linear Flap-Lag Dynamics of Hingeless
    Rotor Blades in Hover. Journal of the American Helicopter Society, vol. 17,
    no. 2, 1972, pp. 2-14.

13. Peters, D. A.: Flap-Lag Stability of Helicopter Rotor Blades in Forward Flight. J. of the Am. Helicopter Soc., Oct. 1975, pp. 2-13.

14. Friedmann, P. P.; and Tong, P.: Nonlinear Flap-Lag Dynamics of Hingeless Helicopter Blades in Hover and Forward Flight. J. of Sound and Vibration, vol. 30, no. 1, 1973, pp. 9-31.

15. Hodges, D. H.; and Ormiston, R. A.: Stability of Elastic Bending and Torsion of Uniform Cantilever Rotor Blades in Hover with Variable Structural Coupling. NASA TN D-8192, 1976.

16. Friedmann, P. P.; and Kottapalli, S. B. R.: Coupled Flap-Lag-Torsional Dynamics of Hingeless Rotor Blades in Forward Flight. J. of the Am. Helicopter Soc., vol. 27, no. 4, Oct. 1982, pp. 28-36.

17. Peters, D. A.; and Gaonkar, G. H.: Theoretical Flap-Lag Damping with Various Inflow Models. J. of the Am. Helicopter Soc., vol. 25, no. 3, 1980, pp. 29-36.

18. Gaonkar, G. H.; and Peters, D. A.: Use of Multiblade Coordinates for Helicopter Flap-Lag Stability with Dynamic Inflow. J. of Aircraft, vol. 17, no. 2, 1980, pp. 112-118.

19. Gaonkar, G. H.; Sastry, V. V. S. S.; Reddy, T. S. R.; Nagabhushanam, J.; and Peters, D. A.: The Use of Actuator-Disk Dynamic Inflow for Helicopter Flap-Lag Stability. J. of the Am. Helicopter Soc., vol. 28, no. 3, 1983, pp. 79-88.

20. Gaonkar, G. H.; Mitra, A. K.; Reddy, T. S. R.; and Peters, D. A.: Sensitivity of Helicopter Aeromechanical Stability to Dynamic Inflow. Vertica, vol. 6, no. 1, 1982, pp. 59-67.

21. Johnson W.: Influence of Unsteady Aerodynamics on Hingeless Rotor Ground Resonance. AIAA J. of Aircraft, vol. 19, no. 8, Aug. 1982, pp. 668-673.

22. Friedmann, P. P.: Formulation and Solution of Rotary Wing Aeroelastic Stability and Response Problems. Paper no. 3.2, 8th European Rotorcraft Forum, 1982, Aix-En-Provence, France.

23. 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, 1974.

24. Kaza, K. R.; and Kvaternik, R. G.: Nonlinear Aeroelastic Equations for Combined Flapwise Bending, Chordwise Bending, Torsion, and Extension of Twisted Nonuniform Rotor Blades in Forward Flight. NASA TM 74059, 1977.

25. Rosen, A.; and Friedmann, P. P.: Nonlinear Equations of Equilibrium for Elastic Helicopter or Wind Turbine Blades Undergoing Moderate Deformation. Report no. 7718, Univ. of California at Los Angles, Calif., Jan. 1977.

26. Johnson, Wayne: A Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics, Part I: Analysis Development. NASA TM 81182, June 1980.

27. Brown, K. M.; and Dennis, J. E.: Derivative-Free Analogues of the Levenberg Marquardt and Gauss Algorithms for Nonlinear Least Squares Approximations. Numerische Mathematik, vol. 18, 1972, pp. 289-297.

28. Johnson, Wayne: Helicopter Theory. Princeton University Press, Princeton, N.J., 1980.

29. Hodges, D. H.; Ormiston, R. A.; and Peters, D. A.: On the Nonlinear Deformation Geometry of Euler-Bernoulli Beams. NASA TP 1566, April 1980.

30. Peters, D. A.; and Ormiston, R. A.: The Effect of Second Order Blade Bending on the Angle of Attack of Hingeless Rotor Blades. J. of the Am. Helicopter Soc., vol. 18, no. 4, 1973, pp. 45-48.

31. Dugundji, John; and Wendell, J. H.: Some Analysis Methods for Rotating Systems with Periodic Coefficients. AIAA J., vol. 21, no. 6, June 1983, pp. 890-897.

TABLE 1.- COMMANDS AND BRIEF DESCRIPTION OF THEIR FUNCTION

Command	Function
Input commands	
1. READ EXPRESSION	Reads an expression name, number of terms, and term details.
2. READ MATRIX	Reads a matrix name, size, and term details for each element.
3. READ TABLE FOR SUBSTITUTION	Reads table name, number of relations, type of table, and relation details.
4. READ DIFFERENTIATION TABLE	Reads differentiation table name, number of relations, and relation details.
5. READ GROUP AND ORDER OF VARIABLES	Reads group of each variable and its order of magnitude.
6. READ ORDERING SCHEME	Reads order up to which variables of each group can be retained.
7. READ VARIABLES FOR COLLECTION OF COEFFICIENTS	Reads a group name and variables for which the coefficients have to be collected.
8. READ TAPE	Reads from a tape (disk), to separate steady, and perturbed terms.
Algebraic manipulation commands	
9. FORM EXPRESSION	Multiplies, adds, or subtracts two or more expressions.
10. FORM MATRIX	Multiplies two matrices including transpose multiplication.
11. ADD MATRICES	Adds or subtracts two compatible matrices.
12. CONSTANT MATRIX	Multiplies the matrix elements with a constant for expression having one term.
13. MATRIX TO EXPRESSION	Converts matrix elements to expression elements.
14. SUBSTITUTE TABLE INTO EXPRESSION	Substitutes a table of relations into an expression.

TABLE 1.- CONCLUDED

Command	Function
Input commands - concluded	
15. DIFFERENTIATE EXPRESSION	Differentiates an expression with respect to a variable.
Application commands	
16. FORM LAGRANGIAN	Uses Lagrangian formulation to generate governing equations.
17. INITIALIZE MULTIBLADE	Generates multiblade functions.
18. PERFORM MULTIBLADE TRANSFORMATION	Generates multiblade equations.
19. WRITE TAPE	Writes on tape (disk) compatible with symbolic program.
20. NEGATE EXPRESSION	Gives the negative of an expression.
21. RESET COUNTER	Erases unnecessary expressions.
22. END OF DATA	Stops execution.
Program TRANS[a]	Codes the equations into FORTRAN subroutines.

[a]Executed separately at present.

TABLE 2.- FORTRAN SYMBOLS

Original variable	FORTRAN symbol	Original variable	FORTRAN symbol
R	RAD	v'	VS
u	U	v_i	LAMB
$\dot{u}$	UD	w	W
U_T	UT	w'	WS
U_P	UP	$\dot{w}$	WD
U_F	UF	x	XCOR
$\dot{U}_F$	UFD	θ	THTA
v	V	ϕ	PHI
$\dot{v}$	VD	Ω	OMEG

```
READ MATRIX
#LAFP0303
03
1.0
-0.5          VS   VS
-0.5          WS   WS
01
1.0           VS
01
1.0           WS
02
-1.0          VS%EONE
-1.0          WS%ETWO
03
1.0        %ETRE
-1.0          PHI THTA
-1.0           VS   WS THTA
02
1.0         PHI
1.0        %EFUR THTA
02
1.0           VS%ETWO
-1.0          WS%EONE
03
-1.0       %ETRE THTA
-1.0          PHI
-1.0           VS   WS
02
-1.0          PHI THTA
1.0        %EFUR
READ MATRIX
# VELO301
03
-1.0           UD
1.0            UFD
1.0            OMEG    V
02
-1.0           VD
-1.0       %XUUF OMEG
02
-1.0           OMEG  RAD%LAMD
-1.0           WD
FORM MATRIX
#LAFP# VEL#AVEL*E2D1
MATRIX EXPRESSION
02#AVEL
%  UTO201%  UP0301
NEGATE EXPRESSION
%  UT
```

TABLE 3.- CONCLUDED

NEGATE EXPRESSION
% UP

Note: %EONE, %ETWO, etc., are expressions
read earlier in the program.

Output of U_T

```
*********************************************
*       DETAILS OF THE EXPRESSION     UT    *
*                      NUMBER OF TERMS  15  *
*********************************************

  1  1.000*  VS*OMEG*    V*
  2  1.000*  VD*
  3  1.000*XCOR*OMEG*
  4  1.000*   U*OMEG*
  5 -1.000*  UF*OMEG*
  6 -0.500*  VS*  VS*  VD*
  7 -0.500*  VS*  VS*XCOR*OMEG*
  8 -0.500* PHI* PHI*  VD*
  9 -0.500* PHI* PHI*XCOR*OMEG*
 10 -1.000* PHI*THTA*  VD*
 11 -1.000* PHI*THTA*XCOR*OMEG*
 12  1.000* PHI*LAMB*OMEG* RAD*
 13  1.000* PHI*  WD*
 14  1.000*THTA*LAMB*OMEG* RAD*
 15  1.000*THTA*  WD*
```
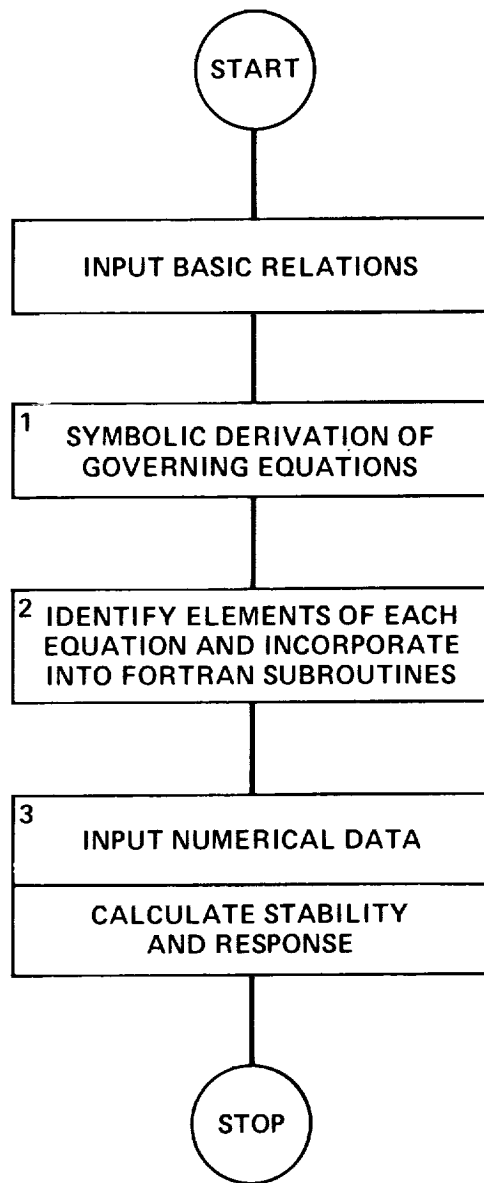
(a) Flow chart.

Figure 1.- Aeroelastic analyses.
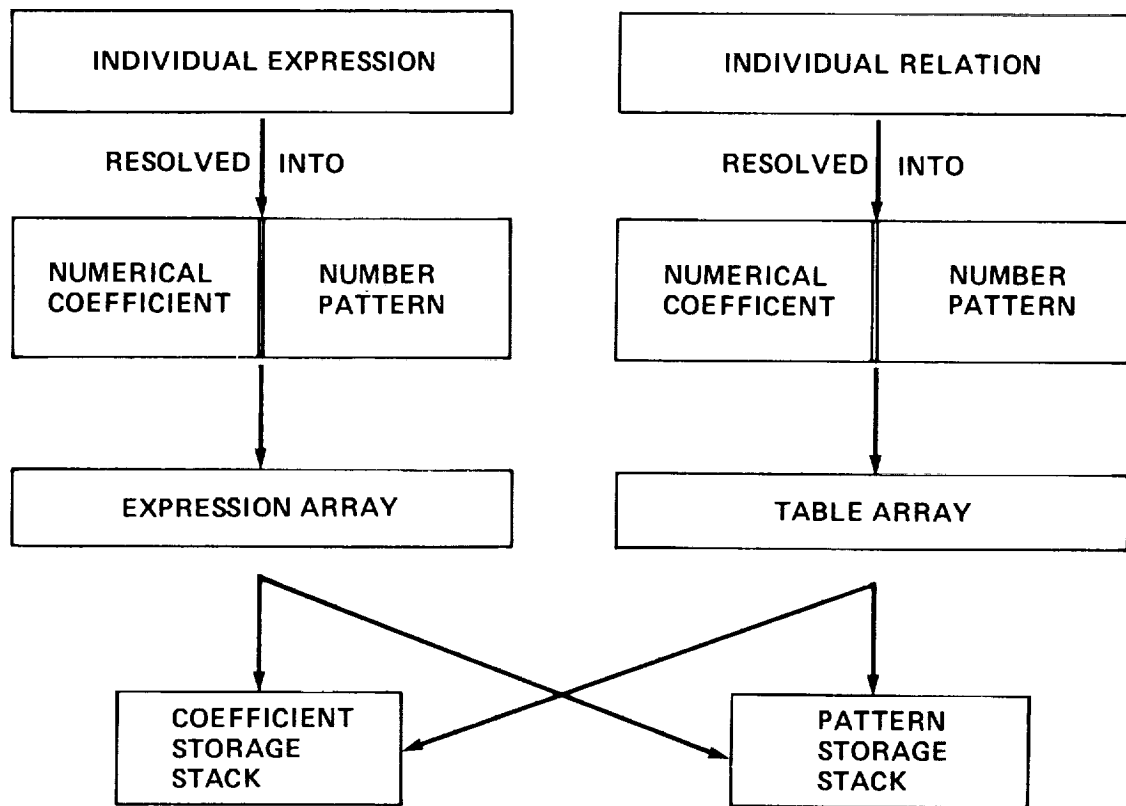
(b) Definition and storing of expressions and relations.

Figure 1.- Concluded.
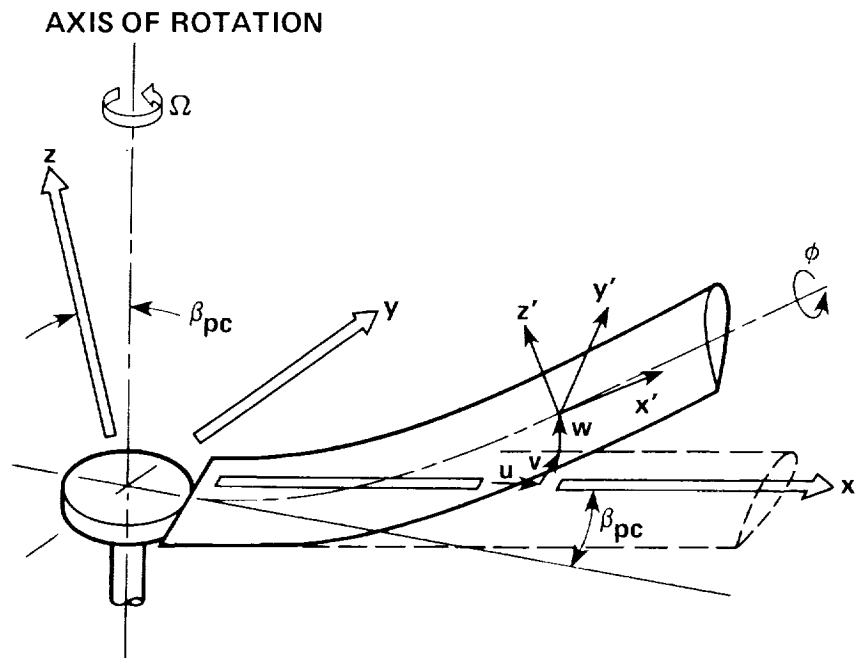
Figure 2.- Rotor blade coordinate systems and deflections
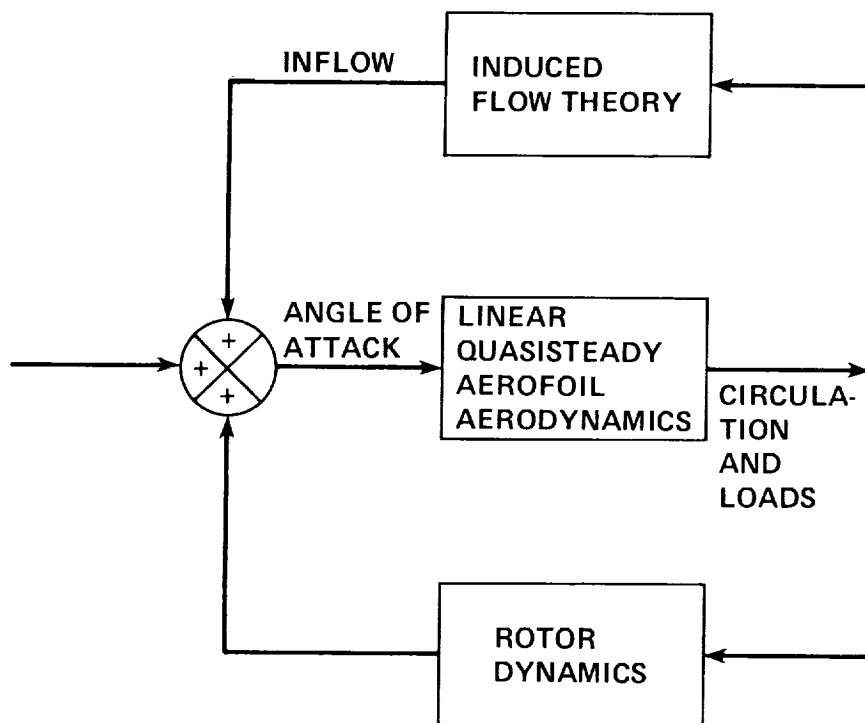
Figure 3.- Inflow dynamics.
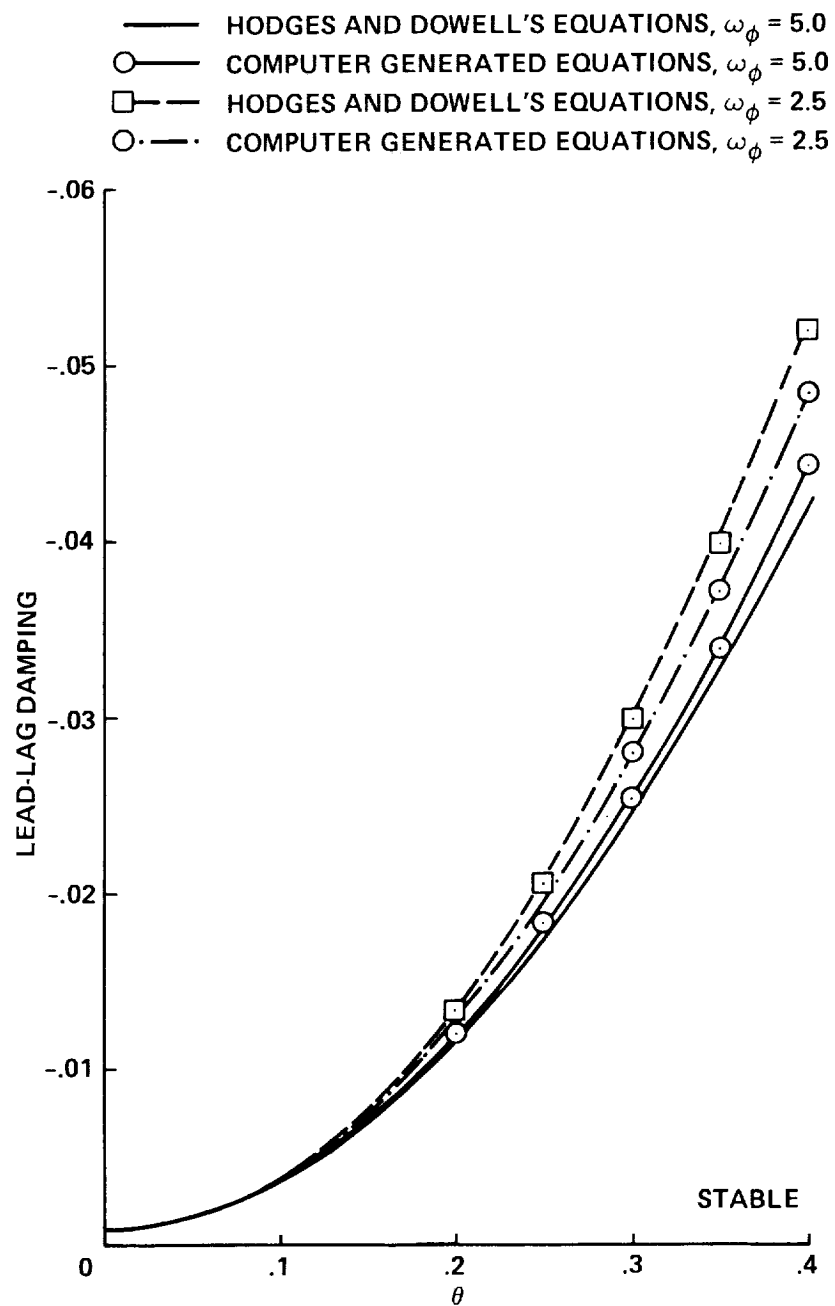
Figure 4.- Comparison of lead-lag damping from two sets of equations for varying pitch angle θ for soft in-plane blade for two torsional frequencies, $\omega_v = 0.7$, $\omega_w = 1.15$, $R = 1.0$, $\sigma = 0.1$.

81
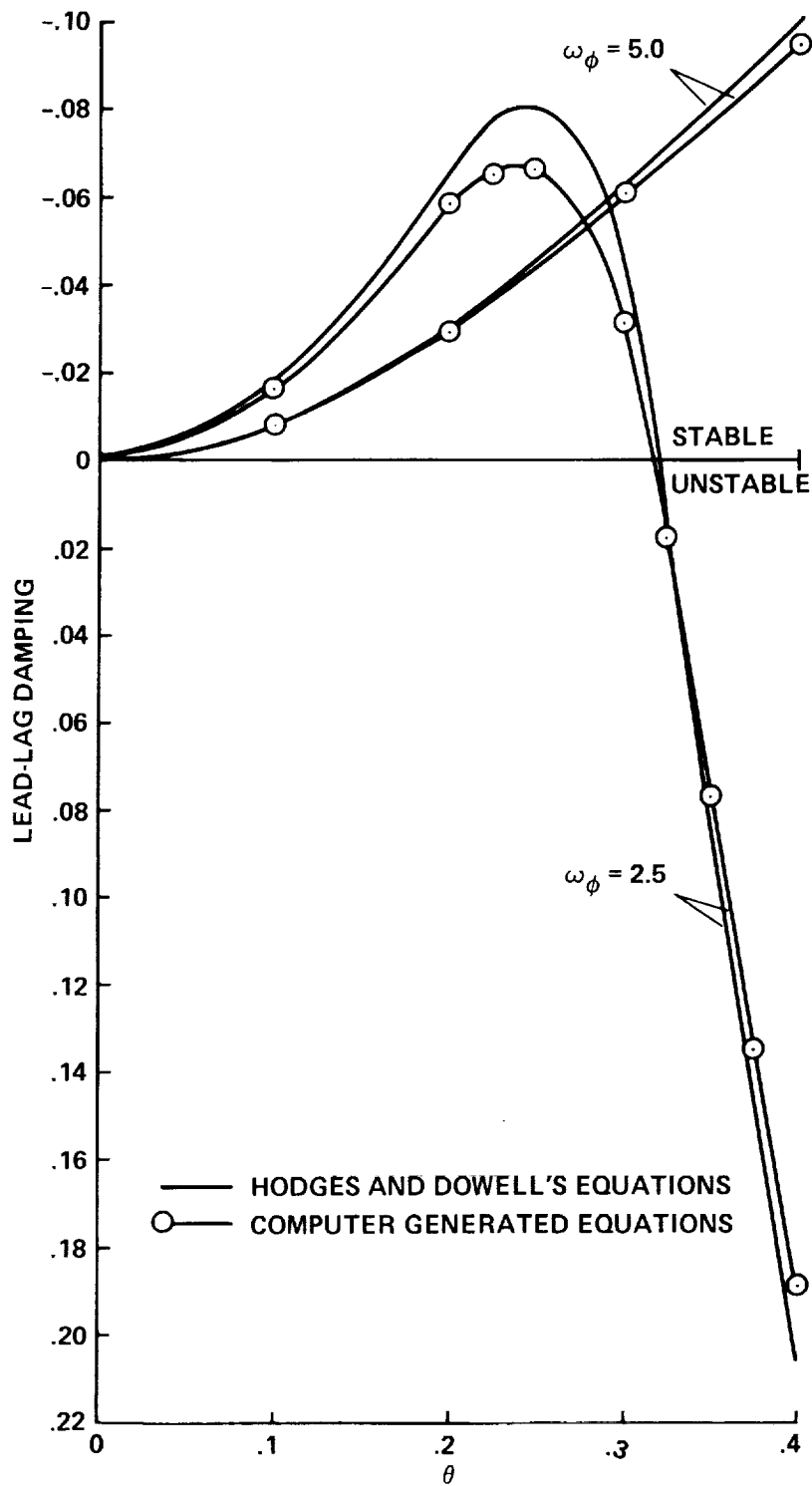
Figure 5.- Comparison of lead-lag damping from two sets of equations for varying pitch angle θ for stiff in-plane blade for two torsional frequencies, $\omega_v = 1.5$, $\omega_w = 1.15$, $R = 1.0$, $\sigma = 0.1$.
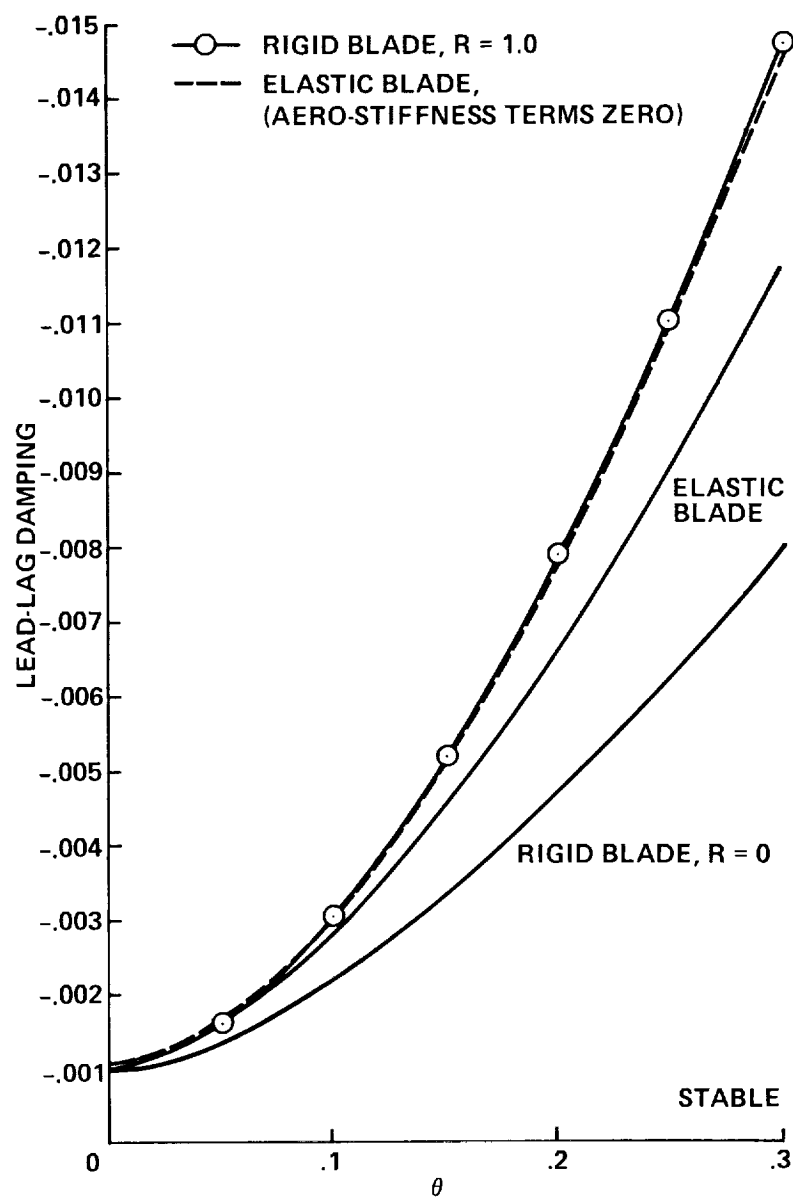
Figure 6.- Lead-lag mode damping versus pitch angle θ, from flap-lag analysis for a rigid blade model and an elastic blade model for a soft in-plane blade, $\omega_v = 0.7$, $\omega_w = 1.15$, $R = 1.0$, $\sigma = 0.05$, $C_{d_o} = 0.01$.

83
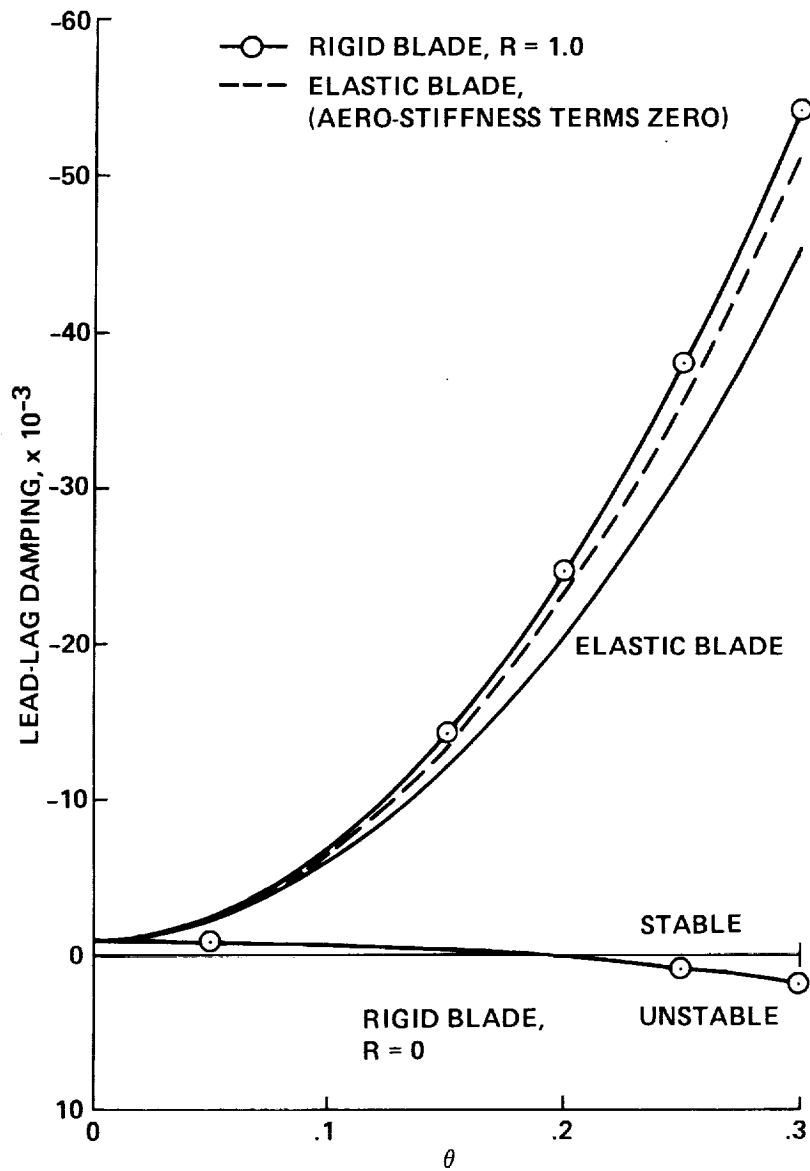
Figure 7.- Lead-lag mode damping versus pitch angle θ, from flap-lag analysis for a rigid blade model and an elastic blade model for a stiff in-plane blade, ω_v = 1.5, ω_w = 1.15, R = 1.0, σ = 0.05, C_{d_o} = 0.01.
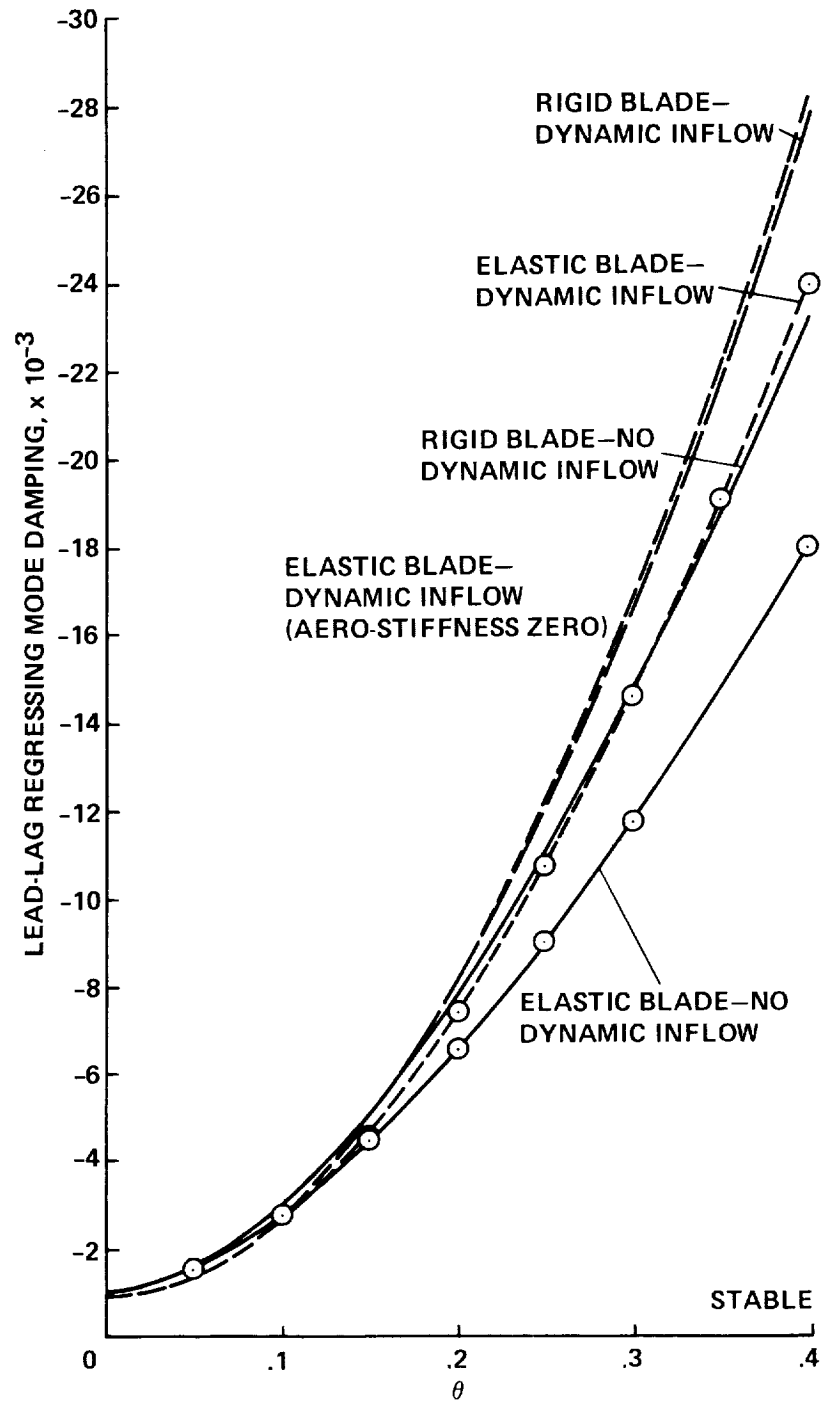
Figure 8.- Lead-lag regressing mode damping with dynamic inflow versus pitch angle θ, from rigid blade model and elastic blade model for a soft in-plane blade, $\omega_V = 0.7$, $\omega_W = 1.15$, $R = 1.0$, $\sigma = 0.05$.
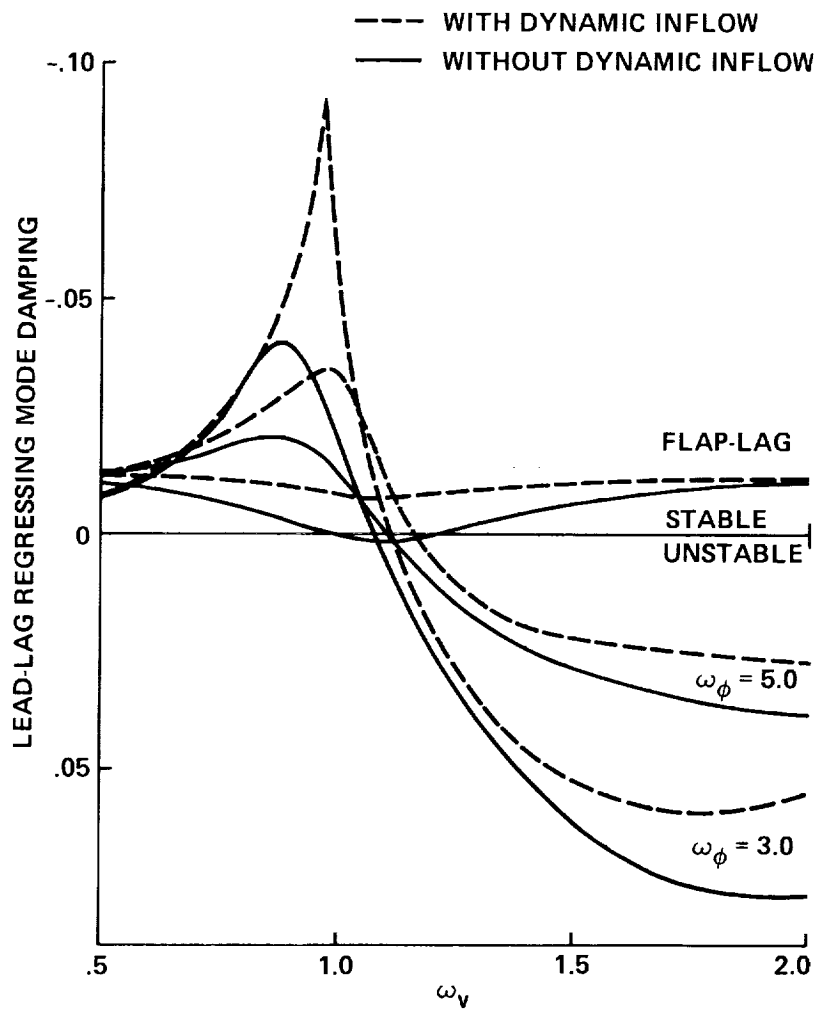
Figure 9.- The effect of torsion frequency and dynamic inflow on lead-lag regressing mode damping versus lead-lag frequency, $\theta = 0.3$ rad, $R = 0.0$, $\omega_w = 1.15$, $\sigma = 0.1$.

86
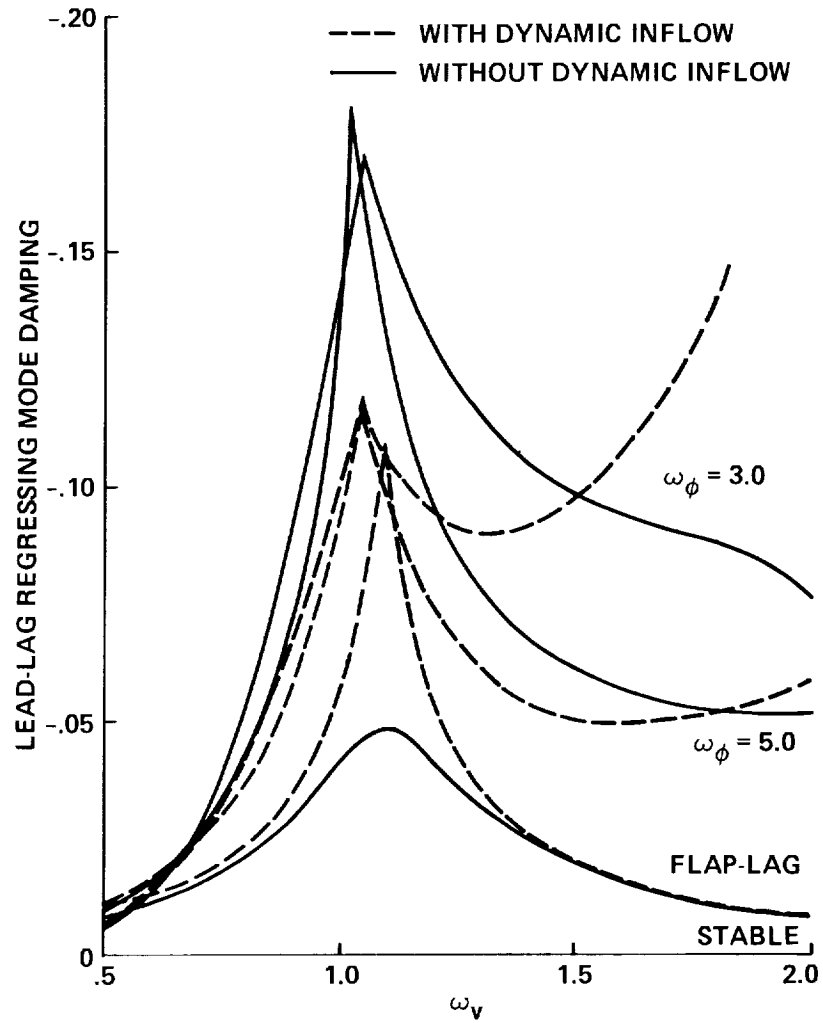
Figure 10.- The effect of torsion frequency and dynamic inflow on lead-lag regressing mode damping versus lead-lag frequency, $\theta = 0.3$ rad, $R = 1.0$, $\omega_w = 1.15$, $\sigma = 0.1$.
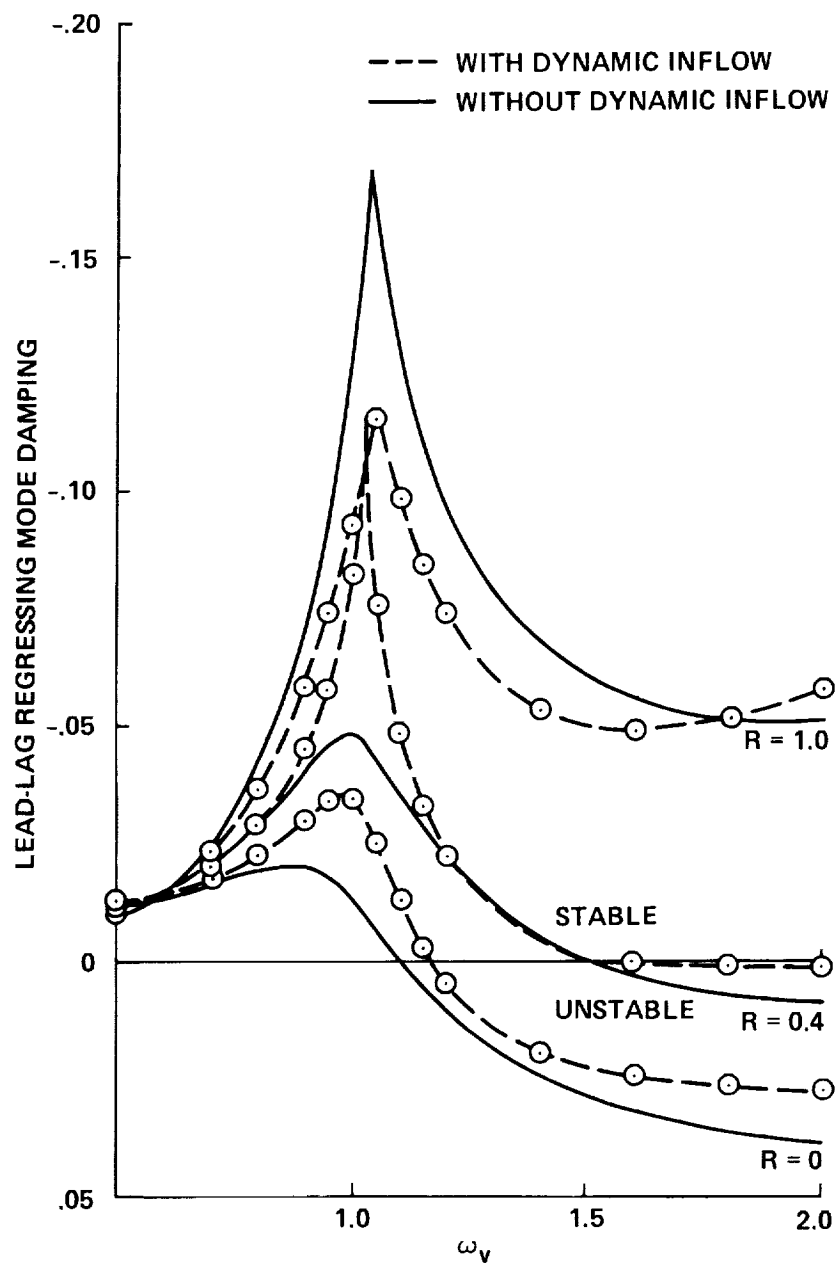
Figure 11.- The effect of elastic coupling and dynamic inflow on lead-lag regressing mode damping versus lead-lag frequency, $\theta = 0.3$ rad, $\omega_\phi = 5.0$, $\omega_w = 1.15$, $\sigma = 0.1$.
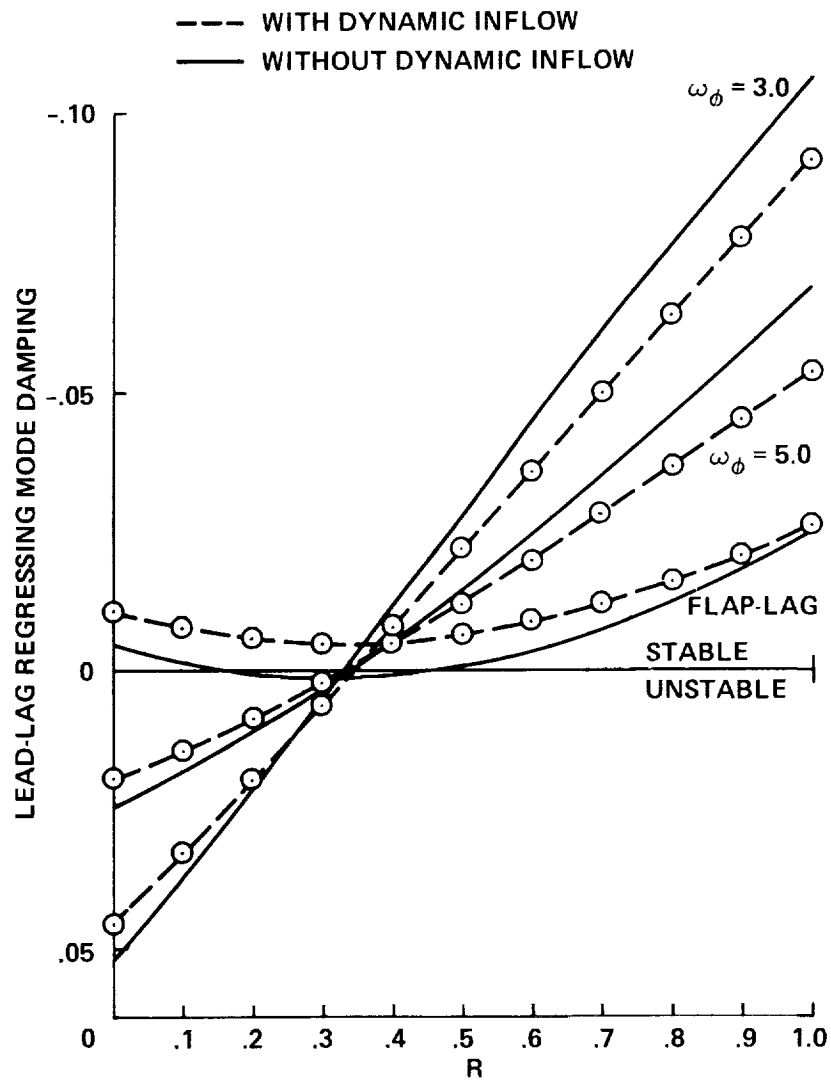
Figure 12.- The effect of torsion frequency and dynamic inflow on lead-lag regressing mode damping versus elastic coupling, $\theta = 0.3$ rad, $\omega_v = 1.4$, $\omega_w = 1.15$, $\sigma = 0.1$.
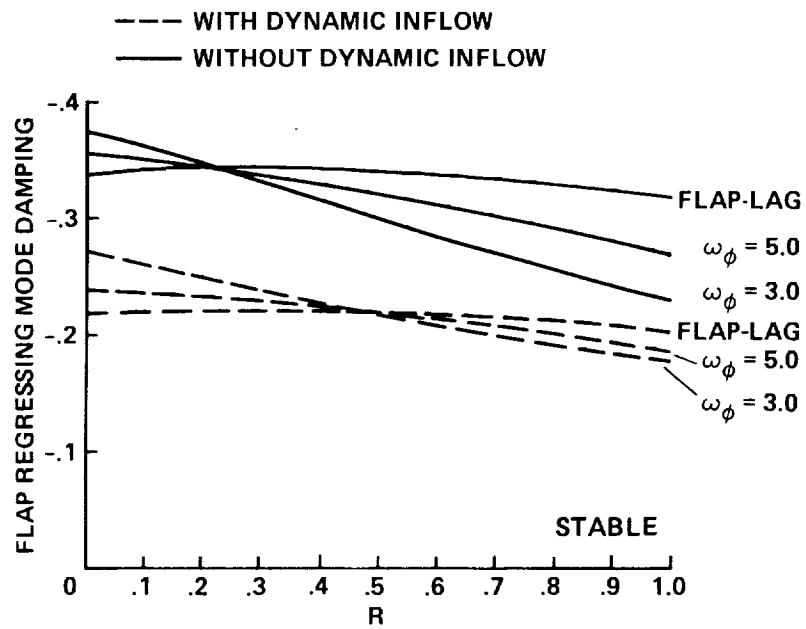
Figure 13.- The effect of torsion frequency and dynamic inflow on flap regressing mode damping versus elastic coupling, $\theta = 0.3$ rad, $\omega_v = 1.4$, $\omega_w = 1.15$, $\sigma = 0.1$.
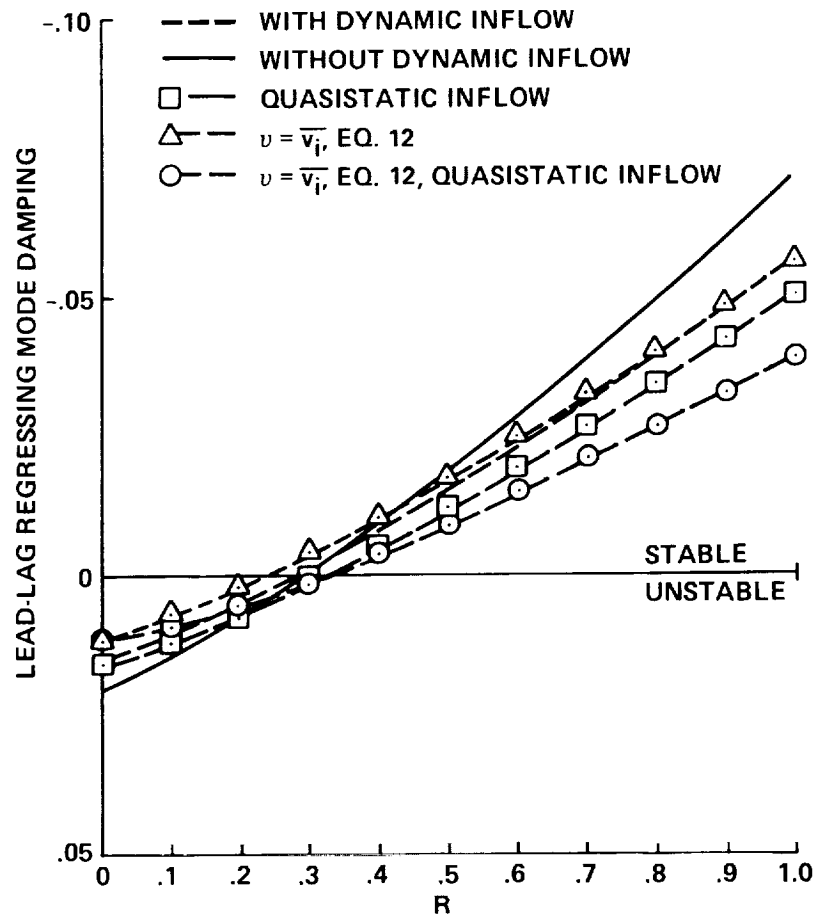
Figure 14.- Dynamic inflow modeling effects on lead-lag regressing mode damping versus elastic coupling. $\theta = 0.3$ rad, $\omega_\phi = 5.0$, $\omega_v = 1.4$, $\omega_w = 1.15$, $\sigma = 0.1$.
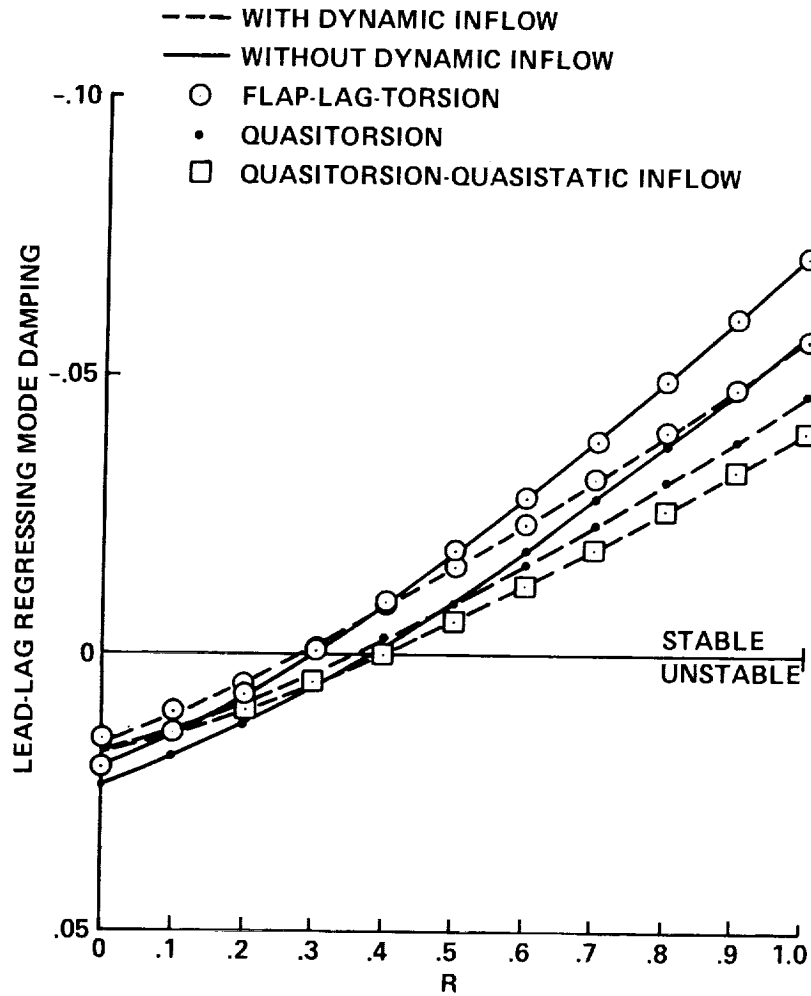
Figure 15.- Lead-lag regressing mode damping from flap-lag-torsion and quasi-static torsion model with and without dynamic inflow versus elastic coupling parameter, $\omega_\phi = 5.0$, $\omega_w = 1.15$, $\omega_v = 1.4$, $\sigma = 0.1$, $\theta = 0.3$ rad.
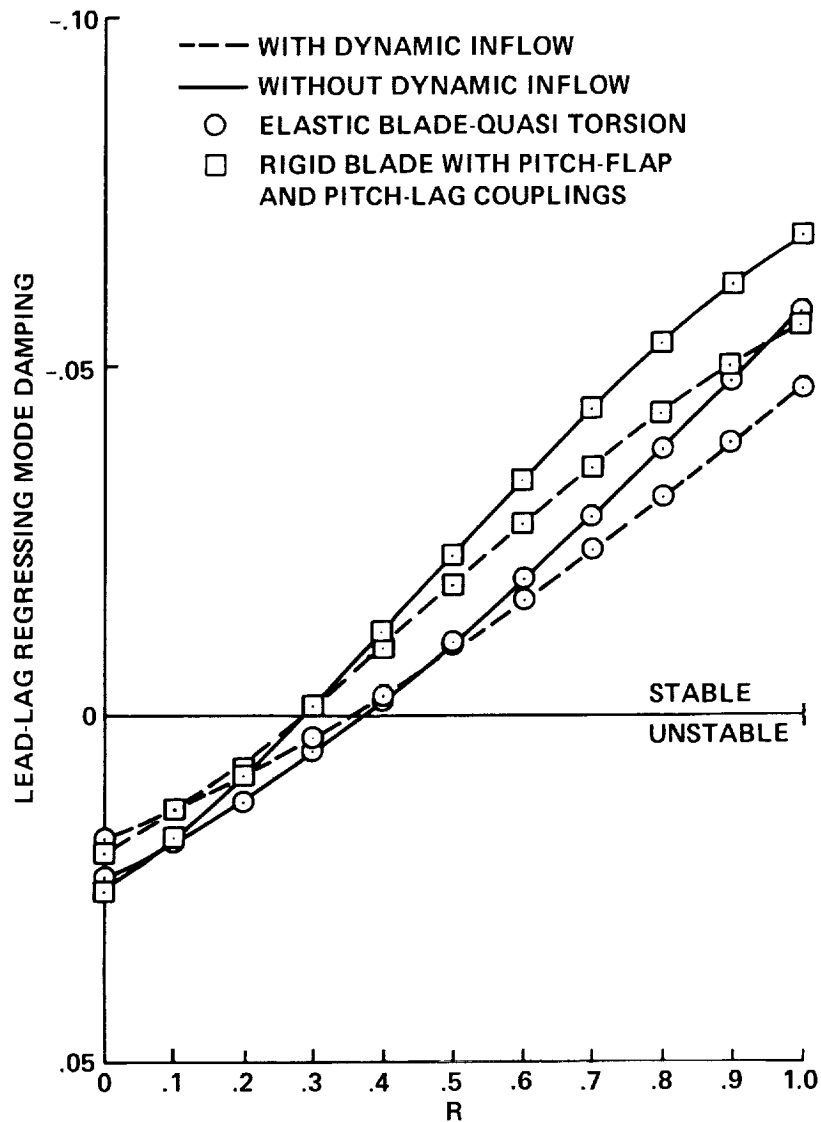
Figure 16.- Lead-lag regressing mode damping from a quasi-static torsion model and a rigid blade flap-lag model with kinematic pitch-flap and pitch-lag couplings, $\theta = 0.3$ rad, $\omega_v = 1.4$, $\omega_w = 1.15$, $\omega_\phi = 5.0$.

C - 2
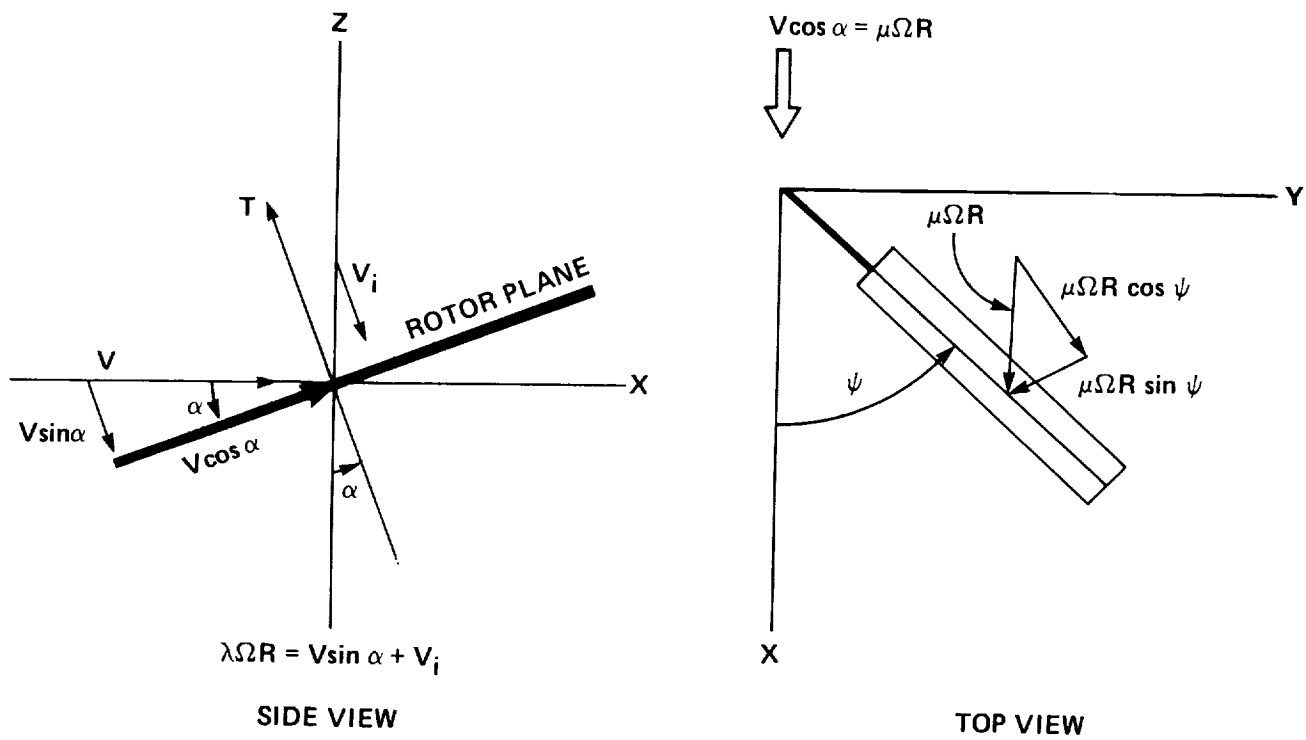
Figure 17.- Aerodynamic velocity components in forward flight.
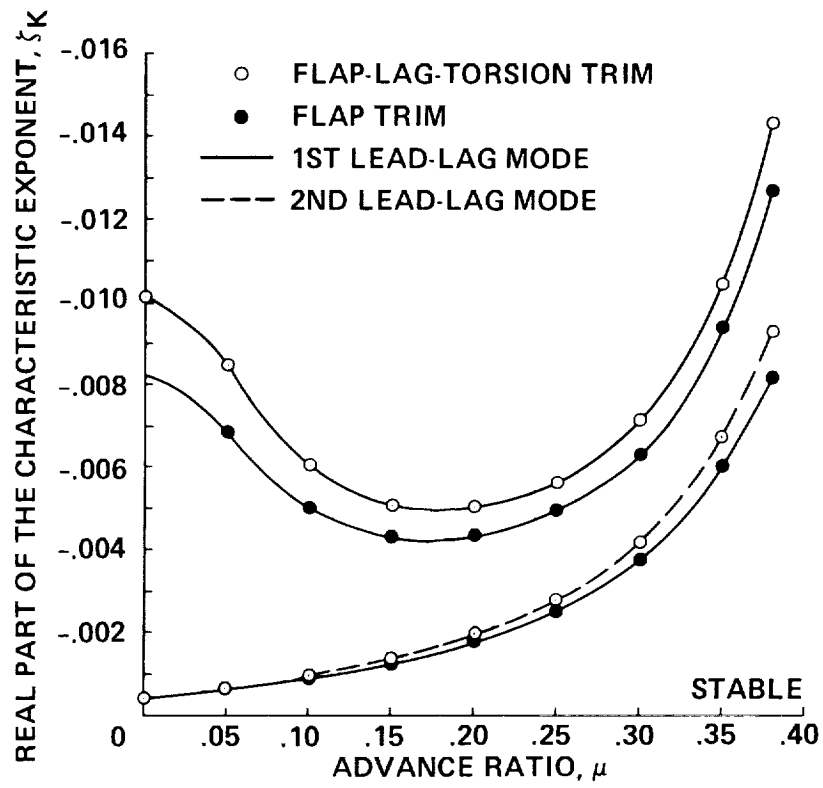
Figure 18.- The effect of number of degrees of freedom used in trim analysis on lead-lag damping versus advance ratio, soft in-plane, $\omega_v = 0.7$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, $R = 1.0$, propulsive trim.
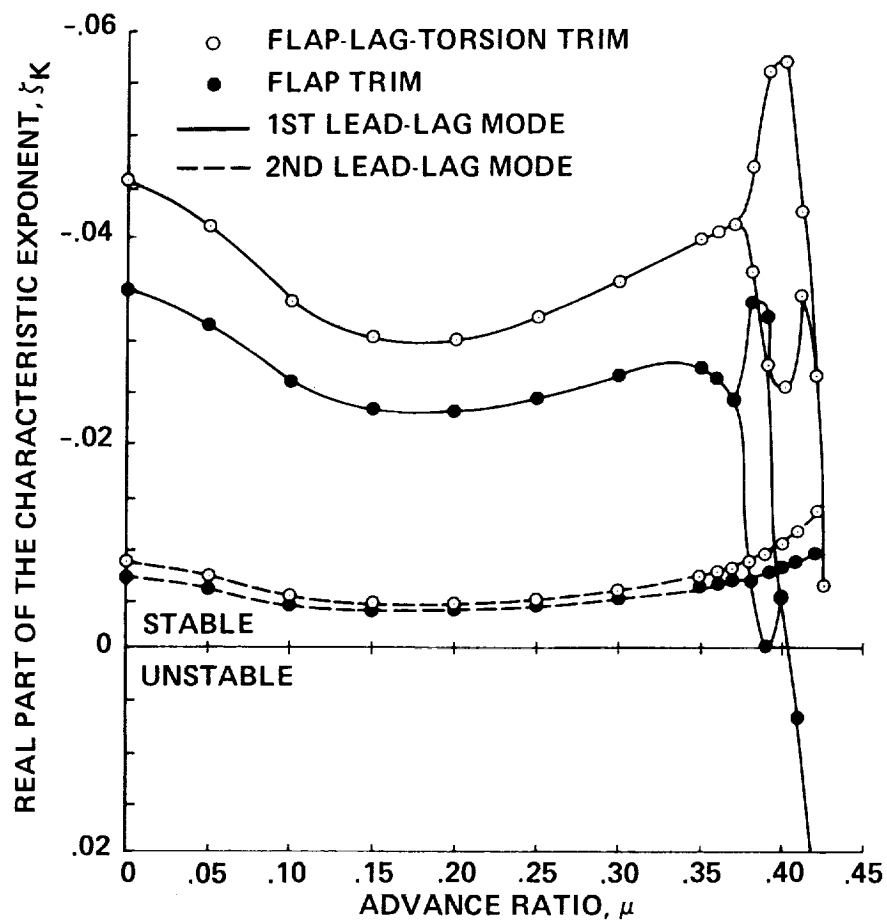
Figure 19.- The effect of number of degrees of freedom used in trim analysis on lead-lag damping versus advance ratio, stiff in-plane, $\omega_v = 1.4$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, $R = 1.0$-, propulsive trim.
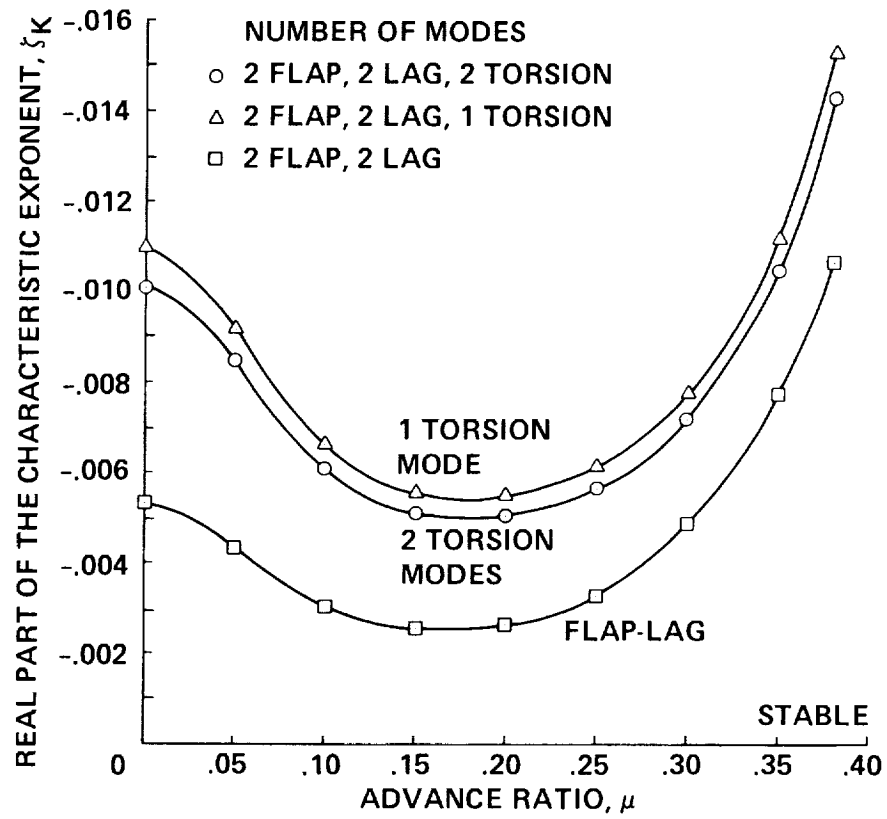
Figure 20.- Lead-lag mode damping versus advance ratio for a flap-lag-torsion model and a flap-lag model, soft in-plane, $\omega_v = 0.7$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, $R = 1.0$.
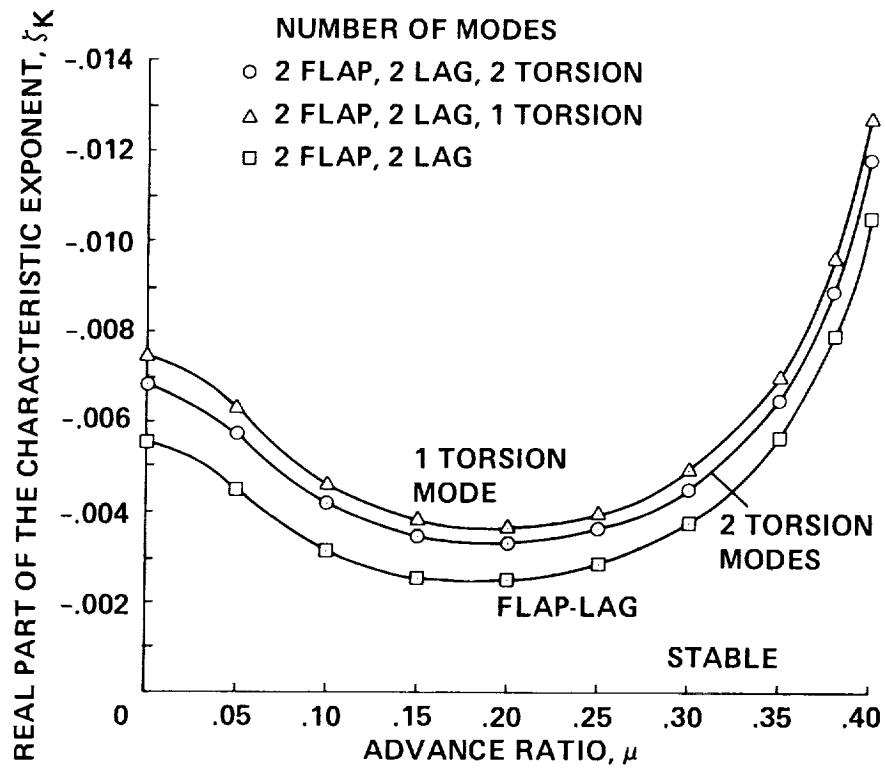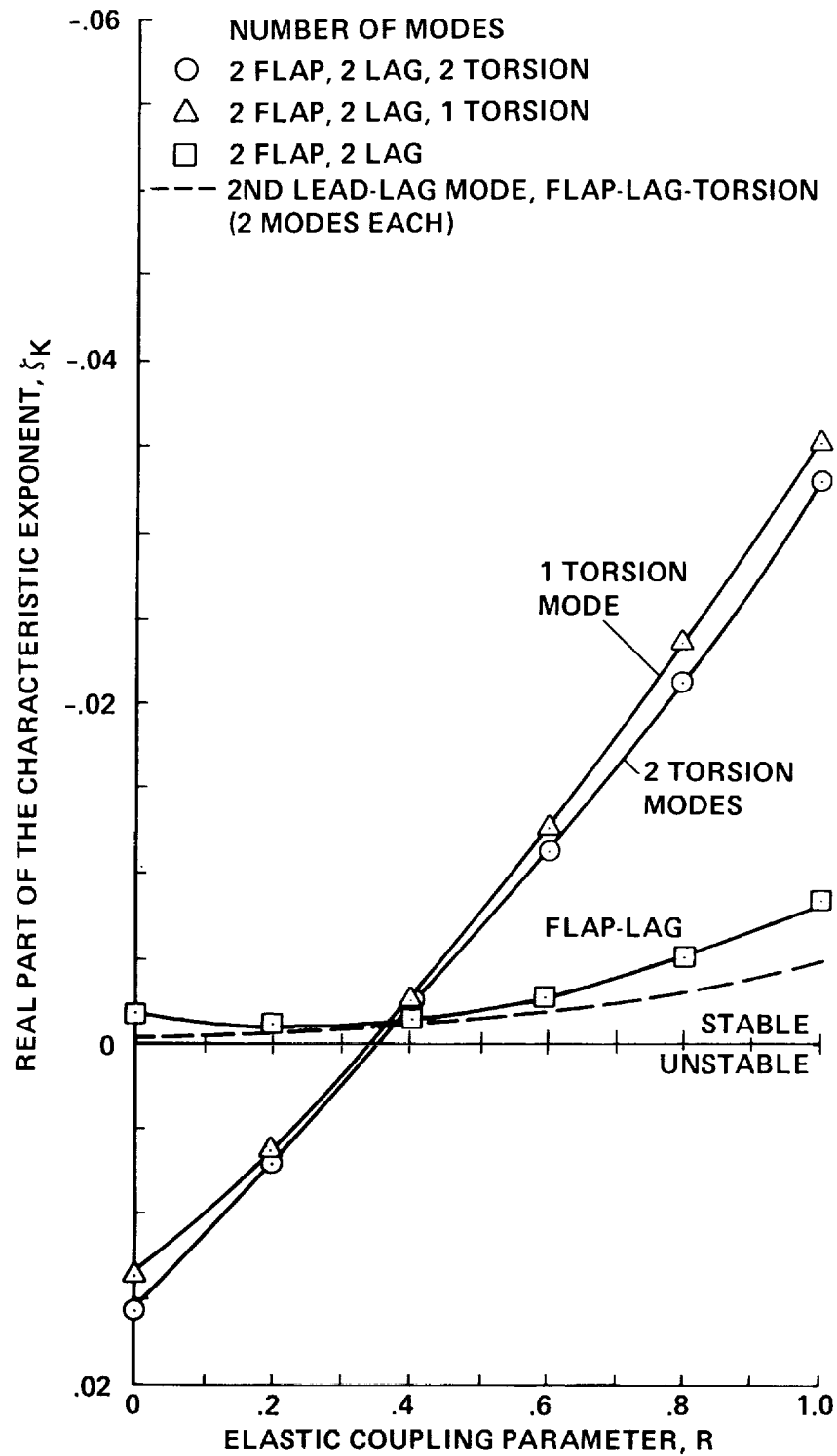
Figure 21.- Lead-lag mode damping versus advance ratio for a flap-lag-torsion model and a flap-lag model, soft in-plane, $\omega_v = 0.7$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, $R = 0.0$.
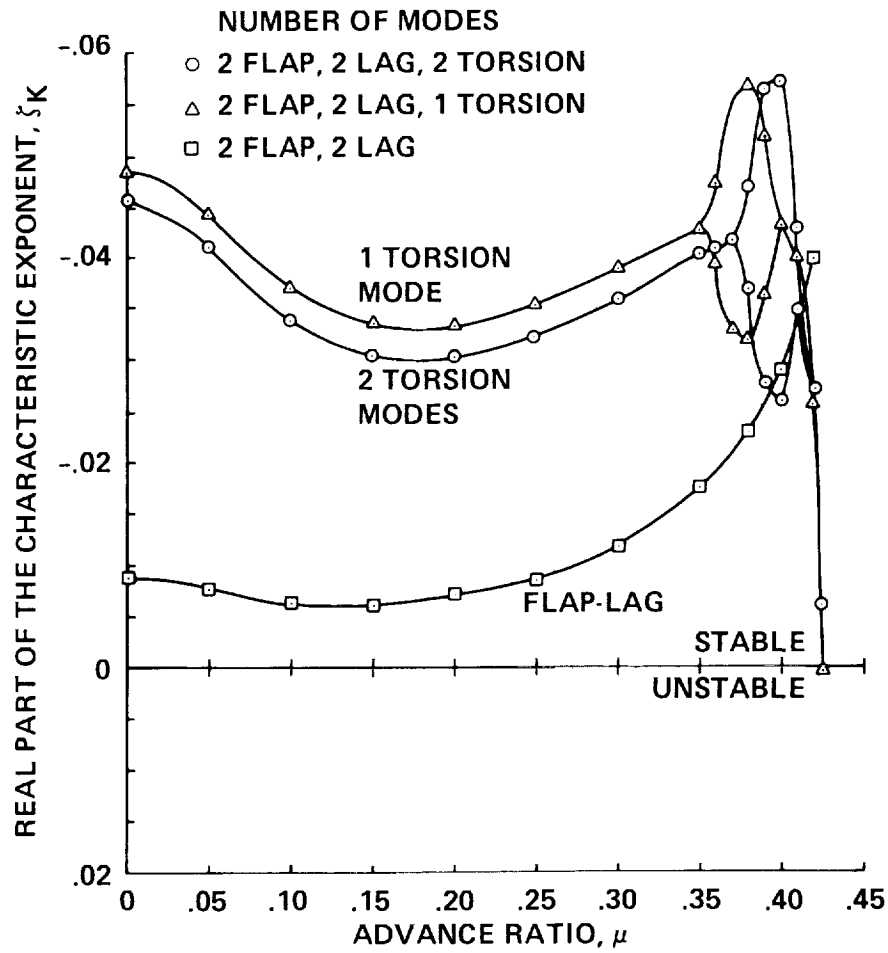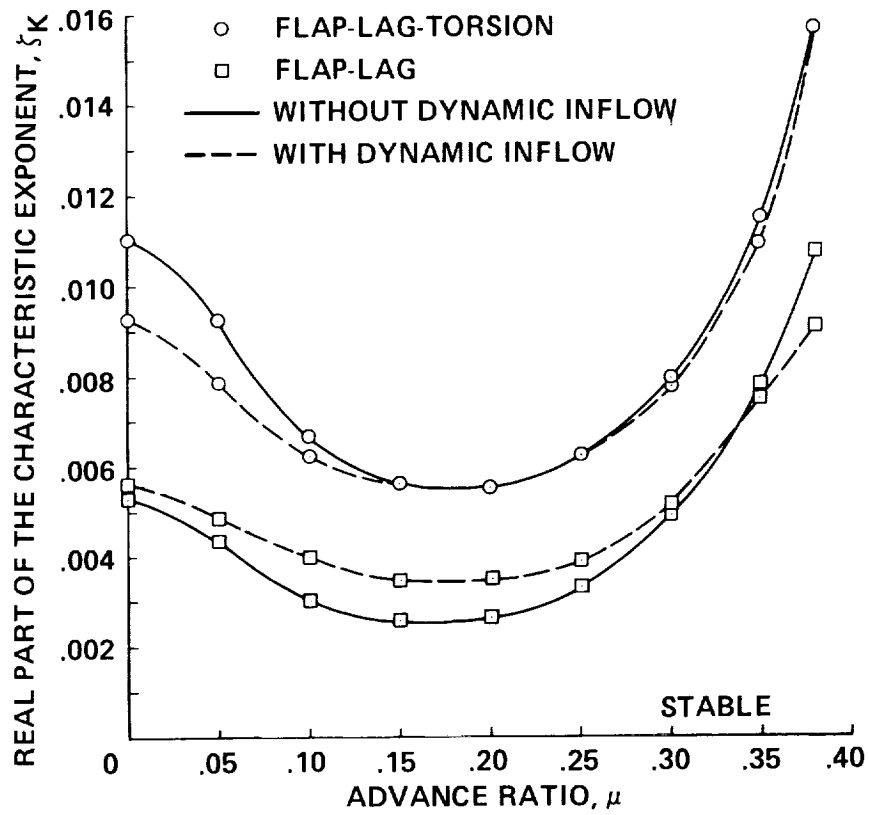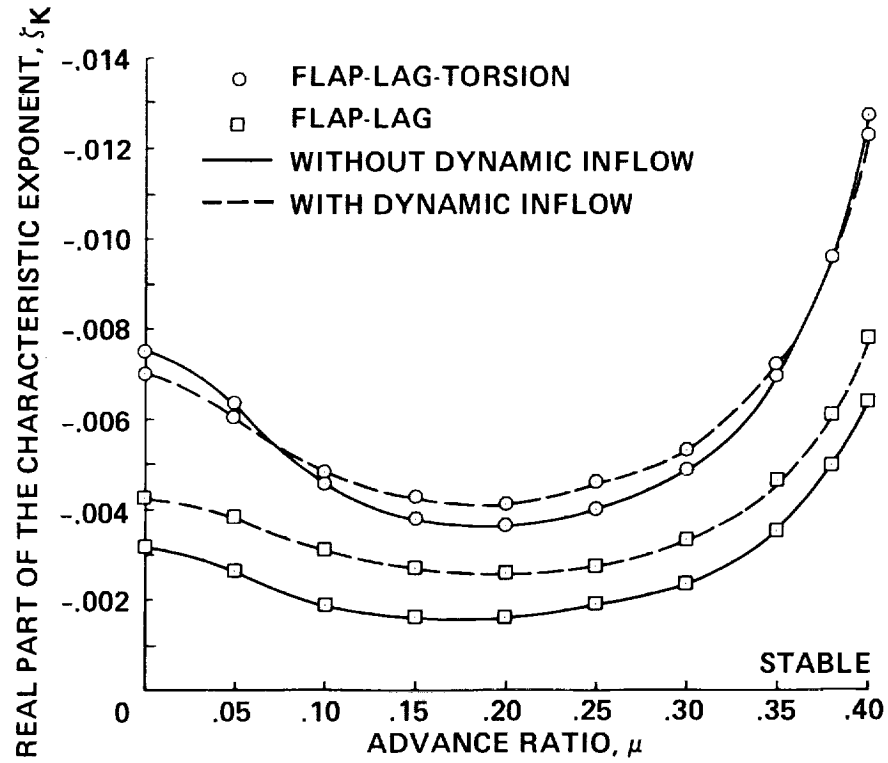
Figure 22.- Lead-lag mode damping versus advance ratio for a flap-lag-torsion model and a slap-lag model, stiff in-plane, ω_v = 1.4, ω_w = 1.15, ω_ϕ = 3.0, σ = 0.1, R = 1.0.
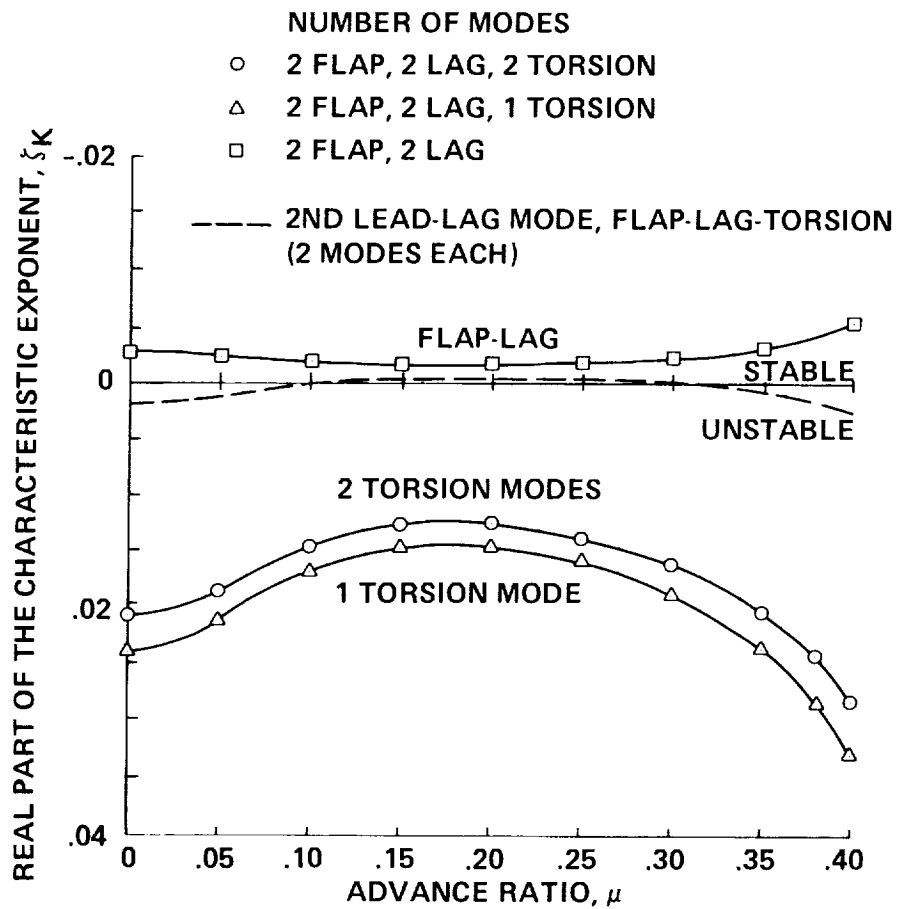
Figure 23.- Lead-lag mode damping versus advance ratio for a flap-lag-torsion model and a flap-lag model, stiff in-plane, $\omega_v = 1.4$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, R = 0.0.

Figure 24.- Lead-lag mode damping versus elastic coupling for a flap-lag-torsion model and flap-lag model, stiff in-plane, $\omega_v = 1.4$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, $\mu = 0.25$.

Figure 25.- The effect of torsion and dynamic inflow on lead-lag regressing mode damping versus advance ratio, soft in-plane, $\omega_v = 0.7$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, $R = 1.0$.
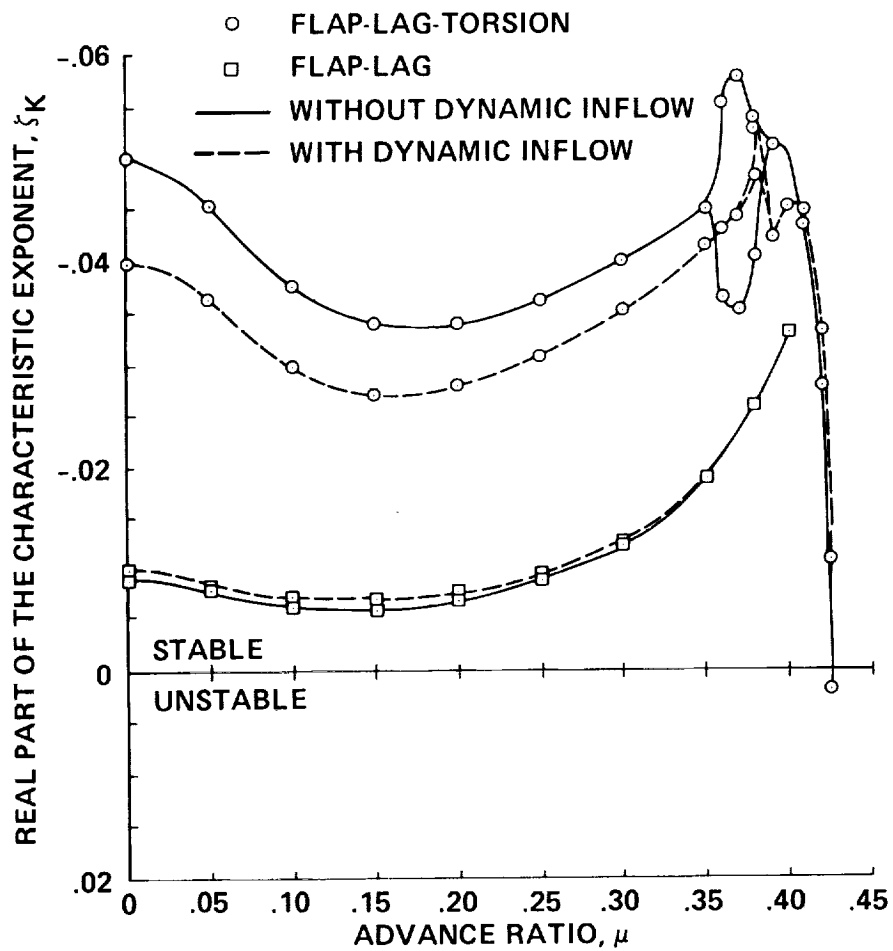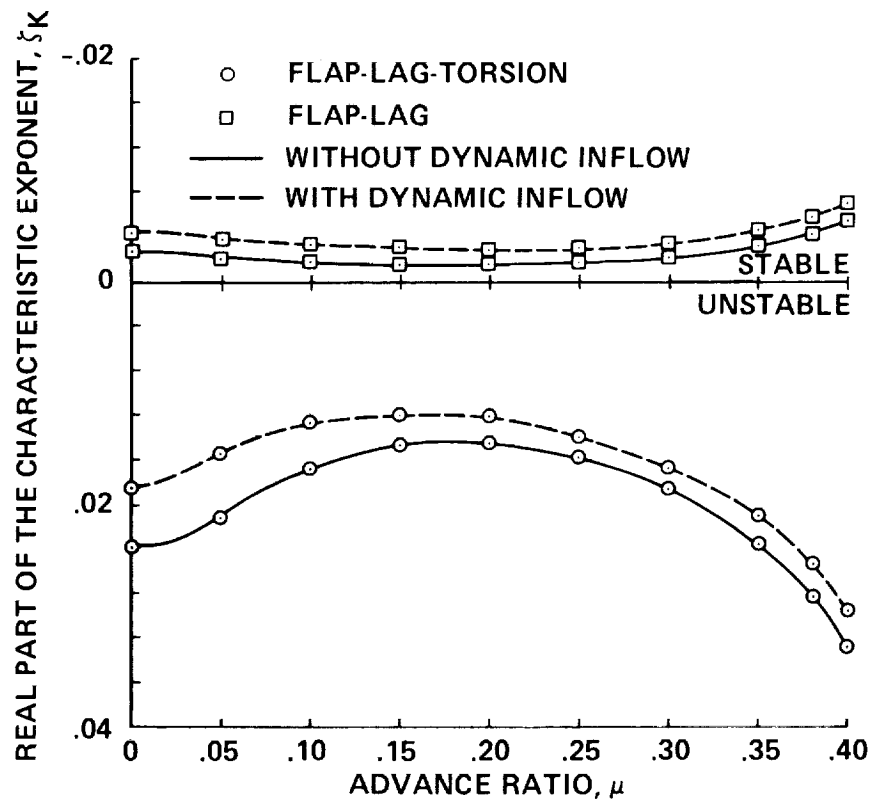
Figure 26.- The effect of torsion and dynamic inflow on lead-lag regressing mode damping versus advance ratio, soft in-plane, $\omega_v = 0.7$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, $R = 0.0$.

Figure 27.- The effect of torsion and dynamic inflow on lead-lag regressing mode damping versus advance ratio, stiff in-plane, $\omega_v = 1.4$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, R = 1.0.

Figure 28.- The effect of torsion and dynamic inflow on lead-lag regressing mode damping versus advance ratio, stiff in-plane, ω_V = 1.4, ω_W = 1.15, ω_ϕ = 3.0, σ = 0.1, R = 0.0.
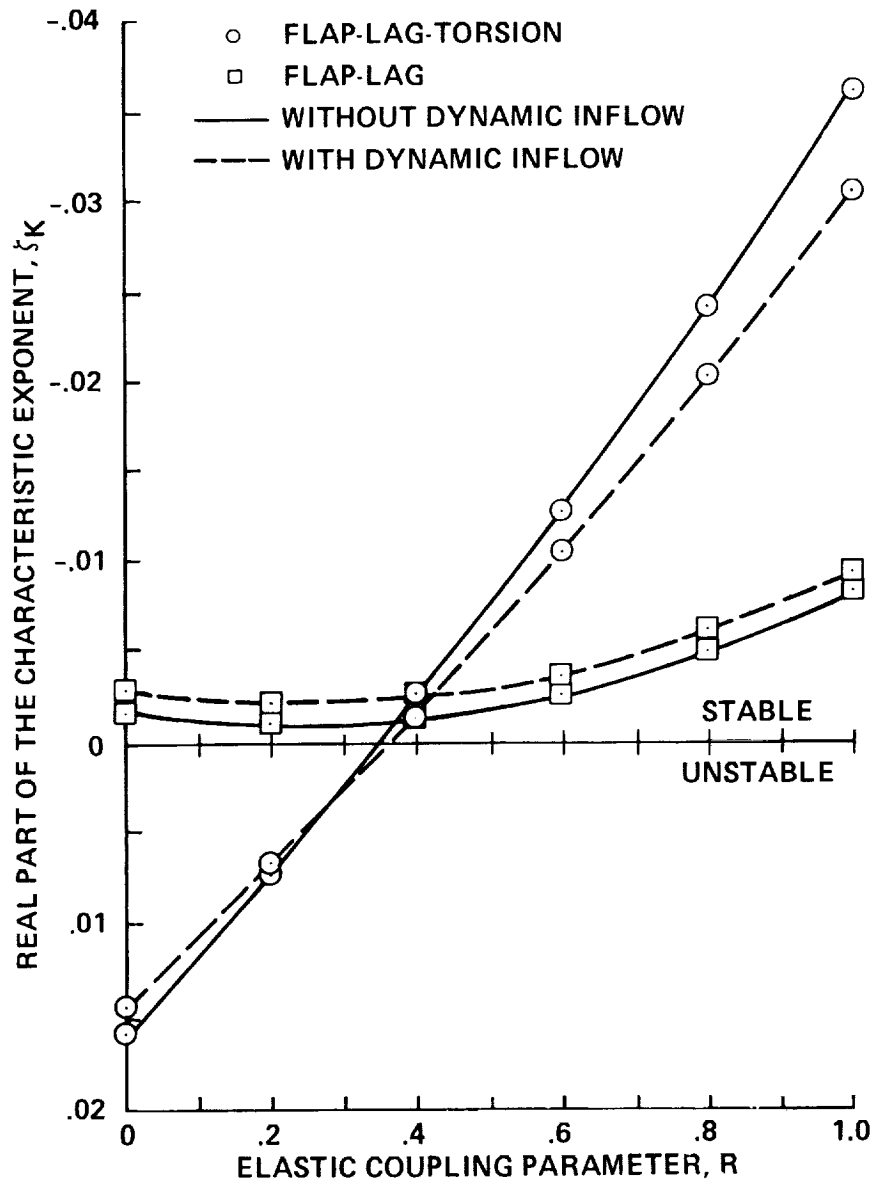
Figure 29.- The effect of torsion and dynamic inflow on lead-lag regressing mode damping versus elastic coupling, stiff in-plane, $\omega_v = 1.4$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, $\mu = 0.25$.
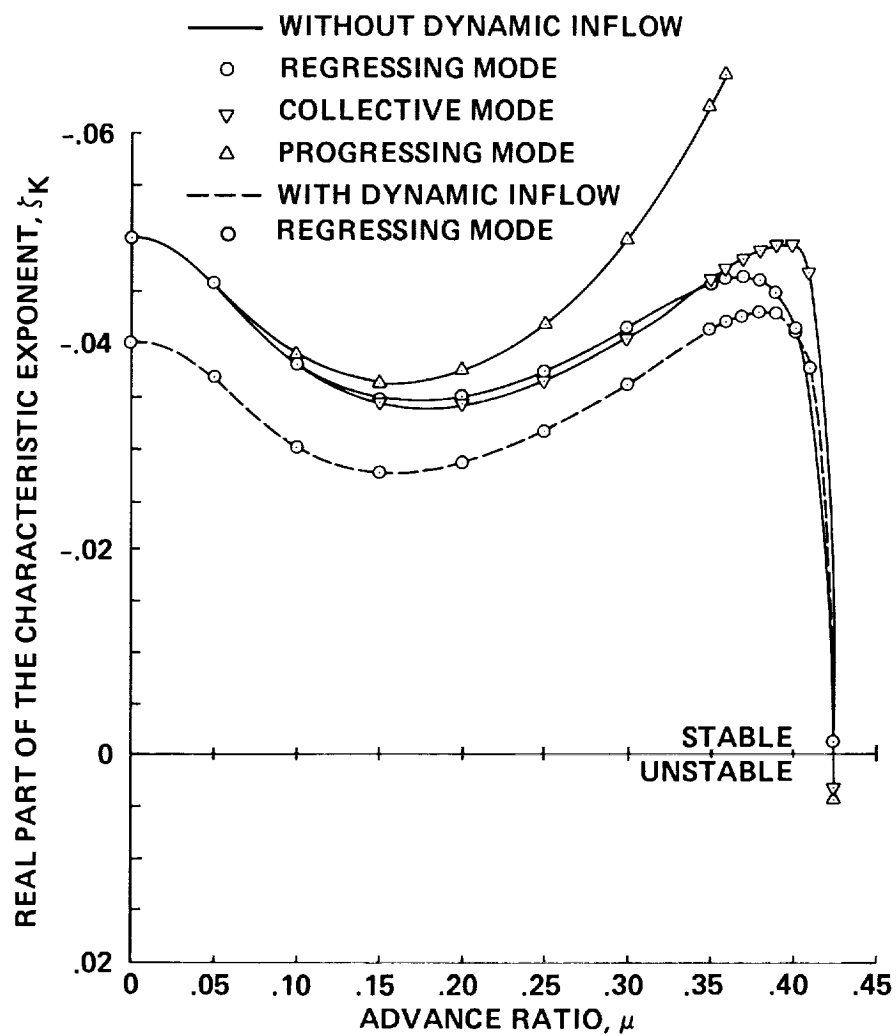
Figure 30.- Comparison of constant coefficient approximation and Floquet analysis, stiff in-plane, $\omega_v = 1.4$, $\omega_w = 1.15$, $\omega_\phi = 3.0$, $\sigma = 0.1$, $R = 1.0$.

1. Report No. NASA TM-86750	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle SYMBOLIC GENERATION OF ELASTIC ROTOR BLADE EQUATIONS USING A FORTRAN PROCESSOR AND NUMERICAL STUDY ON DYNAMIC INFLOW EFFECTS ON THE STABILITY OF HELICOPTER ROTORS		5. Report Date June 1986
		6. Performing Organization Code
7. Author(s) T. S. R. Reddy		8. Performing Organization Report No.
		10. Work Unit No. A-85227
9. Performing Organization Name and Address Ames Research Center Moffett Field, CA 94035		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code 505-42-11

15. Supplementary Notes

Point of Contact: William Warmbrodt, Ames Research Center, MS 247-1, Moffett Field, CA 94035
(415) 694-5642 or FTS 464-5642

16. Abstract

The process of performing an automated aeroelastic stability analysis for an elastic-bladed helicopter rotor is discussed. A symbolic manipulation program, written in FORTRAN, is used to aid in the derivation of the governing equations of motion for the elastic-bladed rotor. The blades undergo coupled bending and torsional deformations. Two-dimensional quasi-steady aerodynamics below stall are used. Although reversed flow effects are neglected, unsteady effects, modeled as dynamic inflow are included. Using a Lagrangian approach, the governing equations are derived in generalized coordinates using the symbolic program. The symbolic program generates the steady and perturbed equations and writes into subroutines to be called by numerical routines. The symbolic program can operate both on expressions and on matrices. For the case of hovering flight, the blade and dynamic inflow equations are converted to equations in a multiblade coordinate system by rearranging the coefficients of the equations. For the case of forward flight, the multiblade equations are obtained through the symbolic program. The final multiblade equations are capable of accommodating any number of elastic blade modes. The computer implementation of this procedure consists of three stages: (1) the symbolic derivation of equations; (2) the coding of the equations into subroutines; and (3) the numerical study after identifying mass, damping, and stiffness coefficients for each equation. Damping results are presented in hover and in forward flight with, and without, dynamic inflow effects for various rotor blade models, including rigid blade lag-flap, elastic flap-lag, flap-lag-torsion, and quasi-static torsion. Results from dynamic inflow effects which are obtained from a lift deficiency function for a quasi-static inflow model in hover are also presented.

17. Key Words (Suggested by Author(s)) Symbolic processing Helicopter Rotary wing Aeroelasticity	18. Distribution Statement Unlimited Subject category - 05
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages	22. Price*