CAMRAD/JA

A COMPREHENSIVE ANALYTICAL MODEL OF ROTORCRAFT AERODYNAMICS AND DYNAMICS

Johnson Aeronautics Version


Wayne Johnson

Johnson Aeronautics
Palo Alto, California
Johnson Aeronautics
Palo Alto, California

CAMRAD/JA

A COMPREHENSIVE ANALYTICAL MODEL OF
ROTORCRAFT AERODYNAMICS AND DYNAMICS

Johnson Aeronautics Version


Wayne Johnson

1988
CAMRAD/JA
A COMPREHENSIVE ANALYTICAL MODEL OF ROTORCRAFT AERODYNAMICS AND DYNAMICS

Johnson Aeronautics Version


Wayne Johnson
Johnson Aeronautics
Palo Alto, California

SUMMARY

The use of a comprehensive analytical model of rotorcraft aerodynamics and dynamics is presented. This analysis is designed to calculate rotor performance, loads, and noise; helicopter vibration and gust response; flight dynamics and handling qualities; and system aeroelastic stability. The analysis is a combination of structural, inertial, and aerodynamic models that is applicable to a wide range of problems and a wide class of vehicles. The analysis is intended for use in the design, testing, and evaluation of rotors and rotorcraft, and to be a basis for further development of rotary wing theories. The analysis is implemented in a digital computer program, called CAMRAD/JA.
CONTENTS

1. Introduction ........................................................................... 1
   1.1 Computational Tasks ...................................................... 1
   1.2 Trim Solution .................................................................. 4
   1.3 Configuration Model ..................................................... 8
   1.4 Rotor Model .................................................................. 9
   1.5 Wake Model .................................................................. 10
   1.6 Aircraft Model ............................................................. 11

2. Code Construction .................................................................. 13
   2.1 Program Organization ................................................... 13
      Rotorcraft analysis subprograms ....................................... 13
      Input file preparation subprograms ................................. 20
      Airfoil file preparation subprograms ............................... 20
      CAMRAD/JA common blocks ......................................... 21
   2.2 Program Files ............................................................... 23
      Rotorcraft analysis ......................................................... 23
      Input file preparation .................................................... 26
      Airfoil file preparation .................................................. 27
   2.3 Program Skeleton .......................................................... 28
      INPUT ........................................................................ 29
      AIRFOIL ....................................................................... 30
      CAMRADJA .................................................................. 31
   2.4 Software ......................................................................... 53
      2.4.1 Software Conventions ........................................... 53
      2.4.2 Software Tool ....................................................... 55

3. Input File Preparation .......................................................... 57
   3.1 Input Data Format ........................................................ 57
   3.2 Job Structure ............................................................... 59
   3.3 Input Variables ............................................................ 61
3.3.1 Namelist NLJOB ................................................. 61

Job description ................................................. 61
Input and output unit numbers ............................... 62

4. Airfoil File Preparation ........................................ 63

4.1 Airfoil Table Format ........................................... 63
4.2 Job Structure ................................................. 66
4.3 Input Variables .............................................. 68

4.3.1 Namelist NLTABL ............................................ 68

Job description ................................................. 68
Angle-of-attack boundaries .................................... 68
Mach number boundaries .................................... 68
Radial segments ............................................. 69
Print and plot definition .................................... 69
Input and output unit numbers ............................ 69

4.3.2 Namelist NLCHAR ............................................ 69
4.3.3 Input airfoil file format ................................. 70

4.4 Notes .......................................................... 72

5. Additional Input File Formats ................................. 79

5.1 Blade Bending Mode File ..................................... 79
5.2 Airframe Interference Velocity File ....................... 81
5.3 Airframe Aerodynamic Coefficient File .................... 83
5.4 Airframe Stability Derivative File ......................... 86
5.5 CFD Interface Input and Output Files ...................... 89

6. Rotorcraft Analysis .............................................. 93

6.1 Overview ..................................................... 93
6.2 Job Structure ................................................. 94
6.3 Input Variables ............................................. 96

6.3.1 Namelist NLCASE ........................................... 98

Job description ................................................. 98
Input and output unit numbers ............................... 98

6.3.2 Namelist NLTRIM ........................................... 100

Case description ................................................. 100
Operating conditions ......................................... 103
Aircraft description ........................................... 104

vi
<table>
<thead>
<tr>
<th>Section Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion analysis</td>
<td>105</td>
</tr>
<tr>
<td>Wake analysis</td>
<td>106</td>
</tr>
<tr>
<td>Trim analysis</td>
<td>107</td>
</tr>
<tr>
<td>Initial control settings</td>
<td>110</td>
</tr>
<tr>
<td>Targets for wind tunnel trim</td>
<td>111</td>
</tr>
<tr>
<td>Trim output control</td>
<td>111</td>
</tr>
<tr>
<td>Self-tuning regulator</td>
<td>112</td>
</tr>
<tr>
<td>Self-tuning regulator in trim analysis</td>
<td>113</td>
</tr>
<tr>
<td>Self-tuning regulator measurement and</td>
<td>114</td>
</tr>
<tr>
<td>control variables</td>
<td></td>
</tr>
<tr>
<td>6.3.3 Namelist NLRTR</td>
<td>121</td>
</tr>
<tr>
<td>Rotor configuration</td>
<td>121</td>
</tr>
<tr>
<td>Higher harmonic forces and control</td>
<td>122</td>
</tr>
<tr>
<td>Aerodynamic model</td>
<td>123</td>
</tr>
<tr>
<td>Aerodynamic model -- Reynolds number correction</td>
<td>123</td>
</tr>
<tr>
<td>Aerodynamic model -- CFD interface</td>
<td>124</td>
</tr>
<tr>
<td>Stall model</td>
<td>125</td>
</tr>
<tr>
<td>Stall model -- dynamic stall model</td>
<td>125</td>
</tr>
<tr>
<td>Inflow model</td>
<td>126</td>
</tr>
<tr>
<td>Dynamic model -- bending and torsion modes</td>
<td>128</td>
</tr>
<tr>
<td>Dynamic model -- gimbal/teeter hinge</td>
<td>129</td>
</tr>
<tr>
<td>Dynamic model -- control system</td>
<td>129</td>
</tr>
<tr>
<td>Dynamic model -- lag damper</td>
<td>130</td>
</tr>
<tr>
<td>Dynamic model -- blade properties</td>
<td>130</td>
</tr>
<tr>
<td>Dynamic model -- pitch-bending coupling</td>
<td>131</td>
</tr>
<tr>
<td>Dynamic model -- root geometry</td>
<td>132</td>
</tr>
<tr>
<td>Dynamic model -- analysis</td>
<td>132</td>
</tr>
<tr>
<td>Blade section aerodynamic characteristics</td>
<td>133</td>
</tr>
<tr>
<td>Blade section inertial and structural</td>
<td>134</td>
</tr>
<tr>
<td>characteristics</td>
<td></td>
</tr>
<tr>
<td>6.3.4 Namelist NLWAKE</td>
<td>135</td>
</tr>
<tr>
<td>Nonuniform inflow model</td>
<td>135</td>
</tr>
<tr>
<td>Computational domain for CFD interface</td>
<td>139</td>
</tr>
<tr>
<td>Prescribed or rigid wake geometry</td>
<td>140</td>
</tr>
<tr>
<td>Free wake geometry</td>
<td>141</td>
</tr>
<tr>
<td>6.3.5 Namelist NLBODY</td>
<td>143</td>
</tr>
<tr>
<td>Airframe configuration</td>
<td>143</td>
</tr>
<tr>
<td>Location of aircraft components</td>
<td>145</td>
</tr>
<tr>
<td>Airframe elastic modes</td>
<td>146</td>
</tr>
<tr>
<td>Auxiliary forces</td>
<td>147</td>
</tr>
<tr>
<td>Control system</td>
<td>148</td>
</tr>
<tr>
<td>Aircraft aerodynamic characteristics</td>
<td>150</td>
</tr>
<tr>
<td>Nonlinear aerodynamic model</td>
<td>151</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

In the design, testing, and evaluation of rotors and rotorcraft, it is necessary to predict and explain the rotor performance, loads, and noise; the helicopter vibration and gust response; the flight dynamics and handling qualities; and the system aeroelastic stability. This capability is required at several levels, including conceptual design; detailed design, development, and modification; and research. A comprehensive analysis makes it possible to perform these tasks with a consistent, balanced, yet high level of technology in a single code.

A comprehensive analysis for rotorcraft was published in 1980. The digital computer program implementing the analysis has acquired the name CAMRAD (for Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics). Johnson Aeronautics has produced CAMRAD/JA, a new software implementation of the analysis, written utilizing a software tool that facilitates modifications, and incorporating major new capabilities. This report documents the use of CAMRAD/JA. The theoretical basis for CAMRAD/JA is described in volume I.

1.1 Computational Tasks

Figure 1-1 shows an outline of the tasks and problems solved by CAMRAD/JA. The structure at this level emphasizes solving the dynamic equations of motion. The first task is the trim analysis; other tasks start from the trim solution. The rotorcraft in trim is in a steady state, unaccelerated flight condition; hence the rotor and airframe motion are periodic. The inverse problem, determining the control required for a specified flight condition, is being solved. The solution involves calculating the periodic rotor motion and the steady trim variables. After the calculation has converged, the performance, loads, and noise may be calculated. In CAMRAD/JA the blades of a rotor are assumed to be identical, with the same periodic motion. The
steady state flight condition
solve for controls and periodic motion
numerical integration of transient response (quasistatic rotor)

stability derivative calculation (quasistatic rotor)
analysis of time-invariant linear differential equations

linear differential equations
analysis (time-invariant or periodic)

Trim

Transient

Performance
Loads
Vibration
Noise

Flight Dynamics

Flutter

Figure 1-1. CAMRAD/JA tasks and solutions.
assumption of periodicity (with a fundamental frequency equal to the rotor rotational speed) excludes a calculation of the vibratory dynamic and aerodynamic interaction between two rotors of unequal rotation rates, such as a main rotor and tail rotor; the static or mean interaction is always taken into account.

The flight dynamics analysis is based on a frequency separation of the motion of the rotor and body, allowing the use of a quasistatic rotor solution. Hence the rotor and airframe stability derivatives are calculated, using prescribed perturbations of the body motion and controls, in the same analysis that is used for the trim solution (where the motion is truly steady state). Time-invariant linear differential equations for the aircraft rigid-body motions are constructed. The poles, zeros, and eigenvectors of these equations define the aircraft flying qualities.

The transient analysis involves an integration in time to obtain the general vehicle response. For CAMRAD/JA, the only transients considered are those produced by rigid body dynamics, pilot inputs, and gusts, all of which are slow relative to the rotor rotational frequency. Hence a quasistatic rotor solution is sufficient, and again the rotor analysis is identical to that used for the trim solution. The rigid-body equations of motion are numerically integrated for prescribed control or gust inputs to calculate a nonequilibrium flight path.

The flutter analysis involves the construction of a set of linear differential equations describing the motion of the rotor and the aircraft (all variables). The eigenvalues of these equations define the system stability. The equations may be time-invariant (for axial flow), or may have periodic coefficients (solved using Floquet theory). A constant coefficient approximation for the periodic coefficient equations, and various quasistatic reductions can be used (as implemented in CAMRAD/JA, neither is applicable for a two-bladed rotor).
1.2 Trim Solution

The structure of the solution of the trim task in CAMRAD/JA is outlined in Figures 1-2 and 1-3. The periodic motion in a steady-state, unaccelerated flight condition is required. The final converged solution, not intermediate transients, is desired. Hence following a strictly physical approach in the solution is not necessary. For efficiency and improved convergence, computationally intensive calculations are moved outside inner loops (if weak coupling allows this approach), and the major iteration loops are split into several levels.

The control required to achieve a specified flight condition is to be calculated (the inverse problem). Hence algebraic equations (for free flight obtained from equilibrium of forces and moments on the helicopter; for a wind tunnel case obtained by setting the thrust, tip-path-plane tilt, etc., equal to target values) are solved for the trim variables (rotor or pilot controls, and aircraft Euler angles). Differential equations are solved for the periodic rotor motion and airframe vibration.

The trim iteration is an outer loop (Figure 1-3). In CAMRAD/JA, the Newton-Raphson method (with a relaxation factor) is used to solve the algebraic equations. The periodic motion for fixed controls is calculated in an inner loop (Figure 1-3). In CAMRAD/JA a harmonic analysis method is used that is equivalent to an integration in time with a filter over the last revolution that forces the solution to be periodic. The analysis advances the rotor around the azimuth, calculating the forcing function in the time-domain and then updating the harmonics of the motion at each time step. The use of the frequency domain (a Fourier series representation) enforces periodicity, and allows the use of a large time step since numerical stability is separated from the physical stability of the system (which often has low-damped or high-frequency modes). There are separate circulation and motion iterations (Figure 1-3). In the circulation loop, the uniform or
uniform inflow stage

find trimmed solution

nonuniform inflow stage

calculate prescribed wake geometry
calculate wake influence coefficients
find trimmed solution
(repeat if wake geometry changes)

nonuniform inflow stage

calculate free wake geometry
calculate wake influence coefficients
find trimmed solution
(repeat if wake geometry changes)

Figure 1-2. Solution of trim task: inflow analysis levels
TRIMMED SOLUTION

iterate controls to trim

find self-tuning regulator solution

iterate regulator controls

solve for periodic motion and airloads

circulation iteration
  calculate induced velocity from loading

motion iteration
  solve for rotor motion

  azimuthal step
  radial integration
  blade section aerodynamics
  update motion harmonics
  calculate total rotor loads

solve for airframe motion

test motion convergence

test circulation convergence

test regulator convergence

test trim convergence

Figure 1-3. Solution of trim task: trim, regulator, circulation, and motion iterations.
nonuniform induced velocity is calculated from the circulation or aerodynamic loading; the motion is calculated for fixed induced velocity; the circulation is reevaluated; and the procedure is repeated until the circulation converges (a relaxation factor on the circulation is used to improve convergence). In the motion loop, there is an iteration between the calculation of the rotor motion and the airframe vibration, to avoid interharmonic coupling and to ensure proper filter of harmonics of the hub forces. A relaxation factor is also used in the motion loop to improve convergence. While the trim loop can be omitted (for a fixed controls solution), the circulation and motion iterations are always required.

The analysis includes a self-tuning regulator, which can automatically adjust controls selected from the rotor and aircraft primary controls, rotor higher harmonic controls, and higher harmonic auxiliary forces. The regulator forms a cost function (to be minimized) using selected rotor response quantities, including: flapping and power; airframe sensor vibratory response; hub and root loads; blade section loads; and rotor noise. For the trim analysis, an iteration to solve for the converged regulator state is introduced between the trim and circulation iterations.

The wake geometry and influence coefficient calculation are computationally expensive; they are therefore moved outside the trim iteration (Figure 1-2). The influence coefficients relate the induced velocity to the rotor blade bound circulation. This approach is possible because of the weak coupling of the influence coefficient calculation and the trim iteration, particularly when the rotor is trimmed to a specified thrust and tip-path-plane orientation. In CAMRAD/JA there are three levels of analysis: uniform inflow, nonuniform inflow with prescribed wake geometry, and nonuniform inflow with free wake geometry. Here "uniform inflow" refers to an empirical model based on momentum theory, and actually includes a linear variation of the inflow over the rotor disk. For accuracy, it is necessary to use the
bound circulation distribution from the nonuniform inflow calculation in
the free wake geometry analysis. For efficiency, the nonuniform inflow
calculation should originate from the trimmed uniform inflow solution.
The wake influence coefficients and geometry (prescribed or free) depend
on the rotor loading, so potentially an iteration between the influence
coefficient calculation and trim solution is necessary (Figure 1-2). In
practice, if the rotor is trimmed to a specified thrust and
tip-path-plane orientation at each level, the remaining influence of the
loading changes on the wake geometry is small, and hence iteration is
seldom necessary. It is most efficient to execute each of the three
levels once and only once to obtain a nonuniform, free wake solution.

It is possible to couple the CAMRAD/JA analysis with an external
solution for the rotor airloads, typically from a computationally
intensive CFD analysis. CAMRAD/JA can calculate a partial
angle-of-attack (excluding wake elements that are inside the CFD domain)
for use by the external analysis. By this means the effects of the
rotor wake and blade motion can be included in the CFD analysis. Then
CAMRAD/JA can read and use the externally calculated blade lift, drag,
and moment coefficients. Since this new rotor loading will change the
wake and motion solution, the process must generally be repeated. If
the physical coupling between CAMRAD/JA and external analyses is weak,
the iteration converges with a reasonably small number of executions of
the CFD analysis.

1.3 Configuration Model

CAMRAD/JA analyzes a general two-rotor aircraft. The
configurations considered are the single main-rotor and tail-rotor
helicopter; tandem main-rotor helicopter; coaxial main-rotor helicopter;
side-by-side main-rotor or tilting proprotor aircraft; and a rotor or
aircraft in a wind tunnel. Auxiliary forces acting on the airframe can
be included. Articulated, hingeless, gimbaled, and teetering rotors with an arbitrary number of blades can be analyzed.

1.4 Rotor Model

The rotor structural model is based on engineering beam theory for rotating wings with large pitch and twist. A single load path is assumed (multiple load path bearingless rotors can not be analyzed). The rotor blade is assumed to have a straight undeformed elastic axis, with specific root geometry possibilities. The blade motion considered includes in-plane and out-of-plane bending, torsion, control system flexibility, flap/lag/gimbal/teeter hinges, and rotor rotational speed. The rotor shaft motion and hub forces are considered. The blade pitch input includes higher harmonic control, from either the rotating or nonrotating frame.

The blade motion is described by rotating, free-vibration modes, equivalent to a Galerkin analysis. The blade mode shapes can be calculated internally or obtained from an external analysis. Nonlinear terms are retained in the equations of motion based on established knowledge of certain important nonlinear effects, and the requirement of consistency in the derivation. A vector formulation of the blade structural dynamics is used. The vector combination of in-plane and out-of-plane moments and deflections eliminates the dependence on the coordinate system, with a simplification of the equations as a consequence.

The rotor aerodynamic model is based on lifting-line theory, using steady two-dimensional airfoil characteristics and a vortex wake. The analysis includes empirical dynamic stall models; yawed-flow and swept-blade corrections; and unsteady aerodynamic forces from thin airfoil theory. The aerodynamic model is applicable to axial and nonaxial flight, with high inflow and large angles. The induced velocity is obtained from momentum theory or a vortex wake model. The
momentum theory model includes a mean term and terms that vary linearly over the rotor disk (produced by forward flight or hub moments); rotor/rotor and rotor/airframe interference; and ground effect.

For the flutter analysis, multiblade coordinates and an inflow dynamics model to represent low-frequency unsteady aerodynamics of the rotor can be used. In the inflow dynamics models, the uniform and linear induced-velocity components are related, by first-order differential equations, to the net aerodynamic thrust and hub moments on the rotor.

The rotor model is characterized by a section analysis, which follows from the assumption of high-aspect ratio: engineering beam theory for the structural model and lifting line theory for the aerodynamic model. The equations of motion are obtained from equilibrium of the inertial, aerodynamic, and elastic forces on the portion of blade outboard of a particular blade section. The interface between the aerodynamics and dynamics models is defined by the section aerodynamic forces and the section velocities.

1.5 Wake Model

The rotor wake model in CAMRAD/JA is usually based on a vortex lattice (straight-line segments) approximation for the wake. A small viscous core radius is used for the tip vortices. Vortex sheet elements can be used to represent the inboard wake, but usually it is sufficient (and more efficient) to approximate the sheets by line segments, with a large core radius to eliminate large velocities. Nonplanar, quadrilateral sheet elements are available if needed. The wake influence coefficients are calculated for incompressible flow. Rotor/rotor interference can be calculated (but only the mean velocities at the hub for the single main rotor and tail rotor case). The mean interference velocities at the airframe can be calculated.
The wake roll-up process is modeled. Eventually the tip vortex has the strength of the peak bound circulation at the azimuth where the wake element was trailed. The possibility of two bound circulation peaks, inboard and outboard peaks of opposite sign, is included in the rolled-up wake model. A number of prescribed parameters allow the tip vortex to have only a fraction of this peak strength when it encounters the following blade. The radial location of the tip vortex at the generating blade is also prescribed.

Blade-vortex interaction loading is calculated using either second-order lifting-line theory (three-quarter-chord collocation point), or using a lifting-surface theory correction. Simply using an artificially large vortex core size is a third possibility. A large core radius can be used for the velocity induced on the inboard part of the blade, in order to suppress the blade-vortex interactions there (as observed in experiment).

The wake geometry models in CAMRAD/JA include simple undistorted models; hover prescribed wake models based on experimental measurements; and a calculated free wake. The free wake analysis calculates the distorted tip vortex geometry for a single rotor in forward flight. This free-wake analysis is very efficient, and has modeling features that are consistent with the CAMRAD/JA wake model. The influence of aircraft turn rate on the undistorted wake convection is included.

1.6 Aircraft Model

The aircraft model in CAMRAD/JA allows for two rotors on a body having both rigid and elastic motion. A wind tunnel configuration (no rigid body motion) is also considered. The elastic airframe modes must be obtained from an external analysis (such as NASTRAN). CAMRAD/JA includes a drive train model, with the engine, governor, shaft flexibility, and rotor rotational speed degrees-of-freedom represented. The airframe auxiliary forces include trim, perturbation, and higher
harmonic terms. Airframe sensors available include accelerometers, airframe angular rate and motion measurements, and air velocity measurements. For the frequency-domain aeroservoelasticity analysis, a control system consisting of several scalar loops can be defined.

The airframe aerodynamic loads are a combination of nonlinear and linearized forces (only the nonlinear forces are considered for trim). The nonlinear terms are obtained from tables and simple equations. The interference velocities produced by the airframe at the rotor position can be calculated for a collection of wings (horseshoe vortices and doublet lines) and nonlifting bodies (ellipsoids and airfoil-shaped bodies of revolution). The interference velocities can also be obtained from an external analysis.
2. CODE CONSTRUCTION

2.1 Program Organization

The CAMRAD/JA computer code consists of three main programs, with the following functions:

(a) Input file preparation.

(b) Airfoil file preparation.

(c) Rotorcraft analysis.

The rotorcraft analysis can be run using namelist input of all variables (excluding tables), but is best to construct an input file. The airfoil tables must be always be put in a special format for use by the rotorcraft analysis.

The following pages list the subprograms that constitute these three programs, and state the primary function of each subprogram. Only the subprograms for rotor#1 are listed (final character of the subprogram name equal to "1"). The subprograms for rotor#2 (final character "2") have identical functions.

Rotorcraft Analysis Subprograms

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAMRADJA</td>
<td>PRIMARY JOB AND ANALYSIS CONTROL</td>
</tr>
<tr>
<td>AEROF1</td>
<td>CALCULATE BLADE AERODYNAMIC FORCES</td>
</tr>
<tr>
<td>AEROS1</td>
<td>CALCULATE BLADE SECTION AERODYNAMIC COEFFICIENTS</td>
</tr>
<tr>
<td>AEROT1</td>
<td>INTERPOLATE AIRFOIL TABLES</td>
</tr>
<tr>
<td>BDIEL</td>
<td>PERTURBATION VELOCITIES OF ELLIPSOID</td>
</tr>
<tr>
<td>BDIMS</td>
<td>PERTURBATION VELOCITIES OF BODY OF REVOLUTION</td>
</tr>
<tr>
<td>BDIMSP</td>
<td>CALCULATE MODIFIED SLENDER BODY THEORY PARAMETERS</td>
</tr>
<tr>
<td>BDISP</td>
<td>PERTURBATION VELOCITIES OF SPHERE</td>
</tr>
<tr>
<td>BESSEL</td>
<td>CALCULATE J BESSEL FUNCTION</td>
</tr>
<tr>
<td>BODYA</td>
<td>CALCULATE BODY AERODYNAMIC FORCES</td>
</tr>
</tbody>
</table>
BODYAT  CALCULATE BODY AERODYNAMIC COEFFICIENTS FROM TABLES
BODYC  INITIALIZE AIRFRAME PARAMETERS AT TRIM
BODYD  CALCULATE BODY LINEARIZED AERODYNAMIC FORCES
BODYF  CALCULATE AIRFRAME GENERALIZED FORCES
BODYFL CALCULATE AIRFRAME AERODYNAMIC FORCES FROM STABILITY DERIVATIVES
BODYFN CALCULATE AIRFRAME NONLINEAR AERODYNAMIC FORCES
BODYI1 CALCULATE AIRFRAME/ROTOR AERODYNAMIC INTERFERENCE
BODYIB CALCULATE BODY AERODYNAMIC INTERFERENCE VELOCITIES
BODYIW CALCULATE WING AERODYNAMIC INTERFERENCE VELOCITIES
BODYM1 CALCULATE AIRFRAME TRANSFER FUNCTION MATRIX
BODYS  CALCULATE AIRFRAME SENSOR MATRICES
BODYV1 CALCULATE HARMONICS OF AIRFRAME MOTION
CASEAN FREQUENCY DOMAIN AEROSERVOELASTICITY ANALYSIS
CHEKR1 CHECK FOR FATAL ERRORS
CONVC1 TEST CIRCULATION CONVERGENCE
CONVM1 TEST MOTION CONVERGENCE
CONVP  PRINT CIRCULATION AND MOTION CONVERGENCE
CSYSAN ANALYZE CONSTANT COEFFICIENT, LINEAR DIFFERENTIAL EQUATIONS
DERED  ELIMINATE EQUATIONS AND VARIABLES FROM DIFFERENTIAL EQUATIONS
DPLL  CALCULATE DIPOLE LINE SEGMENT INDUCED VELOCITY
EIGEN  CALCULATE EIGENVALUES AND EIGENVECTORS OF MATRIX
ENGNC INITIALIZE DRIVE TRAIN PARAMETERS AT TRIM
ENGNM1 CALCULATE DRIVE TRAIN TRANSFER FUNCTION MATRIX
ENGNV1 CALCULATE HARMONICS OF DRIVE TRAIN MOTION
FILEA1 READ OR WRITE AIRFOIL TABLE FILE
FILEB1 READ OR WRITE BLADE BENDING MODE FILE
FILEC  CLOSE FILE (VAX VERSION)
FILED  READ OR WRITE AIRFRAME STABILITY DERIVATIVE FILE
FILEF  READ OR WRITE AIRFRAME AERODYNAMIC COEFFICIENT FILE
FILEI1 READ OR WRITE INPUT FILE
FILEJ1 READ OR WRITE JOB SCRATCH FILE
FILEO  OPEN FILE (VAX VERSION)
FILEP  READ OR WRITE PLOT DATA FILE
FILETD CONSTRUCT TIME/DATA FILE IDENTIFICATION (VAX VERSION)
FILEV1 READ OR WRITE AIRFRAME INTERFERENCE VELOCITY FILE
FILEW1 READ OR WRITE CFD INTERFACE INPUT FILE
FILEX1 READ OR WRITE CFD INTERFACE OUTPUT FILE
FLUT  FLUTTER
FLUTA1 CALCULATE FLUTTER AERODYNAMIC COEFFICIENTS
FLUTB CALCULATE FLUTTER AIRCRAFT MATRICES
FLUTBC PERTURB VARIABLE FOR FLUTTER AIRCRAFT MATRICES
FLUTD1 CALCULATE INERTIAL TERMS IN FLUTTER ROTOR MATRICES
FLUTF1 CALCULATE AERODYNAMIC FORCES IN FLUTTER ROTOR MATRICES
FLUTI1 CALCULATE FLUTTER INERTIA COEFFICIENTS
FLUTL ANALYZE FLUTTER CONSTANT COEFFICIENT LINEAR EQUATIONS
FLUTM CALCULATE FLUTTER MATRICES
FLUTMB FLIGHT DYNAMICS MATRICES FOR FLUTTER
FLUTMC COMPLETION OF FLUTTER MATRICES
FLUTMD DYNAMIC INFLOW FOR FLUTTER MATRICES
FLUTME DRIVE TRAIN EQUATIONS FOR FLUTTER MATRICES
FLUTMI INITIALIZE CALCULATION OF FLUTTER MATRICES
FLUTMM CONSTRUCT COUPLED MATRICES FOR FLUTTER
FLUTMS REDUCE FLUTTER MATRICES TO FINAL EQUATIONS
FLUTR1 CALCULATE FLUTTER ROTOR MATRICES
GEOME1 EVALUATE WAKE GEOMETRY
GEOMF1 CALCULATE FREE WAKE GEOMETRY DISTORTION
GEOMFS SCULLY FREE WAKE GEOMETRY CALCULATION
GEOMP1 PRINTER- PLOT OF WAKE GEOMETRY
GEOMR1 CALCULATE WAKE GEOMETRY DISTORTION
GEOMX CALCULATE VORTEX LINE SEGMENT GEOMETRY INSIDE CFD COMPUTATION DOMAIN
HISTPP PRINTER- PLOT OF AZIMUTHAL TIME HISTORY
INIT  INITIALIZATION
INITA CALCULATE AERODYNAMIC ENVIRONMENT PARAMETERS
INITB  INITIALIZE AIRFRAME PARAMETERS
INITC  INITIALIZE CASE PARAMETERS
INITE  INITIALIZE DRIVE TRAIN PARAMETERS
INITR1 INITIALIZE ROTOR PARAMETERS
INITS  INITIALIZE SELF-TUNING REGULATOR PARAMETERS
INITV  INITIALIZE VARIABLES AT END OF CASE
INPT  INPUT FOR CASE
INPTB  BODY NAMELIST INPUT
INPTC  CASE NAMELIST INPUT
INPTF  FLUTTER NAMELIST INPUT
INPTJ  INPUT FOR JOB
INPTL1 LOADS NAMELIST INPUT
INPTN  READ OR WRITE NAMELIST INPUT
INPTR1 ROTOR NAMELIST INPUT
INPTS  FLIGHT DYNAMICS NAMELIST INPUT
INPTT  TRANSIENT NAMELIST INPUT
INPTW1 WAKE NAMELIST INPUT
INRTC1 CALCULATE BLADE INERTIA COEFFICIENTS
INRTI1 CALCULATE INVERSE OF TRANSFER FUNCTION MATRIX
INRTM1 CALCULATE ROTOR TRANSFER FUNCTION MATRIX
INTERP LINEARLY INTERPOLATE TWO-DIMENSIONAL TABLE
LDAMP1 NONLINEAR LAG DAMPER MOMENT
LDHD1 CALCULATE DIMENSIONAL HUB AND CONTROL LOADS
LDHX1 CALCULATE DIMENSIONLESS HUB AND CONTROL LOADS
LDRM1 CALCULATE AND STORE ROTOR MOTION FOR LOADS CALCULATION
LDSD1 CALCULATE DIMENSIONAL BLADE SECTION LOADS
LDSI1 CALCULATE INERTIA COEFFICIENTS FOR SECTION LOADS
LDSX1 CALCULATE DIMENSIONLESS BLADE SECTION LOADS
LDVS  CALCULATE AIRFRAME SENSOR VIBRATORY RESPONSE
LOAD  LOADS, VIBRATION, AND NOISE
LOADA1 CALCULATE AND PRINT ROTOR AERODYNAMIC LOADS (FUNCTION R AND PSI)
LOADAM  CALCULATE RADIAL AND AZIMUTHAL AVERAGES
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOADAP</td>
<td>PRINT AND WRITE TO PLOT FILE MOTION AND AERODYNAMICS</td>
</tr>
<tr>
<td>LOADD1</td>
<td>CALCULATE AND PRINT ROTOR MOTION (FUNCTION OF PSI)</td>
</tr>
<tr>
<td>LOADH</td>
<td>CALCULATE MEAN AND HALF PEAK TO PEAK</td>
</tr>
<tr>
<td>LOADH1</td>
<td>PRINT AND PLOT HUB AND CONTROL LOADS</td>
</tr>
<tr>
<td>LOADHP</td>
<td>PRINT OR WRITE TO PLOT FILE HUB AND BLADE LOADS</td>
</tr>
<tr>
<td>LOADM</td>
<td>CALCULATE MEAN AND HALF PEAK TO PEAK</td>
</tr>
<tr>
<td>LOADR1</td>
<td>ROTOR LOADS AND NOISE</td>
</tr>
<tr>
<td>LOADS1</td>
<td>PRINT AND PLOT BLADE SECTION LOADS</td>
</tr>
<tr>
<td>LOADV</td>
<td>PRINT AND PLOT AIRFRAME VIBRATION</td>
</tr>
<tr>
<td>MINV</td>
<td>CALCULATE INVERSE OF MATRIX</td>
</tr>
<tr>
<td>MINVC</td>
<td>CALCULATE INVERSE OF COMPLEX MATRIX</td>
</tr>
<tr>
<td>MODE1</td>
<td>BLADE MODES</td>
</tr>
<tr>
<td>MODEA1</td>
<td>CALCULATE ARTICULATED BLADE FLAP AND LAG MODES</td>
</tr>
<tr>
<td>MODEB1</td>
<td>CALCULATE BLADE BENDING MODES</td>
</tr>
<tr>
<td>MODEC1</td>
<td>INITIALIZE BLADE MODE PARAMETERS</td>
</tr>
<tr>
<td>MODED1</td>
<td>CALCULATE BLADE ROOT GEOMETRY</td>
</tr>
<tr>
<td>MODEG</td>
<td>CALCULATE GALERKIN FUNCTIONS FOR BENDING MODES</td>
</tr>
<tr>
<td>MODEK1</td>
<td>CALCULATE KINEMATIC PITCH-BENDING COUPLING</td>
</tr>
<tr>
<td>MODEP1</td>
<td>PRINT BLADE MODES</td>
</tr>
<tr>
<td>MODER1</td>
<td>RADIAL STATIONS FOR BENDING AND TORSION MODES</td>
</tr>
<tr>
<td>MODET1</td>
<td>CALCULATE BLADE TORSION MODES</td>
</tr>
<tr>
<td>MOTNB1</td>
<td>CALCULATE BLADE AND HUB MOTION</td>
</tr>
<tr>
<td>MOTNC1</td>
<td>INITIALIZE ROTOR PARAMETERS AT TRIM</td>
</tr>
<tr>
<td>MOTNF1</td>
<td>CALCULATE ROTOR GENERALIZED FORCES</td>
</tr>
<tr>
<td>MOTNH1</td>
<td>CALCULATE HARMONICS OF HUB MOTION</td>
</tr>
<tr>
<td>MOTNR1</td>
<td>CALCULATE HARMONICS OF ROTOR MOTION</td>
</tr>
<tr>
<td>MOTNS</td>
<td>CALCULATE STATIC ELASTIC MOTION</td>
</tr>
<tr>
<td>NOISF1</td>
<td>CALCULATE FAR FIELD ROTATIONAL NOISE</td>
</tr>
<tr>
<td>NOISR1</td>
<td>PRINT AND PLOT FAR FIELD ROTATIONAL NOISE</td>
</tr>
<tr>
<td>NOIST</td>
<td>CALCULATE NOISE TIME HISTORY AND SPECTRUM</td>
</tr>
<tr>
<td>PERF</td>
<td>PERFORMANCE</td>
</tr>
<tr>
<td>PERFA</td>
<td>CALCULATE AND PRINT AIRFRAME PERFORMANCE</td>
</tr>
</tbody>
</table>
PERFF1  CALCULATE AND PRINT ROTOR FORCES AND POWER
PERFM1  CALCULATE AND PRINT ROTOR AND AIRFRAME MOTION
PERFO   CALCULATE AND PRINT OPERATING CONDITION
PERFR   ROTOR PERFORMANCE
PERFR1  CALCULATE AND PRINT ROTOR PERFORMANCE
PERFS   CALCULATE AND PRINT SELF-TUNING REGULATOR PERFORMANCE
POLRPP  PRINTER- PLOT OF POLAR PLOT
PRNTA   PRINT AEROSERVOELASTICITY INPUT DATA
PRNTB   PRINT BODY INPUT DATA
PRNTC   PRINT TRIM INPUT DATA
PRNTF   PRINT FLUTTER INPUT DATA
PRNTG   PRINT TRANSIENT GUST AND CONTROL INPUT DATA
PRNTH   PRINT CASE HEADER
PRNTHF  PRINT FLUTTER HEADER
PRNTHS  PRINT FLIGHT DYNAMICS HEADER
PRNTHT  PRINT TRANSIENT HEADER
PRNTI   PRINT INPUT DATA
PRNTJ   PRINT JOB INPUT DATA
PRNTL1  PRINT ROTOR LOADS INPUT DATA
PRNTLA  PRINT AIRFRAME LOADS INPUT DATA
PRNTR1  PRINT ROTOR INPUT DATA
PRNTS   PRINT FLIGHT DYNAMICS INPUT DATA
PRNTT   PRINT TRANSIENT INPUT DATA
PRNTW1  PRINT WAKE INPUT DATA
PSYSAN  ANALYZE PERIODIC COEFFICIENT, LINEAR DIFFERENTIAL EQUATIONS
QSTRAIN QUASISTATIC REDUCTION OF LINEAR DIFFERENTIAL EQUATIONS
RAMF    CALCULATE ROTOR/AIRFRAME PERIODIC MOTION AND FORCES
SPANPP  PRINTER- PLOT OF SPANWISE PLOT
STAB    FLIGHT DYNAMICS
STABD   PRINT STABILITY DERIVATIVES
STABDV  TRANSFORM AXES FOR STABILITY DERIVATIVE
STABE   CALCULATE FLIGHT DYNAMICS EQUATIONS
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STABL</td>
<td>ANALYZE FLIGHT DYNAMICS LINEAR EQUATIONS</td>
</tr>
<tr>
<td>STABLA</td>
<td>ACCELERATION FOR NUMERICAL INTEGRATION OF TRANSIENT</td>
</tr>
<tr>
<td>STABL1</td>
<td>FLIGHT DYNAMICS NUMERICAL INTEGRATION OF TRANSIENT</td>
</tr>
<tr>
<td>STABLS</td>
<td>CALCULATE RESPONSE FOR NUMERICAL INTEGRATION OF TRANSIENT</td>
</tr>
<tr>
<td>STABM</td>
<td>CALCULATE FLIGHT DYNAMICS STABILITY DERIVATIVES AND MATRICES</td>
</tr>
<tr>
<td>STABMC</td>
<td>PERTURB VARIABLES FOR STABILITY DERIVATIVE CALCULATION</td>
</tr>
<tr>
<td>STABMII</td>
<td>CONSTRUCT FLIGHT DYNAMICS MATRICES</td>
</tr>
<tr>
<td>STABMPI</td>
<td>PRINT DURING STABILITY DERIVATIVE CALCULATION</td>
</tr>
<tr>
<td>STABP</td>
<td>PRINT FLIGHT DYNAMICS TRANSIENT SOLUTION</td>
</tr>
<tr>
<td>STABPI</td>
<td>PRINT FLIGHT DYNAMICS TRANSIENT HEADER</td>
</tr>
<tr>
<td>STRI</td>
<td>SELF-TUNING REGULATOR ITERATIVE SOLUTION</td>
</tr>
<tr>
<td>STRIC</td>
<td>CALCULATE GAIN MATRIX FOR SELF-TUNING REGULATOR</td>
</tr>
<tr>
<td>STRIJ</td>
<td>CALCULATE COST FUNCTION FOR SELF-TUNING REGULATOR</td>
</tr>
<tr>
<td>STRIT</td>
<td>CALCULATE CONTROLS FOR SELF-TUNING REGULATOR</td>
</tr>
<tr>
<td>STRIZ</td>
<td>CALCULATE OUTPUT FOR SELF-TUNING REGULATOR</td>
</tr>
<tr>
<td>TIMEC</td>
<td>CPU TIME (VAX VERSION)</td>
</tr>
<tr>
<td>TIMER</td>
<td>PROGRAM TIMER</td>
</tr>
<tr>
<td>TRAN</td>
<td>TRANSIENT</td>
</tr>
<tr>
<td>TRANC</td>
<td>CALCULATE TRANSIENT GUST OR CONTROL</td>
</tr>
<tr>
<td>TRANCC</td>
<td>CALCULATE TRANSIENT CONTROL</td>
</tr>
<tr>
<td>TRANCG</td>
<td>CALCULATE TRANSIENT GUST</td>
</tr>
<tr>
<td>TRANCT</td>
<td>CALCULATE TIME HISTORY OF TRANSIENT CONTROL/GUST</td>
</tr>
<tr>
<td>TRANI</td>
<td>CALCULATE TRANSIENT ACCELERATION FOR NUMERICAL INTEGRATION</td>
</tr>
<tr>
<td>TRANP</td>
<td>PRINT TRANSIENT SOLUTION</td>
</tr>
<tr>
<td>TRANPI</td>
<td>PRINT TRANSIENT HEADER AND OUTPUT DEFINITION</td>
</tr>
<tr>
<td>TRANS</td>
<td>CALCULATE RESPONSE FOR TRANSIENT ANALYSIS</td>
</tr>
<tr>
<td>TRCKPP</td>
<td>PRINTER- PLOT TIME HISTORY</td>
</tr>
<tr>
<td>TRIM</td>
<td>TRIM</td>
</tr>
<tr>
<td>TRIMI</td>
<td>CALCULATE TRIM SOLUTION BY ITERATION</td>
</tr>
<tr>
<td>TRIMIC</td>
<td>INCREMENT CONTROL FOR TRIM SOLUTION</td>
</tr>
<tr>
<td>TRIMIM</td>
<td>DEFINE CONTROLS AND TESTS FOR TRIM OPTIONS</td>
</tr>
<tr>
<td>TRIMIT</td>
<td>CALCULATE FORCES FOR TRIM SOLUTION</td>
</tr>
</tbody>
</table>
TRIMIV  CALCULATE AIRCRAFT CONTROLS FROM PILOT'S CONTROLS
TRIMP  PRINT TRIM SOLUTION
TRIMW  TRIM AND WAKE ITERATION
VTXL  CALCULATE VORTEX LINE SEGMENT INDUCED VELOCITY
VTXS  CALCULATE VORTEX SHEET SEGMENT INDUCED VELOCITY
WAKEB1  CALCULATE BLADE POSITION
WAKEC1  CALCULATE INFLUENCE COEFFICIENTS FOR NONUNIFORM INFLOW
WAKEN1  CALCULATE NON-UNIFORM WAKE INDUCED VELOCITY
WAKEU1  CALCULATE UNIFORM WAKE-INDUCED VELOCITY
WAKEX1  CALCULATE EFFECTIVE ANGLE OF ATTACK FOR CFD INTERFACE
WKPA1  INFLUENCE COEFFICIENTS FOR AXISYMMETRIC FAR WAKE
WKPFW1  INFLUENCE COEFFICIENTS FOR FAR WAKE ELEMENT (RW/FW/DW)
WKPNW1  INFLUENCE COEFFICIENTS FOR NEAR WAKE ELEMENT (NW)
WKPRU1  INFLUENCE COEFFICIENTS FOR ROLLING UP WAKE ELEMENT (RW/RU)

Input File Preparation Subprograms

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>INPUT FILE PREPARATION</td>
</tr>
<tr>
<td>INPTJI</td>
<td>READ JOB NAMELIST FOR INPUT PROGRAM</td>
</tr>
</tbody>
</table>

Airfoil File Preparation Subprograms

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIRFOIL</td>
<td>AIRFOIL TABLE PREPARATION</td>
</tr>
<tr>
<td>AFTBPT</td>
<td>READ AND PRINT AIRFOIL TABLE PARAMETERS</td>
</tr>
<tr>
<td>AFTBIT</td>
<td>NAMELIST READ OF AIRFOIL TABLE PARAMETERS</td>
</tr>
<tr>
<td>AFTBCT</td>
<td>CONSTRUCT NEW AIRFOIL TABLE</td>
</tr>
<tr>
<td>AFTBPC</td>
<td>READ AND PRINT AIRFOIL EQUATION PARAMETERS</td>
</tr>
<tr>
<td>AFTBIC</td>
<td>NAMELIST READ OF AIRFOIL EQUATION PARAMETERS</td>
</tr>
<tr>
<td>AFTBOC</td>
<td>PRINT OF AIRFOIL EQUATION PARAMETERS</td>
</tr>
<tr>
<td>AFTBEQ</td>
<td>SECTION AERODYNAMIC CHARACTERISTICS FROM EQUATIONS</td>
</tr>
</tbody>
</table>
AFTBRD READ C81 FORMAT AIRFOIL FILE
AFTBIN INTERPOLATE C81 FORMAT AIRFOIL FILE
AFTBPR PRINT AIRFOIL TABLE DATA
AFTBPP PRINTER- PLOT AIRFOIL AERODYNAMIC CHARACTERISTICS

The list below gives the labels of the common blocks used by the CAMRAD/JA computer code, and states the type of data contained in each. Only the common blocks for rotor#1 are listed ("1" somewhere in the name); the common blocks for rotor#2 ("2" in the name) have identical functions. A complete description of all the variables in these common blocks is provided by the dictionary contents listing.

**CAMRAD/JA Common Blocks**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMDATA</td>
<td>Input trim data</td>
</tr>
<tr>
<td>SRDATA</td>
<td>Input self-tuning regulator data</td>
</tr>
<tr>
<td>R1DATA</td>
<td>Input rotor data</td>
</tr>
<tr>
<td>W1DATA</td>
<td>Input wake data</td>
</tr>
<tr>
<td>G1DATA</td>
<td>Input free wake geometry data</td>
</tr>
<tr>
<td>BDDATA</td>
<td>Input airframe data</td>
</tr>
<tr>
<td>BADATA</td>
<td>Input airframe aerodynamics data</td>
</tr>
<tr>
<td>ENDATA</td>
<td>Input drive train data</td>
</tr>
<tr>
<td>L1DATA</td>
<td>Input rotor loads data</td>
</tr>
<tr>
<td>LADATA</td>
<td>Input airframe loads data</td>
</tr>
<tr>
<td>GCDATA</td>
<td>Input gust and control data</td>
</tr>
<tr>
<td>TNDATA</td>
<td>Input transient data</td>
</tr>
<tr>
<td>STDATA</td>
<td>Input flight dynamics data</td>
</tr>
<tr>
<td>FLDATA</td>
<td>Input flutter data</td>
</tr>
<tr>
<td>HCDATA</td>
<td>Input control system data</td>
</tr>
<tr>
<td>A1TABL</td>
<td>Rotor airfoil tables</td>
</tr>
<tr>
<td>BATAVL</td>
<td>Airframe aerodynamic coefficient table</td>
</tr>
</tbody>
</table>

21
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDTABL</td>
<td>Airframe stability derivative table</td>
</tr>
<tr>
<td>UNITNO</td>
<td>Input/output unit numbers</td>
</tr>
<tr>
<td>CASECM</td>
<td>Job description</td>
</tr>
<tr>
<td>TIMECM</td>
<td>Timer statistics</td>
</tr>
<tr>
<td>TRIMCM</td>
<td>Calculated trim data</td>
</tr>
<tr>
<td>STRCM</td>
<td>Self-tuning regulator data</td>
</tr>
<tr>
<td>RTR1CM</td>
<td>Calculated rotor data</td>
</tr>
<tr>
<td>RH1CM</td>
<td>Transfer function matrices</td>
</tr>
<tr>
<td>BODYCM</td>
<td>Calculated airframe data</td>
</tr>
<tr>
<td>ENGNM</td>
<td>Calculated drive train data</td>
</tr>
<tr>
<td>GUSTCM</td>
<td>Gust and transient control</td>
</tr>
<tr>
<td>CONTCM</td>
<td>Aircraft controls and motion</td>
</tr>
<tr>
<td>CONVCM</td>
<td>Circulation and motion convergence</td>
</tr>
<tr>
<td>MD1CM</td>
<td>Blade modes</td>
</tr>
<tr>
<td>INC1CM</td>
<td>Rotor inertial coefficients</td>
</tr>
<tr>
<td>WKV1CM</td>
<td>Induced velocity</td>
</tr>
<tr>
<td>MNH1CM</td>
<td>Hub motion</td>
</tr>
<tr>
<td>AES1CM</td>
<td>Blade section aerodynamics</td>
</tr>
<tr>
<td>MNR1CM</td>
<td>Rotor motion and airframe vibration</td>
</tr>
<tr>
<td>MNSCM</td>
<td>Static elastic motion</td>
</tr>
<tr>
<td>AEF1CM</td>
<td>Rotor forces</td>
</tr>
<tr>
<td>QR1CM</td>
<td>Rotor generalized forces</td>
</tr>
<tr>
<td>QBDCM</td>
<td>Airframe generalized forces</td>
</tr>
<tr>
<td>WGC1CM</td>
<td>Wake geometry</td>
</tr>
<tr>
<td>WKCLCM</td>
<td>Wake influence coefficients</td>
</tr>
<tr>
<td>AEMNRM</td>
<td>Calculated motion data</td>
</tr>
<tr>
<td>LDMNCM</td>
<td>Calculated loads data</td>
</tr>
<tr>
<td>FLMCM</td>
<td>Flutter matrices</td>
</tr>
<tr>
<td>FLM1CM</td>
<td>Flutter rotor matrices</td>
</tr>
<tr>
<td>FMLACM</td>
<td>Flutter airframe matrices</td>
</tr>
<tr>
<td>FLINCM</td>
<td>Flutter inertial coefficients</td>
</tr>
<tr>
<td>FLAECM</td>
<td>Flutter aerodynamic coefficients</td>
</tr>
</tbody>
</table>
STDCM  Flight dynamics stability derivatives
STMCN  Flight dynamics matrices
TRANCM  Calculated transient data
CSYSCM  Scratch matrices for linear equations analysis
HASECM  Parameters for aeroservoelasticity analysis

2.2 Program Files

This section describes the files used by the CAMRAD/JA programs. The system-specific routines to open and close the files are described in section 7. The program identifies files by logical names (although these may not be needed by some machines). For scratch files (to be deleted at the end of a run) the actual file name is given. File unit numbers are always input variables, with unique default values. Generally the use of a particular file is controlled by various input parameters, depending on the analysis functions.

A. Rotorcraft Analysis

A1) File logical name: ---
Function: job input
Unit number variable: NUIN (namelist NLCASE)
Unit number default value: 5
Subroutine using file: any (no open/close)
Parameter controlling use: ---

A2) File logical name: ---
Function: job output
Unit number variable: NUOUT (namelist NLCASE)
Unit number default value: 6
Subroutine using file: any (no open/close)
Parameter controlling use: ---

A3) File logical name: ---
Function: job debug output
Unit number variable: NUDB (namelist NLCASE)
Unit number default value: 6
Subroutine using file: any (no open/close)
Parameter controlling use: ---
<table>
<thead>
<tr>
<th>File Logical Name</th>
<th>Function</th>
<th>Unit Number Variable</th>
<th>Unit Number Default Value</th>
<th>Subroutine Using File</th>
<th>Parameter Controlling Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4) INPUTFILE</td>
<td>input data</td>
<td>NFDAT (name list NLCA)</td>
<td>40</td>
<td>INPT (through FILEI)</td>
<td>INFILE (name list NLCA)</td>
</tr>
<tr>
<td>A5) AFTABLE1</td>
<td>rotor#1 airfoil table</td>
<td>NFAFL (name list NLCA)</td>
<td>41</td>
<td>INPT (through FILEA1)</td>
<td>AFFILE (name list NLCA)</td>
</tr>
<tr>
<td>A6) AFTABLE2</td>
<td>rotor#2 airfoil table</td>
<td>NFAFL2 (name list NLCA)</td>
<td>42</td>
<td>INPT (through FILEA2)</td>
<td>AFFILE (name list NLCA), NROTOR</td>
</tr>
<tr>
<td>A7) PLOTFILE</td>
<td>plot data</td>
<td>NFPLT (name list NLCA)</td>
<td>43</td>
<td>any (through FILEP)</td>
<td>PLFILE (name list NLCA)</td>
</tr>
<tr>
<td>A8) BENDMODEL1</td>
<td>rotor#1 bending modes</td>
<td>NFBND1 (name list NLCA)</td>
<td>61</td>
<td>INPT (through FILEB1)</td>
<td>HINGE (name list NLTR)</td>
</tr>
<tr>
<td>A9) BENDMODEL2</td>
<td>rotor#2 bending modes</td>
<td>NFBND2 (name list NLCA)</td>
<td>62</td>
<td>INPT (through FILEB2)</td>
<td>HINGE (name list NLTR)</td>
</tr>
<tr>
<td>A10) AEROUNT</td>
<td>rotor#1 body interference vel</td>
<td>NFINT1 (name list NLCA)</td>
<td>63</td>
<td>INPT (through FILEI)</td>
<td>OPINTV (name list NLBDJ)</td>
</tr>
<tr>
<td>A11</td>
<td>File logical name:</td>
<td>Function:</td>
<td>Unit number variable:</td>
<td>Unit number default value:</td>
<td>Subroutine using file:</td>
</tr>
<tr>
<td>-----</td>
<td>-------------------</td>
<td>-----------</td>
<td>-----------------------</td>
<td>---------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>AEROINT2</td>
<td>rotor#2 body interference vel</td>
<td>NFINT2 (namelist NLCASE)</td>
<td>64</td>
<td>INPT (through FILEV2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A12</th>
<th>File logical name:</th>
<th>Function:</th>
<th>Unit number variable:</th>
<th>Unit number default value:</th>
<th>Subroutine using file:</th>
<th>Parameter controlling use:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BODYAERO</td>
<td>airframe aero coefficients</td>
<td>NFBAT (namelist NLCASE)</td>
<td>65</td>
<td>INPT (through FILEF)</td>
<td>OPBAT (namelist NLBODY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A13</th>
<th>File logical name:</th>
<th>Function:</th>
<th>Unit number variable:</th>
<th>Unit number default value:</th>
<th>Subroutine using file:</th>
<th>Parameter controlling use:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STABDERIV</td>
<td>airframe stability derivatives</td>
<td>NFDRV (namelist NLCASE)</td>
<td>66</td>
<td>INPT (through FILED)</td>
<td>OPDRV (namelist NLBODY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A14</th>
<th>File logical name:</th>
<th>Function:</th>
<th>Unit number variable:</th>
<th>Unit number default value:</th>
<th>Subroutine using file:</th>
<th>Parameter controlling use:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CFDINPUT1</td>
<td>rotor#1 CFD interface input</td>
<td>NFCI1 (namelist NLCASE)</td>
<td>71</td>
<td>INPT (through FILEW1)</td>
<td>OPCFD (namelist NLRTR)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A15</th>
<th>File logical name:</th>
<th>Function:</th>
<th>Unit number variable:</th>
<th>Unit number default value:</th>
<th>Subroutine using file:</th>
<th>Parameter controlling use:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CFDINPUT2</td>
<td>rotor#2 CFD interface input</td>
<td>NFCI2 (namelist NLCASE)</td>
<td>72</td>
<td>INPT (through FILEW2)</td>
<td>OPCFD (namelist NLRTR)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A16</th>
<th>File logical name:</th>
<th>Function:</th>
<th>Unit number variable:</th>
<th>Unit number default value:</th>
<th>Subroutine using file:</th>
<th>Parameter controlling use:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CFDOUTPUT1</td>
<td>rotor#1 CFD interface output</td>
<td>NFCO1 (namelist NLCASE)</td>
<td>73</td>
<td>TRIM (through FILEX1)</td>
<td>OPCFD (namelist NLRTR)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A17</th>
<th>File logical name:</th>
<th>Function:</th>
<th>Unit number variable:</th>
<th>Unit number default value:</th>
<th>Subroutine using file:</th>
<th>Parameter controlling use:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CFDOUTPUT2</td>
<td>rotor#2 CFD interface output</td>
<td>NFCO2 (namelist NLCASE)</td>
<td>74</td>
<td>TRIM (through FILEX2)</td>
<td>OPCFD (namelist NLRTR)</td>
</tr>
</tbody>
</table>
### A18) File logical name:
- **Function:** job scratch file
- **Unit number variable:** NFSCRJ (namelist NLCASE)
- **Unit number default value:** 50
- **Subroutine using file:** CAMRADJA (through FILEJ)
- **Parameter controlling use:** ANTYPE (namelist NLTRIM)

### A19) File logical name:
- **Function:** linear systems scratch file
- **Unit number variable:** NFSCRL (namelist NLCASE)
- **Unit number default value:** 51
- **Subroutine using file:** FLUTMB, FLUTMS
- **Parameter controlling use:** OPFLOW, DOF (namelist NLFLUT); OPSYMM (namelist NLFLUT)

### A20) File logical name:
- **Function:** linear systems scratch file
- **Unit number variable:** NFSCRF (namelist NLCASE)
- **Unit number default value:** 52
- **Subroutine using file:** FLUTMS
- **Parameter controlling use:** OPSYMM (namelist NLFLUT)

### B. Input File Preparation

#### B1) File logical name:
- **Function:** job input
- **Unit number variable:** NUIN (namelist NLJOB)
- **Unit number default value:** 5
- **Subroutine using file:** any (no open/close)
- **Parameter controlling use:** ---

#### B2) File logical name:
- **Function:** job output
- **Unit number variable:** NUOUT (namelist NLJOB)
- **Unit number default value:** 6
- **Subroutine using file:** any (no open/close)
- **Parameter controlling use:** ---

#### B3) File logical name:
- **Function:** input data file
- **Unit number variable:** NFDATI (namelist NLJOB)
- **Unit number default value:** 40
- **Subroutine using file:** INPUT (through FILEI)
- **Parameter controlling use:** NFILEI (namelist NLJOB)
### B4) File logical name: INPUTLISTn (n = 1 to 9)
- **Function:** input namelist data
- **Unit number variable:** NFNLI0 (namelist NLJ0B)
- **Unit number default value:** 40 (NFNLI = NFNLI0+n = 41 to 49)
- **Subroutine using file:** INPUT (through INPTN)
- **Parameter controlling use:** NLISTI (namelist NLJ0B)

### B5) File logical name: OUTPUTFILE
- **Function:** output data file
- **Unit number variable:** NFDATO (namelist NLJ0B)
- **Unit number default value:** 50
- **Subroutine using file:** INPUT (through FILEI)
- **Parameter controlling use:** NFILEO (namelist NLJ0B)

### B6) File logical name: OUTPUTLIST
- **Function:** output namelist data
- **Unit number variable:** NFNLO (namelist NLJ0B)
- **Unit number default value:** 51
- **Subroutine using file:** INPUT (through INPTN)
- **Parameter controlling use:** NLISTO (namelist NLJ0B)

### C. Airfoil File Preparation

#### C1) File logical name: ---
- **Function:** job input
- **Unit number variable:** NUIN (namelist NLTABL)
- **Unit number default value:** 5
- **Subroutine using file:** any (no open/close)
- **Parameter controlling use:** ---

#### C2) File logical name: ---
- **Function:** job output
- **Unit number variable:** NUOUT (namelist NLTABL)
- **Unit number default value:** 6
- **Subroutine using file:** any (no open/close)
- **Parameter controlling use:** ---

#### C3) File logical name: AFTABLE
- **Function:** airfoil file
- **Unit number variable:** NFILEO (namelist NLTABL)
- **Unit number default value:** 40
- **Subroutine using file:** AFTBPT, AFTBCT (through FILEA1)
- **Parameter controlling use:** OPREAD (namelist NLTABL)
2.3 Program Skeleton

The following pages present a schematic of the three CAMRAD/JA programs, showing the basic flow of control and the major iterations and options. The schematic also defines the input and output structure of the programs. The appearance of a subprogram name (in capital letters) means that the subprogram is called at that point in the analysis. The contents of a subprogram are shown by means of a three-sided box appearing below the subprogram name. Refer to section 2.1 for a list of subprogram names and principal functions.
call INPTJI (read namelist NLJOB, job input parameters)

if NFILEI ne 0, then
    call FILEI (read CAMRAD/JA input file)

if NLISTI ne 0, then
    for I = 1 to NLISTI (namelist files)
        call FILEO
        read namelist NLREAD (file parameters)
        call INPTN (read CAMRAD/JA namelists)
        call FILEC
    endfor

job namelist input
    read namelist NLREAD
    call INPTN (read CAMRAD/JA namelists)

if NFILEO ne 0, then
    call FILEI (write CAMRAD/JA input file)

if NLISTO ne 0, then
    call FILETD (for identification FILEID)
    call FILEO
    write namelist NLREAD (file parameters)
    call INPTN (write CAMRAD/JA namelists)
    call FILEC

if NPRINT ne 0, then
    call PRNTI (print input parameters)
AIRFOIL

call AFTBPT

  call AFTBIT (read namelist NLTABL)

  if OREAD = 0, then
    call FILEA1 (read airfoil file)
  if OREAD = 0 and AFTOP = 1, then
    call AFTBOC

call AFTBCT

  if OREAD ne 0, then construct new table
    for JR = 1 to NRB (radial segments)
      if OREAD ne 2, then
        call AFTBPC

          call AFTBIC (read namelist NLCHAR)
          call AFTBOC

        if OREAD = 2, then
          call AFTBRE

            call FILEO
            read C81 airfoil file
            call FILEC

            angle of attack iteration
            Mach number iteration
              if OREAD ne 2, then
                call AFTBEQ (equations)
              if OREAD = 2, then
                call AFTBIN (C81 airfoil file)

          call INTERP

    call FILEA1 (write airfoil file)

call AFTBPR

  if OPRRT(1) ne 0, then interpolate and print airfoil data
    call AEROT1
  if OPRRT(2) ne 0, then interpolate and plot airfoil data
    call AEROT1
    call AFTBFP
  if OPRRT(3) ne 0, then list table
read and print job input data
   call INPTJ (read namelist NLCASE)
call PRNTJ
   call FILEP (open plot file)

for JCASE = 1 to NCASES
   call TIMER (initialize)
call TIMER
   call FILETD (for job identification IDENT)

   job input and initialization
      call INPT
      call INIT

   print header and input parameters
      call PRNTH
      call PRNTI
      call FILEP (plot file header)

   trim
      call TRIM
      call FILEJ (write trim data scratch file)

   flutter
      call FLUT
      call FILEJ (read trim data scratch file)

   flight dynamics
      call STAB
      call FILEJ (read trim data scratch file)

   transient
      call TRAN

   call FILEP (close plot file)
call INITV
call TIMER
call TIMER (print)
if (INFILE ge 2) or (INFILE = 1 and first case), then
   call FILEI (read input file)

call INPTN

if first case, then
   if AFFILE = 1 or ge 3, then
      call FILEA1 (read airfoil file for rotor#1)
      if AFFILE ge 2 and NROTOR gt 1, then
         if AFFILE = 2 or 3, then
            call FILEA2 (read airfoil file for rotor#2)
         if AFFILE ge 4, then
            call FILEA2 (read rotor#1 file for rotor#2)

   if HINGE = -1 (rotor#1), then
      call FILEB1 (read bending mode file for rotor#1)
   if HINGE = -1 (rotor#2) and NROTOR gt 1, then
      call FILEB2 (read bending mode file for rotor#2)

   if OPINTV(1) = 1 (rotor#1), then
      call FILEV1 (read airframe interference velocity file)
   if OPINTV(2) = 1 (rotor#2) and NROTOR gt 1, then
      call FILEV2 (read airframe interference velocity file)

   if OPBAT ne 0 then
      call FILEF (read airframe aero coefficient file)

   if OPDRV ge 2 then
      call FILED (read airframe stability derivative file)

   if OPCFD ge 2 (rotor#1), then
      call FILEW1 (read CFD interface input file)
   if OPCFD ge 2 (rotor#2) and NROTOR gt 1, then
      call FILEW2 (read CFD interface input file)
INPTN

call INPTC (read namelist NLTRIM)

if OPREAD(1) ne 0, then
    call INPTR1 (read namelist NLRTR)

if OPREAD(2) ne 0, then
    call INPTW1 (read namelist NLWAKE)

if OPREAD(3) ne 0, then
    call INPTR2 (read namelist NLRTR)

if OPREAD(4) ne 0, then
    call INPTW2 (read namelist NLWAKE)

if OPREAD(5) ne 0, then
    call INPTB (read namelist NLBODY)

if OPREAD(6) ne 0, then
    call INPTL1 (read namelist NLOAD)

if OPREAD(7) ne 0, then
    call INPTL2 (read namelist NLOAD)

if OPREAD(8) ne 0, then
    call INPTF (read namelist NLFLUT)

if OPREAD(9) ne 0, then
    call INPTS (read namelist NLSTAB)

if OPREAD(10) ne 0, then
    call INPTT (read namelist NLTRAN)
INIT

call INITA
call INITC
call INITB
call INITR1
call INITR2
call INITE
call INITS

call STRIC
call STRIT

call CHEKR1
call CHEKR2
TRIM

TIMER

uniform inflow
if ITERU ne 0, then
call TRIMW

nonuniform inflow with prescribed wake geometry
if ITERR ne 0, then
call TRIMW

nonuniform inflow with free wake geometry
if ITERF ne 0, then
call TRIMW

call WAKEX1

call WAKEC1

call FILEX1 (write CFD interface output file)
call WAKEX2

call WAKEC2

call FILEX2 (write CFD interface output file)
call MODEP1
call MODEP2
call TRIMP
call FILEP (status of analysis)
call TRIMIM
call TRIMIT

call PERF
call LOAD
call TIMER
influence coefficient and trim iteration
for IT = 1 to ITMAX (ITERR or ITERF)
call WAKEC1
call WAKEC2
call TRIMI

if IT multiple of NPRNTT
call MODEP1
call MODEP2
call TRIMP

    call FILEP (status of analysis)
call TRIMIM
call TRIMIT

if NPRNTP gt 0, then
call PERF
if NPRNTL gt 0, then
call LOAD
TRIMI

initial control setting
call TRIMIV
call STRI
call TRIMIM

trim iteration
call TRIMIT
for COUNTT = 1 to MTRIM

    if COUNTT - 1 = multiple of MTRIMD, then
        construct derivative matrix by perturbation
        for I = 1 to MT (perturb controls)
            call TRIMIC
        call TRIMIV
        call STRI
        call TRIMIT
        call TRIMIC
        call MINV
    else if OPTIDR ne 0, then
        recursive update of derivative matrix
        call MINV

    call TRIMIC (increment controls)
call TRIMIV
call STRI
call TRIMIT
test trim convergence

print warnings if not converged
MODEP1

    call FILEP

PERF

    call TIMER
    call CONVP
    call PERFO
    call PERFS
    call PERFA
    call PERFR

        call PERFR1
        call PERFR2

    call TIMER

PERFR1

    call FILEP
    call PERFMT

        call FILEP

    call PERFF1
LOAD

    call TIMER
    call LOADR1
    call LOADR2
    call LOADV
        call LDVS
        call FILEP
    call TIMER

LOADR1

    call LDRM1
        call MOTNB1

    if MDLOAD ne 0, then
        call LOADD1
    if MALOAD ne 0, then
        call LOADA1
    if MWAKE gt 0, then
        call GEOMP1
    if MHLOAD ne 0, then
        call LOADH1
    for IR = 1 to MRLOAD
        call LOADS1
    for IN = 1 to MNOISE
        call NOISER1

LOADD1

    call FILEP
    call LOAADP
    call HISTPP
LOADA1

call FILEP
call LOADAM
call LOADAP
call HISTPP
call SPANPP
call POLRPP

GEOMP1

call GEOME1
call FILEP

LOADH1

call FILEP
call LDHX1

call LOADM
call LOADH
call LOADHP
call HISTPP
call LDHD1
call LOADHP
call HISTPP

LOADS1

call FILEP
call LDSX1

call LDS11
call LOADM
call LOADH
call LOADHP
call HISTPP
call LDSD1
call LOADHP
call HISTPP
NOISR1

  call FILEP
  call NOISF1

  call BESSEL
  call NOIST

  call FILEP

LOADAP

  call LOADH
  call FILEP

LOADHP

  call FILEP
initial control setting
  call RAMF

regulator iteration
  call STRIZ
  call STRIJ
  for COUNTS = 1 to MSTR
    if COUNTS-1 = multiple of MIDSTR, then
      construct T-matrix by perturbation
      for I = 1 to NTSTR (perturb controls)
        call STRIT
        call RAMF
        call STRIZ
        call STRIJ
        call STRIC
    else if RIDSTR ne 0, then
      recursive update of T-matrix
      call STRIC
    increment controls
    call STRIT
    call RAMF
    call STRIZ
    call STRIJ
    test regulator convergence

call LDVS
call LDRM1
call LDHX1
call LDRM2
call LDHX2
call LDRM1
call LDSX1
call LDRM2
call LDSX2
call NOISF1
call NOISF2

call MINV
call TIMER
call BODYC
call BODY11
call MOTNC1
call MODE1
call BODY12
call MOTNC2
call MODE2
call BODYM1
call BODYM2

for COUNTC = 1 to ITERC (circulation iteration)
call CONVC1
call CONVC2
call WAKEU1
call WAKEN1
call WAKEU2
call WAKEN2
for COUNTM = 1 to ITERM (motion iteration)
call CONVM1
call INRTM1
call CONVM2
call INRTM2
call ENGN1
call ENGNM1
call ENGN2

def JPSI = 0 to MREV*MPSI by MPSIR (azimuth loop)
call MOTNH1
call MOTNR1
call MOTNH2
call MOTNR2
call BODYV1
call ENGNV1
call MOTNF1
call BODYV2
call ENGNV2
call MOTNF2
call MOTNS

test motion convergence
call CONVM1
call CONVM2
test circulation convergence
call CONVC1
call CONVC2

call BODYF
call BODYS
call CONVP
call TIMER
BODYI1

call WAKEI1

call BODYIW

  call VTXL
  call DPLL


call BODYIB

  call BDIEL
  call BDISP
  call BDIMS

  call BDIMSP

MODE1

call TIMER

call MODEC1

  if collective change gt EPMODE, then
  
    if HINGE ge 1, then
    
      call MODEB1

      call MODEG
      call MINV
      call EIGEN
      call MODER1

    if HINGE = 0, then
    
      call MODEA1

      call MODER1

  call MODEK1
  call MODED1

call MODET1

  call MODER1
  call MINV
  call EIGEN
  call MODER1

  call INRTC1

call TIMER
MOTNR1

call TIMER

for JP = JPSI+1 to JPSI+MPSIR (azimuth step)
call MOTNB1

call AEROF1

for IR = 1 to MRA (radial step)
call AEROS1

call AEROT1

call LDAMPI

call TIMER

BODYF

call BODYFN

call BODYA

call BODYAT

call INTERP

call BODYFL

call BODYD

call INTERP
call GEOMR1

call TIMER
call WAKEB1
call GEOMF1

call GEOMFS

call TIMER

call TIMER

call GEOME1
for I = 1 to MPSI (azimuth loop)
call GEOME1
call WAKEB2

call GEOME1
call VTXL

for M = 1 to NBLADE (blade loop)
call GEOME1
call VTXL

for K = 1 to KFW or KDW (wake age loop)
call GEOME1

call WKPFW1

call VTXL
call VTXS

call WKPRU1

call VTXL
call VTXS

call WKPNW1

call VTXL

call WKPAX1

call VTXL
call VTXS

call TIMER
VTXS

call VTXL

VTXL

call GEOMX

FLUT

call TIMER

if OPFLOW le 0 (constant coefficients), then
  call FLUTM
  call PRNTHF
  call MODEP1
  call MODEP2
  call FLUTL

  call TIMER

  if ANTYPE(1-4) ne 0, then
    call CSYSAN
    call CASEAN

    call TIMER

if OPFDAN ne 0, then
  call STABD

  call STABDV

  call STABE

if OPFLOW gt 0 (periodic coefficients), then
  for NT = 0 to MPSIPC
    call FLUTM
    if NT = MPSIPC
      call PRNTHF
      call MODEP1
      call MODEP2
    call PSYSAN

  TIMER
FLUTM

initialize (names and identifies)
call FLUTMI

blade modes and rotor matrices
call MODE1
call FLUTR1
call MODE2
call FLUTR2

airframe matrices
call FLUTB

coupled matrices
call FLUTMM

completed equations
call FLUTMD
call FLUTME
call FLUTMC

flight dynamics matrices
call FLUTMB

final equations; symmetric/antisymmetric equations
call FLUTMS

FLUTR1

call FLUTI1
call FLUTD1
call FLUTA1

[call AEROS1]
call FLUTF1
FLUTB

- call FLUTBC
- call BODYF
- call FLUTBC

FLUTMB

- call FILEO (scratch file)
- call DERED
- call QSTRAN
- call STABDV
- call FILEC

FLUTMS

- call DERED
- call QSTRAN
- call FILEO (scratch file)
- call DERED
- call QSTRAN
- call FILEO (scratch file)
- call FILEC
- call DERED
- call QSTRAN
- call FILEC
STAB

call TIMER
call PRNTHS

call STABM

stability derivatives
for ID = 1 to 21
  call STABMC (increment control or motion)

  for IT = 1 to ITERS
    call WAKEC1
    call WAKEC2
    call STRI

  call STABMP
  if NPRNTF gt 0, then
    call PERF
  if NPRNTL gt 0, then
    call LOAD

  call STABMC

flight dynamics matrices
  call STABMM

call STABD

call STABDV

call STABE

call TIMER
STABE

equation sets
for IEQ = 1 to 3 (if EQTYPE(IEQ) ne 0)
call DERED

call STABL

call TIMER

if ANTYPE(1-4) ne 0, then
call CSYSAN
call CASEAN

if ANTYPE(5) ne 0, then
call STABLI

call STABPI
call MINV
call STABPS
call STABP

integration
for IT = 1 to TMAX/TSTEP
call TRANC
call STABLA
call STABPS
if IT = multiple of NPRNTT, then
call STABP

call TRCKPP
call FILEP

call TIMER
TRAN

call TIMER
call PRNTH

call TRANPI
call MINV
call TRANS
call TRANP

integration

for IT = 1 to TMAX/TSTEP
    call TRANC

    call TRANI

        for IT = 1 to ITERT
            call WAKE1
            call WAKEC2
            call STRI

        call TRANS
        if IT = multiple of NPRNTT, then
            call TRANP
            if NPRNTP gt 0, then
                call PERF
            if NPRNTL gt 0, then
                call LOAD

    call TRCKPP
    call FILEP
    call TIMER

TRANC

call TRANCC

    call TRANCT

call TRANCG

    call TRANCT
2.4 Software

2.4.1 Software conventions.

Fortran offers common blocks and subroutine arguments as means to transfer data between modules. The use of common blocks has the disadvantage that access to the data base by lower levels of the program is not visible to or controlled by the upper levels. The disadvantage of arguments can be the need to pass them down through many levels before they are used. With very large programs, arguments are not a satisfactory solution: there are too many, they must be passed through too many levels, and there is a concern about the overhead in the calls.

Hence in CAMRAD/JA the data base (meaning here the complete set of global variables used by the program) is defined in terms of common blocks. Any one subroutine is unlikely to use all variables in a common however (the data base would be too fragmented if commons were restricted to a single interface). It is undesirable to have the unused variables present in a subroutine, particularly for very large programs. Moreover, it is desirable to have all the input and output of a subroutine directly evident in the code; and to have the capability to use local names for variables rather than the global names.

These considerations imply a data base structure and access that are not found explicitly in Fortran, but it can be implemented using COMMON and EQUIVALENCE statements. Specifically, commons can be introduced into a subroutine in terms of a dummy vector spanning the complete common, and the variables needed in the subroutine can be accessed by equivalencing them to the appropriate location in the dummy vector.

CAMRAD/JA has the requirement for identical analysis of two rotors. To avoid the overhead of moving data in and out of the rotor subroutines, parallel source is used for the two rotors. The disadvantages of this approach are the memory requirement and the maintenance of parallel source.
Experience writing software following these conventions and requirements demonstrated the need for a software tool to aid in the development and modification of CAMRAD/JA.

Without a software tool, implementing the data base interface in a subprogram in terms of COMMON and EQUIVALENCE statements does not guarantee that the interface will be clear in the resulting source. Implementing changes to this data base requires either revising all appearances of a common in the code, or introducing new commons specifically for the change. The former can be difficult to do correctly in a large program. The latter likely means that the change is only partially integrated with the original model, frequently limited in either scope or function. Multiple changes increase the difficulties geometrically, perhaps exponentially. Changing the dimension of an array can be particularly troublesome, and changing dimensions consistently throughout a large program is one of the most difficult modifications to make.

The INCLUDE capability of certain compilers solves some of these maintenance problems. Among the disadvantages of INCLUDE are that all variables of a common must be declared in a subroutine in order to use a subset of the variables; and renaming the variables is not allowed. Hence the use of INCLUDE would not allow the data base structure and access required here.

Typically the data base information exists in several places, including the user's manual, the programmer's manual, and the code formats. Consequently it is difficult to ensure that changes are reflected everywhere.

A software tool is needed to facilitate the development and maintenance of large Fortran programs, particularly the implementation of major changes. The emphasis should be on the data base structure -- its definition, access, and documentation. The tool must implement a construction that allows Fortran subroutines access to the data base. The information describing the data base must exist in only one place.
The information should be automatically extracted for documentation (both internal and external to the code).

2.4.2 Software tool.

CAMRAD/JA has been written using a software tool designed to handle inter-module communications in large Fortran programs; and to facilitate the development and maintenance of large programs, particularly the implementation of major changes. The tool emphasizes the definition, access, and documentation of a data base consisting of Fortran commons and variables. The principal components of the tool are: (a) a dictionary, which is the sole source of information about the data base; (b) translation of code (written using various constructions) to Fortran source (compilable) based on the dictionary information; (c) documentation of the dictionary contents; and (d) search of the subroutine code for use of dictionary information. The conventions allow for variable dimensions in arrays in the dictionary and code.

The software tool is a combination of a dictionary and a precompiler. The dictionary is the sole location for information (code input and output, input parameter read and print, documentation, and variable dimensions). The software tool, written itself in Fortran, makes extensive use of character string manipulations, and uses an indexed file (keyed access) for the dictionary. The principal components of the tool are as follows.

(a) Dictionary: the single source of information about the data base, consisting of common and variable definitions and dimension values.

(b) Translate: "code" is written using various constructions, and then translated into Fortran source (compilable) based on the dictionary information.

(c) Contents: documentation of the commons and variables is prepared using the dictionary information.
(d) Search: the subroutine code can be searched for titles, calls, the use of common or variable names, and the use of dimensions.

The software tool handles the maintenance problem associated with the use of parallel source for analysis of two rotors. Commons and subroutines are defined for rotor#1 only (with "1" in the names). The corresponding commons and subroutines for rotor#2 have identical structure. Conventions in the translation will produce both rotor#1 and rotor#2 source from the rotor#1 code.

The conventions of the software tool allow variable array dimensions to be designated in either the dictionary or in the code constructions for local or global arrays, with numerical values of the dimensions in the dictionary. These conventions avoid the necessity for translation of dimensions in a general Fortran statement. If the code is properly written to take advantage of the tool, a dimension can be changed everywhere in the program by just changing its value in the dictionary.

The code is written with a prologue that includes a title, subroutine argument definitions, and the principal construction that allows the subroutine to access the data base. This construction (translated to COMMON and EQUIVALENCE statements) permits extraction of variables from the dictionary, perhaps renamed or equivalenced to part of a dictionary array. Other constructions permit the automatic generation of subroutines to read the input parameters (by namelist) or print the input parameters, based entirely on information in the dictionary.

The dictionary includes all information that defines the commons and variables. The variable information includes the common assignment, variable name, type, argument, format for printing, and description. A file of the dictionary contents (all variable definitions and descriptions) accompanies the CAMRAD/JA source. Such information is required for reference when writing or modifying the code.
3. INPUT FILE PREPARATION

3.1 Input Data Format

CAMRAD/JA input data (excluding tables) are defined in two standard forms: an unformatted input file, and a set of namelists. All of the input for a rotorcraft analysis can be obtained from the job namelists, but that approach is not recommended because there are so many parameters. For a particular analysis project, the majority of the input parameters will have a fixed or baseline value. These parameters should be defined in an input file, and the job namelists used to make parameter changes for a specific run. The input file preparation program produces the input file in CAMRAD/JA format (it will also produce the namelists).

The CAMRAD/JA namelist format consists of the following eleven namelists, with the associated common blocks. This set will be referred to as the "CAMRAD/JA namelists."

<table>
<thead>
<tr>
<th>CAMRAD/JA Namelists</th>
<th>Commons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Data</td>
</tr>
<tr>
<td>NLTRIM</td>
<td>Job and trim</td>
</tr>
<tr>
<td>NLRTR</td>
<td>Rotor#1</td>
</tr>
<tr>
<td>NLWAKE</td>
<td>Wake, rotor#1</td>
</tr>
<tr>
<td>NLRTR</td>
<td>Rotor#2</td>
</tr>
<tr>
<td>NLWAKE</td>
<td>Wake, rotor#2</td>
</tr>
<tr>
<td>NLBODY</td>
<td>Airframe and drive train</td>
</tr>
<tr>
<td>NLOAD</td>
<td>Loads, airframe and rotor#1</td>
</tr>
<tr>
<td>NLOAD</td>
<td>Loads, airframe and rotor#2</td>
</tr>
<tr>
<td>NLFLUT</td>
<td>Flutter</td>
</tr>
<tr>
<td>NLSTAB</td>
<td>Flight dynamics</td>
</tr>
<tr>
<td>NLTRAN</td>
<td>Transient</td>
</tr>
</tbody>
</table>
These namelists must appear in the order shown, and the NLTRIM namelist must always be present. The remaining ten need not be present; which of them are being used is determined by a parameter in a preceding namelist (or in NLTRIM).

The CAMRAD/JA input file is constructed by an unformatted write of all of the above common blocks (with names xxDATA). In order to minimize the size of the input file, the following parameters are not always included.

(a) The airframe sensor definition (namelist NLLOAD, common LADATA) is written only if the parameter MVIB or NRVIB is greater than zero.

(b) The control system definition for the aeroservoelasticity analysis (namelist NLFLUT or NLSTAB, common HCDATA) is written only if the parameter MLOOP is greater than zero.

(c) The self-tuning regulator definition (namelist NLTRIM, common SRDATA) is written only if the parameter UFSTR is greater than zero.

The input file preparation program obtains information from any or all of the following sources: (1) an existing input file; (2) namelist files (up to nine), perhaps including initialization or default parameter values; (3) the job namelists. These sources are read in the order indicated, the data being overwritten. The input file preparation program will produce any or all of the following output: (1) a new input file; (2) a namelist file; (3) a listing of all the input parameters.

It is useful to have a namelist file that can be read first to zero all the input data. Then the rotorcraft-specific namelist files need only contain the nonzero data. Such a namelist file can be created by
running the input preparation program on a DEC VAX (which automatically initializes variables to zero), with no namelist or file input. It is necessary to set all character-type variables to blanks in the job namelist. This job must be rerun whenever the data base is changed during a code modification. The resulting namelist file ZEROS.LIST is available with the input preparation program.

Also available are baseline namelist files BASEF.LIST and BASEH.LIST. These files contain typical values of many of the input parameters, with some notes (see section 6.5).

3.2 Job Structure

A job to run the input file preparation program consists of the following steps.
(a) Definition of the files required by the job.
(b) Call of the input file preparation program.
(c) Namelist NLJOB, containing parameters defining job.
(d) Namelist NLREAD, with variable CNTNTS defining "CAMRAD/JA namelists" to follow.
(e) "CAMRAD/JA namelists."

So the job command stream has the input:

&NLJOB job parameters,&END
&NLREAD CNTNTS=...,&END
"CAMRAD/JA namelists"

The following files may be read or written, depending on the parameters in namelist NLJOB.
<table>
<thead>
<tr>
<th>logical name</th>
<th>unit number</th>
<th>format</th>
<th>use</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUTFILE</td>
<td>NFDATI</td>
<td>input file</td>
<td>read</td>
</tr>
<tr>
<td>INPUTLIST1</td>
<td>NFNLI0+1</td>
<td>namelist</td>
<td>read</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>INPUTLIST9</td>
<td>NFNLI0+9</td>
<td>namelist</td>
<td>read</td>
</tr>
<tr>
<td>OUTPUTFILE</td>
<td>NFDATO</td>
<td>input file</td>
<td>written</td>
</tr>
<tr>
<td>OUTPUTLIST</td>
<td>NFNLO</td>
<td>namelist</td>
<td>written</td>
</tr>
</tbody>
</table>

The namelist files (read or written) have the format:

```
&NLREAD CNTNTS=...,&END
"CAMRAD/JA namelists"
```

Note that both the job command stream and the namelist files consist of the namelist NLREAD and the "CAMRAD/JA namelists." NLTRIM is always present in "CAMRAD/JA namelists." The variable CNTNTS determines which other namelists are present:

```
CNTNTS(10) integer; namelists to be read
(0 to suppress read)
```

| CNTNTS(1) | NLRTR, rotor#1 |
| CNTNTS(2) | NLWAKE, rotor#1|
| CNTNTS(3) | NLRTR, rotor#2 |
| CNTNTS(4) | NLWAKE, rotor#2|
| CNTNTS(5) | NBODY          |
| CNTNTS(6) | NLLOAD, rotor#1|
| CNTNTS(7) | NLLOAD, rotor#2|
| CNTNTS(8) | NLFLUT         |
| CNTNTS(9) | NLSTAB         |
| CNTNTS(10)| NLTRAN         |

The input file preparation program reads first the job namelist NLJOB, containing the parameters defining what else to read. Then the order in which the other sources are read is: INPUTFILE, INPUTLIST1, ..., INPUTLIST9, and finally the remainder of the job command stream. Data from each of these sources supersedes that previously read.
A typical input file preparation job has the following form (for the DEC VAX).

```
$ASSIGN ZEROS.LIST INPUTLIST1
$ASSIGN helo.list  INPUTLIST2
$ASSIGN helr1.list INPUTLIST3
$ASSIGN helr2.list INPUTLIST4
$ASSIGN hel.dat OUTPUTFILE
$DEFINE/USER_MODE SYS$OUTPUT hel.out
$RUN INPUT
 &NLJOB NLISTI=4,&END
 &NLREAD CNTNTS=10*0,&END
 &NLTRIM &END
```

This job reads ZEROS.LIST to zero the data base, and then reads three files in namelist format (for the airframe, rotor#1, and rotor#2). It produces an unformatted CAMRAD/JA input file. By default, all the input parameters will be printed.

### 3.3 Input Variables

#### 3.3.1 Namelist NLJOB

<table>
<thead>
<tr>
<th>Job description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPRINT</td>
<td>integer; 1 to print all parameters; 0 to not; default 1</td>
</tr>
<tr>
<td>NFILEI</td>
<td>integer; 1 to read input file, 0 to not; default 0</td>
</tr>
<tr>
<td>NLISTI</td>
<td>integer; n to read n namelist files (maximum 9), 0 to not; default 0</td>
</tr>
<tr>
<td>NFILEO</td>
<td>integer; 1 to write input file, 0 to not; default 1</td>
</tr>
<tr>
<td>NLISTO</td>
<td>integer; 1 to write namelist file, 0 to not; default 0</td>
</tr>
</tbody>
</table>
NLSTC(10)  integer; specification of contents of output namelist file (NLTRIM always present); 0 to suppress read; default 0

<table>
<thead>
<tr>
<th>NLSTC(1)</th>
<th>NLSTC(2)</th>
<th>NLSTC(3)</th>
<th>NLSTC(4)</th>
<th>NLSTC(5)</th>
<th>NLSTC(6)</th>
<th>NLSTC(7)</th>
<th>NLSTC(8)</th>
<th>NLSTC(9)</th>
<th>NLSTC(10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLRTR, rotor#1</td>
<td>NLWAKE, rotor#1</td>
<td>NLRTR, rotor#2</td>
<td>NLWAKE, rotor#2</td>
<td>NLBODY</td>
<td>NLLOAD, rotor#1</td>
<td>NLLOAD, rotor#2</td>
<td>NLFLUT</td>
<td>NLSTAB</td>
<td>NLTRAN</td>
</tr>
</tbody>
</table>

Input and output unit numbers

NFDATI  integer; unit number for read of input file; default 40

NFNLO  integer; unit number for read of namelist files; default 40; unit number of nth namelist file is NFNLO+n (default 41 to 49)

NFDATO  integer; unit number for write of input file; default 50

NFNLO  integer; unit number for write of namelist file; default 51

NUIN  integer; unit number for job input; default 5

NUOUT  integer; unit number for job output; default 6
4. AIRFOIL FILE PREPARATION

4.1 Airfoil Table Format

The CAMRAD/JA rotorcraft analysis requires the blade airfoil tables in a standard form. The airfoil file preparation program produces the airfoil file in CAMRAD/JA format, from either airfoil tables or equations.

The steady, two-dimensional airfoil data (C_l, C_d, and C_m as a function of \( \alpha \), \( M \), and \( r \); see Figure 4-1) are usually obtained in tables from wind tunnel test data. If test data are not available, approximate tables might be synthesized from generalized airfoil equations. For use by this analysis, the tables will be translated to the following form. The data will be defined at a finite set of angle-of-attack points. To facilitate interpolation, these points will consist of several groups, with the same angle-of-attack increment within each group. Then the set of angle-of-attack points are completely specified by the \( \alpha \) at the boundaries between the groups, and the indices of these points: \( N_a, \alpha_{n1} \) to \( \alpha_{nNa} \), and \( n1 \) to \( nNa \) (for \( N_a - 1 \) groups). The organization is illustrated below.

```
+ + + + + + + + + +
+ 1 2 3 4 5 ... Na +
+ n1 n2 n3 n4 n5 ... nNa +
+ a1 a2 a3 a4 a5 ... aNa +
```

angle-of-attack points
boundaries of groups
boundary index
angle-of-attack index at boundary
angle-of-attack at boundary

Note that \( n_k \) is a count of all the angle-of-attack points (not just the boundary points). So \( nNa \) is the total number of values in the table. The angle-of-attack range in the table should be from -180 to 180 deg.
Figure 4-1. Sketch of section aerodynamic characteristics.
(the interpolation routine does not extrapolate if asked for data from outside the domain of the table). With this organization, the interpolation is most efficient: it is only necessary to search the table in terms of the groups; interpolation within a group requires only numerical operations. The reduction in search operations will be significant if a large number of points can be divided into a small number of groups. The organization is similar for the variation with Mach number.

For the radial variation, the blade is divided into segments with the same section, defined by \( r \) at the boundaries: \( r_0 \) to \( r_{N_r+1} \) for \( N_r \) segments. Within each radial segment, a single airfoil table is used; there is no radial interpolation.

For the table of each radial segment, a Reynolds number \( \text{Re}_{t1} \) is specified, corresponding to Mach number \( M = 1 \). Then the Reynolds number of the airfoil data at Mach number \( M \) is \( \text{Re} = M \text{Re}_{t1} \). This parameter will be used for the Reynolds number correction of the airfoil table by the rotorcraft analysis. The value of \( \text{Re}_{t1} \) can be superseded by input directly to the rotorcraft analysis. If \( \text{Re}_{t1} = 0 \), the Reynolds number correction can not be applied.

The input airfoil tables (to be converted to CAMRAD/JA format) are used in C81 format.

It is best to use airfoil tables based on two-dimensional measurements of the section loads. For cases when such measurements are not available, and can not be approximated by data for similar airfoils, the tables can be synthesized from equations that represent typical airfoil characteristics.
The airfoil file preparation program can perform one of three functions:

(a) Read airfoil table files (C81 format, for up to 10 radial stations) and write an unformatted CAMRAD/JA airfoil file. The file data can be printed and plotted.

(b) Construct airfoil tables using equations, and write an unformatted CAMRAD/JA airfoil file. The file data can be printed and plotted.

(c) Read an unformatted CAMRAD/JA airfoil file, to print and plot the file data. The output includes the information in the CAMRAD/JA airfoil file about its construction: the headers and file names of the C81 tables, or the equation parameters.

4.2 Job Structure

A job to run the airfoil file preparation program consists of the following steps.

(a) Definition of the files required by the job.

(b) Call of the airfoil file preparation program.

(c) Namelist NLTABL, containing parameters defining the job and the table.

(d) Namelist NLCHAR (for each radial station), containing airfoil equation parameters; not used if file is constructed from tables.

The following files may be read or written, depending on the parameters in namelist NLTABL.
<table>
<thead>
<tr>
<th>logical name</th>
<th>unit number</th>
<th>format</th>
<th>use</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFTABLE</td>
<td>NFILE0</td>
<td>CAMRAD/JA</td>
<td>read (OPREAD=0)</td>
</tr>
<tr>
<td>AFDECK1</td>
<td>NFILEI</td>
<td>C81</td>
<td>read (OPREAD=2)</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>AFDECK10</td>
<td>NFILEI+9</td>
<td>C81</td>
<td>read (OPREAD=2)</td>
</tr>
<tr>
<td>AFTABLE</td>
<td>NFILE0</td>
<td>CAMRAD/JA</td>
<td>written (OPREAD=1 or 2)</td>
</tr>
</tbody>
</table>

AFDECKn is for radial segment n (defined by R and NRB in namelist NLTABL), with AFDECK1 for the root segment.

Typical airfoil file preparation jobs have the following form (for the DEC VAX).

```
$ASSIGN af.tab AFTABLE
$ASSIGN af1.c81 AFDECK1
$ASSIGN af2.c81 AFDECK2
$DEFINE/USER_MODE SYSS$OUTPUT af.out
$RUN AIRFOIL
  &NLTABL OPREAD=2, TITLE='.....', NRB=2, R=0..5, L, RETABL=..., &END
```

This job reads C81 format tables for two radial stations and produces the unformatted CAMRAD/JA airfoil file. By default, the file data will be printed and plotted.

```
$ASSIGN af.eqs AFTABLE
$DEFINE/USER_MODE SYSS$OUTPUT af.out
$RUN AIRFOIL
  &NLTABL OPREAD=1, TITLE='.....', RETABL=..., &END
  &NLCHAR data, &END
```

This job constructs a table from equations, and produces the unformatted CAMRAD/JA airfoil file. By default, the file data will be printed and plotted.
This job reads an unformatted CAMRAD/JA airfoil file, and prints the information about its construction. By default, the file data will be printed and plotted.

4.3 Input Variables

4.3.1 Namelist NLTABL.

Job description

OPREAD integer; function selection
0 read existing file and print/plot data
1 create table from equations and write file
2 read C81 format airfoil files and write file (default)

TITLE character; title, maximum 80 characters; not used if OPREAD=0

OPPRNT(3) integer; output control, 0 to suppress; default 1
OPPRNT(1) interpolate and print table
OPPRNT(2) interpolate and plot table
OPPRNT(3) list complete file (actual entries, not interpolated)

Angle-of-attack boundaries

NAB integer; number of boundaries, \( N_a \); maximum 20

NA(NAB) integer; indices at boundaries, \( n_k \)

A(NAB) real; \( \alpha \) at boundaries (deg, -180. to 180.)

Mach number boundaries

NMB integer; number of boundaries, \( N_m \); maximum 20

NM(NMB) integer; indices at boundaries, \( n_k \)

M(NMB) real; \( M \) at boundaries (0. to 1.)
Radial segments

NRB integer; number of segments, \( N_\ell \); maximum 10

R(NRB+1) real; boundaries of segments; \( R(1)=0. \),
\( R(NRB+1)=1. \)

RETAB1(NRB) real; Reynolds number \( R_{\ell 1} \) of airfoil table for
\( M = 1; Re = M R_{\ell 1} \) for Mach number = \( M \)

Note: maximum \( NAB\times NMB \times NRB = 10000 \)

Print and plot definition (defaults in code) for OPPRNT(1) or
OPPRNT(2)

NMPRNT integer; number of Mach number values; maximum
10 (used for both print and plot)

MPRNT(NMPRNT) real; Mach number values

NAPRNT integer; number of angle-of-attack values;
maximum 60 (only used for print)

APRNT(NAPRNT) real; angle-of-attack values (deg)

Input and output unit numbers

NFILEO integer; unit number for read or write of airfoil
file; default 40

NFILEI integer; unit number for read of C81 files;
default 51; unit number of nth C81 file is
NFILEI-1+n (default 51 to 60)

NUIN integer; unit number for job input; default 5

NUOUT integer; unit number for job output; default 6

4.3.2 Namelist NLCHAR.

This namelist is read for each radial station, if OPREAD = 1 (defaults
in code).
CLA  real; \( a = c_{1\alpha} \) at \( M = 0 \) (per rad)

MDIV  real; lift divergence Mach number \( M_{\text{div}} \)

CLMAX  real; \( c_{\text{lmax}} \) at \( M = 0 \)

FSTALL  real; factor \( f_s \) for \( c_{\text{lmax}} \)

MSTALL  real; Mach number \( M_s \) for \( c_{\text{lmax}} \)

GSTALL  real; factor \( g_s \) for stall \( c_l \)

HSTALL  real; factor \( h_s \) for stall \( c_l \)

CLF  real; \( c_{1f} \) for stall \( c_l \)

CMAC  real; \( c_{\text{mac}} \)

CMS  real; \( c_{\text{ms}} \)

DELO  real; \( \delta_0 \)

DEL1  real; \( \delta_1 \)

DEL2  real; \( \delta_2 \)

DCDDM  real; \( \partial c_d/\partial M \)

MCRIT  real; critical Mach number at \( \alpha = 0 \)

ACRIT  real; \( \alpha_{\text{crit}} \) where critical Mach number is zero

ALFD  real; drag stall angle (deg)

CDF  real; \( c_{df} \) for stall \( c_d \)

4.3.3 **Input Airfoil File Format.**

The input airfoil files are in C81 format. There is one file per airfoil, containing the lift, drag, and moment coefficient data as a function of angle-of-attack and Mach number. The file consists of a header line, followed by the lift coefficient, drag coefficient, and moment coefficient tables:
The parameters are defined as follows.

NMx number of Mach number entries
NAx number of angle-of-attack entries
Mx(NMx) Mach numbers
Ax(NAx) angles of attack
Cx(NAx,NMx) coefficient

where x = L, D, and M for the lift coefficient, drag coefficient, and moment coefficient respectively. Note the following:

(a) The first line of the file contains a 30-character header, and the number of Mach number and angle-of-attack entries. The maximum values of NMx and NAx are 99; the minimum values are 2.

(b) Next the file contains the Mach numbers for the lift table, and then the angles of attack and lift coefficients. The format of each line is (F7.n, 9F7.n), with the first position occupied only by the angle-of-attack values. There is more than one line per angle-of-attack if NMx is greater than 9. The angle-of-attack and Mach number entries must be in sequential order. The angle-of-attack should range from -180. deg to 180. deg (since the data are not extrapolated beyond the table).

(c) Then the file contains the Mach numbers for the drag table, and the angles of attack and drag coefficients.
(d) Finally the file contains the Mach numbers for the moment table, and the angles of attack and moment coefficients.

4.4 Notes

(1) Default values for namelist NLTABL:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAB</td>
<td>6</td>
</tr>
<tr>
<td>NA</td>
<td>1,16,28,88,100,115</td>
</tr>
<tr>
<td>NMB</td>
<td>3</td>
</tr>
<tr>
<td>NM</td>
<td>1,7,21</td>
</tr>
<tr>
<td>M</td>
<td>0.,.6,.95</td>
</tr>
<tr>
<td>NRB</td>
<td>1</td>
</tr>
<tr>
<td>R</td>
<td>0.,1.</td>
</tr>
<tr>
<td>RETABL</td>
<td>0.</td>
</tr>
</tbody>
</table>

which defines a table with 2415 entries per radial station, using the following increments.

(a) Angle-of-attack: 1 deg increments from -30 to 30; 2 deg increments from 150 to -150; 10 deg increments elsewhere.
(b) Mach number: .1 increments from 0. to .6; .025 increments from .6 to .95.

With the maximum NAB*NMB*NRB=10000 (total number of table entries), four radial stations can be defined (NRB=4). For more radial stations, NAB or NMB must be reduced.
(2) Default values for namelist NLCHAR:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA</td>
<td>5.7</td>
</tr>
<tr>
<td>MDIV</td>
<td>0.75</td>
</tr>
<tr>
<td>CLMAX</td>
<td>1.2</td>
</tr>
<tr>
<td>FSTALL</td>
<td>0.5</td>
</tr>
<tr>
<td>MSTALL</td>
<td>0.4</td>
</tr>
<tr>
<td>GSTALL</td>
<td>1.2</td>
</tr>
<tr>
<td>HSTALL</td>
<td>0.6</td>
</tr>
<tr>
<td>CLF</td>
<td>1.12</td>
</tr>
<tr>
<td>CMAC</td>
<td>0.0</td>
</tr>
<tr>
<td>CMS</td>
<td>-0.07</td>
</tr>
<tr>
<td>DELO</td>
<td>0.0084</td>
</tr>
<tr>
<td>DEL1</td>
<td>-0.0102</td>
</tr>
<tr>
<td>DEL2</td>
<td>0.384</td>
</tr>
<tr>
<td>DCDDM</td>
<td>0.65</td>
</tr>
<tr>
<td>MCRIT</td>
<td>0.8</td>
</tr>
<tr>
<td>AGRIT</td>
<td>33.0</td>
</tr>
<tr>
<td>ALFD</td>
<td>12.0</td>
</tr>
<tr>
<td>CDF</td>
<td>2.05</td>
</tr>
</tbody>
</table>

which represents an NACA 0012 airfoil.

(3) The tables can be synthesized from equations that represent typical airfoil characteristics (OPREAD=1). The equations used in the airfoil file preparation program are as follows (see Figures 4-2 and 4-3).

A. Below stall

\[
c_{l\alpha} = \begin{cases} 
  \frac{a}{\sqrt{1-M^2}} & M < M_{div} \\
  \frac{a(1-M)/((1-M_{div})\sqrt{1-M_{div}^2})}{c_{l_{\alpha}}} & M_{div} < M < M_{div}^+.1 \\
  \frac{a((1-M)/((1-M_{div})\sqrt{1-M_{div}^2})}{c_{l_{\alpha}}} & M \geq M_{div}^+.1 \\
  \frac{(M-M_{div}^-.1)/(1-M_{div}^-.1)}{c_{l_{\alpha}}} & 
\end{cases}
\]

\[
c_l = c_{l\alpha}^\alpha
\]
\[ \begin{align*}
\mathbf{c}_m &= \mathbf{c}_{\text{mac}} \\
\mathbf{c}_d &= \delta_0 + \delta_1 \alpha + \delta_2 \alpha^2 + \Delta\mathbf{c}_d \\
\Delta\mathbf{c}_d &= \max(0, \partial\mathbf{c}_d/\partial (M-M_c)) \\
\mathbf{M}_c &= \max(0, M_{\text{crit}}(1-|\alpha|/\alpha_{\text{crit}}))
\end{align*} \]

B. Stall angle

\[ \mathbf{c}_{\text{ls}} = \mathbf{c}_{\text{lmax}} \min \left( 1, \frac{(1-M) + f_s(M-M_s)}{1-M_s} \right) \]

\[ \alpha_s = \frac{c_{\text{ls}}}{c_{\text{l}\alpha}} \]

C. Stalled lift (\(|\alpha| > \alpha_s\))

\[ \mathbf{c}_l = \text{sign}(\alpha) \max \left[ \frac{(g_s\alpha_s - |\alpha|)c_{\text{ls}} + (|\alpha| - \alpha_s)h_sc_{\text{ls}}}{g_s\alpha_s - \alpha_s}, \max(h_sc_{\text{ls}}, c_{\text{lfs}}\sin2|\alpha|) \right] \]

\[ \mathbf{c}_l = c_{\text{lfs}}\sin2\alpha \quad \text{if } |\alpha| > 45^\circ \]

D. Stalled moment (\(|\alpha| > \alpha_s\))

\[ \mathbf{c}_m = \begin{cases} 
\text{sign}(\alpha) \frac{(60 - |\alpha|)c_{\text{ms}} + (|\alpha| - \alpha_s).75c_{\text{mf}}}{60 - \alpha_s} & |\alpha| < 60^\circ \\
\text{sign}(\alpha) \frac{(90 - |\alpha|).75c_{\text{mf}} + (|\alpha| - 60)c_{\text{mf}}}{30} & |\alpha| > 60^\circ 
\end{cases} \]

\[ \mathbf{c}_{\text{mf}} = -\kappa\mathbf{c}_d(\alpha-90) - \kappa(c_d(\alpha-\alpha_d) + c_{d_f}) \]

74
E. Stalled drag ($|\alpha| > \alpha_d$)

$$c_d = c_d(\alpha - \alpha_d) + c_{df} \sin \left( \frac{|\alpha| - \alpha_d}{90 - \alpha_d} \right)$$

F. Reverse flow ($|\alpha| > 90$)

use effective angle-of-attack and account for moment axis shift

$$\alpha_e = \alpha - \pi \text{sign} \alpha$$

$$c_m = c_m + (\pi \cos \alpha_e) c_i + (\pi \sin \alpha_e) c_d$$
Figure 4-2. Airfoil characteristics -- lift and drag coefficients.
Figure 4-3. Airfoil characteristics -- moment coefficient and lift-curve slope.
5. ADDITIONAL INPUT FILE FORMATS

5.1 Blade Bending Mode File

The contents of this input file are the blade coupled flap/lag bending modes (for rotor #1 or rotor #2). The file characteristics are as follows.

File logical name: BENDMODEL
Function: rotor#1 bending modes
Unit number variable: NFBND1 (namelist NLCASE)
Unit number default value: 61
Subroutine using file: INPT (through FILEB1)
Parameter controlling use: HINGE (namelist NLRTR)

File logical name: BENDMODE2
Function: rotor#2 bending modes
Unit number variable: NFBND1 (namelist NLCASE)
Unit number default value: 62
Subroutine using file: INPT (through FILEB2)
Parameter controlling use: HINGE (namelist NLRTR)

The file has namelist format:

&NLBEND
NU=..., ETA=..., ETAP=..., ETAPP=..., EТАPH=..., 
&END

The input parameters are as follows.

NU(NBM) real; bending mode frequency, per rev
ETA(2,NBM,IR) real; deflection of bending mode
ETAP(2,NBM,IR) real; slope of bending mode
ETAPP(2,NBM,IR) real; curvature of bending mode
ETAPH(2,NBM) real; slope of bending mode at hinge

There are two components for each mode (NBM) and radial station (IR). The first component is out-of-plane motion (flap, positive
upward), the second component is inplane motion (lead, positive forward). This motion is measured relative hub plane axes (not rotated by the blade pitch or twist).

NBM is the number of bending modes required. If modes are skipped (in DOF of namelist NLTRIM) then NBM is the index of the last blade bending mode required.

IR is the index of the radial stations at which the bending modes are used.

<table>
<thead>
<tr>
<th>IR</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>r = r_{FA}</td>
</tr>
<tr>
<td>2</td>
<td>r = r_{PB}</td>
</tr>
<tr>
<td>3</td>
<td>r = r_{root}</td>
</tr>
<tr>
<td>4</td>
<td>r = 1.</td>
</tr>
<tr>
<td>5-15</td>
<td>r = (j - 1)0.1, j = 1 to 11</td>
</tr>
<tr>
<td>16-66</td>
<td>r = (j - 1)\Delta r, j = 1 to MRM+1</td>
</tr>
<tr>
<td></td>
<td>(\Delta r = 1/\text{MRM})</td>
</tr>
<tr>
<td>67-96</td>
<td>r = r_j, j = 1 to MRA (aerodynamic stations)</td>
</tr>
</tbody>
</table>

The aerodynamic radial stations r_j are the midpoints of the aerodynamic panels, defined by the edges RAE(MRA+1). The variables RFA, RPB, RROOT, MRM, RAE, MRA are in namelist NLRTR. The subroutine MODE1 can be used to obtain these radial stations.

The arrays are dimensioned for NBM = 10.

If problems occur using this file format, check the dictionary contents to confirm that the above is the current definition of the bending mode arrays (in common MD1CM).
5.2 Airframe Interference Velocity File

The contents of this input file are the aerodynamic interference velocities produced by the airframe (at rotor #1 or rotor #2). The file characteristics are as follows.

<table>
<thead>
<tr>
<th>File logical name:</th>
<th>AEROINT1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function:</td>
<td>rotor#1 body interference vel</td>
</tr>
<tr>
<td>Unit number variable:</td>
<td>NFINT1 (namelist NLCASE)</td>
</tr>
<tr>
<td>Unit number default value:</td>
<td>63</td>
</tr>
<tr>
<td>Subroutine using file:</td>
<td>INPT (through FILEV1)</td>
</tr>
<tr>
<td>Parameter controlling use:</td>
<td>OPINTV (namelist NLBODY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File logical name:</th>
<th>AEROINT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function:</td>
<td>rotor#2 body interference vel</td>
</tr>
<tr>
<td>Unit number variable:</td>
<td>NFINT2 (namelist NLCASE)</td>
</tr>
<tr>
<td>Unit number default value:</td>
<td>64</td>
</tr>
<tr>
<td>Subroutine using file:</td>
<td>INPT (through FILEV2)</td>
</tr>
<tr>
<td>Parameter controlling use:</td>
<td>OPINTV (namelist NLBODY)</td>
</tr>
</tbody>
</table>

The file has namelist format:

```
&NINT
  NVINT=..., VINTX=..., VINTY=..., VINTZ=..., VINTR=..., VINTT=..., VINTP=..., 
&END
```

The input parameters are as follows.

<table>
<thead>
<tr>
<th>NVINT</th>
<th>integer; specification of coordinate axes of velocity data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 airframe axes</td>
</tr>
<tr>
<td></td>
<td>2 nonrotating shaft axes</td>
</tr>
<tr>
<td></td>
<td>3 rotating shaft axes</td>
</tr>
</tbody>
</table>

<p>| VINTX(MRA,MPSI) | x component, + forward |
| VINTY(MRA,MPSI) | y component, + right  |
| VINTZ(MRA,MPSI) | z component, + down  |</p>
<table>
<thead>
<tr>
<th>MRA</th>
<th>MPSI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VINTX(MRA,MPSI)</td>
<td>x component, + aft</td>
<td>velocity (divided by flight speed V), for nonrotating shaft axes</td>
</tr>
<tr>
<td>VINTY(MRA,MPSI)</td>
<td>y component, + to advancing side</td>
<td></td>
</tr>
<tr>
<td>VINTZ(MRA,MPSI)</td>
<td>z component, + up</td>
<td></td>
</tr>
<tr>
<td>VINTR(MRA,MPSI)</td>
<td>Δur component, + outboard</td>
<td>for rotating shaft axes</td>
</tr>
<tr>
<td>VINTT(MRA,MPSI)</td>
<td>Δut component, + to trailing edge</td>
<td></td>
</tr>
<tr>
<td>VINTP(MRA,MPSI)</td>
<td>Δup component, + down</td>
<td></td>
</tr>
</tbody>
</table>

Note that VINTR, VINTT, and VINTP are equivalenced to VINTX, VINTY, and VINTZ, respectively.

MRA is the number of aerodynamic radial stations (namelist NLRTR), and MPSI is the number of azimuth stations (namelist NLTRIM). The velocities are defined at points \( (r_i, \psi_j) \) on the rotor disk, for \( i = 1 \) to MRA and \( j = 1 \) to MPSI. The radial stations are at the midpoints of the aerodynamic panels, defined by the positions of the panel edges (RAE, namelist NLRTR). The most inboard station is \( i = 1 \), and the most outboard station is \( i = \) MRA. The azimuth stations are \( \psi_j = j(360/\text{MPSI}) \) degrees, measured from downstream in the direction of rotation of the rotor.

The arrays are dimensioned for \( \text{MRA} = 30 \).

If problems occur using this file format, check the dictionary contents to confirm that the above is the current definition of the velocity arrays (in common AESICM).
5.3 Airframe Aerodynamic Coefficient File

The contents of this input file are the lift coefficient, drag coefficient, and moment coefficient of the airframe. The file characteristics are as follows.

<table>
<thead>
<tr>
<th>File logical name:</th>
<th>BODYAERO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function:</td>
<td>airframe aero coefficients</td>
</tr>
<tr>
<td>Unit number variable:</td>
<td>NFBAT (namelist NLCASE)</td>
</tr>
<tr>
<td>Unit number default value:</td>
<td>65</td>
</tr>
<tr>
<td>Subroutine using file:</td>
<td>INPT (through FILEF)</td>
</tr>
<tr>
<td>Parameter controlling use:</td>
<td>OPBAT (namelist NLBODY)</td>
</tr>
</tbody>
</table>

The coefficients are functions of elevator angle (deg), angle-of-attack (deg), and Mach number. The tables are arranged as angle-of-attack vs Mach number arrays, for a set of elevator angles. The file consists of a header line, a line with the reference area and chord, and then the lift, drag, and moment tables:

```
TITLE
AREA   CHORD
LABEL  ND
DI     NMi   NAi
       M(1,i).....M(NMi,i)
A(1,i)  C(1,1,i)....C(NMi,1,i)
       ...                         ....
A(NAi,i) C(1,NAi,i)....C(NMi,NAi,i)
       i = 1 to ND
```

Read Format

A32
2F12.0
A12,I2
F12.0,2I2
12X,8F12.0
9F12.0/(12X,8F12.0)
9F12.0/(12X,8F12.0)

The lines from LABEL on are repeated three times, for the lift coefficient, drag coefficient, and moment coefficient tables respectively (in that order). The lines from DI on are repeated ND times. The format for the M and A-C lines is 9F12.0, with the first position occupied only by the A (angle of attack) values. There is more
than one line per A value if there are more than 8 M (Mach number) values. The input parameters are as follows.

<table>
<thead>
<tr>
<th>HEADER</th>
<th>file header (32 characters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA</td>
<td>reference area S for coefficients; ft(^2) or m(^2)</td>
</tr>
<tr>
<td>CHORD</td>
<td>reference chord c for coefficients; ft or m</td>
</tr>
<tr>
<td>LABEL</td>
<td>CL, CD, or CM</td>
</tr>
<tr>
<td>ND</td>
<td>number of elevator angle entries; i = 1 to ND</td>
</tr>
<tr>
<td>D(i)</td>
<td>elevator angle (deg) for angle-of-attack</td>
</tr>
<tr>
<td></td>
<td>and Mach number array to follow</td>
</tr>
<tr>
<td>N(Mi)</td>
<td>number of Mach number entries</td>
</tr>
<tr>
<td>N(Ai)</td>
<td>number of angle-of-attack entries</td>
</tr>
<tr>
<td>M(N(Mi),N(Di))</td>
<td>Mach numbers</td>
</tr>
<tr>
<td>A(N(Ai),N(Di))</td>
<td>angles of attack (deg); relative to the airframe (F) axes</td>
</tr>
<tr>
<td>C(N(Mi),N(Ai),N(Di))</td>
<td>coefficient</td>
</tr>
</tbody>
</table>

The references S and c are only used with the coefficients in this file. The coefficients are defined as \(SC_L = L/q\), \(SC_D = D/q\), and \(SCC_M = M/q\). Hence if \(S = 1\) and \(c = 1\) are used, the table entries are \(L/q\) (ft\(^2\) or m\(^2\)), \(D/q\) (ft\(^2\) or m\(^2\)), and \(M/q\) (ft\(^3\) or m\(^3\)).

The coefficients are linearly interpolated between the D, A, and M values, without extrapolation beyond the table. ND, NMi, and NAl must be greater than or equal to 1. A coefficient independent of D is obtained using ND = 1. A zero table is obtained using ND = NM = NA = 1 and a single entry of C = 0.

The D, A, and M values must be unique and in sequential order (that successive values are unequal and in order is checked when the file is read).
The LABEL must be CL, CD, or CM, in that order (checked when the file is read).

The arrays are dimensioned for maximum ND, NMi, and NAi = 25; and the maximum number of coefficient array entries is 5000.

If problems occur using this file format, check the dictionary contents to confirm that the above is the current definition of the table (in common BATABL).
5.4 Airframe Stability Derivative File

The contents of this input file are stability derivatives of the airframe. The file characteristics are as follows.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>File logical name:</td>
<td>STABDERIV</td>
</tr>
<tr>
<td>Function:</td>
<td>airframe stability derivatives</td>
</tr>
<tr>
<td>Unit number variable:</td>
<td>NFDRV (namelist NLCASE)</td>
</tr>
<tr>
<td>Unit number default value:</td>
<td>66</td>
</tr>
<tr>
<td>Subroutine using file:</td>
<td>INPT (through FILED)</td>
</tr>
<tr>
<td>Parameter controlling use:</td>
<td>OPDRV (namelist NLBODY)</td>
</tr>
</tbody>
</table>

There are 33 coefficients, each a function of angle-of-attack (deg) and Mach number. The file consists of a header line; a line with the reference area, chord, and span; and then the coefficient tables:

```
Read Format

TITLE               A32
AREA                CHORD  SPAN     3F12.0
LABELi              NMI    NAI      A12.2I2
                   M(1,i)....M(NMI,i)  12X.8F12.0
A(1,i)              C(1,1,i)....C(NMI,1,i)  9F12.0/(12X.8F12.0)
....                 ....
A(NAI,i)            C(1,NAI,i)....C(NMI,NAI,i)  9F12.0/(12X.8F12.0)
                     i = 1 to 33
```

The lines from LABELi on are repeated for each of the 33 tables. The format for the M and A-C lines is 9F12.0, with the first position occupied only by the A (angle of attack) values. There is more than one line per A value if there are more than 8 M (Mach number) values. The input parameters are as follows.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEADER</td>
<td>file header (32 characters)</td>
</tr>
<tr>
<td>AREA</td>
<td>reference area S for coefficients; ft² or m²</td>
</tr>
<tr>
<td>CHORD</td>
<td>reference chord c for coefficients; ft or m</td>
</tr>
<tr>
<td>SPAN</td>
<td>reference span b for coefficients; ft or m</td>
</tr>
</tbody>
</table>
LABEL{i}  table identifier; \( i = 1 \) to 33
N{M_{i}}  number of Mach number entries
N{A_{i}}  number of angle-of-attack entries
M(N{M_{i}},33)  Mach numbers
A(N{A_{i}},33)  angles of attack (deg); relative to the
                airframe (F) axes
C(N{M_{i}},N{A_{i}},33)  coefficient

LABEL{i} identifies the coefficient table, with the following values
and order:

CLA, CLM, CLAD, CLQ, CLDE, CLDF,
CDA, CDM, CDAD, CDQ, CDDE, CDDF,
CMA, CMM, CMAD, CMQ, CMDE, CMDF,
CYB, CYP, CYR, CYDA, CYDR,
CLB, CLP, CLR, CLDA, CLDR,
CNB, CNP, CNR, CNDA, CNDR

These values are checked when the file is read. See namelist NLBODY for
a definition of the derivatives. The derivatives with respect to
angle-of-attack, sideslip, and control angle have units of per-radian or
per-degree depending on the parameter OPDRVU (namelist NLBODY). The
references (S, c, and b) and coefficients in this file and in namelist
NLBODY are entirely independent; only one set is used, as determined by
the parameter OPDRV (namelist NLBODY).

The references S, c, and b are only used with the coefficients in
this file. If \( S = 1 \), \( c = 1 \), and \( b = 1 \) are used, the table entries are
in the form of force/q or moment/q (ft or m to some power).

The coefficients are linearly interpolated between the A and M
values, without extrapolation beyond the table. N{M_{i}} and N{A_{i}} must be
greater than or equal to 1. A zero table is obtained using NM = NA - 1
and a single entry of C = 0.

The A and M values must be unique and in sequential order (that
successive values are unequal and in order is checked when the file is
read).

The arrays are dimensioned for maximum NMi and NAi = 25; and the
maximum number of coefficient array entries is 10000.

If problems occur using this file format, check the dictionary
contents to confirm that the above is the current definition of the
table (in common SDTABL).
5.5 CFD Interface Input and Output Files

The contents of these input and output files are the partial angle-of-attack and the blade loading required for the CFD interface (of rotor #1 or rotor #2). The file characteristics are as follows.

File logical name: CFDINPUT1
Function: rotor#1 CFD interface input
Unit number variable: NFCI1 (namelist NLCASE)
Unit number default value: 71
Subroutine using file: INPT (through FILEW1)
Parameter controlling use: OPCFD (namelist NLRTR)

File logical name: CFDINPUT2
Function: rotor#2 CFD interface input
Unit number variable: NFCI2 (namelist NLCASE)
Unit number default value: 72
Subroutine using file: INPT (through FILEW2)
Parameter controlling use: OPCFD (namelist NLRTR)

File logical name: CFDOUTPUT1
Function: rotor#1 CFD interface output
Unit number variable: NFCO1 (namelist NLCASE)
Unit number default value: 73
Subroutine using file: TRIM (through FILEX1)
Parameter controlling use: OPCFD (namelist NLRTR)

File logical name: CFDOUTPUT2
Function: rotor#2 CFD interface output
Unit number variable: NFCO2 (namelist NLCASE)
Unit number default value: 74
Subroutine using file: TRIM (through FILEX2)
Parameter controlling use: OPCFD (namelist NLRTR)

The files have namelist format:

\&NLCFD
   ALPHAP=...,CLTAB=...,CDTAB=...,CMTAB=..., 
\&END

for the output, and
&NLCFD
  CLOLD=...,CDOLD=...,CMOLD=...,  
  CLEXT=...,CDEXT=...,CMEXT=...,  
&END

for the input. The parameters are as follows.

**CFD interface output**

- **ALPHAP(MRA,MPSI)**
  - real; partial angle-of-attack, excluding the wake inside the CFD domain

- **CLTAB(MRA,MPSI)**
  - lift coefficient

- **CDTAB(MRA,MPSI)**
  - drag coefficient

- **CMTAB(MRA,MPSI)**
  - moment coefficient

**CFD interface input**

- real; old table coefficients
- **CxOLD=CxTAB** of previous cycle (from output file)

- **CLOLD(MRA,MPSI)**
  - lift coefficient

- **CDOLD(MRA,MPSI)**
  - drag coefficient

- **CMOLD(MRA,MPSI)**
  - moment coefficient

- real; coefficients from external calculation

- **CLEXT(MRA,MPSI)**
  - lift coefficient

- **CDEXT(MRA,MPSI)**
  - drag coefficient

- **CMEXT(MRA,MPSI)**
  - moment coefficient

Note that CLOLD, CDOLD, CMOLD in the input file are obtained from CLTAB, CDTAB, CMTAB in the output file of the previous cycle. However, if CLEXT, CDEXT, CMEXT are not to be used on part of the disk, then they must equal CLOLD, CDOLD, CMOLD there (perhaps both set to zero).

MRA is the number of aerodynamic radial stations (namelist NLRTR), and MPSI is the number of azimuth stations (namelist NLTRIM). The parameters are defined at points \((r_i, \varphi_j)\) on the rotor disk, for \(i = 1\)
to MRA and \( j = 1 \) to MPSI. The radial stations are at the midpoints of the aerodynamic panels, defined by the positions of the panel edges (RAE, namelist NLRTR). The most inboard station is \( i = 1 \), and the most outboard station is \( i = \text{MRA} \). The azimuth stations are \( \psi_j = j(360/\text{MPSI}) \) degrees, measured from downstream in the direction of rotation of the rotor.

The arrays are dimensioned for \( \text{MRA} = 30 \).

If problems occur using this file format, check the dictionary contents to confirm that the above is the current definition of the angle-of-attack and blade loading arrays (in common AES1CM).
6. ROTORCRAFT ANALYSIS

6.1 Overview

The CAMRAD/JA rotorcraft analysis accepts data in the following standard forms: unformatted airfoil files (from the airfoil file preparation program; see section 4); an unformatted input file (from the input file preparation program; see section 3), and a set of "CAMRAD/JA namelists." All of the input (excluding tables) for a rotorcraft analysis can be obtained from the job namelists, but that approach is not recommended because there are so many parameters. For a particular analysis project, the majority of the input parameters will have a fixed or baseline value. These parameters should be defined in an input file, and the job namelists used to make parameter changes for a specific run.

More than one case can be run in a single job. The airfoil files are read for the first case only. The input file can be read for the first case only or for every case. Each case must have the "CAMRAD/JA namelists."

The CAMRAD/JA namelist format consists of the following eleven namelists, with the associated common blocks.

<table>
<thead>
<tr>
<th>CAMRAD/JA Namelists</th>
<th>Data</th>
<th>Commons</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLTTRIM</td>
<td>Job and trim</td>
<td>TMDATA, SRDATA</td>
</tr>
<tr>
<td>NLRTR</td>
<td>Rotor#1</td>
<td>R1DATA</td>
</tr>
<tr>
<td>NLWAKE</td>
<td>Wake, rotor#1</td>
<td>G1DATA, W1DATA</td>
</tr>
<tr>
<td>NLRTR</td>
<td>Rotor#2</td>
<td>R2DATA</td>
</tr>
<tr>
<td>NLWAKE</td>
<td>Wake, rotor#2</td>
<td>G2DATA, W2DATA</td>
</tr>
<tr>
<td>NLBODY</td>
<td>Airframe and drive train</td>
<td>BDDATA, BADATA, ENDATA</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>Loads, airframe and rotor#1</td>
<td>LADATA, L1DATA</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>Loads, airframe and rotor#2</td>
<td>LADATA, L2DATA</td>
</tr>
<tr>
<td>NLFLUT</td>
<td>Flutter</td>
<td>FLDATA, HCDATA</td>
</tr>
<tr>
<td>NLSTAB</td>
<td>Flight dynamics</td>
<td>STDATA, GCDATA, HCDATA</td>
</tr>
<tr>
<td>NLTRAN</td>
<td>Transient</td>
<td>TNDATA, GCDATA</td>
</tr>
</tbody>
</table>
These namelists must appear in the order shown, and the NLTRIM namelist must always be present. The remaining ten need not be present; which of them are being used is determined by the parameter OPREAD in namelist NLTRIM.

Optionally, selected output can be directed to a plot data file. This file includes sufficient headers and titles to identify the data. The plot data file can be read using the CAMRAD/JA subroutine FILEP, the prologue of which describes the standard formats of the file. Alternatively, the file can be read directly, using the appropriate formatted or namelist read statements.

Depending on the analysis parameters, additional input files may also be read: (a) blade bending modes; (b) airframe interference velocity files; (c) airframe aerodynamic coefficient file; (d) airframe stability derivative file; (e) CFD interface input files. Depending on the analysis parameters, additional output files may also be written: (a) CFD interface output files. The formats of these files are described in section 5.

6.2 Job Structure

A job to run the rotorcraft analysis program consists of the following steps.
(a) Definition of the airfoil files required by the job.
(b) Optionally, definition of the input file required by the job.
(c) Optionally, definition of the plot data file required by the job.
(d) Optionally, definition of the input bending mode files required by the job.
(e) Optionally, definition of the input airframe interference velocity files required by the job.

(f) Optionally, definition of the input airframe aerodynamic coefficient file required by the job.

(g) Optionally, definition of the input airframe stability derivative file required by the job.

(h) Optionally, definition of the CFD interface input and output files required by the job.

(i) Definition of the scratch files (if required by the computer system).

(j) Call of the rotorcraft analysis program.

(k) Namelist NL CASE, containing the parameters defining the job.

(l) For each case, the "CAMRAD/JA namelists." The namelist NLTRIM must always be present; OPREAD in namelist NLTRIM determines which of the other namelists are read.

The following files may be read or written, depending on the parameters in namelist NL CASE.

<table>
<thead>
<tr>
<th>logical name</th>
<th>unit number</th>
<th>format</th>
<th>use</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUTFILE</td>
<td>NFDAT</td>
<td>input file</td>
<td>read</td>
</tr>
<tr>
<td>AFTABLE1</td>
<td>NFAF1</td>
<td>airfoil table, rotor#1</td>
<td>read</td>
</tr>
<tr>
<td>AFTABLE2</td>
<td>NFAF2</td>
<td>airfoil table, rotor#2</td>
<td>read</td>
</tr>
<tr>
<td>PLOTFILE</td>
<td>NFPLT</td>
<td>plot data file</td>
<td>written</td>
</tr>
<tr>
<td>BENDMODE1</td>
<td>NFBND1</td>
<td>bending modes, rotor#1</td>
<td>read</td>
</tr>
<tr>
<td>BENDMODE2</td>
<td>NFBND2</td>
<td>bending modes, rotor#2</td>
<td>read</td>
</tr>
<tr>
<td>AEROINT1</td>
<td>NFINT1</td>
<td>interference vel, rotor#1</td>
<td>read</td>
</tr>
<tr>
<td>AEROINT2</td>
<td>NFINT2</td>
<td>interference vel, rotor#2</td>
<td>read</td>
</tr>
<tr>
<td>BODYAERO</td>
<td>NFBAT</td>
<td>airframe aero coefficients</td>
<td>read</td>
</tr>
<tr>
<td>STABDERIV</td>
<td>NFDRV</td>
<td>airframe stability deriv</td>
<td>read</td>
</tr>
<tr>
<td>CFDINPUT1</td>
<td>NFCI1</td>
<td>CFD int input, rotor#1</td>
<td>read</td>
</tr>
<tr>
<td>CFDINPUT2</td>
<td>NFCI2</td>
<td>CFD int input, rotor#2</td>
<td>read</td>
</tr>
<tr>
<td>CFDOUTPUT1</td>
<td>NFCO1</td>
<td>CFD int output, rotor#1</td>
<td>written</td>
</tr>
<tr>
<td>CFDOUTPUT2</td>
<td>NFCO2</td>
<td>CFD int output, rotor#2</td>
<td>written</td>
</tr>
<tr>
<td>SCRATCHJ.CAMRAD</td>
<td>NFSCRJ</td>
<td>scratch file</td>
<td>read/write</td>
</tr>
<tr>
<td>SCRATCHL.CAMRAD</td>
<td>NFSCRCL</td>
<td>scratch file</td>
<td>read/write</td>
</tr>
<tr>
<td>SCRATCHF.CAMRAD</td>
<td>NFSCRF</td>
<td>scratch file</td>
<td>read/write</td>
</tr>
</tbody>
</table>

95
A typical rotorcraft analysis job has the following form (for the DEC VAX):

```
$ASSIGN hel.dat INPUTFILE
$ASSIGN af1.dat AFTABLE1
$ASSIGN af2.dat AFTABLE2
$ASSIGN job.plot PLOTFILE
$DEFINE/USER_MODE SYS$OUTPUT job.out
$RUN CAMRADJA
&NLCASE NCASES=2,INFILE=1,AFFILE=3,PLFILE=1,&END
&NLTRIM
   VKTS=x.,VTIP=x..,COLL=x.,LATCYC=x.,LNGCYC=x.,
   PEDAL=x.,APITCH=x.,AROLL=x.,OPREAD=10*0,
&END
&NLTRIM
   VKTS=y.,VTIP=y.,COLL=y.,LATCYC=y.,LNGCYC=y.,
   PEDAL=y.,APITCH=y.,AROLL=y.,OPREAD=10*0,
&END
```

The scratch files need not be explicitly defined for the VAX. For each case it is generally necessary to specify in namelist NLTRIM the aircraft speed (VKTS or VEL; default 0.), rotor tip speed (RPM or VTIP; default value in namelist NLRTR), and the initial control settings for the trim iteration. Note that if the initial control settings are not defined for the second case, the analysis uses the final trimmed values from the first case.

6.3 Input Variables

In this section the input variables for the rotorcraft analysis are defined. The definition is organized according to the namelists:
<table>
<thead>
<tr>
<th>Label</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLCASE</td>
<td>Case</td>
</tr>
<tr>
<td>NLTRIM</td>
<td>Job and trim</td>
</tr>
<tr>
<td>NLRTR</td>
<td>Rotor</td>
</tr>
<tr>
<td>NLWAKE</td>
<td>Wake</td>
</tr>
<tr>
<td>NLBODY</td>
<td>Airframe and drive train</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>Airframe and rotor loads</td>
</tr>
<tr>
<td>NLFLUT</td>
<td>Flutter</td>
</tr>
<tr>
<td>NLSTAB</td>
<td>Flight dynamics</td>
</tr>
<tr>
<td>NLTRAN</td>
<td>Transient</td>
</tr>
</tbody>
</table>

In the description of the input parameters for the rotor (namelist NLRTR), the quantities NBM and NTM are used.

(a) NBM is the index of the highest-frequency blade bending mode used in the analysis.

(b) NTM is the index of the highest-frequency blade torsion mode used in the analysis.

Both parameters are obtained from the specification of the degrees of freedom in the trim or flutter models (DOF, in namelist NLTRIM or NLFLUT).

Dimensional input parameters can be in either English or metric (SI) units, as selected by OPUNIT in namelist NLTRIM. A consistent mass-length-time system is used for the units of all input parameters: foot-slug-second, or meter-kilogram-second. There are two exceptions to this convention.

1. The aircraft gross weight (WEIGHT in namelist NLBODY) is input in pounds or kilograms.

2. The aircraft velocity (VKTS in namelist NLTRIM) is input in knots (or dimensionless, using VEL in namelist NLTRIM).

Dimensionless parameters are based on the air density, rotor rotational speed (rad/sec), and rotor radius. Angles are input in degrees.
6.3.1 Namelist NLCASE

Job description

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCASES</td>
<td>integer; number of cases; default 1</td>
</tr>
<tr>
<td>INFILE</td>
<td>integer; read of input file</td>
</tr>
<tr>
<td>AFFILE</td>
<td>integer; read of airfoil files (first case only)</td>
</tr>
<tr>
<td>PLFILE</td>
<td>integer; write to plot file enabled</td>
</tr>
</tbody>
</table>

Input and output unit numbers

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFDAT</td>
<td>integer; unit number for read of input file; default 40</td>
</tr>
<tr>
<td>NFAF1</td>
<td>integer; unit number for read of rotor#1 airfoil file; default 41</td>
</tr>
<tr>
<td>NFAF2</td>
<td>integer; unit number for read of rotor#2 airfoil file; default 42</td>
</tr>
<tr>
<td>NFPLT</td>
<td>integer; unit number for write of plot file; default 43</td>
</tr>
<tr>
<td>NFBD1</td>
<td>integer; unit number for read of rotor#1 bending mode file; default 61</td>
</tr>
<tr>
<td>NFBD2</td>
<td>integer; unit number for read of rotor#2 bending mode file; default 62</td>
</tr>
<tr>
<td>NFINT1</td>
<td>integer; unit number for read of rotor#1 airframe interference velocity file; default 63</td>
</tr>
</tbody>
</table>
NFINT2  integer; unit number for read of rotor#2 airframe interference velocity file; default 64
NFBAT   integer; unit number for read of airframe aerodynamic coefficient file; default 65
NFDRV   integer; unit number for read of airframe stability derivative file; default 66
NFCII1  integer; unit number for read of rotor#1 CFD interface input file; default 71
NFCII2  integer; unit number for read of rotor#2 CFD interface input file; default 72
NFCO1   integer; unit number for read of rotor#1 CFD interface output file; default 73
NFCO2   integer; unit number for read of rotor#2 CFD interface output file; default 74
NFSCRJ  integer; unit number for scratch file; default 50
NFSCRL  integer; unit number for scratch file; default 51
NFSCRF  integer; unit number for scratch file; default 52
NUIN    integer; unit number for job input; default 5
NUOUT   integer; unit number for job output; default 6
NUDB    integer; unit number for job debug output; default 6
6.3.2 Namelist NLTRIM

Case description

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>character; title for job and case, maximum 80 characters</td>
</tr>
<tr>
<td>CODE</td>
<td>character; job or case identification, maximum 20 characters</td>
</tr>
<tr>
<td>OPUNIT</td>
<td>integer; unit system: 1 for English units (ft-slug-sec), 2 for metric units (m-kg-sec)</td>
</tr>
<tr>
<td>ANTYPE(3)</td>
<td>integer; analysis tasks (0 to suppress)</td>
</tr>
<tr>
<td></td>
<td>ANTYPE(1)</td>
</tr>
<tr>
<td></td>
<td>ANTYPE(2)</td>
</tr>
<tr>
<td></td>
<td>ANTYPE(3)</td>
</tr>
<tr>
<td>OPREAD(10)</td>
<td>integer; namelists to be read for this case (0 to suppress read)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(1)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(2)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(3)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(4)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(5)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(6)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(7)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(8)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(9)</td>
</tr>
<tr>
<td></td>
<td>OPREAD(10)</td>
</tr>
<tr>
<td>NPRNTI</td>
<td>integer; print of input parameters</td>
</tr>
<tr>
<td></td>
<td>0 none</td>
</tr>
<tr>
<td></td>
<td>1 parameters related to current job; determined by: LEVEL, ANTYPE (NLTRIM, NLFLUT, NLSTAB), OPFDAN, MNOISE, MVIB, NRVIB, MHHC, MHHCF, MHHAF, OPASE (NLFLUT, NLSTAB), OPSTR</td>
</tr>
<tr>
<td></td>
<td>2 all parameters, for current configuration; determined by: NROTOR, CONFIG, MPSI, MRA, MRI, NEM, NAF, NWING, NBODY, MLOOP, NTSTR, NZSTR, NPOFF</td>
</tr>
</tbody>
</table>
TRACE integer; print of convergence information for trim, regulator, circulation, and motion iterations
  0 none
  1 trim
  2 plus regulator
  3 plus circulation
  4 plus motion

DBTIME(3) integer; debug print interval; use DEBUG(1) to display counter
  DBTIME(1) option
    0 DEBUG always effective
    1 start debug print at
    2 start debug print at
      DBTIME(2), stop debug print at DBTIME(3)
      DBTIME(2), stop execution at DBTIME(3)

DBTIME(2) counter value at start of debug print

DBTIME(3) counter value at stop of debug print
integer: debug print control (from subroutines indicated in parentheses)
  0  no debug print
  2  low level print
  3  high level print
high level print can produce large amount of output (control with parameter DBTIME)

(1) timer and debug counter, 2 (TIMER)
(2) input, 2-3 (INPTx, FILEx)
(3) initialization, 2 (INITC, INITR, INITB, INITE, INITS)
(4) trim and regulator iterations, 2 (TRIMI, STRI)
(5) loads, 2 (LDSI)
(6) flutter matrices, 2-3 (FLUTMB, FLUTMM, FLUTMS)
(7) flutter coefficients, 2-3 (FLUTI, FLUTA)
(8) flight dynamics, 2-3 (STABM, STABMM, STABMC, STABE)
(9) transient, 2 (TRANI)
(10) rotor/airframe motion and forces, 2-3 (RAMF, CONVC, CONVM)
(11) blade modes, 2 (MODE, MODEx, FILEB)
(12) inertia coefficients, (INRTC)
(13) airframe constants and matrices, 2 (BODYC, ENGNC, MOTNC, BODYM, ENGNM, BODYS)
(14) induced velocity, 2-3 (WAKEU, WAKEN, WAKEX, FILEX, BODYI, BODYIB, BODYIW)
(15) rotor matrices, 2-3 (INRTM)
(16) hub/airframe motion and generalized forces, 2 (MOTNH, BODYV, ENGNV, MOTNF, MOTNS)
(17) rotor motion, 2-3 (MOTNR)
(18) rotor aerodynamics, 2-3 (AEROF)
(19) blade section aerodynamics, 3 (AEROS)
(20) body forces and aerodynamics, 2 (BODYF, BODYA, BODYD, BODYFL, BODYFN)
(21) wake influence coefficients, 2 (WAKEC, WKPA, WKPF, WKPNW, WKPRU)
(22) vortex line and sheet, 2-3 (VTXL, VTXS)
(23) prescribed wake geometry, 2-3 (GEOMR)
(24) free wake geometry, 2-3 (GEOMF, GEOMFS)
Operating conditions

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VKTS</td>
<td>real; aircraft speed V (knots)</td>
</tr>
<tr>
<td>VEL</td>
<td>real; velocity ratio, V/WR; only used if VKTS not input</td>
</tr>
</tbody>
</table>

input either VEL or VKTS by job namelist; if neither parameter is defined, V = 0 is used

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTIP</td>
<td>real; rotor#1 tip speed ( \Omega R ) (ft/sec or m/sec)</td>
</tr>
<tr>
<td>RPM</td>
<td>real; rotor#1 rotational speed (rpm); only used if VTIP not input</td>
</tr>
</tbody>
</table>

input either VTIP or RPM by job namelist; if neither parameter is defined, the normal tip speed VTIPN is used; rotor\#2 speed is calculated from the gear ratio TRATIO

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPDENS</td>
<td>integer; specification of aerodynamic environment</td>
</tr>
<tr>
<td></td>
<td>1 altitude and standard day</td>
</tr>
<tr>
<td></td>
<td>2 altitude and temperature</td>
</tr>
<tr>
<td></td>
<td>3 density and temperature</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTMSL</td>
<td>real; altitude above mean sea level (ft or m); used if OPDENS = 1 or 2</td>
</tr>
<tr>
<td>TEMP</td>
<td>real; air temperature (deg F or deg C); used if OPDENS = 2 or 3</td>
</tr>
<tr>
<td>DENSE</td>
<td>real; air density (slug/ft(^3) or kg/m(^3)); used if OPDENS = 3</td>
</tr>
<tr>
<td>OPGRDND</td>
<td>integer; ground effect analysis: 0 for out of ground effect, ne 0 for in ground effect</td>
</tr>
<tr>
<td>HAGL</td>
<td>real; altitude aircraft center-of-gravity above ground (ft or m), for ground effect analysis</td>
</tr>
</tbody>
</table>
Aircraft description

NROTOR  integer; number of rotors

AFLAP   real; wing flap angle $\delta_f$ (deg)

OPENCN  integer; engine state
0  normal operation
1  autorotation (engine inertia, engine damping, and throttle control torque zero; no engine speed degree of freedom)
2  engine out (engine damping and throttle control torque zero)

DOF(74) integer; vector defining degrees of freedom used in vibratory motion solution; 0 if not used; order:

rotor#1  $q_1$ ... $q_{10}$  $P_0$ ... $P_4$  $\beta_C$

rotor#2  $q_1$ ... $q_{10}$  $P_0$ ... $P_4$  $\beta_C$ (bending)  $\beta_C$ (torsion)  $\beta_C$ (gimbal/teeter)

airframe  $\phi_F$  $\theta_F$  $\psi_F$  $K_F$  $K_F$  $K_F$  $q_{S7}$ ... $q_{S36}$ (rigid body)  (flexible body)

drive train  $\psi_s$  $\psi_1$  $\psi_e$  $\Delta\theta_t$  $\Delta\theta_{govr1}$  $\Delta\theta_{govr2}$ (rotor/engine speed) (governor)

maximum number of bending modes = 10
maximum number of pitch/torsion modes = 5
maximum number of elastic airframe modes = 30

DOFT(8) integer; vector defining blade bending degrees of freedom used for mean deflection in nonlinear rotor equations; 0 if not used (subset of DOF); order:

rotor#1  $q_1$  $q_2$  $q_3$  $q_4$

rotor#2  $q_1$  $q_2$  $q_3$  $q_4$

104
### Motion analysis

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPSI</td>
<td>integer; number of azimuth steps per revolution in motion and loads analysis, maximum 36; for nonuniform inflow must be multiple of number of blade; for free wake geometry, maximum 24</td>
</tr>
<tr>
<td>MHARM(2)</td>
<td>integer; number of harmonics in rotor motion analysis; maximum 20; 0 for only mean response</td>
</tr>
<tr>
<td>MHARM(1)</td>
<td>rotor#1</td>
</tr>
<tr>
<td>MHARM(2)</td>
<td>rotor#2</td>
</tr>
<tr>
<td>MHARMF(2)</td>
<td>integer; number of harmonics in airframe vibration analysis (harmonics of N/rev, N = number of blades, so typically at most MHARM/NBLADE); maximum 10; 0 for only static elastic response</td>
</tr>
<tr>
<td>MHARMF(1)</td>
<td>rotor#1</td>
</tr>
<tr>
<td>MHARMF(2)</td>
<td>rotor#2</td>
</tr>
<tr>
<td>MPSIR</td>
<td>integer; in harmonic motion solution, number of rotor azimuth steps between update of airframe vibration</td>
</tr>
<tr>
<td>MREV</td>
<td>integer; in harmonic motion solution, number of revolutions between tests for motion convergence</td>
</tr>
<tr>
<td>ITERM</td>
<td>integer; maximum number of motion iterations</td>
</tr>
<tr>
<td>EPMOTN</td>
<td>real; tolerance for motion convergence (deg)</td>
</tr>
<tr>
<td>ITERC</td>
<td>integer; maximum number of circulation iterations</td>
</tr>
<tr>
<td>EPCIRC</td>
<td>real; tolerance for circulation convergence ($\Delta C_{T/\sigma}$)</td>
</tr>
</tbody>
</table>
Wake analysis

<table>
<thead>
<tr>
<th>LEVEL(2)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer; rotor wake analysis level (must be consistent with INFLOW): 0 for uniform inflow, 1 for nonuniform inflow with prescribed wake geometry, 2 for nonuniform inflow with free wake geometry</td>
<td></td>
</tr>
<tr>
<td>LEVEL(1)</td>
<td>rotor#1</td>
</tr>
<tr>
<td>LEVEL(2)</td>
<td>rotor#2</td>
</tr>
</tbody>
</table>

integers; number of iterations between trim and wake geometry; 0 to skip a stage

<table>
<thead>
<tr>
<th>ITERU</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>at uniform inflow stage</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITERR</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>at nonuniform inflow, prescribed wake geometry stage</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITERF</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>at nonuniform inflow, free wake geometry stage</td>
<td></td>
</tr>
</tbody>
</table>
Trim analysis

**OPTHM** integer; specification of trim option
0 free flight, no trim
LE 99 free flight
100 wind tunnel, no trim
GE 100 wind tunnel

free flight options

**OPTHM** = 1 trim forces and moments with
\[ \delta_o \delta_c \delta_s \delta_p \theta_{FT} \phi_{FT} \]
2 trim forces and moments with
\[ \delta_o \delta_c \delta_s \delta_p \theta_{FT} \psi_{FP} \]
3 trim forces, moments, and power with
\[ \delta_o \delta_c \delta_s \delta_p \theta_{FT} \phi_{FT} \theta_{FP} \]
4 trim forces, moments, and power with
\[ \delta_o \delta_c \delta_s \delta_p \theta_{FT} \psi_{FP} \theta_{FP} \]
5 trim forces, moments, and power with
\[ \delta_o \delta_c \delta_s \delta_p \delta_t \theta_{FT} \phi_{FT} \]
6 trim forces, moments, and power with
\[ \delta_o \delta_c \delta_s \delta_p \delta_t \theta_{FT} \psi_{FP} \]
7 trim symmetric forces and moment with
\[ \delta_o \delta_s \theta_{FT} \]
8 trim symmetric forces, moment, and power with
\[ \delta_o \delta_s \theta_{FT} \theta_{FP} \]
9 trim symmetric forces, moment, and power with
\[ \delta_o \delta_s \delta_t \theta_{FT} \]
wind tunnel options

**Optrim - TTC**

**trimmed quantities**
- \( L \) = rotor lift \( C_L/\sigma \) (wind axes)
- \( X \) = rotor drag \( C_X/\sigma \) (wind axes)
- \( Y \) = rotor side force \( C_Y/\sigma \) (shaft axes)
- \( T \) = rotor thrust \( C_T/\sigma \) (shaft axes)
- \( P \) = rotor power \( C_P/\sigma \)

<table>
<thead>
<tr>
<th>TT</th>
<th>trim</th>
<th>( \beta_s )</th>
<th>( \beta_c )</th>
<th>( \beta_s )</th>
<th>( \beta_c )</th>
<th>( \beta_s )</th>
<th>( \beta_c )</th>
<th>( \beta_s )</th>
<th>( \beta_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>12</td>
<td>trim P</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>13</td>
<td>trim X</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>14</td>
<td>trim P</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>15</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>16</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>17</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>18</td>
<td>trim P</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>19</td>
<td>trim P</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>20</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>21</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>22</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>23</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>24</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>25</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>26</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
<tr>
<td>27</td>
<td>trim L</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
<td>( \beta_s )</td>
<td>( \beta_c )</td>
</tr>
</tbody>
</table>

**trim variables**

if \( MT = 2 \)
- \( C = 1.4 \) with \( \delta_o \) \( \delta_c \)
- \( C = 2.5 \) with \( \delta_s \) \( \delta_c \)
- \( C = 3.6 \) with \( \beta_T \) \( \delta_c \)

if \( MT = 3 \)
- \( C = 1.4 \) with \( \delta_o \) \( \delta_s \) \( \delta_c \)
- \( C = 2.5 \) with \( \delta_o \) \( \theta_T \) \( \delta_c \)
- \( C = 3.6 \) with \( \delta_s \) \( \theta_T \) \( \delta_c \)

if \( MT = 4 \)
- \( C = 1.4 \) with \( \delta_o \) \( \delta_s \) \( \theta_T \) \( \delta_c \)

only longitudinal trimmed quantities (no \( Y \) or \( \beta_s \))
and trim variables (no \( \delta_c \)) if \( C = 4, 5, \) or \( 6 \)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTRIM</td>
<td>integer; maximum number of iterations on controls to achieve trim</td>
</tr>
<tr>
<td>MTRIMD</td>
<td>integer; number of trim iterations between perturbation identification of derivative matrix</td>
</tr>
<tr>
<td>DELTA</td>
<td>real; control step in perturbation identification of derivative matrix (stick displacement or aircraft attitude, deg)</td>
</tr>
<tr>
<td>OPTIDR</td>
<td>integer; ne 0 for recursive update of trim derivative matrix</td>
</tr>
<tr>
<td>ALPHA</td>
<td>real; weight in recursive update (exponential window)</td>
</tr>
<tr>
<td>FACTOR</td>
<td>real; factor reducing control increment in order to improve trim convergence</td>
</tr>
<tr>
<td>EPTRIM</td>
<td>real; tolerance on trim convergence</td>
</tr>
<tr>
<td>OPGOVT</td>
<td>integer; use of governor in trim</td>
</tr>
<tr>
<td></td>
<td>0 trim collective stick ( \delta_o )</td>
</tr>
<tr>
<td></td>
<td>1 trim rotor#1 governor</td>
</tr>
<tr>
<td></td>
<td>2 trim rotor#2 governor</td>
</tr>
<tr>
<td></td>
<td>3 trim both rotor governors</td>
</tr>
<tr>
<td>OPWT2T</td>
<td>integer; rotor forces and power evaluation for wind tunnel trim options (OPTRIM GT 100)</td>
</tr>
<tr>
<td></td>
<td>1 rotor#1 quantities only</td>
</tr>
<tr>
<td></td>
<td>2 sum of both rotors</td>
</tr>
</tbody>
</table>
Initial control settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLL</td>
<td>real; collective stick displacement $\delta_o$ or $\Delta \theta_{gogr}$ (deg), positive up</td>
</tr>
<tr>
<td>LATCYC</td>
<td>real; lateral cyclic stick displacement $\delta_c$ (deg), positive right</td>
</tr>
<tr>
<td>LNGCYC</td>
<td>real; longitudinal cyclic stick displacement $\delta_s$ (deg), positive forward</td>
</tr>
<tr>
<td>PEDAL</td>
<td>real; pedal displacement $\delta_p$ (deg), positive to right</td>
</tr>
<tr>
<td>THROTL</td>
<td>real; throttle displacement $\delta_t$ (deg)</td>
</tr>
<tr>
<td>APITCH</td>
<td>real; for free flight cases, aircraft pitch angle $\phi_{FT}$ (deg), positive nose up; for wind tunnel cases, rotor shaft angle-of-attack $\phi_T$ (deg), positive nose up</td>
</tr>
<tr>
<td>AROLL</td>
<td>real; for free flight cases, aircraft roll angle $\phi_{FP}$ (deg), positive to right</td>
</tr>
<tr>
<td>ACLIMB</td>
<td>real; for free flight cases, aircraft climb angle $\phi_{FP}$ (deg), positive up</td>
</tr>
<tr>
<td>AYAW</td>
<td>real; for free flight cases, aircraft yaw angle $\psi_{FP}$ (deg), positive to right; for wind tunnel cases, test module yaw angle $\psi_T$ (deg), positive to right</td>
</tr>
<tr>
<td>RTURN</td>
<td>real; trim turn rate $\psi_T$ (deg/sec), positive to right; only used for free flight cases</td>
</tr>
</tbody>
</table>

Initial values of controls, orientation, and motion; trimmed as determined by OPTRIM; free flight cases are OPTRIM le 99, wind tunnel cases are OPTRIM ge 100

$\phi_{FT}$ and $\phi_{FT}$ define orientation of body axes relative to earth axes

$\phi_{FP}$ and $\psi_{FP}$ define orientation of velocity axes relative to earth axes; $V_{climb} = V \sin \phi_{FP}$ and $V_{side} = V \sin \psi_{FP} \cos \phi_{FP}$
Targets for wind tunnel trim

CTTRIM   real; thrust $C_T/\sigma$ (shaft axes) or lift $C_L/\sigma$ (wind axes)

CPTRIM   real; power $C_P/\sigma$

CXTRIM   real; rotor drag $C_X/\sigma$ (wind axes, negative for propulsive force from rotor)

XTRIM    real; rotor drag $X/q$ (ft$^2$ or m$^2$; wind axes, negative for propulsive force from rotor); only used if CXTRIM=0.

CYTRIM   real; side force $C_Y/\sigma$ (shaft axes)

BCTRIM   real; longitudinal tip-path-plane tilt relative shaft $\beta_c$ (deg), positive for forward tilt

BSTRIM   real; lateral tip-path-plane tilt relative shaft $\beta_s$ (deg), positive for advancing side up

Trim output control

NPRNTT   integer; parameter n, trim/performanceeloads printed every nth cycle; le 0 to suppress

NPRNTP   integer; le 0 to suppress performance print

NPRNTL   integer; le 0 to suppress loads print

print control during intermediate cycles of nonuniform inflow and wake geometry iteration (do not influence print of final solution)

NTFILE   integer; if ne 0, write blade motion harmonics to plot file

NEFILE   integer; if ne 0, write blade bending and torsion modes to plot file

111
Self-tuning regulator

OPSTR  integer; regulator defined and active if ne 0
NZSTR  integer; number of measurements; maximum 40
NTSTR  integer; number of controls; le NZSTR, maximum 25
ZTARG(NZSTR)  real; targets for measurements
TZERO(NTSTR)  real; initial values of controls
WTZ(NZSTR)  real; measurement weights in cost function
WTDELT(NTSTR)  real; control increment weights in cost function
QSTR(NTSTR)  real; T-matrix variance for Kalman filter
DELSTR  real; control step in perturbation identification of T-matrix; deg
ALFSTR  real; weight in recursive update of T-matrix (exponential window)
FACTS  real; factor reducing control increment
JTARG  real; convergence criterion
  0  SQRT(cost) less than EPSTR
  1  SQRT(Δcost) less then EPSTR
NTMTRX  integer; T-matrix initialization: 0 for zero,
         1 to use input TMTRX
TMTRX(NZSTR,NTSTR)  real; initial T-matrix value
Self-tuning regulator in trim analysis

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTR</td>
<td>integer; maximum number of iterations on control to achieve convergence; 0 to turn off regulator</td>
</tr>
<tr>
<td>PIDSTR</td>
<td>integer; perturbation identification of T-matrix</td>
</tr>
<tr>
<td></td>
<td>0 never (use input TMTRX)</td>
</tr>
<tr>
<td></td>
<td>1 at beginning of trim, then every MIDSTR regulator iterations</td>
</tr>
<tr>
<td></td>
<td>2 each trim iteration, then every MIDSTR regulator iterations</td>
</tr>
<tr>
<td>MIDSTR</td>
<td>integer; number of regulator iterations between perturbation identification of T-matrix; le 0 for never</td>
</tr>
<tr>
<td>RIDSTR</td>
<td>integer; if ne 0, use recursive update of T-matrix</td>
</tr>
<tr>
<td>EPSTR</td>
<td>real; tolerance on regulator convergence</td>
</tr>
</tbody>
</table>
Self-tuning regulator measurement and control variables

CONSTR(4,NTSTR) character*4; specification of control variables; four keywords - category, rotor, quantity, and harmonic; left-justify each keyword

category = HHC  higher harmonic controls and forces
  rotor = RTR1  rotor #1
  rotor = RTR2  rotor #2
  rotor = SYM   symmetric
  rotor = ANTI  antisymmetric
  quantity = ROT rotating frame control
  quantity = COLL collective control
  quantity = LAT lateral cyclic control
  quantity = LNG longitudinal cyclic control
  quantity = FRCn nth auxiliary force
  harmonic = CSnn nth cosine harmonic
  harmonic = SNnn nth sine harmonic

category = RTR  rotor primary controls
  rotor = RTR1  rotor #1
  rotor = RTR2  rotor #2
  rotor = SYM   symmetric
  rotor = ANTI  antisymmetric
  quantity = T7S collective pitch
  quantity = T1C lateral cyclic pitch
  quantity = T1S longitudinal cyclic pitch

category = BODY  airframe primary controls
  quantity = DELF flaperon
  quantity = DELE elevator
  quantity = DELA aileron
  quantity = DELR rudder
  quantity = THTT throttle
  quantity = FRCn nth auxiliary force

114
summary of allowed keywords:

<table>
<thead>
<tr>
<th>category</th>
<th>rotor</th>
<th>quantity</th>
<th>harmonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>HHC</td>
<td>RTR1</td>
<td>ROT</td>
<td>CSnn</td>
</tr>
<tr>
<td></td>
<td>RTR2</td>
<td>COLL</td>
<td>SNnn</td>
</tr>
<tr>
<td></td>
<td>SYM</td>
<td>LAT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ANTI</td>
<td>LNG</td>
<td>FRcn</td>
</tr>
<tr>
<td>RTR</td>
<td>RTR1</td>
<td>T75</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTR2</td>
<td>T1C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYM</td>
<td>T1S</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ANTI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BODY</td>
<td></td>
<td>DELF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>THTT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FRcn</td>
<td></td>
</tr>
</tbody>
</table>
OUTSTR(6,NZSTR) character*4; specification of measurement variables; six keywords = category, rotor, location, quantity, measure, and harmonic; left-justify each keyword

category = PERF  rotor performance
rotor = RTR1  rotor #1
rotor = RTR2  rotor #2
rotor = SYM  symmetric
rotor = ANTI  antisymmetric
rotor = TOTL  total both rotors

quantity = BC  longitudinal flapping
quantity = BS  lateral flapping
quantity = POWR  power

category = VIB  airframe vibratory response
rotor = RTR1  rotor #1
rotor = RTR2  rotor #2
rotor = SYM  symmetric
rotor = ANTI  antisymmetric
rotor = TOTL  total both rotors

location = nnnn  sensor number
measure = COS  cosine component
measure = SIN  sine component
measure = MAG  magnitude
measure = RMS  root-mean-square
measure = AMP  1.414 * RMS

harmonic = nnnn  harmonic number
   (for COS, SIN, MAG)

category = HUB  nonrotating frame hub loads
rotor = RTR1  rotor #1
rotor = RTR2  rotor #2
rotor = SYM  symmetric
rotor = ANTI  antisymmetric

quantity = H  drag force
quantity = Y  side force
quantity = T  thrust

116
<table>
<thead>
<tr>
<th>quantity</th>
<th>roll moment</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>pitch moment</td>
</tr>
<tr>
<td>MY</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>torque</td>
</tr>
<tr>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>cosine component</td>
</tr>
<tr>
<td>COS</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>sine component</td>
</tr>
<tr>
<td>SIN</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>magnitude</td>
</tr>
<tr>
<td>MAG</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>mean</td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>half peak-to-peak</td>
</tr>
<tr>
<td>HPTP</td>
<td></td>
</tr>
<tr>
<td>harmonic</td>
<td>harmonic number</td>
</tr>
<tr>
<td>nnnn</td>
<td>(for COS, SIN, MAG)</td>
</tr>
</tbody>
</table>

**category - ROOT**  
rotating frame root loads

<table>
<thead>
<tr>
<th>rotor</th>
<th>rotor #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTR1</td>
<td></td>
</tr>
<tr>
<td>rotor</td>
<td>rotor #2</td>
</tr>
<tr>
<td>RTR2</td>
<td></td>
</tr>
<tr>
<td>rotor</td>
<td>symmetric</td>
</tr>
<tr>
<td>SYM</td>
<td></td>
</tr>
<tr>
<td>rotor</td>
<td>antisymmetric</td>
</tr>
<tr>
<td>ANTI</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>inplane shear</td>
</tr>
<tr>
<td>FX</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>radial shear</td>
</tr>
<tr>
<td>FR</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>vertical shear</td>
</tr>
<tr>
<td>FZ</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>flapwise moment</td>
</tr>
<tr>
<td>MX</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>lagwise moment</td>
</tr>
<tr>
<td>MZ</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>control moment</td>
</tr>
<tr>
<td>MC</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>cosine component</td>
</tr>
<tr>
<td>COS</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>sine component</td>
</tr>
<tr>
<td>SIN</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>magnitude</td>
</tr>
<tr>
<td>MAG</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>mean</td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
</tr>
<tr>
<td>measure</td>
<td>half peak-to-peak</td>
</tr>
<tr>
<td>HPTP</td>
<td></td>
</tr>
<tr>
<td>harmonic</td>
<td>harmonic number</td>
</tr>
<tr>
<td>nnnn</td>
<td>(for COS, SIN, MAG)</td>
</tr>
</tbody>
</table>

**category - BLDS**  
blade loads, rotating shaft axes

<table>
<thead>
<tr>
<th>rotor</th>
<th>rotor #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTR1</td>
<td></td>
</tr>
<tr>
<td>rotor</td>
<td>rotor #2</td>
</tr>
<tr>
<td>RTR2</td>
<td></td>
</tr>
<tr>
<td>rotor</td>
<td>symmetric</td>
</tr>
<tr>
<td>SYM</td>
<td></td>
</tr>
<tr>
<td>rotor</td>
<td>antisymmetric</td>
</tr>
<tr>
<td>ANTI</td>
<td></td>
</tr>
<tr>
<td>location</td>
<td>radial station number</td>
</tr>
<tr>
<td>nnnn</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>inplane shear</td>
</tr>
<tr>
<td>FX</td>
<td></td>
</tr>
</tbody>
</table>
quantity = FR radial shear
quantity = FZ vertical shear
quantity = MX flapwise moment
quantity = MZ lagwise moment

measure = COS cosine component
measure = SIN sine component
measure = MAG magnitude
measure = MEAN mean
measure = HPTP half peak-to-peak

harmonic = nnnn harmonic number (for COS, SIN, MAG)

category = BLDP blade loads, principal axes
  rotor = RTR1 rotor #1
  rotor = RTR2 rotor #2
  rotor = SYM symmetric
  rotor = ANTI anti-symmetric

location = nnnn radial station number

quantity = FX inplane shear
quantity = FR radial shear
quantity = FZ vertical shear
quantity = MX flapwise moment
quantity = MZ lagwise moment
quantity = MT torsion moment

measure = COS cosine component
measure = SIN sine component
measure = MAG magnitude
measure = MEAN mean
measure = HPTP half peak-to-peak

harmonic = nnnn harmonic number (for COS, SIN, MAG)

category = NOIS rotor rotational noise
  rotor = RTR1 rotor #1
  rotor = RTR2 rotor #2
  rotor = SYM symmetric
  rotor = ANTI antisymmetric

118
location = nnnn  microphone number
quantity = LIFT  lift noise
quantity = DRAG  drag noise
quantity = RADL  radial force noise
quantity = THCK  thickness noise
quantity = TOTL  total noise

measure = COS  cosine component
measure = SIN  sine component
measure = MAG  magnitude
measure = OCT  octave band spectrum
measure = RMS  rms pressure
measure = MAX  maximum pressure
measure = MIN  minimum pressure
measure = OSPL  overall sound pressure level
measure = DBA  dBA level

harmonic = nnnn  harmonic number
             (for COS, SIN, MAG)
             or octave band (for OCT)
summary of allowed keywords:

category | rotor | location | quantity | measure | harmonic
---|---|---|---|---|---
PERF | RTR1 | BC | | | |
| RTR2 | BS | | | | |
| SYM | | | | | |
| ANTI | | | | | |
| TOTL | | | | | |
| VIB | RTR1 | nnnn | COS | nnnn | |
| RTR2 | | | SIN | | |
| SYM | | | MAG | | |
| ANTI | | | RMS | | |
| TOTL | | | AMP | | |
| HUB | RTR1 | H | COS | nnnn | |
| RTR2 | Y | SIN | | | |
| SYM | T | MAG | | | |
| ANTI | MX | MEAN | | | |
| | MY | HPTP | | | |
| | Q | | | | |
| ROOT | RTR1 | FX | COS | nnnn | |
| RTR2 | FR | SIN | | | |
| SYM | FZ | MAG | | | |
| ANTI | MX | MEAN | | | |
| | MY | HPTP | | | |
| | MC | | | | |
| BLDS | RTR1 | nnnn | FX | COS | nnnn | |
| RTR2 | FR | SIN | | | |
| SYM | FZ | MAG | | | |
| ANTI | MX | MEAN | | | |
| | MY | HPTP | | | |
| BLDP | RTR1 | nnnn | FX | COS | nnnn | |
| RTR2 | FR | SIN | | | |
| SYM | FZ | MAG | | | |
| ANTI | MX | MEAN | | | |
| | MY | HPTP | | | |
| | MT | | | | |
| NOIS | RTR1 | nnnn | LIFT | COS | nnnn | |
| RTR2 | | | DRAG | SIN | |
| SYM | | | RADL | MAG | |
| ANTI | | | THCK | OCT | |
| | | | TOTL | RMS | |
| | | | | MAX | |
| | | | | MIN | |
| | | | | OSPL | |
| | | | | DBA | |
6.3.3 Namelist NLRTR

Rotor configuration

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>character; title for rotor data, maximum 80 characters</td>
</tr>
<tr>
<td>TYPE</td>
<td>character*4; label for rotor; suggest MAIN, FRNT, RCHT, or LOWR for rotor#1; and TAIL, REAR, LEFT, or UPPR for rotor#2</td>
</tr>
<tr>
<td>RADIUS</td>
<td>real; blade radius R (ft or m)</td>
</tr>
<tr>
<td>NBLADE</td>
<td>integer; number of blades</td>
</tr>
<tr>
<td>SIGMA</td>
<td>real; solidity ratio $\sigma = \frac{Nc_m}{\pi R}$ (based on mean chord)</td>
</tr>
<tr>
<td>ROTATE</td>
<td>integer; rotor rotation direction (viewed from above): 1 for counterclockwise, -1 for clockwise</td>
</tr>
<tr>
<td>VTIPN</td>
<td>real; normal tip speed $\Omega R_o$ (ft/sec or m/sec)</td>
</tr>
</tbody>
</table>
Higher harmonic forces and control -- rotating frame control

<table>
<thead>
<tr>
<th>MHHC</th>
<th>integer; number of harmonics, n = 1 to MHHC; maximum 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>THHC(MHHC)</td>
<td>real; pitch amplitude, ( \cos(n\psi) ); 1/rev not used (deg)</td>
</tr>
<tr>
<td>THHS(MHHC)</td>
<td>real; pitch amplitude, ( \sin(n\psi) ); 1/rev not used (deg)</td>
</tr>
</tbody>
</table>

Higher harmonic forces and control -- nonrotating frame control

<table>
<thead>
<tr>
<th>MHHCF</th>
<th>integer; number of harmonics, p = 1 to MHHCF; maximum 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOHHC(MHHCF)</td>
<td>real; collective pitch amplitude, ( \cos(pN\psi) ) (deg)</td>
</tr>
<tr>
<td>TOHHS(MHHCF)</td>
<td>real; collective pitch amplitude, ( \sin(pN\psi) ) (deg)</td>
</tr>
<tr>
<td>TCHHC(MHHCF)</td>
<td>real; lateral cyclic pitch amplitude, ( \cos(pN\psi) ) (deg)</td>
</tr>
<tr>
<td>TCHHS(MHHCF)</td>
<td>real; lateral cyclic pitch amplitude, ( \sin(pN\psi) ) (deg)</td>
</tr>
<tr>
<td>TSHHC(MHHCF)</td>
<td>real; longitudinal cyclic pitch amplitude, ( \cos(pN\psi) ) (deg)</td>
</tr>
<tr>
<td>TSHHS(MHHCF)</td>
<td>real; longitudinal cyclic pitch amplitude, ( \sin(pN\psi) ) (deg)</td>
</tr>
</tbody>
</table>

Higher harmonic forces and control -- airframe forces

<table>
<thead>
<tr>
<th>MHHAF</th>
<th>integer; number of harmonics, p = 1 to MHHAF; maximum 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FHHC(NAF,MHHAF)</td>
<td>real; force amplitude, ( \cos(pN\psi) ) (lb or N)</td>
</tr>
<tr>
<td>FHHS(NAF,MHHAF)</td>
<td>real; force amplitude, ( \sin(pN\psi) ) (lb or N)</td>
</tr>
</tbody>
</table>
Aerodynamic model

**BTIP**
real; tip loss parameter \( B \)

**OPTIP**
integer; tip loss type: 1 for tip loss factor, 2 for Prandtl function

**LINTW**
integer; twist type: 0 for nonlinear (TWISTA and TWISTI), 1 for linear (TWISTL)

**TWISTL**
real; linear twist rate \( \theta_{tw} \) (deg); used to calculate TWISTA and TWISTI, if LINTW = 1

**RGMAX**
real; \( r_{G_{\text{max}}} / R \) (maximum bound circulation for induced velocity calculation found outboard of \( r_{G_{\text{max}}} \))

**OPUSLD**
integer; use of unsteady lift, moment, and circulation terms
0 suppress
1 include
2 include, but zero for stall

**OPCOMP**
integer; 0 for incompressible rotor airloads

Aerodynamic model -- Reynolds number correction

**OPREYN**
integer; Reynolds number correction of airfoil tables
0 none
1 drag coefficient
2 lift coefficient
3 both

**EXPRED**
real; exponent in Reynolds number correction for drag coefficient; typically 0.125 to 0.2

**EXPREL**
real; exponent in Reynolds number correction for lift coefficient; typically 0.125 to 0.2

123
Aerodynamic model -- CFD interface

OPCFD  integer; use of CFD interface
0  none
1  calculate and write to file the partial angle of attack
2  read from file and use the externally calculated blade loading
3  both calculate/write and read/use

LDMCFD(3)  integer; if ne 0, use externally calculated blade loading (to suppress a coefficient on only part of the rotor disk, set \( c_{\text{ext}} = c_{\text{old}} \) in the input file)
LDMCFD(1)  lift coefficient
LDMCFD(2)  drag coefficient
LDMCFD(3)  moment coefficient
Stall model

OPSTLL integer; definition of stall model (the stall delay can be suppressed by setting TAU = 0.)
0 no stall
1 static stall
2 \( \alpha \) stall delay
3 \( \alpha \) stall delay, with dynamic stall vortex loads
4 \( \sqrt{\alpha} \) stall delay
5 \( \sqrt{\alpha} \) stall delay, with dynamic stall vortex loads

OPYAW integer; yawed flow corrections
0 both yawed flow and radial drag included
1 no yawed flow (\( \cos \Lambda = 1 \))
2 no radial drag (\( F_r = 0 \))
3 neither yawed flow nor radial drag included

ADELAY real; maximum angle-of-attack increment produced by stall delay (deg)

AMAXNS real; angle-of-attack in linear range, for no stall model (deg)

Stall model -- dynamic stall model

TAU(3) stall delay time constants for lift, drag, and moment; \( \tau \) calculated if TAU lt 0.

PSIDS(3) dynamic stall vortex load rise and fall time (azimuth increment) for lift, drag, and moment: \( \Delta \psi_{ds} \) (deg)

ALFDS(3) dynamic stall angle-of-attack for lift, drag, and moment: \( \alpha_{ds} \) (deg)

ALFRE(3) stall recovery angle of attack for lift, drag, and moment: \( \alpha_{re} \) (deg)

CLDSP maximum peak dynamic stall vortex-induced lift coefficient: \( \Delta c_{l,ds} \)

CDDSP maximum peak dynamic stall vortex-induced drag coefficient: \( \Delta c_{d,ds} \)

CMDSP maximum peak dynamic stall vortex-induced moment coefficient: \( \Delta c_{m,ds} \)
Inflow model

**INFLOW(6)** integer; definition of nonuniform wake-induced velocity calculation (must be consistent with LEVEL)

<table>
<thead>
<tr>
<th><strong>INFLOW(1)</strong></th>
<th>at this rotor: 0 for uniform, 1 for nonuniform</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INFLOW(2)</strong></td>
<td>at other rotor: 0 for zero, 1 for empirical, 2 for average at hub, 3 for nonuniform (only if $\Omega_2 = \Omega_1$)</td>
</tr>
<tr>
<td><strong>INFLOW(3)</strong></td>
<td>at wing-body: 0 for zero, 1 for empirical, 2 for nonuniform</td>
</tr>
<tr>
<td><strong>INFLOW(4)</strong></td>
<td>at horizontal tail: 0 for zero, 1 for empirical, 2 for nonuniform</td>
</tr>
<tr>
<td><strong>INFLOW(5)</strong></td>
<td>at vertical tail: 0 for zero, 1 for empirical, 2 for nonuniform</td>
</tr>
<tr>
<td><strong>INFLOW(6)</strong></td>
<td>number of points off rotor disk: 0 for none, maximum NPOFF</td>
</tr>
</tbody>
</table>

**KHLMDA** real; empirical inflow correction factor for hover, $\kappa_h$

**KFLMDA** real; empirical inflow correction factor for forward flight, $\kappa_f$

**OPFFLI** integer; model for linear inflow variation over rotor disk in forward flight
0 none
1 White and Blake model (uses FXLMDA and FYLMDA)
2 Coleman and Feingold model (uses FXLMDA and FYLMDA)
3 input KXLMDA and KYLMDA
real; linear inflow variation over rotor disk in forward flight (typically FXLMDA = FYLMDA = 1.)

KXLMDA input longitudinal gradient
KYLMDA input lateral gradient
FXLMDA longitudinal gradient factor \( f_x \)
FYLMDA lateral gradient factor \( f_y \)

FMLMDA real; factor \( f_y \) on linear inflow variation produced by hub moments (typically 1.)

real; factor for interference velocity at other rotor (\( \kappa_{21} \) or \( \kappa_{12} \)); linear variation between KINTH at \( \mu = 0.05 \) and KINTF at \( \mu = 0.10 \) is used

KINTH hover
KINTF forward flight

real; factor for rotor-induced interference velocity at airframe; \( K \) should equal the product of the fraction of fully-developed wake and the maximum fraction of the airframe surface in wake

KINTWB at wing-body, \( K_w \)
KINTHT at horizontal tail, \( K_H \)
KINTVT at vertical tail, \( K_V \)

FACTWU real; relaxation factor, introducing lag in thrust used to calculate induced velocity

OPTZT integer; hover trim near zero thrust, if ne 0 (inflow calculated for fixed, nominal wake vertical convection; uniform inflow analysis only)

CTSTZT real; nominal \( C_T/\sigma \) of fixed wake geometry for hover trim near zero thrust (0.01 used if input value is 0.)
Dynamic model -- bending and torsion modes

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HINGE</td>
<td>integer; specification of blade bending mode type</td>
</tr>
<tr>
<td></td>
<td>-1 read bending modes from file</td>
</tr>
<tr>
<td></td>
<td>0 articulated (rigid flap and lag modes only)</td>
</tr>
<tr>
<td></td>
<td>1 cantilever</td>
</tr>
<tr>
<td></td>
<td>2 hinged</td>
</tr>
<tr>
<td></td>
<td>3 hinged flap and cantilever lag (coning hub)</td>
</tr>
<tr>
<td></td>
<td>4 cantilever flap and hinged lag (lagging hub)</td>
</tr>
<tr>
<td>RCPL</td>
<td>real; structural coupling parameter ( R ) (effective pitch angle ( R\theta ) used to calculate blade bending modes; normally 1.)</td>
</tr>
<tr>
<td>EFLAP</td>
<td>real; hinge offset ( e/R ) (extent of rigid hub for cantilever blade)</td>
</tr>
<tr>
<td>ELAG</td>
<td>flap</td>
</tr>
<tr>
<td></td>
<td>lag</td>
</tr>
<tr>
<td>KFLAP</td>
<td>flap hinge spring (( \text{ft-lb/rad or m-N/rad} ))</td>
</tr>
<tr>
<td>KLAG</td>
<td>lag hinge spring (( \text{ft-lb/rad or m-N/rad} ))</td>
</tr>
<tr>
<td>TSPRNG</td>
<td>real; flap and lag hinge pitch angle at zero collective (deg), ( \theta_{ho} )</td>
</tr>
<tr>
<td>RCPLS</td>
<td>real; flap and lag hinge coupling parameter, ( R_s )</td>
</tr>
<tr>
<td></td>
<td>pitch angle of the flap and lag hinges:</td>
</tr>
<tr>
<td></td>
<td>( \theta_h = \theta_{ho} + R_s \theta_{75} )</td>
</tr>
<tr>
<td>RFA</td>
<td>real; feathering axis radial location, ( r_{FA}/R )</td>
</tr>
<tr>
<td>MRB</td>
<td>integer; number of radial stations for integration in blade modes calculation; maximum 100</td>
</tr>
<tr>
<td>MRM</td>
<td>integer; number of radial stations for integration of inertial coefficients; maximum 50</td>
</tr>
<tr>
<td>EPMODE</td>
<td>real; criterion on change of collective pitch to update blade modes, ( \Delta \theta_{75} ) (deg)</td>
</tr>
<tr>
<td>NONROT</td>
<td>integer; ne 0 to calculate nonrotating bending frequencies</td>
</tr>
</tbody>
</table>
NCOLB integer; number of collocation functions for bending mode calculations (total flap and lag, alternating); maximum 20

NCOLT integer; number of collocation functions for torsion model calculations; maximum 10

Dynamic model -- gimbal/teeter hinge

real; natural frequency (per rev at normal tip speed VTIPN)

NUGC longitudinal gimbal $\nu_{GC}$ or teeter $\nu_{T}$
NUGS lateral gimbal $\nu_{GS}$

real; damping (ft-lb/rad/sec or m-N/rad/sec)

GDAMPC longitudinal gimbal $C_{GC}$ or teeter $C_{T}$
GDAMPS lateral gimbal $C_{GS}$

Dynamic model -- control system

real; control system damping (ft-lb/rad/sec or m-N/rad/sec)

TDAMPO collective
TDAMPC cyclic
TDAMPR rotating

WTIN integer; source of control system stiffness: 1 for stiffness, 2 for frequency

real; control system frequency $\omega_\theta$ (per rev, at normal tip speed VTIPN), used if WTIN = 2

FTO collective
FTC cyclic
FTR reactionless

real; control system stiffness $K_\theta$ (ft-lb/rad or m-N/rad), used if WTIN = 1

KTO collective
KTC cyclic
KTR reactionless

129
Dynamic model -- lag damper

<table>
<thead>
<tr>
<th>LDAMPC</th>
<th>real; linear lag damper coefficient $C_r$ (ft-lb/rad/sec or m-N/rad/sec); estimated damping if a nonlinear damper is used (LDAMPM $\neq 0$); the lag mode has structural damping also (GSB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAMPM</td>
<td>real; maximum moment of nonlinear lag damper, $M_{LD}$ (ft-lb or m-N); linear lag damper used if LDAMPM $= 0$.</td>
</tr>
<tr>
<td>LDAMPR</td>
<td>real; lag velocity $\omega_{LD}$ where maximum moment of lag damper occurs (rad/sec); hydraulic damping below $\omega_{LD}$ and friction damping above.</td>
</tr>
</tbody>
</table>

Dynamic model -- blade properties

<table>
<thead>
<tr>
<th>GSB(NBM)</th>
<th>real; bending mode structural damping $g_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GST(NTM)</td>
<td>real; torsion mode structural damping $g_s</td>
</tr>
<tr>
<td>MBLADE</td>
<td>real; blade mass (slug or kg); if $le 0$, integral of section mass MASS used</td>
</tr>
<tr>
<td>MASST</td>
<td>real; tip mass (slug or kg); the tip mass can also be included directly in the section mass distribution</td>
</tr>
<tr>
<td>XIT</td>
<td>real; chordwise offset of the tip mass center of gravity aft of the elastic axis, $x_T/R$</td>
</tr>
</tbody>
</table>
Dynamic model -- pitch-bending coupling

\textbf{KPIN}\hspace{1cm} \text{integer; source of pitch-bending coupling:} \\
\hspace{1cm} \text{1 for input, 2 for calculated; negative to} \\
\hspace{1cm} \text{suppress } \cos \theta_{75} \text{ factor in pitch-bending and} \\
\hspace{1cm} \text{pitch-gimbal coupling}

\textbf{PHIPH}\hspace{1cm} \text{real; root geometry to calculate pitch-bending} \\
\hspace{1cm} \text{coupling (KPIN = 2 or -2)} \\
\hspace{1cm} \text{pitch horn cant angle, } \phi_{PH} \text{ (deg)}

\textbf{PHIPL}\hspace{1cm} \text{pitch link cant angle, } \phi_{PL} \text{ (deg)}

\textbf{RPB}\hspace{1cm} \text{pitch bearing radial location, } r_{PB}/R

\textbf{RPH}\hspace{1cm} \text{pitch horn radial location, } r_{PH}/R

\textbf{XPH}\hspace{1cm} \text{pitch horn length, } x_{PH}/R

\textbf{ATANKP(NBM)}\hspace{1cm} \text{real; pitch-bending coupling } \tan^{-1}K_p \text{ (deg),} \\
\hspace{1cm} \text{for pitch horn level (KPIN = 1 or -1)}

\textbf{DEL3G}\hspace{1cm} \text{real; pitch-gimbal coupling } \tan^{-1}K_{PG} \text{ (deg),} \\
\hspace{1cm} \text{for pitch horn level}
Dynamic model -- root geometry

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZFA</td>
<td>real; gimbal undersling, $z_{FA}/R$</td>
</tr>
<tr>
<td>XFA</td>
<td>real; torque offset, $x_{FA}/R$</td>
</tr>
<tr>
<td>CONE</td>
<td>real; precone angle $\delta_{FA_1}$ (deg), positive up</td>
</tr>
<tr>
<td>DROOP</td>
<td>real; droop angle $\delta_{FA_2}$ (deg) at $\theta_{75} = 0$, positive down from precone</td>
</tr>
<tr>
<td>SWEEP</td>
<td>real; sweep angle $\delta_{FA_3}$ (deg) at $\theta_{75} = 0$, positive aft</td>
</tr>
<tr>
<td>FDROOP</td>
<td>real; feathering axis droop angle $\delta_{FA_4}$ (deg), positive down from precone</td>
</tr>
<tr>
<td>FSWEEP</td>
<td>real; feathering axis sweep angle $\delta_{FA_5}$ (deg), positive aft</td>
</tr>
</tbody>
</table>

Dynamics model -- analysis

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>
| OPHVIB(3) | integer; control of hub vibration contributions, 0 to suppress; (gravity and static velocity terms always retained; must use OPHVIB(2)=0 if $\Omega_2 \neq \Omega_1$)  
  OPHVIB(1) vibration caused by this rotor  
  OPHVIB(2) vibration caused by other rotor  
  OPHVIB(3) static elastic deflection |
| FACTM    | real; relaxation factor introducing lag in forces for motion iteration; 1. for no lag, less than 1. to improve convergence |
**Blade section aerodynamic characteristics**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRA</td>
<td>integer; number of aerodynamic segments; maximum 30</td>
</tr>
<tr>
<td>RAE(MRA+1)</td>
<td>real; radial stations r/R at edges of aerodynamic segments; sequential, from root to tip</td>
</tr>
</tbody>
</table>

The following quantities are specified at the midpoints of the aerodynamic segments, from root to tip:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHORD(MRA)</td>
<td>real; blade chord, c/R</td>
</tr>
<tr>
<td>TWISTA(MRA)</td>
<td>real; blade twist relative 75% radius, $\theta_{tw}$ (deg)</td>
</tr>
<tr>
<td>THET2L(MRA)</td>
<td>real; incremental pitch of zero lift line, $\theta_{ZL}$ (deg); $\theta_{ZL}$ is the pitch of the axis corresponding to zero angle of attack in the airfoil tables, relative to the twist angle TWISTA</td>
</tr>
<tr>
<td>XA(MRA)</td>
<td>real; offset of aerodynamic center aft of elastic axis, $x_{A}/R$; $x_{A}$ is the point about which the moment data in the airfoil tables are given</td>
</tr>
<tr>
<td>XAC(MRA)</td>
<td>real; offset of aerodynamic center (for unsteady aerodynamics only) aft of elastic axis, $x_{AC}/R$</td>
</tr>
<tr>
<td>ASWEEP(MRA)</td>
<td>real; sweep angle of quarter-chord line relative to reference span line (deg), positive aft</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCORRL(MRA)</td>
<td>real; Mach number correction factor, $M_{eff} = fM$ for lift coefficient</td>
</tr>
<tr>
<td>MCORRD(MRA)</td>
<td>for drag coefficient</td>
</tr>
<tr>
<td>MCORRM(MRA)</td>
<td>for moment coefficient</td>
</tr>
<tr>
<td>DELCD(MRA)</td>
<td>real; drag coefficient increment, $\Delta c_d$</td>
</tr>
<tr>
<td>DELCM(MRA)</td>
<td>real; moment coefficient increment, $\Delta c_m$</td>
</tr>
<tr>
<td>RETABL(MRA)</td>
<td>real; Reynolds number $Re_{t1}$ of airfoil table for Mach number $M$; $Re = M Re_{t1}$ for Mach number $M$; supercedes value in airfoil table (if ne $0$.)</td>
</tr>
</tbody>
</table>
### Blade section inertial and structural characteristics

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRI</td>
<td>integer; number of radial stations where characteristics are defined; maximum 51</td>
</tr>
<tr>
<td>RI(MRI)</td>
<td>real; radial stations r/R where characteristics are defined; sequential, from root to tip, RI(1)=0. and RI(MRI)=1.</td>
</tr>
<tr>
<td>TWIST1(MRI)</td>
<td>real; blade twist relative 75% radius, $\theta_{tw}$ (deg)</td>
</tr>
<tr>
<td>MASS(MRI)</td>
<td>real; section mass, $m$ (slug/ft or kg/m)</td>
</tr>
<tr>
<td>XI(MRI)</td>
<td>real; offset of center of gravity aft of elastic axis, $x_i/R$</td>
</tr>
<tr>
<td>XC(MRI)</td>
<td>real; offset of tension center (modulus-weighted centroid) aft of elastic axis, $x_c/R$</td>
</tr>
<tr>
<td>KP2(MRI)</td>
<td>real; polar radius of gyration about elastic axis, $k_p^2/R^2$</td>
</tr>
<tr>
<td>EIZZ(MRI)</td>
<td>real; flapwise bending stiffness, $EI_{zz}$ (lb-ft$^2$ or N-m$^2$)</td>
</tr>
<tr>
<td>EIXX(MRI)</td>
<td>real; chordwise bending stiffness, $EI_{xx}$ (lb-ft$^2$ or N-m$^2$)</td>
</tr>
<tr>
<td>ITHETA(MRI)</td>
<td>real; section moment of inertia about elastic axis, $I_\theta$ (slug-ft or kg-m)</td>
</tr>
<tr>
<td>GJ(MRI)</td>
<td>real; torsional stiffness, $GJ$ (lb-ft$^2$ or N-m$^2$)</td>
</tr>
</tbody>
</table>
6.3.4 Namelist NLWAKE

Nonuniform inflow model

**OPFW** integer; wake configuration: far wake rollup model
0 single circulation peak, maximum circulation magnitude
1 single, outboard circulation peak
2 two circulation peaks

**OPNW** integer; wake configuration: near wake and lifting line model
0 collocation point at quarter-chord, straight lifting line
1 collocation point at 3c/4, straight lifting line
2 collocation point at quarter-chord, lifting line at aero center (Xa)
3 collocation point at 3c/4, lifting line at aero center (Xa)

integers; extent of wake regions (azimuth increments, wake age $\phi = K\Delta \psi$; ge 1)

**KNW** near wake, $K_{NW}$

**KRW** rolling up wake, $K_{RW}$

**KFW** far wake and tip vortices, $K_{FW}$

**KDW** far wake and tip vortices for collocation points off rotor disk, $K_{DW}$

**RRU** real; rolling up wake model
initial radial station of roll-up, $r_{RU}/R$

**FRU** initial tip vortex fraction of peak circulation, $f_{RU}$

**PRU** extent of roll-up, wake age $\phi_{RU}$ (deg)

**RTVTX** real; radial station of tip vortex at blade (fraction of blade radius)

integers; axisymmetric wake geometry model

**LHW** number of spirals of far wake, $L_{HW}$

**OPHW** axisymmetric wake geometry if 0
CORE(7)  real; vortex core radii, r_c/R
  CORE(1)  tip vortices
  CORE(2)  tip vortices, inboard blade
            collocation points
  CORE(3)  tip vortices, collocation points
            off rotor disk
  CORE(4)  inboard trailed wake lines
            (lt 0. for default = .5*width)
  CORE(5)  inboard shed wake lines
            (lt 0. for default = .5*length)
  CORE(6)  inboard rolled-up trailed wake,
            line model only, OPIVTL gt 0
            (lt 0. for default = .5*width)
  CORE(7)  near wake vortex lines
            (lt 0. for default = .2*width)

OPCORE(2)  integer; vortex core type:  0 for distributed
            vorticity, 1 for concentrated vorticity
     OPCORE(1)  tip vortices
     OPCORE(2)  inboard wake
WKMOLDL(13) integer; definition of wake model: 0 to omit
element, 1 for line segment with stepped
circulation distribution, 2 for line segment with
linear circulation distribution, 3 for vortex
sheet element

WKMOLDL(1) tip vortices (only line models)
WKMOLDL(2) near wake shed vorticity (no sheet
model)
WKMOLDL(3) near wake trailed vorticity (no
sheet model)
WKMOLDL(4) rolling up wake shed vorticity
WKMOLDL(5) rolling up wake trailed vorticity
WKMOLDL(6) far wake shed vorticity
WKMOLDL(7) far wake trailed vorticity
WKMOLDL(8) far wake (off rotor) shed
vorticity
WKMOLDL(9) far wake (off rotor) trailed
vorticity
WKMOLDL(10) bound vortices (no sheet model)
WKMOLDL(11) axisymmetric far wake axial
vorticity (no line model)
WKMOLDL(12) axisymmetric far wake shed
vorticity (no line model)
WKMOLDL(13) axisymmetric far wake ring
vorticity (no line model)

OPIVTL integer; inboard rolled-up trailed wake model;
only used with dual-peak model (OPFW=2)
    0 use WKMOLDL(7)
    1 stepped line segment
    2 linear line segment

OPRGI integer; source of inboard circulation peak
radial stations \( r_GI \) for wake geometry: 0 to
calculate, 1 to use input values; only used with
dual-peak model (OPFW=2)

RGI(MPSI) real; input \( r_GI \) (used only if OPRGI=1); -1. for
single-peak wake model at a particular azimuth

OPTVIC integer; core radius for inboard blade collocation
points; ne 0 to use CORE(2)
RTVIC(2) real; radial stations for transition (between CORE(2) and CORE(1)) of core radius for inboard collocation points (used only if OPTVIC=1)

EPVS real; tolerance in numerical integration of vortex sheet element velocities; ge 1000. for planar-rectangular approximation

DVS real; minimum separation between collocation point and sheet surface (similar to a core radius)

DLS real; lifting surface correction criterion, less than or equal 0. for no correction; correction applied if distance between collocation point and line segment less than \( d_{ls} \) (fraction rotor radius)

OPVXVY integer; if 0, suppress x and y components of induced velocity calculated at rotors

OPWGT integer; wake geometry in steady turn: 0 to suppress effect of turn rate

OPWKBP(4) integer; blade position model for wake analysis
   OPWKBP(1) 0 to suppress inplane motion
   OPWKBP(2) 0 to suppress all harmonics except mean
   OPWKBP(3) 0 for linear from root to tip
   OPWKBP(4) 0 to suppress twist in aero center offset

QDEBUG real; velocity criterion for debug print: print if z velocity component (per unit circulation) greater than QDEBUG

FACTWN real; relaxation factor, introducing lag in bound circulation used to calculated induced velocity
```plaintext
Computational domain for CFD interface (partial angle-of-attack calculation)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOMCFD</td>
<td>integer; specification of what wake elements inside the domain are to be excluded</td>
</tr>
<tr>
<td></td>
<td>0 behind reference blade, for wake age $\phi &lt; \phi_{\text{bound}}$</td>
</tr>
<tr>
<td></td>
<td>1 behind reference blade, for wake age $\phi &lt; \phi_{\text{bound}}$ and all tip vortices</td>
</tr>
<tr>
<td></td>
<td>2 all wake elements</td>
</tr>
<tr>
<td>PHICFD</td>
<td>real; limit on wake age behind reference blade, $\phi_{\text{bound}}$ (deg)</td>
</tr>
<tr>
<td>XCFD(6)</td>
<td>real; extent of domain; relative tip, rotating tip-path-plane axes, divided by rotor radius; all values positive</td>
</tr>
<tr>
<td>XCFD(1)</td>
<td>chordwise, back</td>
</tr>
<tr>
<td>XCFD(2)</td>
<td>chordwise, front</td>
</tr>
<tr>
<td>XCFD(3)</td>
<td>radial, inboard</td>
</tr>
<tr>
<td>XCFD(4)</td>
<td>radial, outboard</td>
</tr>
<tr>
<td>XCFD(5)</td>
<td>vertical, lower</td>
</tr>
<tr>
<td>XCFD(6)</td>
<td>vertical, upper</td>
</tr>
</tbody>
</table>
```
Prescribed or rigid wake geometry

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KRWG</td>
<td>integer; extent of prescribed or rigid wake geometry, $K_{RWG}$ ($\phi = K\Delta\phi$); maximum 144</td>
</tr>
<tr>
<td>OPRWG</td>
<td>integer; wake geometry model</td>
</tr>
<tr>
<td></td>
<td>1 from $K_1 = f_1\lambda$, $K_2 = f_2\lambda$, input $K_3$, input $K_4$</td>
</tr>
<tr>
<td></td>
<td>2 optional, without interference velocity in $\lambda$</td>
</tr>
<tr>
<td></td>
<td>3 from input $K_1$, $K_2$, $K_3$, $K_4$</td>
</tr>
<tr>
<td></td>
<td>4 Landgrebe hover model, from thrust coefficient</td>
</tr>
<tr>
<td></td>
<td>5 Landgrebe hover model, from maximum bound circulation</td>
</tr>
<tr>
<td></td>
<td>6 Kocurek and Tangler hover model, from thrust coefficient</td>
</tr>
<tr>
<td></td>
<td>7 Kocurek and Tangler hover model, from maximum bound circulation</td>
</tr>
<tr>
<td>FK2TWG</td>
<td>real; factor on tip vortex $K_2$ in prescribed wake geometry models ($OPRWG = 4$ to 7)</td>
</tr>
<tr>
<td>FGMXWG</td>
<td>real; factor on equivalent $C_T$ obtained from maximum bound circulation, for prescribed wake geometry models ($OPRWG = 5$ or 7)</td>
</tr>
<tr>
<td></td>
<td>real; factors $f_1$ and $f_2$ for prescribed wake geometry</td>
</tr>
<tr>
<td>FWGT(2)</td>
<td>tip vortex</td>
</tr>
<tr>
<td>FWCSI(2)</td>
<td>inside sheet edge</td>
</tr>
<tr>
<td>FWGSO(2)</td>
<td>outside sheet edge</td>
</tr>
<tr>
<td></td>
<td>real; constants $K_1$, $K_2$, $K_3$, $K_4$ for prescribed wake geometry</td>
</tr>
<tr>
<td>KWGT(4)</td>
<td>tip vortex</td>
</tr>
<tr>
<td>KWCSI(4)</td>
<td>inside sheet edge</td>
</tr>
<tr>
<td>KWGSO(4)</td>
<td>outside sheet edge</td>
</tr>
</tbody>
</table>
Free wake geometry

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KFWG</td>
<td>integer; extent of free wake geometry distortion calculation, $K_{FWG}$ (age $\phi = K\Delta\psi$); maximum 96, multiple MPSI</td>
</tr>
<tr>
<td>OPFWG</td>
<td>integer; wake geometry model</td>
</tr>
<tr>
<td>WGMODL(2)</td>
<td>integer; definition of wake model: 0 to omit element, 1 for line segment, 2 for sheet element</td>
</tr>
<tr>
<td>COREWG(3)</td>
<td>real; vortex core radii, $r_c/R$</td>
</tr>
<tr>
<td>RTWG(2)</td>
<td>real; radial station $r/R$ of trailed vorticity</td>
</tr>
<tr>
<td>MRVBWG</td>
<td>integer; number of wake revolutions used below point where induced velocity is being calculated</td>
</tr>
<tr>
<td>LDMWG</td>
<td>integer; parameter $l_{DM}$; general update every $l_{DM}\Delta\psi$ increment in boundary age</td>
</tr>
<tr>
<td>NDMWG(MPSI)</td>
<td>integer; parameter $n_{DM}(\psi_j)$, $j = 1$ to MPSI; boundary update every $n_{DM}$ increment in age</td>
</tr>
<tr>
<td>DQWG(2)</td>
<td>real; incremental velocity criteria; if magnitude of velocity increment greater than DQWG, then:</td>
</tr>
<tr>
<td>ITERWG</td>
<td>integer; number of wake geometry iterations</td>
</tr>
<tr>
<td>FACTWG</td>
<td>real; relaxation factor, introducing lag in distortion calculation to improve convergence</td>
</tr>
</tbody>
</table>
IPWGDB(2) integer; control of debug level 3 print of wake geometry distortion; 0 to suppress
   IPWGDB(1) azimuth increment in print before general update
   IPWGDB(2) azimuth increment in print after each iteration (last iteration printed in full)

QWGDB real; control of debug level 3 print of wake geometry distortion; induced velocity contribution of wake element printed if greater than QWGDB
6.3.5 *Namelist NLBODY*

Airframe configuration

<table>
<thead>
<tr>
<th>TITLE</th>
<th>character; title for airframe data; maximum 80 characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONFIG</td>
<td>integer; specification of rotorcraft configuration</td>
</tr>
<tr>
<td></td>
<td>0  one rotor</td>
</tr>
<tr>
<td></td>
<td>1  single main rotor and tail rotor</td>
</tr>
<tr>
<td></td>
<td>(tail rotor is rotor#2)</td>
</tr>
<tr>
<td></td>
<td>2  tandem main rotors</td>
</tr>
<tr>
<td></td>
<td>(rear rotor is rotor#2)</td>
</tr>
<tr>
<td></td>
<td>3  tilting proprotor</td>
</tr>
<tr>
<td></td>
<td>(left rotor is rotor#2)</td>
</tr>
<tr>
<td></td>
<td>4  coaxial main rotors</td>
</tr>
<tr>
<td></td>
<td>(upper rotor is rotor#2)</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>real; aircraft gross weight, including rotors (lb or kg)</td>
</tr>
<tr>
<td></td>
<td>real; aircraft moments of inertia, including rotors</td>
</tr>
<tr>
<td></td>
<td>(slug-ft(^2) or kg-m(^2))</td>
</tr>
<tr>
<td></td>
<td>IXX (= I_{xx})</td>
</tr>
<tr>
<td></td>
<td>IYY (= I_{yy})</td>
</tr>
<tr>
<td></td>
<td>IZZ (= I_{zz})</td>
</tr>
<tr>
<td></td>
<td>IXY (= I_{xy})</td>
</tr>
<tr>
<td></td>
<td>IXZ (= I_{xz})</td>
</tr>
<tr>
<td></td>
<td>IYZ (= I_{yz})</td>
</tr>
<tr>
<td>TRATIO</td>
<td>real; ratio of rotor#2 rotational speed to rotor#1</td>
</tr>
<tr>
<td></td>
<td>rotational speed, (\Omega_2/\Omega_1) (transmission gear ratio)</td>
</tr>
<tr>
<td>ASHAFT(2)</td>
<td>shaft angle-of-attack (\theta_R) (deg), positive rearward</td>
</tr>
<tr>
<td></td>
<td>ASHAFT(1) \hspace{1em} rotor#1</td>
</tr>
<tr>
<td></td>
<td>ASHAFT(2) \hspace{1em} rotor#2</td>
</tr>
<tr>
<td>ACANT(2)</td>
<td>shaft cant angle (\phi_R) (deg); positive to right for</td>
</tr>
<tr>
<td></td>
<td>main rotor; positive upward for tail rotor;</td>
</tr>
<tr>
<td></td>
<td>positive inward in helicopter mode for tilt rotor</td>
</tr>
<tr>
<td></td>
<td>ACANT(1) \hspace{1em} rotor#1</td>
</tr>
<tr>
<td></td>
<td>ACANT(2) \hspace{1em} rotor#2</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>ATILT</td>
<td>real; nacelle tilt angle $\alpha_n$ (deg), for tilting proprotor configuration only (CONFIG=3); 0. for airplane mode, 90. for helicopter mode</td>
</tr>
<tr>
<td>HMAST</td>
<td>rotor mast length from pivot to hub (ft or m), for tilting proprotor configuration only (CONFIG=3)</td>
</tr>
<tr>
<td>DPSI21</td>
<td>$\Delta\psi_2$ (deg); rotor#2 azimuth angle $\psi_2$ when rotor#1 azimuth angle $\psi_1 = 0$; must be 0. if $\Omega_2 \neq \Omega_1$</td>
</tr>
<tr>
<td>CANTHT</td>
<td>horizontal tail cant angle $\phi_{HT}$ (deg), positive to left</td>
</tr>
<tr>
<td>CANTVT</td>
<td>vertical tail cant angle $\phi_{VT}$ (deg), positive to right</td>
</tr>
</tbody>
</table>
### Location of aircraft components

Location of aircraft components relative to a body fixed reference system having an arbitrary orientation and origin; fuselage station positive aft, buttline positive to right, and waterline positive up (ft or m)

- **real; aircraft center of gravity**
  - **FSCG**: fuselage station
  - **BLCG**: buttline
  - **WLCG**: waterline

- **real; rotor#1 hub location (for tilting proprotor configuration, right nacelle pivot location)**
  - **FSR1**: fuselage station
  - **BLR1**: buttline
  - **WLR1**: waterline

- **real; rotor#2 hub location (not used for tilting proprotor configuration)**
  - **FSR2**: fuselage station
  - **BLR2**: buttline
  - **WLR2**: waterline

- **real; wing-body center of action**
  - **FSWB**: fuselage station
  - **BLWB**: buttline
  - **WLWB**: waterline

- **real; horizontal tail center of action**
  - **FSHT**: fuselage station
  - **BLHT**: buttline
  - **WLHT**: waterline

- **real; vertical tail center of action**
  - **FSVT**: fuselage station
  - **BLVT**: buttline
  - **WLVT**: waterline

- **NPOFF**: integer; number of points off the rotor, at which nonuniform wake-induced velocity calculated; maximum 20

- **real; location of points off the rotor**
  - **FSOFF(NPOFF)**: fuselage station
  - **BLOFF(NPOFF)**: buttline
  - **WLOFF(NPOFF)**: waterline
### Airframe elastic modes

<table>
<thead>
<tr>
<th>NEM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer; number of airframe modes for which data supplied; maximum 30</td>
<td></td>
</tr>
<tr>
<td>QFREQ(NEM)</td>
<td>real; frequency of modes, $\omega_k$ (Hz)</td>
</tr>
<tr>
<td>QMASS(NEM)</td>
<td>real; generalized mass of modes, including rotors (slug or kg)</td>
</tr>
<tr>
<td>QDAMP(NEM)</td>
<td>real; structural damping, $g_s$</td>
</tr>
<tr>
<td>DOFSYM(NEM)</td>
<td>integer; type of mode: 1 for symmetric, -1 for antisymmetric (only required for flutter analysis with OPSSIMM ne 0, tilting proprotor configuration)</td>
</tr>
<tr>
<td>real; pitch/mast-bending coupling (rad/ft or rad/m)</td>
<td></td>
</tr>
</tbody>
</table>

**KPMCl(NEM)**
- $K_{MC} = - \frac{\partial \theta_{1C}}{\partial q_S}$

**KPMS1(NEM)**
- $K_{MS} = - \frac{\partial \theta_{1S}}{\partial q_S}$

**KPMC2(NEM)**
- $K_{MC} = - \frac{\partial \theta_{1C}}{\partial q_S}$

**KPMS2(NEM)**
- $K_{MS} = - \frac{\partial \theta_{1S}}{\partial q_S}$

**ZETAR1(3,NEM)**
- rotor#1 hub

**ZETAR2(3,NEM)**
- rotor#2 hub

real; linear mode shapes at hub (ft/ft or m/m)

**GAMAR1(3,NEM)**
- rotor#1 hub

**GAMAR2(3,NEM)**
- rotor#2 hub

real; angular mode shapes at hub (rad/ft or rad/m)

**QDAMPA(NEM)**
- real; aerodynamic damping, $F_{q\theta} = \frac{\partial (Q/4\rho V^2)}{\partial (q_S/V)}$ (ft² or m²)

**QCNTRL(4,NEM)**
- real; aerodynamic control derivative, for flaperon, elevator, aileron, and rudder; $F_q\delta = \frac{\partial (Q/4\rho V^2)}{\partial \delta}$ (ft²/rad or m²/rad)
### Auxiliary forces

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAF</td>
<td>integer; number of forces; maximum 5</td>
</tr>
<tr>
<td>FSAF(NAF)</td>
<td>real; location of force on airframe (ft or m)</td>
</tr>
<tr>
<td></td>
<td>fuselage station</td>
</tr>
<tr>
<td>BLAF(NAF)</td>
<td>buttline</td>
</tr>
<tr>
<td>WLAF(NAF)</td>
<td>waterline</td>
</tr>
<tr>
<td>AZAF(NAF)</td>
<td>real; orientation of force (deg)</td>
</tr>
<tr>
<td></td>
<td>azimuth (positive counterclockwise from negative x axis)</td>
</tr>
<tr>
<td>ELAF(NAF)</td>
<td>elevation (positive upward from x-y plane)</td>
</tr>
<tr>
<td>AUXSYM(NAF)</td>
<td>integer; type of force: 1 for symmetric, -1 for antisymmetric (only required for flutter analysis with OPSYMM ne 0, tilting proprotor configuration)</td>
</tr>
<tr>
<td>ZETAAF(3,NEM,NAF)</td>
<td>real; airframe elastic modes, linear mode shape at point of application of force (ft/ft or m/m)</td>
</tr>
</tbody>
</table>
Control system

TCIN integer; control matrix input option
   0 calculate (using swashplate gains and phases)
   1 input (TCNTRL)

TCNTRL(11+NAF,5) real; control matrix, $T_C$

swashplate gains (K's, deg rotor or airframe control per deg pilot's control) and phase angles ($\Delta \psi$'s, deg of azimuth)

real; one rotor configuration, or single main rotor and tail rotor configuration

$K_0$, collective stick to collective pitch

$K_C$, lateral cyclic stick to cyclic pitch

$K_S$, longitudinal cyclic stick to cyclic pitch

$K_P$, pedal to tail rotor collective pitch

$\Delta \psi_C$, lateral cyclic stick to cyclic pitch

$\Delta \psi_s$, longitudinal cyclic stick to cyclic pitch

real; tilting proprotor configuration

$K_0$, collective stick to collective pitch

$K_C$, lateral cyclic stick to differential collective pitch

$K_S$, longitudinal cyclic stick to cyclic pitch

$K_P$, pedal to differential cyclic pitch

$\Delta \psi_s$, longitudinal cyclic stick to cyclic pitch

$\Delta \psi_P$, pedal to differential cyclic pitch
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KOCFE</strong></td>
<td>$K_o$, collective stick to collective pitch</td>
</tr>
<tr>
<td><strong>KCCFE</strong></td>
<td>$K_c$, lateral cyclic stick to cyclic pitch</td>
</tr>
<tr>
<td><strong>KSCFE</strong></td>
<td>$K_s$, longitudinal cyclic stick to cyclic pitch</td>
</tr>
<tr>
<td><strong>KPCFE</strong></td>
<td>$K_p$, pedal to differential collective pitch</td>
</tr>
<tr>
<td><strong>PCCFE</strong></td>
<td>$\Delta \psi_c$, lateral cyclic stick to cyclic pitch</td>
</tr>
<tr>
<td><strong>PSCFE</strong></td>
<td>$\Delta \psi_s$, longitudinal cyclic stick to cyclic pitch</td>
</tr>
<tr>
<td><strong>KFOCFE</strong></td>
<td>$K_{F_0}$, collective stick to front collective pitch</td>
</tr>
<tr>
<td><strong>KROCFE</strong></td>
<td>$K_{R_0}$, collective stick to rear collective pitch</td>
</tr>
<tr>
<td><strong>KFCCFE</strong></td>
<td>$K_{F_c}$, lateral cyclic stick to front cyclic pitch</td>
</tr>
<tr>
<td><strong>KRCCFE</strong></td>
<td>$K_{R_c}$, lateral cyclic stick to rear cyclic pitch</td>
</tr>
<tr>
<td><strong>KFSCFE</strong></td>
<td>$K_{F_s}$, longitudinal cyclic stick to front collective pitch</td>
</tr>
<tr>
<td><strong>KRSCFE</strong></td>
<td>$K_{R_s}$, longitudinal cyclic stick to rear collective pitch</td>
</tr>
<tr>
<td><strong>KFPCE</strong></td>
<td>$K_{F_p}$, pedal to front cyclic pitch</td>
</tr>
<tr>
<td><strong>KRPCE</strong></td>
<td>$K_{R_p}$, pedal to rear cyclic pitch</td>
</tr>
<tr>
<td><strong>PFCCFE</strong></td>
<td>$\Delta \psi_{F_c}$, lateral cyclic stick to front cyclic pitch</td>
</tr>
<tr>
<td><strong>PRCCFE</strong></td>
<td>$\Delta \psi_{R_c}$, lateral cyclic stick to rear cyclic pitch</td>
</tr>
<tr>
<td><strong>FPFCFE</strong></td>
<td>$\Delta \psi_{F_p}$, pedal to front cyclic pitch</td>
</tr>
<tr>
<td><strong>PRPCFE</strong></td>
<td>$\Delta \psi_{R_p}$, pedal to rear cyclic pitch</td>
</tr>
</tbody>
</table>
real; aircraft controls (all configurations)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KFCFE</td>
<td>$K_f$, collective stick to flaperon</td>
</tr>
<tr>
<td>KECEF</td>
<td>$K_e$, longitudinal cyclic stick to elevator</td>
</tr>
<tr>
<td>KACF</td>
<td>$K_a$, lateral cyclic tick to ailerons</td>
</tr>
<tr>
<td>KRCF</td>
<td>$K_r$, pedal to rudder</td>
</tr>
<tr>
<td>KTCF</td>
<td>$K_t$, collective stick to engine</td>
</tr>
<tr>
<td>KTTCF</td>
<td>$K_{tt}$, throttle to engine</td>
</tr>
<tr>
<td>KATCF</td>
<td>$K_{at}$, throttle to auxiliary thrust force</td>
</tr>
<tr>
<td>KAPCF</td>
<td>$K_{ap}$, pedal to auxiliary torque force</td>
</tr>
</tbody>
</table>

CNTRLZ(11) real; rotor and aircraft control inputs (deg) with all sticks centered ($v_p = 0$)

FORCEZ(NAF) real; auxiliary forces (lb or N) with all sticks centered ($v_p = 0$)

Aircraft aerodynamic characteristics

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPBAT</td>
<td>integer; airframe aerodynamic coefficients table</td>
</tr>
<tr>
<td></td>
<td>0 not use</td>
</tr>
<tr>
<td></td>
<td>1 read file</td>
</tr>
<tr>
<td>OPDRV</td>
<td>integer; airframe stability derivatives (flutter, flight dynamics, and transient analyses)</td>
</tr>
<tr>
<td></td>
<td>0 not use</td>
</tr>
<tr>
<td></td>
<td>1 use namelist parameters</td>
</tr>
<tr>
<td></td>
<td>2 read file</td>
</tr>
<tr>
<td>OPDRVU</td>
<td>integer; units of angle-of-attack, sideslip, and control derivatives (namelist or file); not used if OPDRV=0</td>
</tr>
<tr>
<td></td>
<td>0 per-radian</td>
</tr>
<tr>
<td></td>
<td>1 per-degree</td>
</tr>
</tbody>
</table>
Nonlinear aerodynamic model (equations)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>real; wing-body angles</td>
<td></td>
</tr>
<tr>
<td>incidence angle, $i_{WB}$ (deg)</td>
<td></td>
</tr>
<tr>
<td>incidence angle for drag, $i_{WBd}$ (deg)</td>
<td></td>
</tr>
<tr>
<td>maximum angle-of-attack (deg)</td>
<td></td>
</tr>
<tr>
<td>real; wing-body lift</td>
<td></td>
</tr>
<tr>
<td>derivative with angle-of-attack, $L_\alpha/q$ (ft$^2$/rad or m$^2$/rad)</td>
<td></td>
</tr>
<tr>
<td>derivative with flap, $L_{\delta_F}/q$ (ft$^2$/rad or m$^2$/rad)</td>
<td></td>
</tr>
<tr>
<td>derivative with flaperon, $L_{\delta_f}/q$ (ft$^2$/rad or m$^2$/rad)</td>
<td></td>
</tr>
<tr>
<td>real; wing-body drag</td>
<td></td>
</tr>
<tr>
<td>base value, $f_{WB} = D_0/q$ (ft$^2$ or m$^2$)</td>
<td></td>
</tr>
<tr>
<td>vertical drag, $f_{vert}$ (ft$^2$ or m$^2$)</td>
<td></td>
</tr>
<tr>
<td>induced drag, $\delta(D_1/q)/\delta(L/q)^2$ (ft$^2$ or m$^2$)</td>
<td></td>
</tr>
<tr>
<td>derivative with flap, $D_{o\delta_F}/q$ (ft$^2$/rad or m$^2$/rad)</td>
<td></td>
</tr>
<tr>
<td>derivative with flaperon, $D_{o\delta_f}/q$ (ft$^2$/rad or m$^2$/rad)</td>
<td></td>
</tr>
<tr>
<td>real; wing-body pitch moment</td>
<td></td>
</tr>
<tr>
<td>base value, $M_0/q$ (ft$^3$ or m$^3$)</td>
<td></td>
</tr>
<tr>
<td>derivative with angle-of-attack, $M_\alpha/q$ (ft$^3$/rad or m$^3$/rad)</td>
<td></td>
</tr>
<tr>
<td>derivative with flap, $M_{\delta_F}/q$ (ft$^3$/rad or m$^3$/rad)</td>
<td></td>
</tr>
<tr>
<td>derivative with flaperon, $M_{\delta_f}/q$ (ft$^3$/rad or m$^3$/rad)</td>
<td></td>
</tr>
</tbody>
</table>
real; wing-body side force
derivative with sideslip, \( Y_\beta/q \)
\((\text{ft}^2/\text{rad} \text{ or } \text{m}^2/\text{rad})\)

SIDEP
real; wing-body roll moment
derivative with rolling, \( Y_\alpha/q \)
\((\text{ft}^2/\text{rad} \text{ or } \text{m}^2/\text{rad})\)

SIDER
real; wing-body yaw moment
derivative with sideslip, \( N_x \beta/q \)
\((\text{ft}^2/\text{rad} \text{ or } \text{m}^2/\text{rad})\)

SIDEA
derivative with rolling, \( V_{Yp}/q \)
\((\text{ft}^2/\text{rad} \text{ or } \text{m}^2/\text{rad})\)

derivative with yawing, \( V_{Yr}/q \)
\((\text{ft}^2/\text{rad} \text{ or } \text{m}^2/\text{rad})\)

derivative with aileron, \( Y_\delta_a/q \)
\((\text{ft}^2/\text{rad} \text{ or } \text{m}^2/\text{rad})\)

derivative with rolling, \( V_{Nxp}/q \)
\((\text{ft}^4/\text{rad} \text{ or } \text{m}^4/\text{rad})\)

derivative with yawing, \( V_{Nx}/q \)
\((\text{ft}^4/\text{rad} \text{ or } \text{m}^4/\text{rad})\)

derivative with aileron, \( N_{x\delta_a}/q \)
\((\text{ft}^2/\text{rad} \text{ or } \text{m}^2/\text{rad})\)

derivative with rolling, \( V_{Nzp}/q \)
\((\text{ft}^4/\text{rad} \text{ or } \text{m}^4/\text{rad})\)

derivative with yawing, \( V_{Nz}/q \)
\((\text{ft}^4/\text{rad} \text{ or } \text{m}^4/\text{rad})\)

derivative with aileron, \( N_{z\delta_a}/q \)
\((\text{ft}^2/\text{rad} \text{ or } \text{m}^2/\text{rad})\)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFTAHL</td>
<td>real; horizontal tail lift derivative with angle of attack, ( L_\alpha/q ) (ft²/rad or m²/rad)</td>
</tr>
<tr>
<td>LFTEH</td>
<td>derivative with elevator, ( L_{\delta e}/q ) (ft²/rad or m²/rad)</td>
</tr>
<tr>
<td>AMAXH</td>
<td>maximum angle of attack (deg)</td>
</tr>
<tr>
<td>IHT</td>
<td>incidence angle, ( i_{HT} ) (deg)</td>
</tr>
<tr>
<td>LFTAUV</td>
<td>real; vertical tail lift derivative with angle of attack, ( L_\alpha/q ) (ft²/rad or m²/rad)</td>
</tr>
<tr>
<td>LFTRV</td>
<td>derivative with rudder, ( L_{\delta r}/q ) (ft²/rad or m²/rad)</td>
</tr>
<tr>
<td>AMAXV</td>
<td>maximum angle of attack (deg)</td>
</tr>
<tr>
<td>IVT</td>
<td>incidence angle, ( i_{VT} ) (deg)</td>
</tr>
<tr>
<td>OPTINT</td>
<td>integer; 0 to suppress airframe/tail aerodynamic interference (( \epsilon = 0 ) and ( \sigma = 0 ))</td>
</tr>
<tr>
<td>FETAIL</td>
<td>real; airframe/tail interference tail angle of attack change, ( (\partial \epsilon/(\partial (L/q)))^{-1} ) (ft or m)</td>
</tr>
<tr>
<td>LHTAIL</td>
<td>horizontal tail length ( l_{HT} ) (ft or m)</td>
</tr>
<tr>
<td>HVTAIL</td>
<td>vertical tail height ( h_{VT} ), positive up (ft or m)</td>
</tr>
</tbody>
</table>
Stability derivative aerodynamic model

SDAREA real; reference area $S$ for coefficients (ft$^2$ or m$^2$)
SDCORD real; reference chord $c$ for coefficients (ft or m)
SDSPAN real; reference span $b$ for coefficients (ft or m)

The references $S$, $c$, and $b$ are only used with the following stability derivatives. If $S = 1$, $c = 1$, and $b = 1$ are used, the derivatives are in the form of force/q or moment/q (ft or m to some power).

CLA real; airframe lift coefficient derivative with angle of attack, $C_{\alpha}$
CLM derivative with Mach number, $C_{M}$
CLAD derivative with $\dot{\omega}$, $C_{\dot{\alpha}}$
CLQ derivative with pitch rate, $C_{\dot{q}}$
CLDE derivative with elevator, $C_{\delta_e}$
CLDF derivative with flaperon, $C_{\delta_f}$

CDA real; airframe drag coefficient derivative with angle of attack, $C_{\alpha}$
CDM derivative with Mach number, $C_{M}$
CDAD derivative with $\dot{\omega}$, $C_{\dot{\alpha}}$
CDQ derivative with pitch rate, $C_{\dot{q}}$
CDDE derivative with elevator, $C_{\delta_e}$
CDDF derivative with flaperon, $C_{\delta_f}$
CMA  real; airframe pitch moment coefficient derivative with angle of attack, $C_{M\alpha}$

CMM  derivative with Mach number, $C_{M_M}$

CMAD  derivative with $\dot{\theta}$, $C_{M\dot{\alpha}}$

CMQ  derivative with pitch rate, $C_{Mq}$

CMDE  derivative with elevator, $C_{M\delta_e}$

CMDF  derivative with flaperon, $C_{M\delta_f}$

\[
C_L = \frac{\text{(lift)}}{(qS)}
\]

\[
C_D = \frac{\text{(drag)}}{(qS)}
\]

\[
C_M = \frac{\text{(pitch moment)}}{(qSc)}
\]

\[
(C)_{\dot{\alpha}} = \frac{\partial C}{\partial (\dot{\alpha})} \left( \frac{\dot{\alpha}c}{2V} \right)
\]

\[
(C)_{\dot{q}} = \frac{\partial C}{\partial (\dot{q})} \left( \frac{qc}{2V} \right)
\]

CYB  real; airframe side force coefficient derivative with sideslip, $C_{Y\beta}$

CYP  derivative with roll rate, $C_{Y_p}$

CYR  derivative with yaw rate, $C_{Y_r}$

CYDA  derivative with aileron, $C_{Y\delta_a}$

CYDR  derivative with rudder, $C_{Y\delta_r}$

CLB  real; airframe roll moment coefficient derivative with sideslip, $C_{L\beta}$

CLP  derivative with roll rate, $C_{L_p}$

CLR  derivative with yaw rate, $C_{L_r}$

CLDA  derivative with aileron, $C_{L\delta_a}$

CLDR  derivative with rudder, $C_{L\delta_r}$
real; airframe yaw moment coefficient derivative with sideslip, \( C_{n\beta} \)

dervative with roll rate, \( C_{np} \)

\( C_{n} \) = derivate with yaw rate, \( C_{nr} \)

\( C_{\gamma} \) = derivative with aileron, \( C_{n\delta a} \)

\( C_{n\delta r} \) = derivative with rudder, \( C_{n\delta r} \)

\( C_y \) = (side force)/(qS)

\( C_I \) = (roll moment)/(qSb)

\( C_n \) = (yaw moment)/(qSb)

(\( C_p \))\(_p \) = \( \partial C/\partial (pb/2V) \)

(\( C_r \))\(_r \) = \( \partial C/\partial (rb/2V) \)

Airframe/rotor aerodynamic interference

\textbf{OPINTV(2)}

integer; analysis control: 0 for no interference; 1 for velocities read from file; 2 for calculated velocities

\textbf{OPINTV(1) } rotor#1

\textbf{OPINTV(2) } rotor#2

\textbf{OPI1BP(4)}

integer; blade position model for interference calculation, rotor#1

\textbf{OPI1BP(1) } eq 0 to suppress inplane motion
\textbf{OPI1BP(2) } eq 0 to suppress all harmonics except mean
\textbf{OPI1BP(3) } eq 0 for linear from root to tip
\textbf{OPI1BP(4) } eq 0 to suppress twist in aerodynamic center offset

\textbf{OPI2BP(4)}

integer; blade position model for interference calculation, rotor#2

\textbf{OPI2BP(1) } eq 0 to suppress inplane motion
\textbf{OPI2BP(2) } eq 0 to suppress all harmonics except mean
\textbf{OPI2BP(3) } eq 0 for linear from root to tip
\textbf{OPI2BP(4) } eq 0 to suppress twist in aerodynamic center offset
Airframe/rotor aerodynamic interference -- wings

NWING
integer; number of wings; maximum 5

real; three points defining wing quarter-chord line: left tip, middle, right tip; ft or m

FSWING(3,NWING)
fuselage station, positive aft

BLWING(3,NWING)
butline, positive right

WLWING(3,NWING)
waterline, positive up

WCIRC(NWING)
real; baseline wing bound circulation,
\( \Gamma_0 / V = (L/q)/(2b_0) = c_w C_L / 2 \); ft or m

WCIRCF(3,NWING)
real; factor on contribution of wing-body, horizontal tail, and vertical tail L/q's to wing bound circulation \( \Gamma / V \)

WAXS(NWING)
real; wing airfoil cross-section area; ft\(^2\) or m\(^2\)

WRCORE(NWING)
real; horseshoe vortex core size, fraction of wing span; \( \leq 0 \). for default = .05

WXCIRC(NWING)
real; position of circulation line behind wing leading edge, fraction of wing chord

WXTHCK(NWING)
real; position of thickness line behind wing leading edge, fraction of wing chord

WISPAN(NWING)
real; reference span for circulation; ft or m

WICORD(NWING)
real; reference chord for circulation; ft or m
Airframe/rotor aerodynamic interference -- bodies

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBODY</td>
<td>integer; number of bodies; maximum 5</td>
</tr>
<tr>
<td>FSBODY(NBODY)</td>
<td>real; position of body center; ft or m</td>
</tr>
<tr>
<td>BLBODY(NBODY)</td>
<td>fuselage station, positive aft</td>
</tr>
<tr>
<td>WLBODY(NBODY)</td>
<td>buttline, positive right</td>
</tr>
<tr>
<td></td>
<td>waterline, positive up</td>
</tr>
<tr>
<td>BYAW(NBODY)</td>
<td>real; orientation of body axis relative to fuselage coordinates; deg</td>
</tr>
<tr>
<td>BPITCH(NBODY)</td>
<td>yaw angle $\psi_A$, positive right</td>
</tr>
<tr>
<td></td>
<td>pitch angle $\theta_A$, positive up</td>
</tr>
<tr>
<td>BLNTH(NBODY)</td>
<td>real; body length; ft or m</td>
</tr>
<tr>
<td>BTHICK(NBODY)</td>
<td>real; body thickness ratio (maximum thickness divided by length)</td>
</tr>
<tr>
<td>BSHAPE(NBODY)</td>
<td>integer; body shape: 1 for ellipsoid (sphere if thickness ratio ge 1.); 2 for sphere; 3 for airfoil-shaped body of revolution</td>
</tr>
</tbody>
</table>
Engine and drive train model

**ENGPOS** integer; drive train configuration
- 0 one rotor
- 1 asymmetric, engine by rotor#1
- 2 asymmetric, engine by rotor#2
- 3 symmetric

**THRTLC** real; engine power/throttle derivative $\frac{\partial P_E}{\partial \theta_t}$ (dimensional); for both engines if ENGPOS=3

if the throttle variable $\theta_t$ is only used for the governor, then just the products

$$K_P \frac{\partial P_E}{\partial \theta_t} = -\frac{\partial P}{\partial \psi_s}$$

$$K_I \frac{\partial P_E}{\partial \theta_t} = -\frac{\partial P}{\partial \psi_s}$$

must be correct ($P = \Omega_R Q_R = \Omega_E Q_E$), where $K_P$ and $K_I$ are the throttle governor gains

**KEDAMP** real; engine damping factor $\kappa$; typically 1.0
for turboshaft engines, or 10. for induction electric motors

**IENG** real; engine rotational inertia $r_E^2 I_E$ (slug-ft$^2$ or kg-m$^2$); for both engines if ENGPOS=3

real; drive train spring constants (ft-lb/rad or m-N/rad)

**KMAST1**
- rotor#1 shaft, $K_{M1}$ or $K_M$

**KMAST2**
- rotor#1 shaft, $K_{M2}$

**KICS**
- interconnect shaft, $r_{I2}^2 K_I$ or $r_I^2 K_I$

**KENG**
- engine shaft, $r_E^2 K_E$

**GSE** real; engine shaft structural damping $g_s$ ($\psi_e$ degree of freedom)

**GSI** real; interconnect shaft structural damping $g_s$ ($\psi_I$ degree of freedom)

159
Governor parameters

real; governor proportional feedback gains (sec)
throttle, $K_p = -\delta \theta_t / \delta \psi_s$

KPGOVE
KPGOV1 rotor#1 collective, $K_p = \delta \theta / \delta \psi_s$
KPGOV2 rotor#2 collective, $K_p = \delta \theta / \delta \psi_s$

real; governor integral feedback gains
throttle, $K_I = -\delta \theta / \delta \psi_s$

KIGOVE
KIGOV1 rotor#1 collective, $K_I = \delta \theta / \delta \psi_s$
KIGOV2 rotor#2 collective, $K_I = \delta \theta / \delta \psi_s$

real; governor time lag $\tau_1 = 2\zeta / \omega_n$ (sec)

T1GOVE throttle
T1GOV1 rotor#1 collective
T1GOV2 rotor#2 collective

real; governor time lag $\tau_2 = 1 / \omega_n^2$ (sec$^2$)

T2GOVE throttle
T2GOV1 rotor#1 collective
T2GOV2 rotor#2 collective
6.3.6  Namelist NLLOAD

Airframe sensors

**MVIB**  integer; number of sensors; maximum 30

**TYPEV(MVIB)**  integer; sensor type

1  airframe accelerometer
2  airframe angular velocity
3  body angular displacement
4  body angular rate
5  body angular acceleration
6  airframe air velocity

**LOCATV(MVIB)**  integer; location number; 1 to NRVIB

**AXISV(MVIB)**  integer; coordinate system (for TYPEV = 1-5)

0  fuselate (F) axes
1  stability (V) axes
or normalization (for TYPEV = 6)
0  divided by flight speed
1  divided by rotor tip speed

**COMPV(MVIB)**  integer; sensor component: 1 for x, 2 for y, 3 for z

**SENSYM(MVIB)**  integer; type of sensor: 1 for symmetric, -1 for antisymmetric, 0 for both (only required for tilting prorotor configuration, CONFIG=3)

**AZMUTV(MVIB)**  real; azimuth angle of coordinates (rotation to right about z axis); deg

**ELVATV(MVIB)**  real; elevation angle of coordinates (rotation up about y axis); deg
Location of airframe sensors

NRVIB integer; number of locations; maximum 30

real; airframe position (ft or m)

FSVIB(NRVIB) fuselage station
BLVIB(NRVIB) buttline
WLVIB(NRVIB) waterline

ZETAV(3,NEM,NRVIB) real; linear mode shape (ft/ft or m/m)
GAMAV(3,NEM,NRVIB) real; angular mode shape (rad/ft or rad/m)

Airframe vibratory response

MVLOAD integer; number of sensors for which harmonic vibration calculated; le 0 for none

OPDRES integer; sensor response: 0 for dimensionless, ne 0 for dimensional

NVPRNT integer; if ne 0, print vibratory response

NVFILE integer; if ne 0, write vibratory response to plot file
Wake geometry

MWAKE integer; number of azimuth stations at which wake geometry output, maximum MPSI; le 0 for none

NWAKE(4) integer; printer-plot selection; 0 to suppress

NWAKE(1) top view
NWAKE(2) side view
NWAKE(3) back view
NWAKE(4) vertical convection

JWAKE(MWAKE) integer; azimuth stations for which wake geometry is output ($\psi = JWAKE \times \Delta\psi$)

NCPLOT integer; if ne 0, printer-plot wake geometry

NGFILE integer; if ne 0, write wake geometry to plot file

Output format

NPSI integer; azimuth increment in all time history printed output and spanwise printer-plots (1 to MPSI); 1 for all azimuths, MPSI for only one azimuth; 0 or MPSI+1 for just mean and half peak-to-peak loads

NPOLAR integer; parameter n in polar printer-plot format: symbol printed if value/increment multiple of n (typically n = 2)

NSSPAN integer; number of radial stations in spanwise printer-plot (equally spaced from r = 0. to 1., linear interpolation); le 0 for aerodynamic radial stations
Blade motion and rotor aerodynamics (function of azimuth)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDLOAD</td>
<td>integer; output control; le 0 to suppress</td>
</tr>
<tr>
<td>MZHARMD</td>
<td>integer; number of harmonics in output; maximum 30; lt 0 for no harmonic analysis</td>
</tr>
<tr>
<td>NDPRNT(5)</td>
<td>integer; print&lt;br&gt;0 for none&lt;br&gt;1 for time history&lt;br&gt;2 for harmonics&lt;br&gt;3 for both</td>
</tr>
<tr>
<td>NDPLT(5)</td>
<td>integer; printer-plot&lt;br&gt;0 for none&lt;br&gt;1 for time history</td>
</tr>
<tr>
<td>NDFILE(5)</td>
<td>integer; write to plot file&lt;br&gt;0 for none&lt;br&gt;1 for time history&lt;br&gt;2 for harmonics&lt;br&gt;3 for both</td>
</tr>
</tbody>
</table>

print, printer-plot, file write options

1. bending modes, gimbal/teeter, and total tip deflection
2. pitch and torsion modes, pitch control, and total tip pitch
3. linear and angular hub motion, rotational speed perturbation
4. maximum bound circulation
5. induced velocity at wing-body, horizontal tail, vertical tail, off rotor, and other rotor hub
Rotor aerodynamics (blade section; function of \( \psi \) and \( r \))

**MALOAD**
integer; output control; le 0 to suppress

**MHAHRMA**
integer; number of harmonics in output; maximum 30; lt 0 for no harmonic analysis

**DAPLOT**
real; polar printer-plot increment; plot last digit of integer part of value/increment (if multiple of NPOLAR); 0. for default value

**NAPRNT**
integer; print
  0 for none
  1 for time history
  2 for harmonics
  3 for both

**NAPLOT**
integer; printer-plot; 0 for none; sum:
  1 for time history
  + 2 for spanwise
  + 4 for polar

**NAFILE**
integer; write to plot file
  0 for none
  1 for time history
  2 for harmonics
  3 for both

print, printer-plot, file write options

- **aerodynamic environment**
  - angle of attack, \( \alpha \)
  - Mach number, \( M \)
  - yaw angle, \( \Lambda \)

- **section coefficients**
  - lift, \( c_l \)
  - drag, \( c_d \)
  - moment, \( c_m \)
  - radial drag, \( c_{d_r} \)

- **section velocity**
  - perpendicular, \( u_p \)
  - tangential, \( u_T \)
  - radial, \( u_r \)
  - resultant, \( U \)
  - pitch angle, \( \theta \)
(14) inflow angle, \( \phi \)
displacement

(15) lag

(16) flap
effective angle of attack, \( \alpha_{eff} \)

(17) lift

(18) drag

(19) moment
effective Mach number, \( M_{eff} \)

(20) lift

(21) drag

(22) moment

(23) angle of attack rate, \( \dot{\alpha}_c/V \)

(24) cosine of yaw angle, \( \cos \Lambda \)
dynamic stall state

(25) lift

(26) drag

(27) moment
peak dynamic stall vortex load

(28) lift

(29) drag

(30) moment
dynamic stall vortex load increment

(31) lift

(32) drag

(33) moment
induced velocity

(34) \( \lambda_x \), longitudinal

(35) \( \lambda_y \), lateral

(36) \( \lambda_z \), vertical
rotor interference induced velocity

(37) \( \lambda_x \), longitudinal

(38) \( \lambda_y \), lateral

(39) \( \lambda_z \), vertical
gust velocity

(40) \( u_G \)

(41) \( v_G \)

(42) \( w_G \)
dimensionless section loads

(43) lift, \( L/c \)

(44) drag, \( D/c \)

(45) moment, \( M/c \)

(46) radial drag, \( D_r/c \)
dimensionless section forces

(47) inplane, \( F_x/c \)

(48) radial, \( F_r/c \)

166
(49) normal (thrust) \( F_z/c = \frac{d(C_T/\sigma)}{dr} \)
(50) pitch moment, \( M_a/c \)
(51) radial, \( F_r/c \) (shaft axes)

power coefficient

(52) total, \( \frac{d(C_p/\sigma)}{dr} \)
(53) induced, \( \frac{d(C_{p_i}/\sigma)}{dr} \)
(54) rotor interference, \( \frac{d(C_{p_{int}}/\sigma)}{dr} \)
(55) body interference, \( \frac{d(C_{p_{int}}/\sigma)}{dr} \)
(56) profile power, \( \frac{d(C_{p_o}/\sigma)}{dr} \)

dimensional section loads

(57) lift, \( L \)
(58) drag, \( D \)
(59) moment, \( M \)
(60) radial drag, \( D_r \)

dimensional section forces

(61) inplane, \( F_x \)
(62) radial, \( F_r \)
(63) normal (thrust), \( F_z = \frac{dT}{dr} \)
(64) pitch moment, \( M_z \)
(65) radial, \( F_r \) (shaft axes)

power

(66) total, \( \frac{dP}{dr} \)
(67) induced, \( \frac{dP_i}{dr} \)
(68) rotor interference, \( \frac{dP_{int}}{dr} \)
(69) body interference, \( \frac{dP_{int}}{dr} \)
(70) profile, \( \frac{dP_o}{dr} \)

input airframe interference velocity

(71) VINTX or VINTR
(72) VINTY or VINTT
(73) VINTZ or VINTP

airframe interference velocity

(74) radial, \( \Delta u_r \)
(75) tangential, \( \Delta u_t \)
(76) perpendicular, \( \Delta u_p \)

CFD interface output

(77) partial \( \alpha \)
(78) \( c_l_{tab} \)
(79) \( c_d_{tab} \)
(80) \( c_m_{tab} \)

CFD interface input

(81) \( c_{l_{old}} \)
(82) \( c_{d_{old}} \)
(83) \( c_{m_{old}} \)
(84) \( c_{l_{ext}} \)
(85) \( c_{d_{ext}} \)
(86) \( c_{m_{ext}} \)
Hub and control loads

MHLOAD
integer; output control; le 0 to suppress

MHARMH
integer; number of harmonics in output; maximum 30; lt 0 for no harmonic analysis

NHPRINT(4)
integer; print
  0 for none
  1 for time history
  2 for harmonics
  3 for both

NH PLOT(4)
integer; printer-plot
  0 for none
  1 for time history

NH FILE(4)
integer; write to plot file
  0 for none
  1 for time history
  2 for harmonics
  3 for both

print, printer-plot, file write options

(1) rotating frame root shears, bending moments, and control moment; dimensionless
(2) nonrotating frame hub reactions; dimensionless
(3) rotating frame root shears, bending moments, and control moment; dimensional
(4) nonrotating frame hub reactions, dimensional
Blade section loads

**MRLOAD**
integer; number of radial stations; maximum 20; le 0 to suppress

**RLOAD(MRLOAD)**
real; blade radial stations r/R

**OPBTC(3)**
integer; selection of blade moment calculation method; 0 for integrated forces, 1 for modal deflection
- OPBTC(1) bending moments
- OPBTC(2) torsion moment
- OPBTC(3) control load (MHLOAD)

**MHARMR**
integer; number of harmonics in output; maximum 30; lt 0 for no harmonic analysis

**NRPRNT(4)**
integer; print
- 0 for none
- 1 for time history
- 2 for harmonics
- 3 for both

**NRPLOT(4)**
integer; printer-plot
- 0 for none
- 1 for time history

**NRFIELD(4)**
integer; write to plot file
- 0 for none
- 1 for time history
- 2 for harmonics
- 3 for both

print, printer-plot, file write options

1. section forces and bending moments (rotating shaft axes); dimensionless
2. section forces, bending moments, and torsion moment (blade principal axes); dimensionless
3. section forces and bending moments (rotating shaft axes); dimensional
4. section forces, bending moments, and torsion moment (blade principal axes); dimensional

169
Far field rotational noise

**MNOISE**
integer; number of microphones; maximum 10; 0 for no noise analysis

**RANGE(MNOISE)**
real; position of microphone relative to rotor hub
range (ft or m)

**ELVATN(MNOISE)**
elevation, positive above rotor disk (deg)

**AZMUTH(MNOISE)**
azimuth, defined as for rotor azimuth (deg)

**MHARMN(3)**
integer; number of harmonics
MHARMN(1) in noise calculation;
maximum 500
MHARMN(2) in aerodynamic load analysis;
maximum 30
MHARMN(3) in print of noise harmonics

**MTIMEN(3)**
integer; number of time steps
MTIMEN(1) in period of noise calculation;
maximum 500
MTIMEN(2) increment in time history print
MTIMEN(3) increment in time history printer-plot

**MRN**
integer; number of radial stations in spanwise integration

**RROOTN**
real; inboard radial station r/R for spanwise integration

**AXS(MRA)**
real; blade cross section area $A_{xs}/c^2$ at aerodynamic segments, for thickness noise (typically 0.685 times airfoil thickness ratio)

**OPNOIS(4)**
integer; noise calculation; 0 to omit, 1 for impulsive chordwise loading, 2 for distributed chordwise loading
OPNOIS(1) lift noise
OPNOIS(2) drag noise
OPNOIS(3) radial force noise
OPNOIS(4) thickness noise
NNPRINT integer; print
  0 for none
  1 for time history
  2 for harmonics
  3 for both

NNPLOT integer; printer-plot
  0 for none
  1 for time history

NNFILE integer; write to plot file
  0 for none
  1 for time history
  2 for harmonics
  3 for both
6.3.7 Namelist NLFLUT

Flutter analysis

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPFLOW</td>
<td>integer; analysis type</td>
</tr>
<tr>
<td></td>
<td>-1 constant coefficient approximation</td>
</tr>
<tr>
<td></td>
<td>0    axial flow</td>
</tr>
<tr>
<td></td>
<td>1    periodic coefficients</td>
</tr>
<tr>
<td>OPSYMM</td>
<td>integer; ne 0 for separate analysis of symmetric and antisymmetric equations (only for tilting</td>
</tr>
<tr>
<td></td>
<td>proprotor configuration, CONFIG=3</td>
</tr>
<tr>
<td>OPFDAN</td>
<td>integer; ne 0 for flight dynamics analysis of flutter equations</td>
</tr>
<tr>
<td>NBLDFL</td>
<td>integer; 1 for independent rotor blade analysis</td>
</tr>
<tr>
<td>OPUSLD</td>
<td>integer; use of unsteady lift and moment in flutter analysis</td>
</tr>
<tr>
<td></td>
<td>0    suppress</td>
</tr>
<tr>
<td></td>
<td>1    include</td>
</tr>
<tr>
<td></td>
<td>2    include, but zero for stall</td>
</tr>
<tr>
<td>DALPHA</td>
<td>real; angle of attack increment Δα (deg) for calculation of lift, drag, and moment coefficient</td>
</tr>
<tr>
<td></td>
<td>derivatives in rotor aerodynamic coefficients</td>
</tr>
<tr>
<td>DMACH</td>
<td>real; Mach number increment ΔM/M for calculation of lift, drag, and moment coefficient</td>
</tr>
<tr>
<td></td>
<td>derivatives in rotor aerodynamic coefficients</td>
</tr>
<tr>
<td>DELTA</td>
<td>real; control and motion increment for aircraft stability derivative calculation (dimensionless)</td>
</tr>
<tr>
<td>OPDYN1</td>
<td>integer; dynamic inflow model</td>
</tr>
<tr>
<td></td>
<td>1    perturbation empirical model (using parameters in namelist NLRTR)</td>
</tr>
<tr>
<td></td>
<td>2    Pitt and Peters model</td>
</tr>
<tr>
<td>OPGRND</td>
<td>integer; ground effect analysis</td>
</tr>
<tr>
<td></td>
<td>0    out of ground effect</td>
</tr>
<tr>
<td></td>
<td>1    in ground effect</td>
</tr>
</tbody>
</table>

172
KASGE real; factor for antisymmetric ground effect model; 0. to suppress, 1. for unstable roll moment caused by ground effect (only for tilting proprotor configuration, CONFIG=3)

OPRINT integer; 0 to suppress rotor/body aerodynamic interference in flutter analysis

MPSICC integer; number of azimuth stations (per revolution) in evaluation of average coefficients for constant coefficient approximation (OPFLOW=-1); \( \Delta \psi = 360/M \)

MPSIPC integer; number of azimuth steps in period for nonaxial flow, periodic coefficient analysis (OPFLOW=1); \( \Delta \psi = 360/(N*M) \) for N odd, \( \Delta \psi = 720/(N*M) \) for N even, N = number of blades

NINTPC integer; numerical integration method for periodic coefficient analysis (OPFLOW=1)
   1 modified trapezoidal method
   2 Runge-Kutta method

OPTORS(2) integer; 0 for rigid pitch model (infinite control system stiffness, no \( p_0 \) degrees of freedom)
   OPTORS(1) rotor#1
   OPTORS(2) rotor#2

OPDRES integer; sensor response: 0 for dimensionless, ne 0 for dimensional

OPAXFV integer; airframe rigid body degrees of freedom:
   0 for fuselage (F) axes, 1 for stability (V) axes
Degrees of freedom, controls, and gust

DOF(100) integer; vector defining degrees of freedom for flutter analysis; 0 if not used, 1 if used, 2 if quasistatic variable; order:

rotor#1 $\beta_0, \beta_1, \beta_{1S}, \beta_{N/2}, \theta_0, \theta_{1C}, \theta_{1S}, \theta_{N/2}, \beta_{GC}, \beta_{GS}, \psi_S, \lambda_u, \lambda_x, \lambda_y$

rotor#2 $\beta_0, \beta_1, \beta_{1S}, \beta_{N/2}, \theta_0, \theta_{1C}, \theta_{1S}, \theta_{N/2}, \beta_{GC}, \beta_{GS}, \psi_I, \lambda_u, \lambda_x, \lambda_y$

bending (15) pitch/torsion (9) gimbal (9) rotor inflow (9) teeter speed

airframe $\phi_F, \theta_F, \psi_F, x_F, y_F, z_F, q_S, \cdots, q_{S36}, \psi_e, \Delta \theta_e, \Delta \theta_{govr1}, \Delta \theta_{govr2}$

rigid body flexible engine governor body (30) speed

maximum number of bending degrees of freedom = 15;
maximum number of pitch/torsion degrees of freedom = 9;
maximum number of elastic airframe degrees of freedom = 30;
for two-bladed rotor, $\beta_{GC}$ is replaced by $\beta_T$

CON(31) integer; vector defining control variables; ne 0 if used; order:

rotor#1 $\theta_0, \theta_{1C}, \theta_{1S}, \cdots, \theta_{N/2}$ pitch (8)

rotor#2 $\theta_0, \theta_{1C}, \theta_{1S}, \cdots, \theta_{N/2}$ pitch (8)

airframe $\delta_f, \delta_e, \delta_a, \delta_r, \theta_t$

aux forces $f_1, f_2, f_3, f_4, f_5$

pilot $\delta_o, \delta_c, \delta_s, \delta_p, \delta_t$

number of pitch controls = number of blades (maximum 8);
for two-bladed rotor, pitch controls are $\theta_0, \theta_{1C}, \theta_{1S}, \theta_{1}$

GUS(3) integer; vector defining gust components; ne 0 if used; order: $u_G, v_G, w_G$
Sensors

SEN(MVIB) integer; vector defining sensor output for response analysis; ne 0 if used; sensors defined by parameters in namelist NLOAD

Output selected from degrees of freedom, controls, and gust

OUTX(100) integer; vector defining displacement output for response analysis; ne 0 if used; same definition and order as DOF; displacement of first order states not available

OUTDX(100) integer; vector defining velocity output for response analysis; ne 0 if used; same definition and order as DOF

OUTDDX(100) integer; vector defining acceleration output for response analysis; ne 0 if used; same definition and order as DOF

OUTV(34) integer; vector defining control and gust variables as output for response analysis; ne 0 if used; same definition and order as CON+GUS

for tiltrotor configuration with OPSYMM ne 0, rotor#1/rotor#2 are symmetric/antisymmetric in the output definitions
Linear system analysis

ANTYPE(4) integer; linear system analysis tasks; 0 to suppress
   ANTYPE(1) eigenanalysis
   ANTYPE(2) frequency response
   ANTYPE(3) time history
   ANTYPE(4) rms gust response

NLFILE integer; if ne 0, write eigenvalues to plot file, for ANTYPE(1) ne 0

NBFILE integer; if ne 0, write frequency response to plot file; for ANTYPE(2) ne 0

NTFILE integer; if ne 0, write time history response to plot file; for ANTYPE(3) ne 0

NMFILE integer; write matrices to plot file
   0 none
   1 first order form
   2 second order form
   3 both

Eigenanalysis, ANTYPE(1)

NSYSAN integer; calculation control
   0 eigenvalues
   1 eigenvalues and eigenvectors
   10 eigenvalues and zeros
   11 eigenvalues, eigenvectors, and zeros
Frequency response, ANTYPE(2)

OPSTEP integer; if ne 0, static response calculated and printed

NFREQ integer; number of frequencies for which frequency response calculated and printed; le 0 to suppress; maximum 100

FREQ(NFREQ) real; frequencies for which response calculated (NFREQ gt 0), per rev

OPBODE integer; frequency response using calculated scales (FOPLT,FIPLT,NFOPLT,NFIPLT,SCALE)
0 suppress
1 printer-plot
2 print
3 both

MBODE integer; calculation method
1 from matrices
2 from poles and zeros
3 from modes

Time history, ANTYPE(3)

OPTIME integer; control input type
1 step
2 impulse
3 cosine impulse
4 sine doublet
5 square impulse
6 square doublet
7 ramp
8 triangular impulse

PERIOD real; period T (sec) for impulse or doublet (OPTIME = 3 to 6)

DELT real; time step (sec)

MAXT real; maximum time (sec)
Rms gust response, ANTYPE(4)

LGUST(3) real; gust correlation lengths for three gust components (longitudinal, lateral, and vertical)
    gt 0. length L (ft or m), time constant = L/V
    eq 0. L = 400. (time constant = 0.1 sec if speed V = 0)
    lt 0. magnitude is time constant (sec)

GRMS(3) real; gust component rms magnitude: absolute (divided by rotor#1 tip speed for dimensionless response) or relative (rms response is per unit gust magnitude); for three gust components (longitudinal, lateral, and vertical)

MGUST integer; calculation method
    0 stochastic (from modes)
    1 integral of transfer function, from matrices
    2 integral of transfer function, from poles and zeros
    3 integral of transfer function, from modes

OPSPEC integer; gust spectrum type: 1 for Dryden, 2 for von Karman (transfer function method only; stochastic method uses Dryden spectrum)

    transfer function frequency range and resolution defined by NF0PLT, NF1PLT, FOPLT, F1PLT, SCALE
Frequency response and gust response

**FOPLT**  
real; beginning frequency, for linear scale

**FIPLT**  
real; end frequency, for linear scale

**NF0PLT**  
integer; exponent (base 10) of beginning frequency, for log scale

**NF1PLT**  
integer; exponent (base 10) of end frequency, for log scale

**SCALE(6)**  
integer; definition of scales
- **magnitude scale**
  - **SCALE(1)** = 1 log (base 10)
  - 2 dB (20 log10)
  - 3 linear
  - **SCALE(2)** = 1 relative maximum
  - 2 relative 10**k
  - 3 relative 10.
- **frequency scale**
  - **SCALE(3)** = 1 log (base 10)
  - 2 linear
  - **SCALE(4)** = 1 per-rev
  - 2 Hz
  - 3 rad/sec
  - **SCALE(5)** = ND, frequency steps per decade
    (log scale; maximum (NF1-NF0)ND - 300) or frequency label increment (linear scale)
  - **SCALE(6)** = number of frequency increments (linear scale, maximum 300)
Aeroservoelasticity analysis

OPASE integer; analysis tasks
  0 none
  1 stability (from open loop Bode)
  2 stability and closed-loop response
     (static response, frequency
     response, or rms gust response)

the closed-loop analysis also uses the following parameters from the open-loop analysis: KFILE (#2 only), OPDRES, ANTYPE (#2 and #4), OPSTEP, NFREQ, FREQ, OPBODE, MBODE, LGUST, GRMS, MGUST (not equal to 0), OPSPEC, NF0PLT, NF1PLT, FOPLT, F1PLT, SCALE

Control system definition

MLOOP integer; number of control system loops; maximum 4

CLOSE(MLOOP) integer; order of loop closure (loop number, from 1 to MLOOP; no loop used if 0)

NFFDCL integer; number of feedforward variables; maximum 4

Feedback

NYCL(MLOOP) integer; number of output variables; maximum 40

NAMYCL(NYCL,MLOOP) character*8; names of output variables
       (left justified)

Control

NVCL(MLOOP) integer; number of input variables; maximum 40

NAMVCL(NVCL,MLOOP) character*4; names of input variables
       (left justified)

TVCL(NVCL,MLOOP) real; control matrix, relating actuator
       variable to control variables
Transfer functions -- actuator

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHACT(MLOOP)</td>
<td>integer; number of elements</td>
</tr>
<tr>
<td>IDHACT(MHACT, MLOOP)</td>
<td>integer; element identification</td>
</tr>
<tr>
<td>HREACT(MHACT, MLOOP)</td>
<td>real; real part parameter</td>
</tr>
<tr>
<td>HIMACT(MHACT, MLOOP)</td>
<td>real; imaginary part parameter</td>
</tr>
</tbody>
</table>

Transfer functions -- feedback

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHOUT(NYCL, MLOOP)</td>
<td>integer; number of elements</td>
</tr>
<tr>
<td>IDHOUT(MHOUT, NYCL, MLOOP)</td>
<td>integer; element identification</td>
</tr>
<tr>
<td>HREOUT(MHOUT, NYCL, MLOOP)</td>
<td>real; real part parameter</td>
</tr>
<tr>
<td>HIMOUT(MHOUT, NYCL, MLOOP)</td>
<td>real; imaginary part parameter</td>
</tr>
</tbody>
</table>

Transfer functions -- feedforward (3 paths)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHDEL(3, NFFDCL, MLOOP)</td>
<td>integer; number of elements</td>
</tr>
<tr>
<td>IDHDEL(MHDEL, 3, NFFDCL, MLOOP)</td>
<td>integer; element identification</td>
</tr>
<tr>
<td>HREDEL(MHDEL, 3, NFFDCL, MLOOP)</td>
<td>real; real part parameter</td>
</tr>
<tr>
<td>HIMDEL(MHDEL, 3, NFFDCL, MLOOP)</td>
<td>real; imaginary part parameter</td>
</tr>
</tbody>
</table>

MHxxx = number of elements in transfer function definition; maximum 40; 0 for an unused path

IDHxxx = vector identifying type of each element

HRExxx = vector of parameters required for elements

HIMxxx = vector of parameters required for elements each element contributes a factor to the transfer function; the convention for the element identifiers and parameters is described below
<table>
<thead>
<tr>
<th>Element</th>
<th>IDHxxx</th>
<th>HRFxxx</th>
<th>HIMxxx</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td>0</td>
<td>not used</td>
<td>none</td>
</tr>
<tr>
<td>gain</td>
<td>1</td>
<td>K</td>
<td>$e^{-s}$</td>
</tr>
<tr>
<td>lag</td>
<td>2</td>
<td>r, sec</td>
<td>not used</td>
</tr>
<tr>
<td>real pole</td>
<td>10</td>
<td>Re, sec</td>
<td>-1</td>
</tr>
<tr>
<td>complex pole</td>
<td>11</td>
<td>$\frac{(s - p)^{-1}}{(s - p_1)^{-1}(s - p_2)^{-1}}$</td>
<td>$\frac{(s - p)^{-1}}{(s - p_1)^{-1}(s - p_2)^{-1}}$</td>
</tr>
<tr>
<td>complex pole</td>
<td>12</td>
<td>$\omega_n$, sec</td>
<td>$\omega_n$, sec</td>
</tr>
<tr>
<td>real zero</td>
<td>20</td>
<td>Re, sec</td>
<td>0</td>
</tr>
<tr>
<td>complex zero</td>
<td>21</td>
<td>$\omega_n$, sec</td>
<td>$\omega_n$, sec</td>
</tr>
<tr>
<td>complex zero</td>
<td>22</td>
<td>$\omega_n$, sec</td>
<td>$\omega_n$, sec</td>
</tr>
</tbody>
</table>

**Pole/Zero**

- $p = -\frac{1}{r}$
- $p = (\text{Re}, \text{Im})$
- $z = -\frac{1}{r}$
- $z = (\text{Re}, \text{Im})$

**Library**

- NLFLUT
6.3.8 Namelist NLSTAB

Flight dynamics analysis

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPRNTP</td>
<td>integer; le 0 to suppress performance print during stability derivative calculation</td>
</tr>
<tr>
<td>NPRNTL</td>
<td>integer; le 0 to suppress loads print during stability derivative calculation</td>
</tr>
<tr>
<td>ITERS</td>
<td>integer; number of iterations between wake geometry and motion/force calculations</td>
</tr>
</tbody>
</table>
| OPLMDA | integer; induced velocity calculation  
0 update influence coefficients and inflow  
1 only update inflow  
2 update neither influence coefficients nor inflow |
| DELTA  | real; control and motion increment for stability derivative calculation (dimensionless) |
| OPPRNT(4) | integer; print of stability derivatives; 0 to suppress  
OPPRNT(1) rotor coefficient form, dimensionless  
OPPRNT(2) rotor coefficient form, dimensional  
OPPRNT(3) stability derivative form, dimensionless  
OPPRNT(4) stability derivative form, dimensional |
| OPDRES | integer; sensor response: 0 for dimensionless, ne 0 for dimensional         |
| OPAXFV | integer; airframe rigid body degrees of freedom:  
0 for fuselage (F) axes, 1 for stability (V) axes |

183
Degrees of freedom, controls, and gust

DOF(7)  integer; vector defining degrees of freedom, ne 0 if used; order: $\phi_F, \theta_F, \psi_F, x_F, y_F, z_F, \psi_S$

CON(21) integer; vector defining control variables; ne 0 if used; order:

rotor#1 $\theta_o \theta_1C \theta_1S$
rotor#2 $\theta_o \theta_1C \theta_1S$
airframe $\delta_f \delta_e \delta_a \delta_r \theta_t$
aux forces $f_1 f_2 f_3 f_4 f_5$
pilot $\delta_o \delta_c \delta_s \delta_p \delta_t$

GUS(3) integer; vector defining gust components; ne 0 if used; order: $u_G, v_G, w_G$

Sensors

SEN(MVIB) integer; vector defining sensor output for response analysis; ne 0 if used; sensors defined by parameters in namelist NLLOAD

Output selected from degrees of freedom, controls, and gust

OUTX(7) integer; vector defining displacement output for response analysis; ne 0 if used; same definition and order as DOF; displacement of first order states not available

OUTDX(7) integer; vector defining velocity output for response analysis; ne 0 if used; same definition and order as DOF

OUTDDX(7) integer; vector defining acceleration output for response analysis; ne 0 if used; same definition and order as DOF

OUTV(24) integer; vector defining control and gust variables as output for response analysis; ne 0 if used; same definition and order as CON+GUS
Self-tuning regulator during flight dynamics

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTR</td>
<td>integer; maximum number of iterations on control to achieve convergence; 0 to turn off regulator</td>
</tr>
<tr>
<td>PIDSTR</td>
<td>integer; perturbation identification of T-matrix</td>
</tr>
<tr>
<td></td>
<td>0 never (use T-matrix from trim)</td>
</tr>
<tr>
<td></td>
<td>1 at start of each stability derivative iteration, then every MIDSTR regulator iterations</td>
</tr>
<tr>
<td>MIDSTR</td>
<td>integer; number of regulator iterations between perturbation identification of T-matrix; 1 is 0 for never</td>
</tr>
<tr>
<td>RIDSTR</td>
<td>integer; recursive update of T-matrix, if ne 0</td>
</tr>
<tr>
<td>EPSTR</td>
<td>real; tolerance on regulator convergence</td>
</tr>
</tbody>
</table>
Linear system analysis

ANTYPE(5) integer; linear system analysis tasks; 0 to suppress
  ANTYPE(1) eigenanalysis
  ANTYPE(2) frequency response
  ANTYPE(3) time history
  ANTYPE(4) rms gust response
  ANTYPE(5) numerical integration of transient

EQTYPE(3) integer; specification of equations to be analyzed; rotor speed included in all cases (if DOF(7) ne 0); 0 to suppress
  EQTYPE(1) complete set
  EQTYPE(2) symmetric
  EQTYPE(3) antisymmetric

NLFILE integer; if ne 0, write eigenvalues to plot file, for ANTYPE(1) ne 0

NBFILE integer; if ne 0, write frequency response to plot file; for ANTYPE(2) ne 0

NTFILE integer; if ne 0, write time history response to plot file; for ANTYPE(3) or ANTYPE(5) ne 0

NMFILE integer; write matrices to plot file
  0 none
  1 first order form
  2 second order form
  3 both

Eigenanalysis, ANTYPE(1)

NSYSAN integer; calculation control
  0 eigenvalues
  1 eigenvalues and eigenvectors
  10 eigenvalues and zeros
  11 eigenvalues, eigenvectors, and zeros
Frequency response, ANTYPE(2)

**OPSTEP**
integer; if ne 0, static response calculated and printed

**NFREQ**
integer; number of frequencies for which frequency response calculated and printed; le 0 to suppress; maximum 100

**FREQ(NFREQ)**
real; frequencies for which response calculated (NFREQ gt 0), per rev

**OPBODE**
integer; frequency response using calculated scales (F0PLT,F1PLT,N0PLT,N1PLT,SCALE)
  - 0 suppress
  - 1 printer-plot
  - 2 print
  - 3 both

**MBODE**
integer; calculation method
  - 1 from matrices
  - 2 from poles and zeros
  - 3 from modes

Time history, ANTYPE(3)

**OPTIME**
integer; control input type
  - 1 step
  - 2 impulse
  - 3 cosine impulse
  - 4 sine doublet
  - 5 square impulse
  - 6 square doublet
  - 7 ramp
  - 8 triangular impulse

**PERIOD**
real; period T (sec) for impulse or doublet (OPTIME = 3 to 6)

**DELT**
real; time step (sec)

**MAXT**
real; maximum time (sec)
Rms gust response, ANTYPE(4)

**LGUST(3)**

real; gust correlation lengths for three gust components (longitudinal, lateral, and vertical)

- $lt\ 0$. length $L$ (ft or m), time constant $= L/V$
- $eq\ 0$. $L = 400$. (time constant $= 0.1$ sec if speed $V = 0$)
- $gt\ 0$. magnitude is time constant (sec)

**CRMS(3)**

real; gust component rms magnitude: absolute (divided by rotor#1 tip speed for dimensionless response) or relative (rms response is per unit gust magnitude); for three gust components (longitudinal, lateral, and vertical)

**MGUST**

integer; calculation method

- $0$ stochastic (from modes)
- $1$ integral of transfer function, from matrices
- $2$ integral of transfer function, from poles and zeros
- $3$ integral of transfer function, from modes

**OPSPEC**

integer; gust spectrum type: $1$ for Dryden, $2$ for von Karman (transfer function method only; stochastic method uses Dryden spectrum)

transfer function frequency range and resolution defined by NF0PLT, NF1PLT, FOPLT, F1PLT, SCALE
Frequency response and gust response

**FOPLT** real; beginning frequency, for linear scale

**FIPLT** real; end frequency, for linear scale

**NF0PLT** integer; exponent (base 10) of beginning frequency, for log scale

**NF1PLT** integer; exponent (base 10) of end frequency, for log scale

**SCALE(6)** integer; definition of scales

  magnitude scale
  - **SCALE(1)** = 1 log (base 10)
  - 2 dB (20 log-10)
  - 3 linear

  **SCALE(2)** = 1 relative maximum
  - 2 relative 10.**K
  - 3 relative 10.

  frequency scale
  - **SCALE(3)** = 1 log (base 10)
  - 2 linear
  - **SCALE(4)** = 1 per-rev
  - 2 Hz
  - 3 rad/sec

  **SCALE(5)** = ND, frequency steps per decade
  (log scale; maximum (NF1-NF0)ND = 300) or frequency label;
  increment (linear scale)

  **SCALE(6)** = number of frequency increments
  (linear scale, maximum 300)
### Numerical integration of transient

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTEP</td>
<td>real; time step (sec)</td>
</tr>
<tr>
<td>TMAX</td>
<td>real; maximum time (sec)</td>
</tr>
<tr>
<td>NPRNTT</td>
<td>integer; print of transient data every NPRNTT-th time step; le 0 to suppress</td>
</tr>
<tr>
<td>NTPLT</td>
<td>integer; if ne 0, printer-plot time history of system response</td>
</tr>
</tbody>
</table>

Parameters defining prescribed controls and gust; see namelist NLTRAN for description

OPTRAN
OPHIST
CTIME
CMAG(5)
GTIME
CMAG(3)
GDIST(2)
VFLG
PSIG
OPGUST(3)
Aeroservoelasticity analysis

<table>
<thead>
<tr>
<th>OPASE</th>
<th>integer; analysis tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>1</td>
<td>stability (from open loop Bode)</td>
</tr>
<tr>
<td>2</td>
<td>stability and closed-loop response</td>
</tr>
<tr>
<td></td>
<td>(static response, frequency</td>
</tr>
<tr>
<td></td>
<td>response, or rms gust response)</td>
</tr>
</tbody>
</table>

the closed-loop analysis also uses the following parameters from the open-loop analysis: KFILE (#2 only), OPDRES, ANTYPE (#2 and #4), OPSTEP, NFFREQ, FREQ, OPBODE, MBODE, LGUST, GRMS, MGUST (not equal to 0), OPSPEC, NF0PLT, NF1PLT, F0PLT, F1PLT, SCALE

parameters defining control system for aeroservoelasticity analysis; see namelist NLFLUT for description

MLOOP
CLOSE(MLOOP)
NFFDCL
NYCL(MLOOP)
NAMYCL(NYCL,MLOOP)
NVCL(MLOOP)
NAMVCL(NVCL,MLOOP)
TVCL(NVCL,MLOOP)
MHACT(MLOOP)
IDHACT(MHACT,MLOOP)
HREACT(MHACT,MLOOP)
HIMACT(MHACT,MLOOP)
MHOUT(NYCL,MLOOP)
IDHOUT(MHOUT,NYCL,MLOOP)
HREOUT(MHOUT,NYCL,MLOOP)
HIMOUT(MHOUT,NYCL,MLOOP)
MHDEL(3,NFFDCL,MLOOP)
IDHDEL(MHDEL,3,NFFDCL,MLOOP)
HREDEL(MHDEL,3,NFFDCL,MLOOP)
HIMDEL(MHDEL,3,NFFDCL,MLOOP)
6.3.9 Namelist NLTRAN

Transient analysis

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPRNTP</td>
<td>integer; le 0 to suppress performance print during transient</td>
</tr>
<tr>
<td>NPRNTL</td>
<td>integer; le 0 to suppress loads print during transient</td>
</tr>
<tr>
<td>ITERT</td>
<td>integer; number of iterations between wake geometry and motion/force calculations</td>
</tr>
<tr>
<td>OPLMDA</td>
<td>integer; induced velocity calculation</td>
</tr>
<tr>
<td></td>
<td>0 update influence coefficients and inflow</td>
</tr>
<tr>
<td></td>
<td>1 only update inflow</td>
</tr>
<tr>
<td></td>
<td>2 update neither influence coefficients nor inflow</td>
</tr>
<tr>
<td>OPDRES</td>
<td>integer; sensor response: 0 for dimensionless, ne 0 for dimensional</td>
</tr>
<tr>
<td>OPAXFV</td>
<td>integer; airframe rigid body degrees of freedom: 0 for fuselage (F) axes, 1 for stability (V) axes</td>
</tr>
</tbody>
</table>

Degrees of freedom

| DOF(7)    | integer; vector defining degrees of freedom in numerical integration; 0 to suppress acceleration; order: $\phi_F$, $\dot{\phi}_F$, $\psi_F$, $x_F$, $y_F$, $z_F$, $\psi_S$ |

Sensors

| SEN(MVIB) | integer; vector defining sensor output for response analysis; ne 0 if used; sensors defined by parameters in namelist NLLOAD |
Output selected from degrees of freedom, controls, and gust

OUTX(7) integer; vector defining displacement output for response analysis; ne 0 if used; same definition and order as DOF

OUTDX(7) integer; vector defining velocity output for response analysis; ne 0 if used; same definition and order as DOF

OUTDDX(7) integer; vector defining acceleration output for response analysis; ne 0 if used; same definition and order as DOF

OUTV(8) integer; vector defining control and gust variables as output for response analysis; ne 0 if used; order: $\delta_o$, $\delta_c$, $\delta_s$, $\delta_p$, $\delta_t$, $u_C$, $v_C$, $w_C$

Self-tuning regulator during transient

MSTR integer; number of iterations on control (4*MSTR per time step); 0 to turn off regulator

PIDSTR integer; perturbation identification of T-matrix
0 never (use T-matrix from trim)
1 at beginning of transient, then every MIDSTR regulator iterations

MIDSTR integer; number of regulator iterations between perturbation identification of T-matrix; 1e 0 for never

RIDSTR integer; recursive update of T-matrix, if ne 0
Numerical integration of transient

TSTEP  real; time step (sec)
TMAX  real; maximum time (sec)
NPRNTT  integer; print of transient, performance, and loads data every NPRNTT-th time step; le 0 to suppress

NTPLOT  integer; if ne 0, printer-plot time history of system response
NTFILE  integer; if ne 0, write time history of system response to plot file

Transient gust and control

OPTRAN  integer; transient option
  1 control
  2 uniform gust
  3 convected gust

OPHIST  integer; control or gust input type
  1 step
  2 impulse
  3 cosine impulse
  4 sine doublet
  5 square impulse
  6 square doublet
  7 ramp
  8 triangular impulse

Transient gust and control -- transient control

CTIME  real; period T (sec)
CMAG(5)  real; control magnitude (deg);
order: \( \delta_o, \delta_c, \delta_s, \delta_p, \delta_t \)
Transient gust and control -- uniform gust

GTIME  real; period T (sec)

GMAG(3)  real; gust magnitude (ft/sec or m/sec);
          order: $u_G, v_G, w_G$

Transient gust and control -- convected gust

GDIST(2)  real; lengths for convected gust (ft or m)
          GDIST(1)  wavelength L
          GDIST(2)  starting position $L_0$
          (set equal to rotor radius to
          start wave at edge of rotor disk)

GMAG(3)  real; gust magnitude (ft/sec or m/sec);
          order: $u_G, v_G, w_G$

VELG  real; gust convection velocity $V_g$ (ft/sec or
       m/sec)

PSIG  real; azimuth angle of convected gust wave front

OPGUST(3)  integer; convected gust model
            OPGUST(1)  convected (at rate $V_g$) relative
            aircraft if 0; relative fixed
            frame if 1
            OPGUST(2)  use gust at hub if 0; use gust
            distributed over disk if 1
            (rotor#1)
            OPGUST(3)  use gust at hub if 0; use gust
            distributed over disk if 1
            (rotor#2)
6.4 Notes

This section provides equations and figures to further define the input variables. Refer to Volume I for additional details.

6.4.1 Hub geometry.

Figures 6-1 and 6-2 illustrate the rotor hub geometry considered. The following input parameters, in namelist NLRTR, are involved: RFA, ZFA, XFA, CONE, DROOP, SWEEP, FDROOP, FSWEEP. See volume I, section 2.2.1.

Figure 6-3 illustrates the root geometry for calculation of the kinematic pitch/bending coupling. The following input parameters, in namelist NLRTR, are involved: KPIN, PHIPH, PHIPL, RPB, RPH, XPH. These variables are only used to calculate the pitch/bending coupling; they are not required if KPIN = 1 (then ATANKP is used). See volume I, section 2.3.4.

6.4.2 Aircraft description.

The aircraft model is described in volume I, section 4.1.5. The input parameters involved are in namelist NLBODY.

The aircraft geometrical description consists of the location of the following relative to the center of gravity: rotor #1; rotor #2; the point of application of airframe aerodynamic forces, from the wing/body, the horizontal tail, and the vertical tail; and the point of application of auxiliary forces. The orientation and position of the aircraft components is defined in a body axis system (the F frame) with origin at an arbitrary reference point, as in Figure 6-4. The F system has the x axis forward, the y axis to the right, and the z axis downward (flight dynamics convention). The position data is input in terms of fuselage station (FS, positive aft), buttline (BL, positive to the right), and
Figure 6.1. Schematic of the rotor hub and root geometry (side view). Only a single, undistorted blade is shown, without the gimbal undersling. The gimbal is omitted from the model for articulated and hingeless rotors.
Figure 6-2. Schematic of the rotor hub and root geometry (top view). Only a single, undistorted blade is shown, without the gimbal undersling. The gimbal is omitted from the model for articulated and hingeless rotors.
Figure 6-3. Schematic of blade root and control system geometry for calculation the kinematic pitch-bending coupling.
Figure 6-4. Definition of aircraft geometry.
waterline (WL, positive upward) relative to this arbitrary reference point (Figure 6-4).

The mode shapes of the airframe elastic motion are described by the six components of linear and angular hub motion: $\xi_k$ and $\gamma_k$ at each rotor hub. The components of the mode shapes are defined relative to the F system. Assuming that the generalized coordinate $q_k$ has dimensions of m or ft, it follows that the generalized mass $M_k$ has dimensions of kg or slug; that the hub linear motion $\xi_k$ is dimensionless (i.e., m/m or ft/ft); and that the hub angular motion $\gamma_k$ has dimensions of rad/m or rad/ft. The following input parameters are involved: QMASS, ZETAR1, GAMAR1, ZETAR2, and GAMAR2.

These elastic vibration modes can be arbitrarily scaled. If $\xi$ and $\gamma_k$ are multiplied by a factor $S$, then $M_k$ should be multiplied by $S^2$ and the solution for $q_k$ will be divided by $S$. Airframe finite element analyses (such as NASTRAN) typically use inches for length dimensions. To convert to feet, as required here, the mode shapes for angular motion, $\gamma_k$, and the generalized mass, $M_k$, must then be multiplied by 12. The typical coordinate system for the finite element analysis has the x axis positive aft, the y axis positive to the right, and the z axis positive upward. Then to convert to the F frame used here, the signs of the x and z components of the linear and angular modes shapes, $\xi_k$ and $\gamma_k$, must be changed.

For the wind tunnel trim cases, the rotor system is mounted on a test stand and turntable with pitch and yaw capability (APITCH and AYAW in namelist NLTRIM). The flight path angles and trim Euler angles are not used. The wind axes and body axes (F system) coincide, with the x axis upstream, the y axis to the right, and the z axis downward. The geometry (rotor position and orientation, mode shapes, etc.) is defined for zero pitch and yaw angles, relative to a reference point at the center of the rotation. Then the input geometry is transformed by the program to the F system.
6.4.3 Auxiliary forces.

The auxiliary force model is described in volume I, section 4.1.5. The input parameters involved are in namelist NLBODY.

The auxiliary forces, $f_l$, directly generate generalized forces on the airframe, without dynamics (degrees of freedom). The magnitude of the auxiliary force consists of a constant term, a term proportional to the pilot’s control input, and higher harmonic terms. The constant term can represent a fixed auxiliary lift or propulsive force. The control term can be used to trim the aircraft, and as a perturbation for the transient, flight dynamics, and flutter analyses. The higher harmonic terms can represent effects such as fuselage loads produced by rotor wake impingement. The higher harmonic control terms may be associated with either or both rotors.

The position of the auxiliary force point of application is defined by its fuselage station, buttline, and waterline (as for other fuselage positions). The orientation of the force vector is defined by the azimuth angle $\psi_F$ and elevation angle $\theta_F$. These angles are measured relative to the fuselage axis system (F frame): $\psi_F$ is in the x-y plane, positive counterclockwise from the negative x axis; $\theta_F$ is from the x-y plane, positive upward. For example, a propulsive force would be obtained with $\psi_F = 180$ and $\theta_F = 0$. To determine the generalized force for the elastic airframe modes, the linear mode shape $\xi_k$ is required at the point of application of the auxiliary force. The following input parameters are involved: NAF, FSAF, BLAF, WLAf, AZAF, ELAF, ZETAAS, and AUXSYM.

6.4.4 Control system.

The control system model is described in volume I, section 4.1.6. The input parameters involved are in namelist NLBODY.
The control variables included in the rotorcraft model are collective and cyclic pitch of the two rotors; the aircraft controls, which consist of engine throttle, wing flaperon angle, wing aileron angle, elevator angle, and rudder angle; and the airframe auxiliary forces. The control vector is thus

\[ v^T = [(\theta_o \ \theta_1C \ \theta_1S)_1 \ (\theta_o \ \theta_1C \ \theta_1S)_2 \ \delta_f \ \delta_e \ \delta_a \ \delta_r \ \delta_t \ (f_1)] \]

Except for the auxiliary forces, the units of \( v \) are radians internally, and degrees for input/output. The units of the auxiliary forces \( f_1 \) are lb or N.

The pilot's controls consist of collective stick (positive upward), lateral cyclic stick (positive to the right), longitudinal cyclic stick (positive forward), pedal (positive yaw right), and the throttle:

\[ v_p^T = [\delta_o \ \delta_c \ \delta_s \ \delta_p \ \delta_t] \]

It is often convenient to be able to directly associate \( v_p \) with the corresponding rotor controls. Hence the units of \( v_p \) are treated as radians internally, and degrees for input/output (i.e., converted using a factor of 57.29578). Other units (such as inches of stick deflection) can be obtained by proper definition of the control matrix.

A linear relation between the pilot's control inputs and the rotor and aircraft control variables is used:

\[ v = T_C v_p + v_o \]

where \( v_o \) is the control input and auxiliary forces with all sticks centered (\( v_p = 0 \)), and \( T_C \) is a transformation matrix defined by the control system geometry. In terms of the input variables, \( v_o \) is
\( v_o^T = (\text{CNTRLZ}(11), \text{FORCEZ(NAF)}) \)

The control matrix \( T_C \) can be input (as TCNTRL, if TCIN = 1), or defined in terms of swashplate gains and phases (if TCIN = 0). For the rotor and airframe controls, the gain factors (parameters KxCFE) have units of degrees per unit stick deflection, or deg/deg when \( v_p \) is interpreted in terms of degrees of control. The phases (parameters PxCFE) are swashplate lead angles, in degrees of azimuth. The parameter KATCFE makes the auxiliary force \( f_1 \) available for propulsion (connected to the throttle). The parameter KAPCFE makes the auxiliary force \( f_2 \) available for antitorque (connected to the pedal). These gains have units of lb/deg or N/deg (hence are multiplied by 57.29578 in constructing \( T_C \)). The control matrices for the various rotorcraft configurations are defined in volume I, section 4.1.6.

For the trim iteration, the most important requirement is that the coupling between redundant rotor and aircraft controls be accounted for; then the trim solution will produce the correct positions of the individual controls.

6.4.5 Airframe aerodynamics.

The airframe aerodynamic model is described in volume I, section 4.2.6. The corresponding input parameters are in NLBODY.

The aerodynamic loads are a combination of nonlinear and linearized forces. The nonlinear terms are evaluated from simple equations and possibly tables. The linearized terms are evaluated from stability derivatives. For the trim analysis, or generally in the absence of input stability derivatives, only the nonlinear term is used. For the flutter, flight dynamics, and transient analyses, the nonlinear term is evaluated with the trim motion and the linearized term is evaluated with the difference between the current and trim motion.
The nonlinear aerodynamic model calculates the forces using the following expressions.

\[
\begin{bmatrix}
    M_y/q \\
    D/q \\
    L/q
\end{bmatrix} = \begin{bmatrix}
    M_\alpha/q & M_\delta_F/q & M_{\delta F}/q \\
    0 & D_\alpha_\delta F/q & D_\delta F/q \\
    L_\alpha/q & L_\delta F/q & L_{\delta F}/q
\end{bmatrix}
\begin{bmatrix}
    \alpha_{WB} + i_{WB} \\
    \delta_F \\
    \delta_F
\end{bmatrix} + \begin{bmatrix}
    Sc_{CM} \\
    Sc_D \\
    Sc_L
\end{bmatrix}
\]

\[
\begin{bmatrix}
    M_\chi/q \\
    M_\zeta/q \\
    Y/q
\end{bmatrix} = \begin{bmatrix}
    N_x\beta/q & VN_xp/q & VN_x r/q & N_x\delta a/q \\
    N_z\beta/q & VN_zp/q & VN_z r/q & N_z\delta a/q \\
    Y_\beta/q & VY_p/q & VY_r/q & Y_\delta a/q
\end{bmatrix}
\begin{bmatrix}
    \beta_{WB} \\
    p/V \\
    r/V \\
    s_a
\end{bmatrix}
\]

and for the horizontal tail and vertical tail loads:

\[
L_{HT}/q = (L_\alpha/q)(\alpha_{HT} + i_{HT}) + (L_\delta e/q)\delta_e
\]

\[
L_{VT}/q = (L_\alpha/q)(\alpha_{VT} + i_{VT}) + (L_\delta r/q)\delta_r
\]

\[
\epsilon = \frac{(L/q)_{WB}}{f_\epsilon} - \frac{(L_\alpha/q)_{WB} L_{HT}}{V \alpha_{WB}}
\]

\[
\sigma = \frac{z_{VT}}{V} p
\]

Here \( \delta_F \) is a wing flap angle (AFLAP in namelist NLTRIM). Note that both the second and third terms in \( D/q \) (vertical and induced drag) give an \( \alpha^2 \) dependence. The zero lift angles relative to the fuselage axis
system are \( i_{W} \), \( i_{HT} \), and \( i_{VT} \); \( M_{o}/q \) is the moment at zero lift; and \( i_{D} \) is the angle of minimum parasite drag. The airframe aerodynamic coefficients \( (C_{L}, C_{D}, \text{and} C_{M}) \) are obtained from tables (described in section 5.3). These tables are functions of angle-of-attack (measured relative to the fuselage axes), Mach number, and elevator angle.

The linearized aerodynamic model calculates forces that are perturbations from trim, evaluated in terms of stability derivatives:

\[
\begin{bmatrix}
M/q \\
-X/q \\
-Z/q \\
\end{bmatrix} = \\
\begin{bmatrix}
VM_{a}/q & 2M/q+VM_{u}/q & M_{a}/q & VM_{q}/q & M_{b}/q & M_{b}/q \\
VD_{a}/q & 2D/q+VD_{u}/q & D_{a}/q-L/q & VD_{q}/q & D_{b}/q & D_{b}/q \\
VL_{a}/q & 2L/q+VL_{u}/q & L_{a}/q+D/q & VL_{q}/q & L_{b}/q & L_{b}/q \\
\end{bmatrix} \begin{bmatrix}
\frac{z}{V^2} \\
\frac{x}{V} \\
\frac{\dot{x}}{V} \\
q/V \\
\delta_{e} \\
\delta_{f} \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\frac{1}{2} \rho S C_{M_a} & Sc(2C_{M}+MC_{MM}) & ScC_{M_a} & \frac{1}{2} \rho S C_{M_q} & ScC_{M_b} & ScC_{M_b} \\
\frac{1}{2} \rho S C_{D_a} & S(2C_{D}+MC_{MD}) & S(C_{D_a}+C_{L}) & \frac{1}{2} \rho S C_{D_q} & SC_{D_b} & SC_{D_b} \\
\frac{1}{2} \rho S C_{L_a} & S(2C_{L}+MC_{ML}) & S(C_{L_a}+C_{D}) & \frac{1}{2} \rho S C_{L_q} & SC_{L_b} & SC_{L_b} \\
\end{bmatrix} \begin{bmatrix}
\frac{z}{V^2} \\
\frac{x}{V} \\
\frac{\dot{x}}{V} \\
q/V \\
\delta_{e} \\
\delta_{f} \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
L/q \\
N/q \\
Y/q \\
\end{bmatrix} = \\
\begin{bmatrix}
N_{x}/q & VN_{x}/q & VN_{x}r/q & N_{x}/q & N_{x}/q \\
N_{z}/q & VN_{z}/q & VN_{z}r/q & N_{z}/q & N_{z}/q \\
Y_{r}/q & VY_{r}/q & Y_{r}/q & Y_{r}/q & Y_{r}/q \\
\end{bmatrix} \begin{bmatrix}
\gamma/V \\
p/V \\
r/V \\
\delta_{a} \\
\delta_{f} \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\frac{1}{2} \rho S b C_{l}b & \frac{1}{2} \rho S b C_{l}r & \frac{1}{2} \rho S b C_{n}b & \frac{1}{2} \rho S b C_{n}r \\
\frac{1}{2} \rho S b C_{n}b & \frac{1}{2} \rho S b C_{n}r & \frac{1}{2} \rho S b C_{r}b & \frac{1}{2} \rho S b C_{r}r \\
\end{bmatrix} \begin{bmatrix}
\gamma/V \\
p/V \\
r/V \\
\delta_{a} \\
\delta_{f} \\
\end{bmatrix}
\]

The coefficients are based on a reference area \( S \), chord \( c \), and span \( b \):

\( C_{L} = \text{lift}/qS \), \( C_{D} = \text{drag}/qS \), \( C_{M} = \text{pitch-moment}/qSc \), \( C_{y} = \text{side-force}/qS \),

\( C_{l} = \text{roll-moment}/qSb \), \( C_{n} = \text{yaw-moment}/qSb \). The dimensionless rate derivatives are \( (C)_{a} = \partial C/\partial (\dot{a}/c/2V) \), \( (C)_{q} = \partial C/\partial (\dot{q}/c/2V) \), \( (C)_{p} = \partial C/\partial (\dot{p}/b/2V) \), \( (C)_{r} = \partial C/\partial (\dot{r}/b/2V) \). The stability derivative coefficients

206
are either constants, or are evaluated from tables (described in section 5.4). The tables are functions of angle-of-attack (measured relative to the fuselage axes), and Mach number.

For the nonlinear equations, the angles are in radians. For the stability derivative coefficients, the derivatives with respect to respect to angle-of-attack, sideslip, and control angle have units of per-radian or per-degree.

The sign conventions of the loads and motion are as follows.

<table>
<thead>
<tr>
<th>quantity</th>
<th>+ direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>wing-body loads</td>
<td>up</td>
</tr>
<tr>
<td>lift</td>
<td>up</td>
</tr>
<tr>
<td>drag</td>
<td>aft</td>
</tr>
<tr>
<td>side force</td>
<td>right</td>
</tr>
<tr>
<td>pitch moment</td>
<td>nose up</td>
</tr>
<tr>
<td>roll moment</td>
<td>right</td>
</tr>
<tr>
<td>yaw moment</td>
<td>nose right</td>
</tr>
<tr>
<td>tail forces</td>
<td>up</td>
</tr>
<tr>
<td>horizontal tail lift</td>
<td>up</td>
</tr>
<tr>
<td>vertical tail lift</td>
<td>left</td>
</tr>
<tr>
<td>tail cant angle</td>
<td>left</td>
</tr>
<tr>
<td>horizontal tail</td>
<td>left</td>
</tr>
<tr>
<td>vertical tail</td>
<td>right</td>
</tr>
<tr>
<td>motion</td>
<td>aircraft to right</td>
</tr>
<tr>
<td>sideslip</td>
<td>to right</td>
</tr>
<tr>
<td>roll rate</td>
<td>nose up</td>
</tr>
<tr>
<td>pitch rate</td>
<td>nose right</td>
</tr>
<tr>
<td>yaw rate</td>
<td></td>
</tr>
<tr>
<td>controls (down, lift increase)</td>
<td></td>
</tr>
<tr>
<td>flaperon</td>
<td>lift up</td>
</tr>
<tr>
<td>elevator</td>
<td>pitch down</td>
</tr>
<tr>
<td>aileron</td>
<td>roll left</td>
</tr>
<tr>
<td>rudder</td>
<td>yaw right</td>
</tr>
</tbody>
</table>

Hence the following signs are expected for the control derivatives.
nonlinear equations

\[ \text{SIDEB} = \frac{Y_\beta}{q} \quad \text{negative} \]
\[ \text{YAWB} = \frac{N_{z\beta}}{q} \quad \text{negative} \]
\[ \text{ROLLA} = \frac{N_x}{\delta a} \quad \text{negative} \]
\[ \text{LFTEH} = \frac{L_{HT}}{\delta} \quad \text{positive} \]
\[ \text{LFTRV} = \frac{L_{VT}}{\delta} \quad \text{positive} \]

stability derivatives

\[ \text{CYB} = C_{y\beta} \quad \text{negative} \]
\[ \text{CNB} = C_{n\beta} \quad \text{negative} \]
\[ \text{CLDA} = C_{l} \quad \text{negative} \]
\[ \text{CMDE} = C_m \quad \text{negative} \]
\[ \text{CNDR} = C_n \quad \text{positive} \]

It is not necessary that these sign conventions for the aircraft controls be followed, but it is essential that a consistent convention be used for all input parameters (including the definition of the coupling of the aircraft and rotor controls to the pilot's controls).
6.4.6 Airframe elastic mode equations.

The equations of motion for the airframe elastic modes are as follows, in dimensional form:

\[
M_k \ddot{q}_{sk} + g_s \omega_k \dot{q}_{sk} + \omega_k^2 q_{sk} = (Q_k)_{\text{rotors}} \\
+ \frac{\Delta \rho}{2} V^2 \left[ F_{qk} \dot{q}_{sk} - \frac{F_{qk}}{V} + F_{qk} \delta \begin{bmatrix} \delta f \\ \delta e \\ \delta a \\ \delta r \end{bmatrix} \right] + (Q_k)_{f}
\]

The corresponding input variables, in namelist NLBODY, are:

- **QMASS** = \( M_k \) (slug or kg)
- **QDAMP** = \( g_s \)
- **QFREQ** = \( \omega_k \) (input in Hz, used in rad/sec)
- **QDAMPA** = \( F_{qk} \) (ft\(^2\) or m\(^2\))
- **QCNTRL** = \( F_{qk} \delta \) (ft\(^2\)/rad or m\(^2\)/rad)

See volume I, section 4.2.7.

6.4.7 Rotor hub vibration.

The two rotors produce static and vibratory motion of the airframe, which then produces hub motion that influence the rotor dynamics and aerodynamics. Figure 6-5 outlines the analysis procedure. It is useful to be able to suppress the feedback of the nonrotating frame vibration to either or both rotors. This can be done by setting the hub motion to zero (see Figure 6-5). The input variable OPHVIB in namelist NLRTR controls this option for each rotor. Note that if the rotational speeds of the two rotors are not equal, the vibration caused by the other rotor must always be suppressed (dotted lines in Figure 6-5). By using
Figure 6-5. Outline of dynamic interaction of the two rotors.
OPHVIB=0, the rotor excitation of the airframe is calculated, but it is not fed back to the rotor. The calculation of the airframe motion can be suppressed entirely by omitting the degrees of freedom (DOF in namelist NLTRIM). See volume I, section 5.1.1.

6.5 Guidelines

This section provides guidelines for the use of the input parameters in the rotorcraft analysis. It is organized according to the namelists.

6.5.1 General.

(1) Namelist files of baseline values of the input parameters are available for use with the input file preparation program. See section 6.5.8.

(2) Convergence may not be achieved for the initial runs of a new project. There are four major iterations: trim, regulator, circulation, and motion. A warning is printed if an iteration does not converge (search for "WARNING" to find it). Unconverged iterations are also identified in the trim header (search for "AIRCRAFT TRIM"). The parameter TRACE can be used to obtain more information about the convergence. Very severe divergence (especially in the motion iteration) will produce floating-point overflow, generally in the subroutines AEROF1 or AEROF2. If recursive updating of the derivative matrix is used for the trim or regulator iterations, the parameter variance (P-matrix) may diverge, producing floating-point overflow in the subroutines TRIMI or STRI respectively.

The principal parameters controlling convergence are the relaxation factors, tolerances, and maximum number of iterations. The following actions typically improve convergence. The first action should always
be to reduce the relaxation factor (if it is above about 0.1); then one of the other actions can be tried.

a) Trim iteration.
   i) Reduce the relaxation factor. FACTOR should be just small enough to prevent oscillations.
   ii) Increase the maximum number of iterations (MTRIM).
   iii) Decrease the tolerances on the motion and circulation iterations.
   iv) Use recursive updating of the derivative matrix (OPTIDR), or change the weight (ALPHA). If the parameter variance (P-matrix) of the recursive update diverges, perform the perturbation identification more often (MTRIMD).
   v) Recalculate the derivative matrix more often (reduce MTRIMD) or differently (change DELTA).
   vi) Use better initial control settings (obtained from a nearby operating condition, or found using OPSTLL = 0).

a) Regulator iteration.
   i) Reduce the relaxation factor (FACTS) or increase the control weight in the cost function (WTDELT).
   ii) Increase the maximum number of iterations (MSTR).
   iii) Use recursive updating of the T-matrix (RIDSTR), or change the weight (ALFSTR, QSTR). If the parameter variance (P-matrix) of the recursive update diverges, perform the perturbation identification more often (MIDSTR).
   iv) Recalculate the T-matrix more often (reduce MIDSTR) or differently (change DELSTR, PIDSTR).
   v) Use better initial control settings (obtained from a nearby operating condition).
vi) Increase the tolerance (EPSTR) or change the convergence
criterion (JTARG), if it appears that the requested cost
function value is not attainable.

c) Circulation iteration.
   i) Reduce the relaxation factor. FACTWU (uniform inflow) and
      FACTWN (nonuniform inflow) values as low as 0.1 are not
      uncommon.
   ii) Increase the maximum number of iterations (ITERC).
   iii) Revise the aerodynamic radial stations (RAE), if the panel
        widths do not change smoothly.
   iv) Increase the near wake core size (CORE(7)), or change the near
       wake model (OPNW).

d) Motion iteration.
   i) Reduce the relaxation factor. FACTM = 1. can often be used,
      but certain problems may require a lower value (e.g. rotor
      operation with significant stall, or coupled rotor/body
      motion).
   ii) Increase the maximum number of iterations (ITERM).
   iii) Increase the number of revolutions of rotor solution between
        body motion update (MPSIR), if body degrees of freedom are
        present.

Increasing the tolerance (EPTRIM, EPCIRC, or EPMOTN) is not
generally an appropriate way to achieve convergence. Indeed, if a
relaxation factor is reduced (slowing the rate of convergence), the
corresponding tolerance in the test for convergence should also be
reduced. Otherwise, the iteration may stop simply because the
relaxation factor does not allow a significant change from one cycle to
the next. (Note that at the end of a long trim iteration, the control
changes are quite small, so effectively the circulation and motion
iterations get run beyond their converged point.) In addition, if the
circulation and motion iterations are not sufficiently converged, the
trim iteration may become erratic or even diverge. In such a case, the trim convergence can be improved by reducing the circulation and motion tolerances (EPCIRC and EPMOTN).

Convergence problems may also reflect physical limits of the aircraft, not problems with the numerical solution. Typical cases encountered are the following.

(a) Trimming the aircraft in free flight (or the rotor to a specified thrust) at operating conditions near the limits of the rotor load capability. In this case it would be appropriate to try trimming to constant power instead (or at fixed collective).

(b) Trimming a single main-rotor helicopter in free flight at the limit of the tail-rotor antitorque capability. It is possible to artificially increase the tail-rotor thrust limit (such as by suppressing stall, OPSTLL = 0).

(c) Trimming an aircraft with the pilot's controls inappropriately connected to the rotor controls. For example, if trim of thrust and flapping (a wind tunnel trim case) is attempted for a tiltrotor, both collective stick and lateral stick may be connected to the rotor collective pitch, and neither connected to the lateral cyclic pitch. In such a case, use direct input of the pilot/rotor control coupling matrix (TCNTRL).

(d) Duplication of controls between the trim iteration and the regulator. For example, using main rotor cyclic pitch in a flapping controller as well as in trim.

(e) Inappropriate definition of the regulator. Very nonlinear input/output relations in particular can lead to divergence with many regulator definitions. For example, the cost function objective defined by the tolerance may not be physically achievable, or the regulator being used may not be capable of finding the minimum.
(3) A trace of the trim iteration is always printed. More information about the trim iteration, and traces of the regulator, circulation and motion iterations can be obtained using the parameter TRACE. The resulting output provides information about convergence of the iterations. The parameter TRACE will also operate during the flight dynamics and transient analyses.

If more information is required, it can be obtained using the debug print (DEBUG = 2 or 3). The debug output is in namelist format, and can be interpreted by reference to the source code, the dictionary of analysis variables, and the theory documentation. Progress through the analysis can be displayed by turning on the timer, DEBUG(I), which also prints the debug counter.

(4) The circulation iteration is only asymptotically convergent for a hovering rotor at zero thrust (see volume I, section 5.1.12). A special procedure has been implemented to calculate the induced velocity in the uniform inflow analysis for this case (controlled by parameters OPTZT and CTSTZT in namelist NLRTR, discussed further below). This special procedure will not produce correct inflow values, so it must not be used in general. The special procedure should only be used when circulation convergence problems (not eliminated using smaller FACTWU or FACTOR) have been encountered for a hovering rotor at low thrust. For the nonuniform inflow analysis, the equivalent procedure is implemented by using input values of the prescribed wake geometry constants (OPRWG = 3), so the wake geometry does not change during the analysis. However, the wake model is not appropriate for the case of a hovering rotor at zero thrust, so even with fixed wake geometry the results may not be acceptable.

(5) The blade modes used as degrees of freedom are rotating, coupled flap/lag bending modes, and nonrotating elastic torsion modes.
Fully-coupled free vibration modes can be obtained by running a flutter analysis (ANTYPE=1 in namelist NLTRIM) without aerodynamics. The aerodynamics are removed by setting the air density to a small number (DENSE = 0.000001 slug/ft$^3$) and suppressing the circulation iteration (ITERC = 1) and trim iteration (OPTRIM = 0 or 100). For modes about the undeflected blade position, the trim blade motion would be suppressed as well (DOF = 74*0, ITERM = 1 in namelist NLTRIM). The appropriate rotor degrees of freedom are selected for the flutter analysis (DOF in namelist NLFLUT). The frequencies of a single blade can be calculated (NBLDFL = 1), perhaps with root boundary conditions appropriate for collective or cyclic modes; or an entire rotor can be analyzed (with a gimbal or teeter hinge, and the drive train), using multiblade coordinates. The flutter analysis gives the coupled mode frequencies (ANTYPE = 1, NSYSAN = 0 in namelist NLFLUT). Eigenvectors can also be obtained (NSYSAN = 1), but radial modes shapes for the fully coupled modes are not automatically generated.

(6) Write of the plot file is enabled by the parameter PLFILE in namelist NLCASE. The data written is selected by various parameters in the other namelists. The plot file is written by a single subroutine, FILEP, which can also be used to read the file. Use of FILEP, and the format of the plot file, are described in the prologue of that subroutine. The header of the printed output is duplicated in the plot file, to identify the case. Titles are printed before each data set in the file. The titles are sufficient to identify the data (perhaps for automatic search), but do not provide as much detailed description as does the printed output.

6.5.2 Namelist NLTRIM

(1) The operating condition is specified principally by the the following data:
(a) aircraft speed (VKTS or VEL; if VKTS is zero, VEL is used; if both are zero, the speed is zero)

(b) rotor#1 rotational speed (VTIP or RPM; if VTIP is zero, RPM is used; if both are zero, the default is the normal tip speed VTIPN in namelist NLRTR)

(c) atmosphere (OPDENS, and as required ALTMSL, TEMF, and DENSE)

(d) trim turn rate (RTURN)

(2) Initial guesses are required for the control positions at the start of the trim iteration: some subset of COLL, LATCYC, LNGCYC, PEDAL, THROTL, APITCH, AROLL, ACLIMB, AYW. (Calculations can also be performed for fixed controls.) Typically projects require many calculations (varying some parameters) for a limited set of operating conditions. At the beginning of the project it is necessary to establish the control positions required for trim at those operating conditions. If an operating condition has not been analyzed before, the initial control positions must be estimated (from a nearby operating conditions, another job, or other information). If the initial guess is too far off, trim convergence problems are likely. When the control positions have been established for the required set of operating conditions, they should always be used as initial conditions for the remainder of the project (to minimize computation time).

Note that the final control positions of a case are used as initial conditions for the next case in a job (unless superseded by namelist input or reading the input file).

(3) OPTRIM. This parameter selects the trim option. It also distinguishes between free flight cases and wind tunnel cases. For free flight cases, the Euler angles (APITCH and AROLL), flight path angles (ACLIMB and AYW), and turn rate (RTURN) are used. For wind tunnel cases, the test module pitch and yaw angles (APITCH and AYW) are used.
For free flight, the forces and moments are trimmed to zero. Some of the options trim the aircraft power to the value specified by CTRIM. The wind tunnel trim options use the appropriate targets. The rotor drag force (CTRIM or XTRIM) is positive aft; the rotor side force is positive toward the advancing side. The longitudinal flapping angle (CTRIM) is positive for forward tip-path plane tilt; the lateral flapping angle (BSTRIM) is positive for tip-path plane tilt toward the retreating side.

The pilot's stick positions are presented as degrees. With unit gains in the matrix relating the pilot and rotor controls (useful but not required), the stick positions will be identical to the rotor collective and cyclic pitch angles. Note however that positive cyclic stick deflections (LNGC and LATC) produce negative main rotor cyclic pitch changes (TIS and T1C).

(4) FACTOR, MTRIMD, OPTIDR, ALPHA. The relaxation factor for the trim iteration (FACTOR) should be as large as possible (for efficiency). The derivative matrix is usually obtained initially by perturbation, and then recursively updated. Typical values:

a) FACTOR = .5
b) MTRIMD = MTRIM
c) OPTIDR = 1
d) ALPHA = .5

The recursive update of the derivative matrix may help trim convergence, and usually does not hurt. Note that if the exponential-window weighting factor (ALPHA) is set to 1., the recursive update is gradually turned off.

(5) Typical tolerances:

a) motion iteration, EPMOTN = 0.02
b) circulation iteration, EPCIRC = 0.001
c) trim iteration, EPRIM = 0.01 for free flight; 0.005 for symmetric free flight; 0.001 for wind tunnel
See volume I, sections 5.1.4, 5.1.12, 5.3.1 respectively for details on use of these tolerances. Sometimes it is necessary to use smaller EPMOTN and EPCIRC (e.g. a factor of ten smaller than above) in order to improve trim convergence. The only way to be sure that the proper value is being used is to try a smaller value, and check that the results of interest do not change significantly.

(6) If only rotor degrees of freedom are being used, set MPSIR = MPSI and MREV = 1. If airframe degrees of freedom are used, it is generally necessary to have 2 complete revolutions of the rotor analysis between updates of the airframe vibration: MPSIR = 2*MPSI and MREV = 2.

(7) DOF. The rotor modes are ordered by frequency. For a soft in-plane or articulated rotor then, the first bending mode is fundamental lag motion, and the second mode is fundamental flap motion. Typically need 4-5 bending modes and 1-2 torsion modes for a rotor in order to calculate blade bending and torsion moments. Fewer modes (perhaps just rigid flap) may be sufficient for aerodynamic loading calculations, although there may be significant effects of elastic twist and other degrees of freedom even in normal operating conditions. The airframe rigid body and elastic modes must be used in order to calculate airframe vibration; these modes also can influence the rotor loads.

(8) DOFT. This parameter controls the description of the blade bending position, which enters the nonlinear forces on the blade. It must be a subset of the degrees of freedom used in DOF. Typically the first two modes for each rotor (fundamental flap and lag) are sufficient.

(9) MHARM. Use 7-10 when calculating blade bending and torsion moments; the value should be consistent with the frequency range of the
bending and torsion modes used, and the frequency content implied by MPSI. MHARM = 2-3 might be sufficient for airloads (HARM controls the frequency content of the motion; the higher harmonic loading produced by the wake will still be obtained). MHARM = 1 gives just first harmonic motion. Using MHARM = 0 gives just the static blade deflection; this is appropriate and efficient for hover; but will give wrong results in forward flight (or even for a complete aircraft in hover, when the tip-path-plane is tilted relative to the shaft).

10) MHARMF. Typically use 1-2 when include airframe degrees of freedom in DOF.

11) LEVEL. Nonuniform inflow almost always has an influence. Generally the free wake geometry is important at low speeds (below an advance ratio of about 0.25). Typically ITERU = 1, ITERR = 1, and ITERF = 1 are used. These statements are based on experience with the analysis, and should be checked for new problems.

12) Self-tuning regulator. The regulator is defined if OPSTR = 1, and then can be separately turned on for the trim, flight dynamics, and transient analyses (MSTR in namelists NLTRIM, NLSTAB, NLTRAN). If an error is found in the specification of the controls or measurements (CONSTR and OUTSTR), the regulator is turned off and the analysis continues. An error message, listing the bad controls and measurements, appears on the header page.

The measurements selected (OUTSTR) must generally be defined by the appropriate input parameters in namelist NLLOAD. The analysis only checks that the location (sensor number, radial station number, or microphone number) and harmonic number are less than the maximum values allowed, not for consistency with the other input parameters. Note that
the harmonic number is divided by the number of blades only for the airframe vibratory response. The measurements have the following units.

a) flapping: deg
b) power, hub loads, root loads, blade loads: rotor coefficient divided by solidity
c) noise: N/m² or dB
d) airframe vibratory response: dimensionless or dimensional, as selected by OPDRES (namelist NLOAD)

The targets (ZTARG) and cost function weights (WTZ) must reflect these units.

The selected measurements should generally be calculated and printed separately, as controlled by the parameters in namelist NLOAD. The printed output will include explanatory information, which will facility checking that the desired quantity was selected, and that the units are correctly interpreted.

The controls selected (CONSTR) must be independent of the variables used by the trim iteration (the analysis does not check for independence however). If an auxiliary force or higher harmonic control is selected, it must also be defined by the appropriate input parameters (NAF, MHHC, MHHCF, MHHAF, etc). The analysis only checks that the force number and harmonic number are less than the maximum values allowed, not for consistency with the other input parameters. Note that the harmonic number is divided by the number of blades for the nonrotating frame higher harmonic control and forces (refer to the definitions in namelist NLTR). The controls have units of degrees, except the auxiliary forces, which are in lb or N. The initial control values (TZERO) and cost function weights (WTDELT) must reflect these units.

Note that the final control positions and T-matrix of a case are used as initial conditions (TZERO and TMTRX) for the next case in a job (unless superseded by namelist input or reading the input file).

For a transient analysis, the actual regulator should be modeled, but for the trim and flight dynamics analyses the regulator is just a solution procedure for obtaining the converged response. Hence during
trim the use of perturbation and recursive identification, and choice of
the exponential window or Kalman filter, are governed by convergence and
efficiency. Typical values:

a) PIDSTR = 1, RIDSTR = 0 or 1, MIDSTR = 0
b) DELSTR = .5, ALFSTR = .5, QSTR = 0.
c) WTZ = 1. for flapping, 10. for vibration
d) FACTS = .8, WTDELT = .1
e) MSTR = 20, EPSTR = .05 * NZSTR
f) TRACE = 2

Using an input T-matrix (PIDSTR=0) is fastest; perturbation
identification every trim iteration (PIDSTR=2) is much slower. If
convergence problems occur, PIDSTR = 2 or MIDSTR gt 0 can be used. The
increment during perturbation identification, DELSTR, has units of deg
for controls, and 100 lb or 100 N for auxiliary forces. When using
recursive identification, the variance (P-matrix) of the recursive
update can diverge; performing the perturbation identification more
often is an option (MIDSTR gt 0). Note that DELSTR is used to calculate
the initial value of the P-matrix, so should be defined even if
perturbation identification is not used. FACTS and WTDELT act in a
similar manner to prevent divergence, but can significantly slow down
the convergence if not correctly chosen. The regulator may not converge
with JTARG = 0, if the desired value of the cost function J (defined by
EPSTR) can not be achieved; in such cases JTARG = 1 may be more
appropriate.

As examples, the measurements and controls to introduce a flapping
controller are:

```
OUTSTR(1,1)='PERF','RTR1',' ','BC ','
OUTSTR(1,2)='PERF','RTR1',' ','BS ','
CONSTR(1,1)='RTR ',''RTR1','T1C ',
CONSTR(1,2)='RTR ',''RTR1','T1S ',
```
or to introduce a vibration controller:

```
OUTSTR(1,1)='VIB ',''RTR1','l ','','COS ','l ','
OUTSTR(1,2)='VIB ',''RTR1','l ','','SIN ','l ','
CONSTR(1,1)='HHC ',''RTR1','COLL','CS1 ',
CONSTR(1,2)='HHC ',''RTR1','COLL','SN1 ',
```

222
(assuming that the higher harmonic control and the airframe vibratory response are appropriately defined).

6.5.3 Namelist NLRTR

(1) INFLOW. This parameter must be set appropriately when using LEVEL = 1 or 2.

(2) OPSTLL. The intended dynamic stall models are OPSTLL = 3 (\(\hat{\alpha}\) stall delay with dynamic stall vortex loads) and OPSTLL = 4 ((\(\hat{\alpha}\))^2 stall delay). Options OPSTLL = 2 and 5 are included for completeness. A value of FACTM less than 1 (typically .3, perhaps smaller) will probably be needed to achieve convergence with significant rotor stall, particularly for the blade pitch and torsion motion with dynamic stall loads.

(3) OPREYN and RETAB1. OPREYN controls the Reynolds number correction of the drag and lift coefficients. The correction is based on \(K = (Re/Re_t)^N\), where Re is the actual Reynolds number of the aerodynamic station and \(Re_t = M Re_t^1\) is the Reynolds number of the airfoil table data. The parameter \(Re_t^1\) is specified as part of the airfoil file; if RETAB1 in namelist NLRTR is nonzero, it supersedes the value in the airfoil file. If \(Re_t^1 = 0\) (i.e. RETAB1 = 0 in both namelist NLRTR and the airfoil file), the Reynolds number correction will not be applied. RETAB1 is only used for the Reynolds number correction. For comparison with the airfoil table Reynolds number RETAB1, the rotor blade Reynolds number corresponding to the mean chord and \(M = 1\) is \(Re_o/M_{tip}\), where both \(M_{tip}\) and \(Re_o\) are printed on the analysis header page.

(4) OPFFLI; FXLMDA and FYLMDA; KXLMDA and KYLMDA. These parameters define a linear variation of the induced velocity over the rotor disk,
for the "uniform inflow" analysis. The baseline values of the factors FXLMDA and FYLMDA are 1.

(5) OPTZT, CTSTZT. In order to achieve convergence of the circulation iteration for a hovering rotor at zero thrust, OPTZT = 1 must be used, with CTSTZT set to a small but nonzero value. The relaxation factor FACTWU must still be set to a value less than 1 for convergence (the smaller CTSTZT, the smaller FACTWU must be). Note that a large value of CTSTZT will simply result in the rotor induced velocity being set nearly zero. In any case, the inflow being calculated when OPTZT = 1 is not accurate; the results are acceptable only for hover at small thrust, where the inflow effects should be small. OPTZT = 1 must not be used for other operating conditions. See volume I, sections 2.4.3 and 5.1.12.

(6) HINGE = 0 (articulated) will give only the two fundamental flap and lag modes. These modes have a prescribed shape (rigid rotation about the hinge, pure out-of-plane and pure in-plane). The modal equations are not solved. See volume I, section 2.3.2.

(7) TSPRNG and RCPLS. These parameters define the pitch angle of the flap and lag hinges (with or without springs), relative to the hub plane. With the pitch bearing outboard of the hinges, RCPLS = 0 (the hinges do not rotate with the blade), and probably TSPRNG = 0 as well. Both the flap hinge and the lag hinge are rotated by the same angle; the case of a pitch bearing in between the flap and lag hinges can not be modeled. TSPRNG and RCPLS are only used if HINGE = 2, 3, or 4.

(8) MRB. Use a value large enough to obtain accurate values for the blade frequencies (typically MRB = 50 or more).
(9) NCOLB and NCOLT. The number of collocation functions should be about twice the number of modes used. Note that the number of out-of-plane collocation functions and in-plane collocations functions both equal NCOLB/2; so NCOLB/2 should be about twice the maximum of the number of flap or lag modes used.

(10) EPMODE. Typically 0.5 deg is used.

(11) The rotor analysis assumes a straight elastic axis. Generally the first priority in modeling a blade is to ensure that the relationship between the feathering axis, aerodynamic center, and center of gravity is preserved. Hence the following procedure is used to obtain the input parameters from the blade geometry.

(a) The actual position of the blade feathering axis, elastic axis, aerodynamic center, center of gravity, and tension center are defined as a function of radial station.

(b) The location of the feathering axis is defined relative the hub by the undersling (ZFA), torque offset (XFA) and precone (CONE).

(c) The radial location of the pitch bearing (RFA) must be correct relative to the flap and lag hinges. All pitch and torsion mode shapes are zero inboard of RFA, so the torsion properties (GJ, ITHETA, KP2) are not used inboard of RFA. The elastic axis is a straight line outboard of RFA, with droop or sweep relative to the feathering axis (DROOP and SWEEP). Note that for an articulated rotor, the droop and sweep just define the position of zero flap and zero lag deflection.

(d) A plot of the elastic axis is approximated by a straight line from r = RFA to the tip (defining the droop and sweep).

(e) The chordwise offsets of the aerodynamic center (XA and XAC), center of gravity (XI), and tension center (XC) are measured relative this approximate, straight elastic axis.
(12) The nonlinear lag damper is implemented in subroutines LDAMP1 and LDAMP2, which can be modified as required. See volume I, section 2.2.16.

(13) OPCFD, LDMCFD. The CFD interface is intended to be used with the nonuniform inflow calculation (LEVEL ge 1), but will also function for uniform inflow. The partial angle-of-attack is calculated after the last wake cycle; hence ALPHAP is not available for display (using MALOAD, item #77) until the end of all the wake and trim iterations. For a uniform inflow case, ALPHAP will equal the blade angle of attack. The wake geometry is not recalculated to evaluate ALPHAP, and the influence coefficient matrices are overwritten (so the transient and flight dynamics analyses should not be used in some cases).

The parameter LDMCFD determines whether the externally calculated lift, drag, and moment coefficients are used. To use the externally calculated coefficient over only part of the disk, set CxEXT = CxOLD where the internally calculated coefficients are to be used.

(14) The analysis requires the spanwise distribution of the blade inertial and structural properties. If these properties are not known, or a representative blade is to be analyzed, the properties may be estimated as follows. A uniform blade is assumed, and the solidity σ and Lock number γ must be given.

\[ \frac{c}{R} = \pi \frac{\sigma}{N} \]
\[ m = 3I_b/R^3 - 3\rho acR/\gamma - 0.1278\sigma R^2/\gamma N \]
\[ I_\theta = (\sigma/N)^2 I_b/R - (\sigma/N)^2 \rho acR^3/\gamma - 0.0426(\sigma/N)^3 R^4/\gamma \]
\[ GJ = (0.6366\omega_\theta \Omega R)^2 I_\theta \]
\[ EI_{zz} = m\bar{r}^2 R^4/k_z \quad (k_z = 40 \text{ to } 1000, \text{ 300 typical}) \]
\[ EI_{xx} = m\bar{r}^2 R^4/k_x \quad (k_x = 2 \text{ to } 30, \text{ 10 typical}) \]

Here \( N \) is the number of blades; \( R \) is the rotor radius; \( \Omega R \) is the rotor tip speed; and \( \omega_\theta \) is the first torsion frequency (per rev).
6.5.4 Namelist NLWAKE.

(1) OPFW. The options for the far wake roll-up model are: single peak, using either the maximum circulation or the outboard peak; and dual peak, with the inboard trailed wake rolled up or not (OPIVTL).

For the dual peak model (OPFW=2), the radial station of the inboard peak that is used for the wake geometry and influence coefficient calculation should be consistent with that of the current circulation solution. This can be checked by setting MDLOAD=1, NDPRNT(4)=1 (namelist NLLOAD), and comparing RI with RGI in the output. Note that a value of -1 means that an inboard peak was not found; GI and GO will be of the same sign when an inboard peak was not found and so GI is obtained from GO by interpolation. Generally the radial station of the inboard peak converges with one or two wake iterations (ITERR=2 or ITERF=2). The wake iteration updates the rigid or distorted geometry also of course; the analysis can be run with the radial station of the inboard peak fixed (OPRGI=1) to determine the relative importance of these changes.

For the inboard rolled-up trailed wake model, OPIVTL overrides the value of WKMODL(7) for this wake panel, using OPCORE(1) and CORE(6).

(2) OPNW. The options for the near wake model are: collocation points at quarter-chord or three-quarter-chord; and a straight or swept lifting line. The 3c/4 collocation point improves the blade-vortex interaction loads (discussed below).

For a swept tip, the intended model is OPNW = 3. The 3c/4 collocation point must be used if the swept lifting line is used (OPNW = 2 is not a consistent option, but is included for completeness). The swept tip model also requires the correct values for XA and ASWEEP (namelist NLRTR). The lifting line position is defined by XA and OPWKBP(4).
With the 3c/4 collocation point (OPNW = 1 or 3), a smaller value of FACTWN may be required for convergence (compared to that for the c/4 collocation point).

(3) CORE(7). The default value for the near wake core size is 20% of the distance between the aerodynamic radial stations (typically .004 at the tip). Too small a value for CORE(7), and sometimes even the default value, can lead to circulation convergence problems, particularly at high speed. Ideally, a value is needed that is small enough to not effect the results, and large enough to avoid convergence problems. The default value can be checked, and if necessary an appropriate value established, by varying CORE(7).

(4) Blade-vortex interaction. There are three approaches for modeling blade-vortex interaction.

(a) 3c/4 collocation point: OPNW = 1 or 3, DLS = -1., CORE(1) = .01-.035
(b) lifting surface correction: OPNW = 0, DLS = .5, CORE(1) = .01-.035
(c) an empirical vortex core radius: OPNW = 0, DLS = -1., CORE(1) = .02-.05

The 3c/4 collocation point and lifting surface correction account for the same effects, and must not be used together. The last approach should be used for hover (c/4 collocation point, without lifting surface correction).

The tip vortex core sizes recommended (with either the 3c/4 collocation point or the lifting surface correction) are typically 20% of the blade chord for full scale, and 50% of the blade chord for model scale. Multiplying by c/R then gives the above values for CORE(1). The lifting surface theory correction is roughly equivalent to increasing the core radius by 15-20% of the blade chord; so with the c/4 collocation point and no lifting surface correction, the core size
should be 35-70% of the chord. Note that wake measurements suggest a radius for the vortex core of about 10-30% chord.

Lifting line theory calculations of blade-vortex interaction show that the radial and azimuthal resolution in the wake should be about equal to the blade-vortex separation, which has an effective minimum equal the core radius. A radial resolution of .02R is typical at the blade tip, and .04R at 75% radius. These values are larger than the actual core radius (perhaps .005-.02). The resolution in the discretized wake being too large has the effect of increasing the blade-vortex interaction loads, which may be countered by empirically increasing the core radius. The 3c/4 collocation point and lifting surface correction are available to improve the accuracy of the lifting line theory, but neither changes the accuracy associated with discretization of the wake. Hence the core radius CORE(1) will probably always be larger than the physical value, for each of the approaches available.

(5) DLS. The lifting surface theory correction is applied if the middle of the vortex line segment is closer than d (fraction of rotor radius) to the downwash point. DLS should be 5 or 10 times the mean blade chord (divided by radius); a typical value is 0.5. To avoid a large increase in computation time, do not use a value larger than necessary.

(6) KNW. The extent of the near wake should be at least KNW = 2. The required size of the near wake influence coefficient array is NCNW = MRA**2*MPST*(KNW+1). With MPST = 24 and NCNW = 45000, the maximum number of radial stations (MRA) for a given KNW are:

229
<table>
<thead>
<tr>
<th>KNW</th>
<th>MRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
</tr>
</tbody>
</table>

The required size of the far wake influence coefficient matrix is \( NC = MPSI \times 2 \times MR \). With \( MPSI = 24 \) and \( NC = 30000 \), the maximum number of radial stations (MR, both on and off the rotor disk) is 52. The rotocraft analysis checks the required values of NCNW and NC against the actual array dimensions in the code.

(7) KRW, RRU, FRU, PRU. The rolling up wake model can be used if enough information is available to specify these parameters. Otherwise, choose these parameters so that the wake has rolled up by the time it reaches the next blade (KRW less than MPSI/NBLADE).

(8) KFW. At least two revolutions of wake should always be used, \( KFW = 2 \times MPSI \). At low speeds, more revolutions are required: typically about \( .4/(\text{advance ratio}) \) revolutions.

(9) RTVTX. This parameter is the radial station of the tip vortex at the blade trailing edge, with values .985-.99 typical. It is used like the tip loss factor (BTIP in namelist NLRTR, for uniform inflow). RTVTX should be greater than RAE(MRA) (the tip is at RAE(MRA+1) = 1.), so the roll-up occurs in the middle of the last aerodynamic panel. If RTVTX is less than RAE(MRA), then the last panel is outboard of the wake and the tip vortex induces a large positive angle of attack on it, significantly affecting the drag of the tip panel.
(10) DVS, EPVS. The nonplanar, quadrilateral sheet model is very expensive. EPVS is chosen to balance accuracy and efficiency, with .01 a typical value. If EPVS is too small, the sheet numerical integration will not converge (the maximum number of steps is 512; if DEBUG(22)=1, a message is printed when the numerical integration does not converge). Often planar, rectangular sheet elements are a satisfactory approximation (EPVS = 1000.). DVS eliminates the edge singularity, for both nonplanar-quadrilateral and planar rectangular models. It is similar to a core radius, or a finite thickness of the sheet. A typical value is .01. With the nonplanar-quadrilateral model and DVS too small, then for small distances from the sheet the numerical integration will not converge (use DEBUG(22)=1 to see if it occurs), and large velocities will be produced near the edges.

(11) Recommended values:

(a) OPFW = 2 (OPFW = 0 is faster, if a single peak model is acceptable)
(b) OPNW = 3, DLS = -1. (3c/4 collocation point, no lifting surface correction; OPNW = 0 for hover)
(c) CORE(4-7) = -1. (default values internally calculated)
(d) OPCORE = 2*0 (distributed-vorticity core model)
(e) WKMODL = 10*2,3*3 (line segments, except for the far wake model)
(f) OPTLTL = 0 (inboard trailed wake not rolled up)
(g) OPRG1 = 0 (use the calculated inboard peak radial location)
(h) OPWKB = 4*1 (complete blade motion description)
(i) EPVS = 1000., DVS = .01 (planar rectangular sheets)
(j) OPTVIC = 1, CORE(2) = .1, RTVIC = .76,.88 (decrease the effect of blade-vortex interaction on the inboard part of the blade)

(12) DOMCFD. See the discussion of OPCFD in namelist NLRTR. Only line elements are tested to see if they are inside the CFD computational domain, so WKMODL must not include sheets inside the domain. Note that the wake from the other rotor is retained in calculating the partial angle-of-attack. The CFD domain is defined as a box enclosing the
rotating blade. Other geometries can be treated by revising the subroutine GEOMX appropriately. Note however, that the boundary of the CFD domain is here only required where it intersects the rotor wake; and the partial angle-of-attack is probably not very sensitive to small errors in that intersection.

(13) Axisymmetric wake (OPHW = 0) must be used for hover (to get the far wake model). The influence coefficients are calculated for only one azimuth angle, so the computation is efficient. Typically LHW = 30 is used. The shed wake elements can be turned off (WKMODL) if the loading is independent of azimuth (not necessarily true for hover).

(14) KRWG. The wake geometry extent should be the same as KFW or KDW. The extrapolation of the undistorted helices is exact, so the upper limit on KRWG is no problem.

(15) OPRWG. The intended wake models for hover are prescribed models: OPRWG = 4 (Landgrebe), 6 (Kocurek and Tangler), or 3 (input K's). The K's calculated for the prescribed models are printed in the rotor performance section.

The intended wake model for forward flight is rigid geometry, with just vertical convection by the mean inflow: OPRWG = 1, FWGT = FWGSI = FWGSO = 2*1., KWGT = KWGSI = KWGSO = 4*1.

(16) OPWGT. The intended model is OPWGT = 1, which accounts for the effect of aircraft turning on the wake convection. The self-induced wake geometry distortion (prescribed or calculated) does not include the effect of the turn rate however.

(17) FK2TWG, FGMXWG. These factors allow adjustment of the hover prescribed wake model, to achieve good performance prediction. Limited
correlation has determined typical values of \( FK2TWG = 0.9 \) at moderate to high thrust; and \( FOMXWG = 1 \). Extensive correlation is needed to fully define the proper use of these parameters.

(18) Free wake geometry. The wake geometry calculation used is only valid for forward flight, and it only calculates the distortion of the tip vortices of a single rotor. Problems with accuracy or convergence may be encountered for advance ratios less than about 0.1. Note that the maximum number of azimuth stations (MPSI in namelist NLTRIM) is 24 when the free wake geometry is used.

(19) OPFWG, KFWG. The intended wake model is OPFWG = 1. At least two revolutions of wake should always be used, KFWG = 2*MPSI. At low speeds, more revolutions are required: typically about \( .4/(\text{advance ratio}) \) revolutions.

(20) Recommended values:
(a) COREWG: consistent with influence coefficient parameters (CORE)
(b) WGMODL: consistent with influence coefficient parameters (WKMODL)
(c) ITERWG = 2 or 3
(d) FACTWG = 0.5
(e) RTWG(2) = 0.4 for line
(f) MRVBWG = 2
(g) LDW = 180°/Δψ = MPSI/2
(h) NDMW = 90°/Δψ = MPSI/4 fore and aft, 45°/Δψ = MPSI/8 on the sides of the rotor disk; so for MPSI = 24, NDMW = 3*6, 6*3, 6*3, 6*3
(i) DOWG = 0.04\( \lambda \) to 0.08\( \lambda \) (large for efficiency, small for accuracy)
(j) QWDB = .5\( \lambda \) to \( \lambda \) (large to limit debug output)

6.5.5 Namelist NLBODY.
(1) CONFIG. This parameter specifies the rotorcraft configuration, which influences the following.
(a) Airframe geometry: calculation of the matrix RSF transforming from fuselage frame (F axes) to shaft frame (S axes), using ASHAFT, ACANT, and ATILT.

(b) Control system: calculation of the control matrix relating rotor controls to the pilot’s controls.

(c) Names of variables in flutter and flight dynamics analyses (see NLFLUT and NLSTAB input).

There are also calculations that are required only for the tiltrotor configuration (CONFIG = 3):

(a) Calculation of the hub position from the input pivot position and mast length (FSR1, BLR1, WLR1, and HMAST).

(b) Use of asymmetric ground effect in dynamic inflow model of flutter analysis (KASGE in namelist NLFLUT).

(c) Use of symmetric rotor azimuth perturbation in governor for flutter analysis.

(d) Symmetric/antisymmetric split of equations in flutter analysis (OPSYM in namelist NLFLUT).

(2) OPBAT, OPDRV, OPDRVU. The airframe aerodynamic model for trim uses the nonlinear equations, and optionally the airframe coefficient tables. The tables are read and used if OPBAT = 1. If only the table coefficients are to be used, not the equations, then the parameters defining the equations must be set to 0. The derivatives in the nonlinear equations always have units of per-radian.

The airframe aerodynamic model for flutter, flight dynamics, and transient analyses uses the nonlinear equations if OPDRV = 0; and uses the linearized equations if OPDRV = 1 (stability derivatives constant, from the namelist input) or OPDRV = 2 (stability derivative tables from a file). The derivatives with respect to angle-of-attack, sideslip, and control angle have units of per-radian if OPDRVU = 0, or units of
per-degree if OPDRVU = 1. The reference area, chord, and span are defined separately for the namelist input and the table input.

(3) Airframe/rotor interference. The calculated interference velocities are updated once per trim iteration; if there is no trim iteration, there is no update. When the airframe interference velocities are included, the rotor unsteady aerodynamics (nonuniform inflow model) should also be included.

A wing consists of a vortex and doublet line, defined by three points on the wing quarter-chord: the left tip, middle, and right tip (FSWING, BLWING, WLWING). The designations left, middle, and right need not refer to the actual geometry (the wing is not necessarily horizontal or symmetric). The right-hand rule for a vector from the left tip to the right tip defines the direction of positive bound circulation. The interference produced by the wing lift can be suppressed by setting the circulation to zero (WCIRC = WCIRCF = 0). The wing cross-section area can be estimated from $0.68r_w c_w^2$, where $r_w$ is the wing thickness ratio and $c_w$ is the wing chord. The interference produced by the wing thickness can be suppressed by setting the area to zero (WAXS = 0). For an airfoil-shaped body (BSHAPE = 3), the interference velocities are calculated using modified slender-body theory, which requires much more computation than for an ellipsoid or sphere (BSHAPE = 1 or 2). The velocities should not be calculated too near the surface of the airfoil-shaped body. The rotor blade position options (OPI1BP and OPI2BP) are similar to those for the rotor wake analysis (OPWKB in namelist NLWAKE). See volume I, section 4.4.

6.5.6 Namelist NLOAD.

(1) MVIB, NRVIB, MVLOAD. The airframe sensors are defined by the parameters in namelist NLOAD. MVIB is the number of sensors defined. The location of each sensor on the airframe is specified by the
parameter LOCATV. The number of possible sensor locations is NRVIB. The set of sensor locations is defined separately since more than one sensor may be located at the same fuselage point. The orientation of each sensor is specified by the parameters AXISV, AZMUTV, ELVATV, and COMPV. The coordinate system begins with x forward, y to the right, and z down; it is rotated by the azimuth angle about the z axis, and by the elevation angle about the y axis; then one of the three final axes is chosen as the sensor direction.

The harmonic responses of the first MVLOAD sensors is calculated from the periodic airframe motion. For this response to exist, the airframe degrees of freedom must be active (parameters DOF and MHARMF in namelist NLTRIM).

The flutter, flight dynamics, and transient analyses can calculate the responses for these sensors, as selected by the parameter SEN (in namelists NLFLUT, NLSTAB, and NLTRAN).

Note that the accelerometer sensors response to inertial motion of the airframe, but not to gravity.

(2) To be consistent with the harmonic content of the blade motion solution, the aerodynamic loads are smoothed when used to calculate the blade and hub loads (controlled by MRLOAD, MHLOAD, etc.). This smoothing, and hence the resulting loads, depends on the number of harmonics in the motion solution (MHARM, from namelist NLTRIM). Separate parameters in namelist NLLOAD determine the number of harmonics of the blade and hub loads to be calculated.

(3) OPBTC. There are two methods available for calculating the blade bending moments, torsion moment, and control load: by integrating the inertial and aerodynamic forces acting on the blade, and from the product of the stiffness and the modal deflection. Each of the two methods of calculating the bending moments has potential sources of
computational errors. The integrated force method obtains the structural bending moment as the difference between the inertial and aerodynamic loads. Since these nearly cancel on a rotor blade, the bending moment calculation is sensitive to small errors in the inertial and aerodynamic loads. The modal deflection method will not be accurate near a step change in the stiffness distribution (requiring a step change in the curvature, so the moment is continuous) or near a lag damper (requiring a non-zero curvature near the blade hinge).

In particular, the blade and hub moments are very sensitive to the accuracy of the modal frequencies. For a rotor with a flap hinge, the accuracy can be assessed by comparing the blade root flapwise moment ($M_x$, rotating frame) with the product of the flap hinge offset and the root vertical shear force, $eF_z$. Also, the blade section flapwise moment should be zero at the hinge. The accuracy of the modal frequencies (and hence the hub moments) depends on the number of radial stations in the integration (MRB, in namelist NLRTR).

For articulated rotors with offset hinges, correlation with measured blade bending moments has been best using the modal deflection method. Similarly, the hub moments can be obtained from the product of the flap hinge offset and the root vertical shear force. Generally the torsion moments calculated by the two methods are similar, and the control loads calculated by the two methods are nearly identical.

(4) NPSI. This parameter controls the printed time history output, and the spanwise printer-plots. It has no effect on the data written to the plot files. NPSI = MPSI is appropriate for hover. If NPSI is outside the range 1 to MPSI, then just the mean and half peak-to-peak values are obtained for the hub and blade loads (MHLOAD, MRLOAD); and for the aerodynamic output (MALOAD, MDLOAD) the nearest of 1 or MPSI is used. Hence to obtain just mean and half peak-to-peak loads, use NPSI = 0 (to also obtain all azimuths for aerodynamics

237
output), or NPSI = MPSI+1 (to obtain a single azimuth for aerodynamics output).

6.5.7 Namelists NLFLUT, NLSTAB, and NLTRAN.

(1) The parameter DOF in namelist NLFLUT selects the degrees of freedom for the flutter analysis. The rotor bending and torsion degrees of freedom are multiblade coordinates, hence occur in sets, each of which has N degrees of freedom (where N is the number of blades). As an example, for a three-bladed rotor the first three degrees of freedom are the collective and cyclic multiblade coordinates of the first bending mode (of rotor#1), the next three degrees of freedom are the multiblade coordinates of the second bending modes, and so on. The maximum total number of bending degrees of freedom is fifteen per rotor, so for a three-bladed rotor at most five bending modes can be analyzed. The first pitch/torsion mode (subscript 0) is the rigid pitch degree of freedom, and the higher modes are elastic torsion degrees of freedom. The maximum total number of pitch/torsion degrees of freedom is nine per rotor.

The parameter OPTORS in namelist NLFLUT must be consistent with DOF. The rigid pitch degrees of freedom can be dropped by setting DOF=0 for all of them. In addition, OPTORS=0 is required, order to obtain the correct control matrices for the limit of infinite control-system stiffness.

A single, independent blade can be analyzed (even if the rotor has more than one blade), by setting NBLDFL=1. With OPFDAN=1, an additional analysis path is produced: the flutter equations are quasistatically reduced to just the airframe rigid body degrees of freedom, and then are analyzed as defined by the parameters in namelist NLSTAB.

(2) For the flutter, flight dynamics, and transient analyses, the system response (output) is selected by the vectors SEN, OUTX, OUTDX,
OUTDDX, and OUTV (separately for each namelist). Dimensionless or
dimensional output is selected by the parameter OPDRES. The airframe
rigid body degrees of freedom can be relative fuselage (F) axes or
stability (V) axes (parameter OPAXFV). The airframe sensors are
selected in SEN from those defined by the parameters in namelist NLLOAD.

(3) There are three ways to specify the frequencies for the
frequency response calculation: an input list (NFREQ), calculated
scales (OPBODE), or zero (OPSTEP). Of the three methods for calculating
the frequency response (MBODE), from matrices is probably most accurate;
from poles and zeros is probably least accurate; and from modes is
probably fastest for a very large number of inputs and outputs.
Obtaining the same results three ways serves as a check on the
calculation of the zeros and modes.

(4) The zeros calculation (NSYSAN) produces the static response.
As a check on accuracy, this may be compared with the static response
calculated directly from the equations (OPSTEP).

(5) There are two methods for calculating the rms gust response:
a stochastic method, with a Markov process gust model; and integrating
the product of the gust spectrum and the transfer function (which can be
calculated three ways again). The integration method can use either a
Dryden gust spectrum or a von Karman gust spectrum, but is very
sensitive to the range and increment of the frequency scale. The
stochastic method can only use the Dryden spectrum. The required
frequency range and increment can be established by comparing the two
methods with the Dryden spectrum, and then the integration method can be
switched to the von Karman spectrum if desired. Another check is to
identify the gust variables as output parameters (OUTV); then the rms
response of the gust variables should be the input value (GRMS). For
the integration method, this integration of the gust spectrum is always performed (and printed on the last line). The system response spectra are not as smooth as the gust spectra however (because of system resonances), so accurate evaluation of the gust rms response is not sufficient to ensure that the integration of the system spectra is accurate.

Note that the rms gust response is not properly defined for a system with unstable modes.

(6) A time history response can be calculated six ways.

(a) Transient analysis (namelist NLTRAN): Runge-Kutta numerical integration of the aircraft rigid body degrees of freedom with a quasistatic rotor analysis.

(b) Flight dynamics analysis (namelist NLSTAB): Linear equations for the aircraft rigid body degrees of freedom produced using numerical perturbation of a quasistatic rotor analysis; followed by evaluation of the time history from the modes (ANTYPE(3)), or Runge-Kutta numerical integration (ANTYPE(5)).

(c) Flutter analysis (namelist NLFLUT): Linear equations for the entire system produced by analytical perturbation; followed by evaluation of the time history from the modes (ANTYPE(3)). Optionally, the system equations can be quasistatically reduced to just the aircraft rigid body degrees of freedom, and the flight dynamics analysis performed.

Evaluating the time history from the modes can be done at an arbitrary time; the results are not dependent on the time step size (DELT). The Runge-Kutta numerical integration will depend on the time step size (TSTEP); the analysis should be run with progressively smaller step sizes to define the values required for accuracy.
(7) The transient gust or control used during the numerical integration is defined by OPTRAN and related parameters. Depending on OPTRAN, the subroutine TRANCC is called to calculate the control input; the subroutine TRANCG is called to calculate the uniform gust input; or the subroutine TRANCG is called to calculate the convected gust input. In each case, the subroutine TRANCT is called to calculate the transient input or gust amplitude shape, as selected by OPHIST (eight shapes are programmed; other shapes can be obtained by modifying TRANCT).

A quasistatic feedback system can be implemented in subroutine TRANCC as required. For this purpose, the control vector VSAS is to be calculated in TRANCC. The aircraft rigid body motion has been made available in TRANCC. Both this feedback control VSAS and the self-tuning regulator feedback can be active in the transient analysis (namelist NLTRAN). For a time history calculated by the flight dynamics analysis, the feedback control VSAS is not used; and the effect of the self-tuning regulator feedback is contained in the stability derivatives (or absent).

(8) OPASE. The aeroservoelasticity analysis requires a control system definition (see volume I, section 7.4). The same definition is used for all flutter and flight dynamics tasks. The aircraft input and output variables used by the control system are defined in terms of their names. To determine the proper names to use, run the rotorcraft analysis to obtain the open-loop frequency response; the names appear as labels for the printed output. If a name identified in the control system definition is not present in the aircraft model, that control system variable is not used in the closed-loop analysis. Hence a single control system definition can be used for analysis of both symmetric and antisymmetric equations; the control loops associated with the deleted variables will not be closed.
The control system definition must be dimensional or nondimensional, depending on the parameter OPDRES. To determine the units of the aircraft input and output variables, run the rotorcraft analysis to obtain the open-loop response; the units of the variables being used are printed in the header section.

The labels of the feedforward path input variables (typically the pilot's controls) are X1, X2, X3, and X4. Note that unneeded parts of the control system can be suppressed by using NFFDCL = 0, NYCL = 0, or MHxxx = 0 as appropriate.

The behavior available from the closed-loop analysis includes the static response, frequency response, and rms gust response. Note that many of the parameters that define the open-loop response calculation (including ANTYPE(2) and ANTYPE(4)) are also used for the closed-loop response. If only the stability (open-loop Bode) is required for the closed-loop system, use OPASE = 1 (then ANTYPE only defines the open-loop response required).

The gain margin and phase margin are obtained automatically by search the open-loop transfer function for -180 deg phase and unity magnitude respectively. The true occurrences might not be found however, because the transfer function is evaluated only at discrete frequency points over a specified range; and because the phase is a multivalued function of frequency. This difficulty can be avoided by calculating the open-loop Bode (perhaps using NFFREQ and FREQ) with a smaller frequency resolution around the phase = -180 and magnitude = 1 points; typically Δω/ω = 0.005 is sufficient. See also the discussion of gain margin and phase margin in volume I, section 7.4.

(9) Self-tuning regulator. Use of the regulator is separately specified for the trim, flight dynamics, and transient analyses, by the parameters MSTR, PIDSTR, MIDSTR, RIDSTR, and EPSTR (in namelists NLTRIM, NLSTAB, NLTRAN).
For the transient analysis, it is appropriate to model the actual regulator (e.g. using a Kalman filter). The regulator is executed for MSTR iterations, without testing for convergence (there is no EPSTR for NLTRAN). Note that the Runge-Kutta integration used four force evaluations per time step (TSTEP), so there will be 4*MSTR regulator iterations per time step.

6.5.8 Baseline namelist file.

A namelist file of baseline values of the input parameters is available for use with the input file preparation program. It is assumed that all parameters are zeroed before using the baseline file.

This file contains typical values of many of the input parameters, with some notes. It is intended to be a starting point in the development of the input data for a particular aircraft and task. The baseline file is not complete, and the values given are not appropriate in all circumstances.

The forward flight version of the baseline file, BASEF.LIST, is given below. The hover version is BASEH.LIST.

$NLREAD CNTNTS=10*1,$END
$NLTRIM
! Trim analysis parameters; see the user's manual,
! sections 6.5.1 and 6.5.2.
    TITLE='BASELINE INPUT',
!
! Use OPUNIT=2 for metric units.
! Use DEBUG(1)=2 for trace of analysis progress.
! Use TRACE=4 for complete trace of convergence, TRACE=2 when
! have self-tuning regulator.
    OPUNIT=1,NPRNTI=1,TRACE=0,DEBUG=24*0,
!
! Two-rotor aircraft.
    NROTOR=2,
!
! Altitude and standard day.
    OPDENS=1,TEMP=59.,DENSE=.002378,
Azimuth increment of 15 deg.
Use MPSI=MPSI, MREV=1 with only rotor degrees of freedom.
Use MPSI=2*MPSI, MREV=2 with airframe degrees of freedom
also (DOF(33-62)).
  MPSI=24,
  DOFT=2*1, 2*0, 2*1, 2*0, MHARM=2*10, MHARMF=2*2,
  MPSI=24, MREV=1,

Trim solution. Use EPTRIM=.01 for free flight trim,
EPTRIM=.005 for symmetric free flight trim, EPTRIM=.001 for
wind tunnel trim.
  OPTRIM=1, MTRIM=20, MTRIMD=20, EPTRIM=.01,
  DELTA=1., FACTOR=.5, ALPHA=.5, OPWT2T=1,

Output during trim analysis.
  NPRNNT=1, NPRNTP=1, NPRNTL=1,

Circulation and motion convergence. Much smaller values of
EPMOTN and EPCIRC are sometimes needed for trim convergence.
  ITERM=20, EPMOTN=.02, ITERC=20, EPCIRC=.001,

Wake analysis. One cyclic each of uniform inflow, prescribed
wake, and free wake is normally sufficient and efficient.
Hover with fixed collective will require wake geometry
iteration, typically ITERR=4. Dual peak wake model (OPFW=2)
may require two iterations (ITERR=2 or ITERF=2, whichever is
the last step).
  ITERU=1, ITERR=1, ITERF=1,

Self-tuning regulator (use TRACE=2).
  MSTR=20, PIDSTR=1, RIDSTR=1, MIDSTR=0,
  DELSTR=5., ALFSTR=.5, WTZ=40*1., WTDEL=25*1.,
  FACTS=.8, EPSTR=.1,

$END
$NLRT

Rotor#1 parameters; see the user's manual, section 6.5.3.
  TITLE='BASELINE INPUT',

  For rotor#1 use TYPE = 'MAIN', 'FRNT', 'RGHT', or 'LOWR',
  for rotor#2 use TYPE = 'TAIL', 'REAR', 'LEFT', or 'UPPR',
  (as appropriate).
  TYPE='MAIN', ROTATE=1,
! Aerodynamic segments; revise as required to obtain airloads
! at specific radial stations.
MRA=19
RAE=.14,.22,.30,.37,.44,.50,.56,.61,.66,.71,
    .75,.79,.83,.86,.89,.92,.94,.96,.98,1.0,

! Aerodynamic model.
OPTIP=1,BTIP=.98,OPUSLD=2,OPCOMP=1,RGMAX=0.,
OPSTLL=1,OPYAW=0,OPREYN=0,EXPRED=.2,EXPREL=.2,
ADELAY=15.,AMAXNS=4.,TAU=3*1.,PSIDS=3*15.,ALFDS=3*15.,
ALFRE=3*12.,CLDSP=2.,CDDSP=0.,CMDSP=.65,
KHLMDA=1.1,KFILMDA=1.8,
OPFFLI=1,FXLMDA=1.,FYLMDA=1.,FMLMDA=1.,
MCORRL=30*1.,MCORRD=30*1.,MCORRM=30*1.,
LDMCFD=1.2*0,
FACTWU=.5,

! Inflow model. Turn on rotor/rotor and rotor/body
! interference as appropriate.
INFLOW=1.5*0,

! Dynamic model.
! The number of collocation functions (NCOLB or NCOLT) should
! be about twice the number of degrees of freedom used.
FACTM=1.,OPHHIB=3*1.,
NUGC=1.,NUGS=1.,LDAMPR=1.,GSB=10*.01,GST=5*.01,
RCP=1.,MBLADE=1.,MRB=50,MRM=50,
HINGE=2,EPMODE=.5,NCOLB=8,NCOLT=3,
$END
$NLWAKE
! Wake parameters for rotor#1; see the user's manual,
! section 6.5.4.

! Far wake model. Use OPFW=2 for dual-peak wake model; use
! single-peak model when valid (more efficient).
OPFW=0,

! Near wake model. Three options for blade-vortex interaction:
! 3c/4 collocation point (OPNW=3, DLS=.1.); lifting-surface
! correction (OPNW=0, DLS=.5); or larger core radius (increase
! CORE(1) by .01-.015). Use OPNW=3 for swept tip; never use
! OPNW=2. For hover use OPNW=0 and DLS=.1.
! OPNW=0,DLS=.1., ! hover
OPNW=3,DLS=.1., ! forward flight
Extent of wake: KNW at least 2 (limited by size of near wake influence coefficient matrix). KFW at least 2 revolutions (2*MPsI), typically .4/(advance ratio) revolutions.

Rolling up wake model not being used.

Axisymmetric wake geometry used for hover.

LHW=30,OPHW=0,KFW=120, ! hover
LHW=30,OPHW=1,KFW=48, ! forward flight
KNW=4, KRW=4, KDW=96,
RRU=.8, FRU=.8, PRU=60.,

Tip vortex roll-up at blade typically RTVTX = .985 to .99 for rectangular planform.

RTVTX=.985,

Core radii. Typically CORE(1)=.01-.035 with OPNW=3 or DLS=.5; or CORE(1)=.02-.05 with OPNW=0 and DLS=.1.

Suppress inboard blade-vortex interaction in forward flight.

OPTVIC=0, RTVIC=.76,.88, ! hover
OPTVIC=1, RTVIC=.76,.88, ! forward flight
CORE=.02,.1,.02,4*1.,
OPCORE=2*0,

Wake model. Delete shed wake elements for axisymmetric geometry cases (hover), unless blade loading varies with azimuth. EPVS=.01 to use nonplanar quadrilateral sheet elements.

WKMODL=2,0,2,0,2,0,2,2,3,0,3 ! hover
WKMODL=10*2,3*3 ! forward flight
OPIVT=0, OPRGI=0,
DVS=.01, EPVS=1000.,

Wake analysis.
FACTWN=.1, OPVXY=1, OPWKP=4*1, QDEBUG=1000.,

Rigid wake geometry. KRWG should be maximum of KFW and KDW.

Use OPRWG = 1 for forward flight, OPRWG = 3 (input K's), 4 (Landrege prescribed), or 6 (Kocurek and Tangler prescribed) in hover. FK2TWG and FGMMWG require calibration based on hover performance correlation.

OPRWG=6 ! hover
OPRWG=1 ! forward flight
KRWG=96, OPRWG=1,
FWGT=2*1., FWGSI=2*1., FWGSO=2*1.,
KWGT=4*1., KWGSI=4*1., KWGSO=4*1.,
FK2TWG=.9, FGMMWG=1.,
Free wake geometry. KFWG should be at least 2 revolutions
(2*MPS), typically .4/(advance ratio) revolutions.
WGMODL and COREWG should be consistent with WGMODL and CORE.

KFWG=.48,OPFWG=1,
WGMODL=2*1,COREWG=.02,2*1,
RTWG=1,4,MRVWBG=2,LDWG=3*6,6*3,6*6,6*3,3*6,
DOWG=2*.0005,ITERWG=2,FACTWG=.5,IPWGBK=2*6,FAQSC=.1,

$END

Rotor#2 parameters; see the user's manual, section 6.5.3.
TITLE='BASELINE INPUT',

For rotor#1 use TYPE = 'MAIN', 'FRNT', 'Rght', or 'LOWR',
for rotor#2 use TYPE = 'TAIL', 'REAR', 'LEFT', or 'UPPR',
(as appropriate).
TYPE='TAIL',ROTATE=1,

Aerodynamic segments; revise as required to obtain airloads
at specific radial stations.
MRA=19
RAE=14,.22,.30,.37,.44,.50,.56,.61,.66,.71,
.75,.79,.84,.86,.91,.92,.94,.96,.98,1.0,

Aerodynamic model.
OPTIP=1,BTIP=.98,OPUSLD=2,OPCOMP=1,RGMAX=0.,
OPSTLL=1,OPYAW=0,OPREY=.0,EXPRED=.2,EXPREL=.2,
ADELY=15.,AMAXN=4.,TAU=3*1.,PSIDS=3*15.,ALFDS=3*15.,
ALFRE=3*12.,CLDSP=2.,CDDSP=0.,CMDSP=.65,
KHLMDA=1.1,KFILMDA=1.8,
OPPFLI=1,FYLMDA=1.,FYLMDA=1.,FMYMDA=1.,
MCORRL=30*1.,MCORRD=30*1.,MCORRM=30*1.,
LDMCFD=1.2*0,
FACTWU=.5,

Inflow model. Turn on rotor/rotor and rotor/body
interference as appropriate.
INFLOW=1.5*0,

Dynamic model.
The number of collocation functions (NCOLB or NCOLT) should
be about twice the number of degrees of freedom used.
FACTM=1.,OPHVIB=3*1,
NUCC=1,NUGS=1.,LDAMPR=1.,GSA=10*.01,GST=5*.01,
RCPL=1.,MLAD=1.,MRB=50.,MRM=50,
HINGE=2,EPMOD=.5,NCOLB=8,NCOLT=3,

$END
$NLWAKE
!
Wake parameters for rotor#2; see the user’s manual,
! section 6.5.4.
!
Far wake model. Use OPFW=2 for dual-peak wake model; use
! single-peak model when valid (more efficient).

OPFW=0,
!
Near wake model. Three options for blade-vortex interaction:
! 3c/4 collocation point (OPNW=3, DLS=.1.); lifting-surface
! correction (OPNW=0, DLS=.5); or larger core radius (increase
! CORE(1) by .01-.015). Use OPNW=3 for swept tip; never use
! OPNW=2. For hover use OPNW=0 and DLS=.1.

OPNW=0,DLS=.1., ! hover
OPNW=3,DLS=.1., ! forward flight
!
Extent of wake: KNW at least 2 (limited by size of near wake
! influence coefficient matrix). KFW at least 2 revolutions
! (2*MPSI), typically .4/(advance ratio) revolutions.
! Rolling up wake model not being used.
Axisymmetric wake geometry used for hover.

LHW=30,OPHW=0,KFW=120, ! hover
LHW=30,OPHW=1,KFW=48, ! forward flight
KNW=4,KRW=4,KDW=96,
RRU=.8,FRU=.8,PRU=60.,
!
Tip vortex roll-up at blade typically RTVTX = .985 to .99 for
! rectangular planform.

RTVTX=.985,
!
Core radii. Typically CORE(1)=.01-.035 with OPNW=3 cr
! DLS=.5; or CORE(1)=.02-.05 with OPNW=0 and DLS=.1.
! Suppress inboard blade-vortex interaction in forward flight.

OPTVIC=0,RTVIC=.76,.88, ! hover
OPTVIC=1,RTVIC=.76,.88, ! forward flight
CORE=.02,.1,.02,4*1.,
OPCORE=2*0,
!
Wake model. Delete shed wake elements for axisymmetric
! geometry cases (hover), unless blade loading varies with
! azimuth. EPVS=.01 to use nonplanar quadrilateral sheet
! elements.

WKMODL=2,0,2,0,2,0,2,2,2,3,0,3 ! hover
WKMODL=10*2,3*3
OPIVTL=0,OPRGI=0,
DVS=.01,EPVS=1000.,
!
Wake analysis.

FACTWN=.1,OPVXVY=1,OPWKBP=4*1,QDEBUG=1000.,

Rigid wake geometry. KRWG should be maximum of KFW and KDW.
Use OPRWG = 1 for forward flight, OPRWG = 3 (input K’s), 4
(Landgrebe prescribed), or 6 (Kocurek and Tangler prescribed)
in hover. FK2TWG and FGMXWG require calibration based on
hover performance correlation.

OPRGW=6    ! hover
OPRGW=1    ! forward flight
KRWG=96,OPWG=1,
FWGT=2*1.,FWGSI=2*1.,FWGSO=2*1.,
KWGT=4*1.,KWGSI=4*1.,KWGSO=4*1.,
FK2TWG=9,FGMXWG=1.,

Free wake geometry. KFWG should be at least 2 revolutions
(2*MPSI), typically .4/(advance ratio) revolutions.
WGMODL and COREWG should be consistent with WKMODL and CORE.
KFWG=48,OPFWG=1,
WGMODL=2*1,COREWG=.02,2*-1.,
RTWG=.1,4,MRVWBG=2,LDWBG=12,NDWBG=3*6,6*3,6*3,3*6,
DQWG=2*.0005,ITERWG=2,FATWBG=.5,IPWGDB=2*6,QWGDB=.1.

$END
$NLBODY

Airframe parameters; see the user’s manual, section 6.5.5.
TITLE='BASELINE INPUT',

Single main-rotor and tail-rotor configuration.
CONFIG=1,TRATIO=1.,ENPGP=1,

Calculate the control matrix.
TCIN=0,

Unit gains between pilot’s controls and rotor controls,
so pilot’s stick deflections equal rotor controls in deg.
KOCF=1.,KCCFE=1.,KSCFE=1.,KCFCF=1.,
KFOCFE=1.,KFCFE=1.,KFSCFE=1.,KFPCFE=1.,
KRCF=1.,KRCF=1.,KRCF=1.,KRSCF=1.,KRCPF=1.,

Rotor/airframe aerodynamic interference model.
WXIRC and WXTMAX given far field values.
OPI1BP=4*1,OPI2BP=4*1,WXIRC=5*.25,WXTMAX=5*.375,

$END
$NLLOAD

Load parameters for rotor#1; see the user’s manual,
section 6.5.6.
! Vibration print; turn on with MVLOAD.
NVPRNT=1,

! Format of output; use NPSI=MPSI for hover.
NPOLAR=2,NSSPAN=25,NPSI=24, ! hover
NPOLAR=2,NSSPAN=25,NPSI=1, ! forward flight

! Hub loads; turn on with MHLOAD.
Print everything, plot dimensionless loads.
MHARMH=10,NHPRNT=4*3,NHPLOT=2*1,2*0.

! Blade loads; turn on with MRLOAD.
Print everything, plot dimensionless, principal axis loads.
Blade bending moments from modal deflection.
MHARMR=10,NRPRNT=4*3,NRPLOT=0,1,2*0,OPBTC=1,2*0,

! Blade motion and aerodynamics; turn on with MDLOAD.
Print bound circulation -- useful for nonuniform inflow.
MHARMD=10,NDPRNT=3*0,1,0,

! Rotor aerodynamics; turn on with MALOAD (and NPRNTA for
print). Polar plot of angle of attack (#1); time history
and spanwise plot of section lift d(CT/s)/dr (#49).
MHARMA=10,NAPLOT(1)=4,NAPLOT(49)=3,

! Wake geometry plot; turn on with MWAKE.
All four views, for psi = 90, 180, 270, 360 deg.
NGPLOT=1,NWAKE=4*1,JWAKE=6,12,18,24,

! Rotor noise analysis; turn on with MNOISE.
OPNOIS=4*2,NNPRNT=3,NNPLOT=1,MRN=25,RROOTN=.3,
MHARMN=25,15,25,MTIMEN=100,5,5,

$END
$NLLOAD
! Load parameters for rotor#2; see the user's manual,
! section 6.5.6.

! Vibration print; turn on with MVLOAD.
NVPRNT=1,

! Format of output; use NPSI=MPSI for hover.
NPOLAR=2,NSSPAN=25,NPSI=24, ! hover
NPOLAR=2,NSSPAN=25,NPSI=1, ! forward flight
Hub loads; turn on with MHLOAD.
Print everything, plot dimensionless loads.
MHARMH=10,NHPRNT=4*3,NHPLPT=2*1,2*0.

Blade loads; turn on with MRLOAD.
Print everything, plot dimensionless, principal axis loads.
Blade bending moments from modal deflection.
MHARMR=10,NRPRNT=4*3,NRPLTO=0,1,2*0,OPBTC=1,2*0,

Blade motion and aerodynamics; turn on with MDLOAD.
Print bound circulation -- useful for nonuniform inflow.
MHARMD=10,NDPRNT=3*0,1,0,

Rotor aerodynamics; turn on with MALOAD (and NPRNTA for print). Polar plot of angle of attack (#1); time history and spanwise plot of section lift d(CT/s)/dr (#49).
MHARMA=10,NAPLOT(1)=4,NAPLOT(49)=3,

Wake geometry plot; turn on with MWAKE.
All four views, for psi = 90, 180, 270, 360 deg.
NGPLOT=1,NWAKE=4*1, JWAKE=6,12,18,24,

Rotor noise analysis; turn on with MNOISE.
OPNOIS=4*2,NNPRNT=3,NNPLOT=1,MRN=25,RRROOTN=.3,
MHARMN=25,15,25,MTIMEN=100,5,5,

Flutter analysis parameters; see the user's manual, section 6.5.7.

Most common: constant coefficient eigenvalues.
Separate symmetric and antisymmetric analyses for tiltrotor.
OPFLOW=1,OPSYM=1,

Model. Use OPTORS(1)=0 or OPTORS(2)=0 if rigid pitch degree of freedom is absent (DOF(16)=0 or DOF(46)=0).
OPTORS=2*1,OPUSLD=2,OPDYNI=1,OPGRND=0,KASEG=1.,OPRINT=1,

Evaluation of differential equations.
DALPHA=1.,DMACH=.05,DELTA=.01,
MPSICC=12,MPSIPC=720,NINTPC=1,
! Analysis definition.
ANTIPE=1,3*0,NSYSAN=0,
OPBODE=1,MBODE=1,
OPTIME=3,
LGUST=3*0,GRMS=3*1,GMUST=0,OPSPEC=1,
FOPLT=0,F1PLT=2,NFOPLT=-2,NF1PLT=1,SCALE=1,2,1,1,15,40,
$END
$NLSTAB
! Flight dynamics analysis parameters; see the user's manual,
! section 6.5.7.
!
! Six rigid body degrees of freedom.
DOF=6*1,0,
!
! Evaluation of stability derivatives.
ITERS=3,OPLMDA=0,DELTAT=.01,
OPPRNT=3*0,1,NPRNTP=0,NPRNTL=0,
!
! Analysis definition.
EQTYPE=3*1,ANTIPE=1,4*0,NSYSAN=0,
OPBODE=1,MBODE=1,
OPTIME=3,
LGUST=3*0,GRMS=3*1,GMUST=0,OPSPEC=1,
FOPLT=0,F1PLT=2,NFOPLT=-2,NF1PLT=1,SCALE=1,2,1,1,15,40,
NPRNTT=10,NTPLOT=1,
$END
$NLTRAN
! Transient analysis parameters; see the user's manual,
! section 6.5.7.
!
! Six rigid body degrees of freedom.
DOF=6*1,0,
!
! Model.
ITERT=3,OPLMDA=0,NPRNTP=0,NPRNTL=0,
!
! Analysis definition.
NPRNTT=10,NTPLOT=1,
OPTRAN=1,OPHIST=3,
$END

252
7. SYSTEM-SPECIFIC SUBROUTINES

The following subroutines involve calls to machine-dependent functions.

File management

FILEO       Open a file
FILEC       Close a file

Alphanumeric time and date for identification

FILETD      Construct time/date file identification

CPU time

TIMEC       CPU time

Versions of these subroutines are available for the DEC VAX. Their prologues define the input and output arguments.
8. INPUT AND OUTPUT UTILITIES

8.1 Plot File Translation

The program PLOTTER is a utility that extracts data from the CAMRAD/JA-generated plot files, as well as from other input and output files. It is an interactive program, written for the DEC VAX. Because interaction with the user, file manipulation, and graphics formats are machine dependent, this program is primarily intended as a template for the development of the utility at the user's site.

The PLOTTER utility extracts data from the following CAMRAD/JA input and output files.

(a) Input file. Preparation of this unformatted file is described in section 3.
(b) Airfoil file. Preparation of this unformatted file is described in section 4.
(c) Airframe aerodynamic coefficient file. The format of this file is described in section 5.3.
(d) Airframe stability derivative file. The format of this file is described in section 5.4.
(e) Plot file (output). The generation of this file is described in sections 6.1 and 6.5.1.

Generally, after the desired data is extracted it is written to a new file, that can then be read by an appropriate graphics routine. The subroutine PLTFIL produces these new files, and its prologue describes their format. The data are sent to PLTFIL through the common PLTD. The subroutine PLTFIL can be modified as required for specific graphics routines. Exceptions are the linear-system matrices in the plot file, which are written to files in a form suitable for printing.
The data that can be extracted from the input file are the radial distributions of aerodynamic and inertial/structural blade properties, input through namelist NLRTR (see section 6.3.3). Additional quantities that can be generated are the tip loss factors (defined by BTIP and RTVTX); and the blade leading and trailing edge positions (calculated assuming that XA defines the position of the quarter-chord).

The data that can be extracted from the airfoil file are the lift, drag, or moment coefficient as a function of angle of attack or Mach number; or the drag or moment coefficient as a function of lift coefficient.

The data that can be extracted from the airframe aerodynamic coefficient file are the lift, drag, or moment coefficient, as a function of angle of attack, Mach number, or elevator angle; or the drag or moment coefficient as a function of lift coefficient. In addition, the file is checked for proper format.

The data that can be extracted from the airframe stability derivative file are any of the coefficients, as a function of angle of attack or Mach number. In addition, the file is checked for proper format.

The plot file is produced by the rotorcraft analysis, as controlled by the input parameter PLFILE in namelist NLCASE, and by input parameters NxFILE in the other namelists. The following data can be present in the plot file.
<table>
<thead>
<tr>
<th>Namelist</th>
<th>Parameter</th>
<th>Subroutine</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLTRIM</td>
<td>NTFILE</td>
<td>PERFR1,PERFR2</td>
<td>motion harmonics</td>
</tr>
<tr>
<td>NLTRIM</td>
<td>NEFILE</td>
<td>MODEP1,MODEP2</td>
<td>blade modes</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>NVFILE</td>
<td>LOADV</td>
<td>vibration</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>NGFILE</td>
<td>GEOMP1,GEOMP2</td>
<td>wake geometry</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>NDFILE</td>
<td>LOADD1,LOADD2</td>
<td>motion and aero</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>NAFILE</td>
<td>LOADA1,LOADA2</td>
<td>aerodynamics</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>NHFILE</td>
<td>LOADH1,LOADH2</td>
<td>hub loads</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>NRFILE</td>
<td>LOADS1,LOADS2</td>
<td>blade loads</td>
</tr>
<tr>
<td>NLLOAD</td>
<td>NNFILE</td>
<td>NOISR1,NOISR2</td>
<td>noise</td>
</tr>
<tr>
<td>NLFLUT</td>
<td>NLFILE</td>
<td>FLUT</td>
<td>Floquet eigenvalues</td>
</tr>
<tr>
<td>NLFLUT</td>
<td>NLFILE</td>
<td>FLUTL</td>
<td>eigenvalues</td>
</tr>
<tr>
<td>NLFLUT</td>
<td>NBFILe</td>
<td>FLUTL</td>
<td>frequency response</td>
</tr>
<tr>
<td>NLFLUT</td>
<td>NTFILE</td>
<td>FLUTL</td>
<td>linear time history</td>
</tr>
<tr>
<td>NLFLUT</td>
<td>NMFILE</td>
<td>FLUTL</td>
<td>matrices</td>
</tr>
<tr>
<td>NLSTAB</td>
<td>NLFILE</td>
<td>STABL</td>
<td>eigenvalues</td>
</tr>
<tr>
<td>NLSTAB</td>
<td>NBFILe</td>
<td>STABL</td>
<td>frequency response</td>
</tr>
<tr>
<td>NLSTAB</td>
<td>NTFILE</td>
<td>STABL</td>
<td>linear time history</td>
</tr>
<tr>
<td>NLSTAB</td>
<td>NMFILE</td>
<td>STABL</td>
<td>matrices</td>
</tr>
<tr>
<td>NLSTAB</td>
<td>NTFILE</td>
<td>STABLI</td>
<td>time history</td>
</tr>
<tr>
<td>NLTRAN</td>
<td>NTFILE</td>
<td>TRAN</td>
<td>time history</td>
</tr>
</tbody>
</table>

The plot file is written using the subroutine FILEP, which is also used by PLOTTER to read the file. The format of each section of the plot file is described in the prologue of the subroutine that reads it.

8.2 Old Input Conversion

The program OLDINPUT is a utility that converts old format CAMRAD input to CAMRAD/JA format input. It is an interactive program, written for the DEC VAX. Because file manipulation and interaction with the user are machine dependent, and because the original CAMRAD input format
may have been modified by the user, this program is primarily intended
as a template for the development of the utility at the user's site.

The CAMRAD input that can be converted by the OLDINPUT program
takes one (or all) of the following forms.

(a) Old format blockdata subroutines, which must be compiled and
linked with the OLDINPUT program.

(b) Old format input file. This file can be written by CAMRAD (if
BLKDAT=1 and CAMRAD was linked with the blockdata), or by an
appropriate utility.

(c) Old format job (namelist input). Note that if the namelist
read ignores the job control statements and namelist NLCASE,
then the old job can be read without editing.

The OLDINPUT program produces one or more files of CAMRAD/JA namelists.
These files can then be used to prepare an input file (see section 3),
or can be used directly by the rotorcraft analysis (see section 6).

The conversion process includes deletion, revision, and addition of
input parameters, as described in the prologue of the OLDINPUT program.
In the old format, the input and output variables of the linear system
analysis (for either the flutter task or the flight dynamics task) are
determined by the alphanumeric arrays NAMEVP and NAMEXP; while for
CAMRAD/JA the input and output variables are selected by the integer
arrays OUTX, OUTDX, OUTDDX, OUTV, and SEN. The OLDINPUT program does
not convert NAMEVP and NAMEXP to the CAMRAD/JA format (because the names
used are configuration dependent).

The conversion produces CAMRAD/JA input for a job that will be
nearly equivalent to the old job. The input must be further changed by
the user in order to make full use of CAMRAD/JA capabilities.
INDEX OF COMPUTER NAMES

A
A, 68, 72
A1TABL, 21
ACANT, 143, 234
ACLIMB, 110, 217
ACRIT, 70, 73
ADELAY, 125, 245, 247

AER1CM, 22
AEMNCM, 22
AEROF, 102
AEROF1, 13, 45, 211
AEROF2, 211
AERINT1, 24, 81, 95
AERINT2, 25, 81, 95
AEROS, 102
AEROS1, 13, 45, 48
AEROT1, 13, 30, 45
AER1CM, 22, 82, 91
AFDECK, 28
AFDECK1, 67
AFFILE, 24, 32, 98
AFLAP, 104, 205
AFTABLE, 27, 67
AFTABLE1, 24, 95
AFTABLE2, 24, 95
AFTBCT, 20, 27, 30
AFTBEQ, 20, 30
AFTBIC, 20, 30
AFTBIN, 21, 30
AFTBIT, 20, 30
AFTBIC, 20, 30
AFTBPC, 20, 30
AFTBPP, 21, 30
AFTBPR, 21, 30
AFTBPT, 20, 27, 30
AFTBRT, 21, 28, 30
AFTOP, 30
AIRFOIL, 20, 30
ALFD, 70, 73
ALFDS, 125, 245, 247
ALFRE, 125, 245, 247
ALFSTR, 112, 212, 222, 244
ALPHA, 109, 212, 218, 244
ALPHA5, 89, 90, 226
ALTMSL, 103, 217
AMAXH, 153

AMAXNS, 125, 245, 247
AMAXV, 153
AMAXW, 151
ANTYPE, 26, 47, 51, 100, 176, 177,
178, 180, 186, 187, 188, 191, 216,
240, 242, 252
APITC, 110, 201, 217
APRNT, 69
AROLL, 110, 217
ASHAFT, 143, 234
ASWEEP, 133, 227
ATANKP, 131, 196
ATILT, 144, 234
AUXSYM, 147, 202
AXISV, 161, 236
AXS, 170
AYAW, 110, 201, 217
AZAF, 147, 202
AZMUTH, 170
AZMUTV, 161, 236

B
BADATA, 21, 57, 93
BASEF.LIST, 59, 243
BASEH.LIST, 59, 243
BATAVL, 21, 85
BCTRIM, 111, 218
BDDATA, 21, 57, 93
BDIEL, 13, 44
BDIMS, 13, 44
BDIMSP, 13, 44
BDISP, 13, 44
BENDMODE1, 24, 75, 95
BENDMODE2, 24, 75, 95
BESSEL, 13, 41
BLAF, 147, 202
BLBODY, 158
BLCG, 145
BLHT, 145
BLKD, 258
BLNTH, 158
BLOFF, 145
BLR1, 145, 234
BLR2, 145
BLVIB, 162
BLV, 145

259
BLWB, 145
BLWING, 157, 235
BODYA, 13, 45, 102
BODYAERO, 25, 83, 95
BODYAT, 14, 45
BODYC, 14, 43, 102
BODYCM, 22
BODYD, 14, 45, 102
BODYF, 14, 43, 45, 49, 102
BODYFL, 14, 45, 102
BODYFN, 14, 45, 102
BODYI, 102
BODYII, 14, 43, 44
BODYI2, 43
BODYIIB, 14, 44, 102
BODYIW, 14, 44, 102
BODYM, 102
BODYM1, 14, 43
BODYM2, 43
BODYS, 14, 43, 102
BODYV, 102
BODYVI, 14, 43
BODYV2, 43
BPITCH, 158
BSHAPE, 158, 235
BSTRIM, 111, 218
BTHICK, 158
BTIP, 123, 230, 245, 247, 256
BYAW, 158

C
CAMRADJA, 13, 26, 31
CANTHT, 144
CANTVT, 144
CASEAN, 14, 47, 51
CASECM, 22
CDA, 154
CDAD, 154
CDEE, 154
CDDF, 154
CDDS, 125, 245, 247
CDEXT, 90
CDF, 70, 73
CDM, 154
CDOLD, 90
CDQ, 154
CDTAB, 89, 90
CFDINPUT1, 25, 89, 95
CFDINPUT2, 25, 89, 95
CFDOUTPUT1, 25, 89, 95
CFDOUTPUT2, 25, 89, 95
CHEKR1, 14, 34
CHEKR2, 34
CHORD, 133
CLA, 70, 73, 154
CLAD, 154
CLB, 155
CLDA, 155
CLDE, 154
CLDF, 154
CLDR, 155
CLDSP, 125, 245, 247
CLEXT, 90
CLF, 70, 73
CLM, 154
CLMAX, 70, 73
CLOLD, 90
CLOSE, 180, 191
CLP, 155
CLQ, 154
CLR, 155
CLTAB, 89, 90
CMA, 155
CMAC, 70, 73
CMAD, 155
CMAG, 190, 194
CMDE, 155
CMDF, 155
CMDSP, 125, 245, 247
CMEXT, 90
CMM, 155
CMOLD, 90
CMQ, 155
CMS, 70, 73
CNTAB, 89, 90
CNB, 156
CND, 156
CNDL, 156
CNP, 156
CNR, 156
CNTINS, 59, 60
CNTRLZ, 150
CNTRLZ, 204
CODE, 100
COLL, 110, 217
COMPV, 161, 236
CON, 174, 175, 184
CONE, 132, 196, 225
CONFIG, 100, 143, 144, 161, 172, 173, 233, 249
CONSTR, 114, 220, 221, 222
CONTCM, 22
CONVC, 102
CONVC1, 14, 43
CONVC2, 43
CONVCM, 22
CONVM, 102
CONVM1, 14, 43
CONVM2, 43
CONVP, 14, 38, 43
CORE, 136, 137, 213, 227, 228, 231, 233, 245, 246, 247, 248, 249
COREWG, 141, 233, 247, 249
COUNTC, 43
CONTMT, 43
COUNTS, 42
COUNTT, 37
CPTRIM, 111, 218
CSYSAN, 14, 47, 51
CSYSCM, 23
CTIME, 190, 194
CTSTZT, 127, 215, 224
CTTRIM, 111
CxEXT, 226
CxoLD, 226
CXTRIM, 111, 218
CYB, 155
CYDA, 155
CYDR, 155
CYP, 155
CYR, 155
CYTRIM, 111

D
DALPHA, 172, 251
DAPLOT, 165
DBTIME, 101
DCDDM, 70, 73
DEBUG, 101, 215, 243
DEBUG, 231, 243
DEL, 70, 73
DEL3G, 131
DELCD, 133
DELCM, 133
DELSTR, 112, 212, 222, 244
DELT, 177, 187, 240
DELTA, 109, 172, 183, 212, 244, 251, 252
DENSE, 103, 216, 217, 243
DERED, 14, 49, 51
DLS, 138, 228, 229, 231, 245, 246, 248
DMACH, 172, 251
DOF, 26, 80, 97, 104, 174, 175, 184, 186, 192, 193, 211, 216, 219, 236, 238, 244, 251, 252
DOFSYM, 146
DOFT, 104, 219, 244
DOMCFD, 139, 231
DPLL, 14, 44
DPS121, 144
DQWG, 141, 233, 247, 249
DROGDW, 151
DRGFW, 151
DRGFW, 151
DRGFW, 151
DROOPT, 132, 196, 225
DVS, 138, 231, 246, 248

E
EFLAP, 128
EIGEN, 14, 44
EIIXX, 134
EIIZZ, 134
ELAF, 147, 202
ELAG, 128
ELVATN, 170
ELVATV, 161, 236
ENDATA, 21, 57, 93
ENGNCE, 14, 43, 102
ENGNCE, 22
ENGNM, 102
ENGNM1, 14, 43
ENGNM2, 43
ENGNV, 102
ENGNV1, 14, 43
ENGNV2, 43
ENGNPOS, 159, 249
EPDATA, 105, 213, 218, 219, 244
EPDATA, 44, 128, 225, 245, 247
EPFXTN, 105, 213, 218, 219, 244
EPSTR, 112, 113, 185, 213, 222, 242, 244
EPTRIM, 109, 213, 218, 244

261
EPVS, 138, 231, 246, 248
EQTYPE, 51, 186, 252
ETA, 79
ETAP, 79
ETAPH, 79
ETAPP, 79
EXPRED, 123, 245, 247
EXPREL, 123, 245, 247

F
FOPLT, 177, 178, 179, 180, 187, 188, 189, 191, 252
F1PLT, 177, 178, 179, 180, 187, 188, 189, 191, 252
FACTM, 132, 213, 223, 245, 247
FACTOR, 109, 212, 215, 218, 244
FACTS, 112, 212, 222, 244
FACTWC, 141, 233, 247, 249
FACTWN, 138, 213, 228, 246, 249
FACTWU, 127, 213, 215, 224, 245, 247
FDROOP, 132, 196
FETAIL, 153
FGMXWG, 140, 232, 233, 246, 249
FHHC, 122
FHHH, 122
FILE, 102
FILEA1, 14, 24, 27, 30, 32
FILEA2, 24, 32
FILEB, 102
FILEB1, 14, 24, 32, 79
FILEB2, 24, 32, 79
FILEC, 14, 29, 30, 49, 253
FILED, 14, 25, 32, 86
FILEF, 14, 25, 32, 83
FILEI, 14, 24, 27, 29, 32
FILEID, 29
FILEJ, 14, 26, 31
FILEO, 15, 29, 30, 49, 253
FILEP, 15, 24, 31, 35, 36, 38, 39, 40, 41, 52, 94, 216, 257
FILETD, 15, 29, 31, 253
FILEVI, 15, 24, 32, 81
FILEV2, 25, 32, 81
FILEVI, 15, 25, 32, 89
FILEW2, 25, 32, 89
FILEX, 102
FILEX1, 15, 25, 35, 89
FILEX2, 25, 35, 89
FK2TWG, 140, 232, 233, 246, 249

FIAECM, 22
FLDATA, 21, 57, 93
FLINC, 22
FLM1CM, 22
FLMACM, 22
FLMCM, 22
FLUT, 15, 31, 47, 257
FLUTA, 102
FLUTA1, 15, 48
FLUTB, 15, 48, 49
FLUTBG, 15, 49
FLUTD1, 15, 48
FLUTF1, 15, 48
FLUTI, 102
FLUTII, 15, 48
FLUTL, 15, 47, 257
FLUTM, 15, 47, 48
FLUTMB, 15, 26, 48, 49, 102
FLUTMC, 15, 48
FLUTMD, 15, 48
FLUTME, 15, 48
FLUTMI, 15, 48
FLUTMM, 15, 48, 102
FLUTMS, 15, 26, 48, 49, 102
FLUTRI, 15, 48
FLUTR2, 48
FMLMDA, 127, 245, 247
FORCEZ, 150, 204
FREQ, 177, 180, 187, 191, 242
FRNT, 244, 247
FRU, 135, 230, 246, 248
FSAF, 147, 202
FSBODY, 158
FSCHG, 145
FSHT, 145
FSP, 145
FSR1, 145, 234
FSR2, 145
FSTALL, 70, 73
FSVIB, 162
FSVT, 145
FSWB, 145
FSWEEP, 132, 196
FSWING, 157, 235
FT0, 129
FTC, 129
FTR, 129
FWCSC, 140, 232, 246, 249
FWGSO, 140, 232, 246, 249
FWGT, 140, 232, 246, 249
FXLMDA, 127, 223, 245, 247
FYLMDA, 127, 223, 245, 247

G
G1DATA, 21, 57, 93
G2DATA, 57, 93
GAMAR, 146
GAMAR1, 146, 201
GAMAR2, 201
GAMAV, 162
GCDATA, 21, 57, 93
GDAMPC, 129
GDAMPS, 129
GDIST, 190, 195
GEOME1, 15, 40, 46
GEOMF, 102
GEOMF1, 15, 46
GEOMFS, 15, 46, 102
GEOMP1, 15, 39, 40, 257
GEOMP2, 257
GEOMR, 102
GEOMR1, 15, 46
GEOMX, 15, 47, 232
GJ, 134, 225
GMAG1, 190, 195
GRMS, 178, 180, 188, 191, 239, 252
GSB, 130, 245, 247
GSE, 159
GSI, 159
GST, 130, 245, 247
GSTALL, 70, 73
GTIME, 190, 195
GUS, 174, 175, 284
GUSTCM, 22

H
HAGL, 103
HARM, 220
HASECM, 23
HCDATA, 21, 57, 58, 93
HIMACT, 181, 191
HIMDEL, 181, 191
HIMOUT, 181, 191
HINGE, 24, 32, 44, 79, 128, 224, 245, 247
HISTPP, 15, 39, 40
HMAST, 144, 234
HREACT, 181, 191

HREDL, 181, 191
HREOUT, 181, 191
HSTALL, 70, 73
HVTAIL, 153

I
IDENT, 31
IDHACT, 181, 191
IDHDEL, 181, 191
IDHOUT, 181, 191
IENG, 159
IHT, 153
INC1CM, 22
INFILE, 24, 32, 98
INFLOW, 106, 126, 223, 245, 247
INIT, 15, 31, 34
INITA, 15, 34
INITB, 16, 34, 102
INITC, 16, 34, 102
INITE, 16, 34, 102
INITR, 102
INITR1, 16, 34
INITR2, 34
INITS, 16, 34, 102
INITV, 16, 31
INPT, 16, 24, 25, 31, 32, 79, 81, 83, 86, 89, 102
INPTB, 16, 33
INPTC, 16, 33
INPTF, 16, 33
INPTJ, 16, 31
INPTJ1, 20, 29
INPTL1, 16, 33
INPTL2, 33
INPTN, 16, 27, 29, 32, 33
INPTR1, 16, 33
INPTR2, 33
INPTS, 16, 33
INPTT, 16, 33
INPTW1, 16, 33
INPTW2, 33
INPUT, 20, 27, 29
INPUTFILE, 24, 26, 60, 95
INPUTLIST, 27, 60
INRTC, 102
INRTCL, 16, 44
INRTI, 16
INRTM, 102
INRTML, 16, 43

263
<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>INRTM2</td>
<td>43</td>
</tr>
<tr>
<td>INTERP</td>
<td>16, 30, 45</td>
</tr>
<tr>
<td>IPWDB</td>
<td>142, 247, 249</td>
</tr>
<tr>
<td>ITRC</td>
<td>43, 105, 213, 216, 244</td>
</tr>
<tr>
<td>INTERF</td>
<td>35, 36, 106, 220, 227, 244</td>
</tr>
<tr>
<td>INTERM</td>
<td>43, 105, 213, 216, 244</td>
</tr>
<tr>
<td>ITRR</td>
<td>35, 36, 106, 220, 227, 244</td>
</tr>
<tr>
<td>ITERS</td>
<td>50, 183, 252</td>
</tr>
<tr>
<td>IERT</td>
<td>52, 192, 252</td>
</tr>
<tr>
<td>ITERU</td>
<td>35, 106, 220, 244</td>
</tr>
<tr>
<td>ITERWG</td>
<td>141, 233, 247, 249</td>
</tr>
<tr>
<td>ITHETA</td>
<td>134, 225</td>
</tr>
<tr>
<td>ITMAX</td>
<td>36</td>
</tr>
<tr>
<td>IVT</td>
<td>153</td>
</tr>
<tr>
<td>IWB</td>
<td>151</td>
</tr>
<tr>
<td>IWBD</td>
<td>151</td>
</tr>
<tr>
<td>IXX</td>
<td>143</td>
</tr>
<tr>
<td>IXY</td>
<td>143</td>
</tr>
<tr>
<td>IXZ</td>
<td>143</td>
</tr>
<tr>
<td>IYY</td>
<td>143</td>
</tr>
<tr>
<td>IYZ</td>
<td>143</td>
</tr>
<tr>
<td>IZZ</td>
<td>143</td>
</tr>
<tr>
<td>J</td>
<td></td>
</tr>
<tr>
<td>JCASE</td>
<td>31</td>
</tr>
<tr>
<td>JFSI</td>
<td>43, 45</td>
</tr>
<tr>
<td>JTAG</td>
<td>112, 213, 222</td>
</tr>
<tr>
<td>JWAKE</td>
<td>163, 250, 251</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td></td>
</tr>
<tr>
<td>KOCFE</td>
<td>148, 149, 249</td>
</tr>
<tr>
<td>KACFE</td>
<td>150</td>
</tr>
<tr>
<td>KAPCFE</td>
<td>150, 204</td>
</tr>
<tr>
<td>KASCE</td>
<td>173, 234, 251</td>
</tr>
<tr>
<td>KATCFE</td>
<td>150, 204</td>
</tr>
<tr>
<td>KCCFE</td>
<td>148, 149, 249</td>
</tr>
<tr>
<td>KDW</td>
<td>46, 135, 232, 246, 248, 249</td>
</tr>
<tr>
<td>KECEFE</td>
<td>150</td>
</tr>
<tr>
<td>KEDAMP</td>
<td>159</td>
</tr>
<tr>
<td>KENG</td>
<td>159</td>
</tr>
<tr>
<td>KFOCFE</td>
<td>149, 249</td>
</tr>
<tr>
<td>KFCCFE</td>
<td>149, 249</td>
</tr>
<tr>
<td>KFCFE</td>
<td>150</td>
</tr>
<tr>
<td>KFILE</td>
<td>180, 191</td>
</tr>
<tr>
<td>KFLAP</td>
<td>128</td>
</tr>
<tr>
<td>KFLMDA</td>
<td>126, 245, 247</td>
</tr>
<tr>
<td>KFPCEFE</td>
<td>149, 249</td>
</tr>
<tr>
<td>KFSCEFE</td>
<td>149, 249</td>
</tr>
<tr>
<td>KFW</td>
<td>46, 135, 230, 232, 246, 248, 249</td>
</tr>
<tr>
<td>KFWG</td>
<td>141, 233, 247, 249</td>
</tr>
<tr>
<td>KHLMDA</td>
<td>126, 245, 247</td>
</tr>
<tr>
<td>KIC</td>
<td>159</td>
</tr>
<tr>
<td>KGOV1</td>
<td>160</td>
</tr>
<tr>
<td>KGOV2</td>
<td>160</td>
</tr>
<tr>
<td>KGOVE</td>
<td>160</td>
</tr>
<tr>
<td>KINTF</td>
<td>127</td>
</tr>
<tr>
<td>KINTH</td>
<td>127</td>
</tr>
<tr>
<td>KINTHT</td>
<td>127</td>
</tr>
<tr>
<td>KINTTV</td>
<td>127</td>
</tr>
<tr>
<td>KINTWB</td>
<td>127</td>
</tr>
<tr>
<td>KLAG</td>
<td>128</td>
</tr>
<tr>
<td>KMAST1</td>
<td>159</td>
</tr>
<tr>
<td>KMAST2</td>
<td>159</td>
</tr>
<tr>
<td>KNW</td>
<td>135, 229, 246, 248</td>
</tr>
<tr>
<td>KP2</td>
<td>134, 225</td>
</tr>
<tr>
<td>KPCFE</td>
<td>148, 149, 249</td>
</tr>
<tr>
<td>KPGOV1</td>
<td>160</td>
</tr>
<tr>
<td>KPGOV2</td>
<td>160</td>
</tr>
<tr>
<td>KPGOVE</td>
<td>160</td>
</tr>
<tr>
<td>KPIN</td>
<td>131, 196</td>
</tr>
<tr>
<td>KPMC1</td>
<td>146</td>
</tr>
<tr>
<td>KPMC2</td>
<td>146</td>
</tr>
<tr>
<td>KPM1S</td>
<td>146</td>
</tr>
<tr>
<td>KPM2S</td>
<td>146</td>
</tr>
<tr>
<td>KROCFE</td>
<td>149, 249</td>
</tr>
<tr>
<td>KRCCFE</td>
<td>149, 249</td>
</tr>
<tr>
<td>KRCFE</td>
<td>150</td>
</tr>
<tr>
<td>KRPCEF</td>
<td>149, 249</td>
</tr>
<tr>
<td>KRSCEF</td>
<td>149, 249</td>
</tr>
<tr>
<td>KRW</td>
<td>135, 230, 246, 248</td>
</tr>
<tr>
<td>KRWG</td>
<td>140, 232, 246, 249</td>
</tr>
<tr>
<td>KSCFE</td>
<td>148, 149, 249</td>
</tr>
<tr>
<td>KTO</td>
<td>129</td>
</tr>
<tr>
<td>KTC</td>
<td>129</td>
</tr>
<tr>
<td>KTCFE</td>
<td>150</td>
</tr>
<tr>
<td>KTR</td>
<td>129</td>
</tr>
<tr>
<td>KTTCFE</td>
<td>150</td>
</tr>
<tr>
<td>KWASI</td>
<td>140, 232, 246, 249</td>
</tr>
<tr>
<td>KWGSI</td>
<td>140, 232, 246, 249</td>
</tr>
<tr>
<td>KWGSO</td>
<td>140, 232, 246, 249</td>
</tr>
<tr>
<td>KWGT</td>
<td>140, 232, 246, 249</td>
</tr>
<tr>
<td>KXLMDA</td>
<td>127, 223</td>
</tr>
<tr>
<td>KYLMDA</td>
<td>127, 223</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td></td>
</tr>
<tr>
<td>LDATA</td>
<td>21, 57, 93</td>
</tr>
<tr>
<td>L2DATA</td>
<td>57, 93</td>
</tr>
<tr>
<td>LADATA</td>
<td>21, 57, 58, 93</td>
</tr>
<tr>
<td>LATCYC</td>
<td>110, 217, 218</td>
</tr>
</tbody>
</table>

264
| LDAMP1, 16, 45, 226 | LOCATV, 161, 236 |
| LDAMP2, 226 | M |
| LDAMP3, 130 | M, 68, 72 |
| LDAMP4, 130 | MALOAD, 39, 165, 226, 237, 250, 251 |
| LDAMP8, 130, 245, 247 | MASS, 130, 134 |
| LDHD1, 16, 40 | MAST, 130 |
| LDHX1, 16, 40, 42 | MAXT, 177, 187 |
| LDHX2, 42 | MBLADE, 130, 245, 247 |
| LDMDFD, 124, 226, 245, 247 | MBOSE, 177, 180, 187, 191, 239, 252 |
| LDNRCM, 22 | MCEORR, 133, 245, 247 |
| LDNGWC, 141, 233, 247, 249 | MCEORHL, 133, 245, 247 |
| LDRL1, 16, 39, 42 | MCEORRM, 133, 245, 247 |
| LDRL2, 42 | MCRIT, 70, 73 |
| LDSD1, 16, 40 | MD1CM, 22, 80 |
| LDS1, 102 | MDIV, 70, 73 |
| LDS11, 16, 40 | MDLOAD, 39, 164, 227, 237, 250, 251 |
| LDSX1, 16, 40, 42 | MGUST, 178, 180, 188, 191, 252 |
| LDSX2, 42 | MHACT, 181, 191 |
| LDVS, 16, 39, 42 | MHALARM, 105, 219, 236, 244 |
| LEVEL, 100, 106, 126, 220, 223, 226 | MHALRMA, 165, 250, 251 |
| LFTAH, 153 | MHALRMD, 164, 250, 251 |
| LFTAV, 153 | MHALRHF, 105, 220, 236, 244 |
| LFTAW, 151 | MHALRH, 168, 250, 251 |
| LTTEH, 153, 208 | MHALRMN, 170, 250, 251 |
| LFTFW, 151 | MHALRMR, 169, 250, 251 |
| LFTRU, 153, 208 | MHDEL, 181, 191 |
| LGUST, 178, 180, 188, 191, 252 | MHHAF, 100, 122, 221 |
| LHTAIL, 153 | MHHC, 100, 122, 221 |
| LHW, 135, 232, 246, 248 | MHHCF, 100, 122, 221 |
| LINTW, 123 | MHLOAD, 39, 168, 169, 236, 237, 250, 251 |
| LNGCVC, 110, 217, 218 | MHHOUT, 181, 191 |
| LOAD, 16, 35, 36, 39, 50, 52 | MIDSTR, 42, 113, 185, 193, 212, 222, 242, 244 |
| LOADAD1, 16, 39, 40, 257 | MINV, 17, 37, 42, 44, 51, 52 |
| LOADAD2, 257 | MINVC, 17 |
| LOADAM, 16, 40 | MLOOP, 58, 100, 180, 181, 191 |
| LOADAP, 17, 39, 40, 41 | MNHLCM, 22 |
| LOADD1, 17, 39, 257 | MNOISE, 39, 100, 170, 250, 251 |
| LOADD2, 257 | MNR1CM, 22 |
| LOADH, 17, 40, 41 | MNSCM, 22 |
| LOADH1, 17, 39, 40, 257 | MODE, 102 |
| LOADH2, 257 | MODEL1, 17, 43, 44, 48 |
| LOADHP, 17, 40, 41 | MODE2, 43, 48 |
| LOADM, 17, 40 | MDEA1, 17, 44 |
| LOADRI, 17, 39 | MDEB1, 17, 44 |
| LOADR2, 39 | MDEC1, 17, 44 |
| LOADS1, 17, 39, 40, 257 | MDEO1, 17, 44 |
| LOADS2, 257 | |
| LOADV, 17, 39, 257 | |
| NVFILE, 162, 257 |
| NVINT, 81 |
| NVPRNT, 162, 250 |
| NWAKE, 163, 250, 251 |
| NWING, 100, 157 |
| NxFILE, 256 |
| NYCL, 180, 181, 191, 242 |
| NZSTR, 100, 112, 222 |
| OREYN, 123, 223, 245, 247 |
| OPGI, 137, 227, 231, 248 |
| OPRINT, 173, 251 |
| OPRWG, 140, 215, 232, 246, 249 |
| OPB2EC, 178, 180, 188, 191, 252 |
| OPSTEP, 177, 180, 187, 191, 239 |
| OPSTLL, 125, 212, 214, 223, 245, 247 |
| OSTR, 100, 112, 220 |
| OSYM, 26, 146, 147, 172, 175, 234, 251 |
| OPTIN, 37, 109, 212, 218 |
| OPTIME, 177, 187, 252 |
| OPTINT, 153 |
| OPTIP, 123, 245, 247 |
| OPTORS, 173, 238, 251 |
| OPTTRAN, 190, 194, 241, 252 |
| OPTRMM, 107, 108, 109, 110, 216, 217, 244 |
| OPTVIC, 137, 231, 246, 248 |
| OPTZT, 127, 215, 224 |
| OPUNIIT, 97, 100, 243 |
| OPUSLD, 123, 172, 245, 247, 251 |
| OPVXVY, 138, 246, 249 |
| OPWGT, 138, 232, 246, 249 |
| OPWKBP, 138, 227, 231, 235, 246, 249 |
| OPWT2T, 109, 244 |
| OPYW, 125, 245, 247 |
| OUTDDX, 175, 184, 193, 239, 258 |
| OUTDX, 175, 184, 193, 238, 258 |
| OUTPUTFILE, 27, 60 |
| OUTPUTLIST, 27, 60 |
| OUTSTR, 116, 220, 222 |
| OUTV, 175, 184, 193, 239, 258 |
| OUTX, 175, 184, 193, 238, 258 |

**P**

| PCCFE, 148, 149 |
| PEDAL, 110, 217 |
| PERF, 17, 35, 36, 38, 50, 52 |
| PERFA, 17, 38 |
| PERFF1, 18, 38 |
| PERFM1, 18, 38 |
| PERF0, 18, 38 |
| PERFR, 18, 38 |
| PERFR1, 18, 38, 257 |
| PERFR2, 38, 257 |
| PERFS, 18, 38 |
| PERIOD, 177, 187 |
| PPCFE, 149 |
PFPCFE, 149
PHICFD, 139
PHIPH, 131, 196
PHIPL, 131, 196
PHIDSTR, 113, 185, 193, 212, 222, 242, 244
PLFILE, 24, 98, 216, 256
PLOTFILE, 24, 95
PLOTTER, 255, 257
PLTD, 255
PLTFIL, 255
POLRPP, 18, 40
PFCFE, 148
PRCCFE, 149
PRNTA, 18
PRNTB, 18
PRNTC, 18
PRNTF, 18
PRNTG, 18
PRNTH, 18, 31
PRNTFR, 18, 47
PRNTHS, 18, 50
PRNTHT, 18, 52
PRNTI, 18, 29, 31
PRNTJ, 18, 31
PRNTL1, 18
PRNTLA, 18
PRNTRL, 18
PRNTS, 18
PRNTT, 18
PRNTWL, 18
PRPCFE, 149
PRU, 135, 230, 246, 248
PSCFE, 148, 149
PSIDS, 125, 245, 247
PSIG, 190, 195
PSYSAN, 18, 47
Q
QBDCM, 22
QCNTRL, 146
QDAMP, 146
QDAMPF, 146
QDEBUG, 138, 246, 249
QFREQ, 146
QMASS, 146, 201
QR1CM, 22
QSTR, 112, 212, 222
QSTRAIN, 18, 49
QWGDB, 142, 233, 247, 249
R
R, 69, 72
R1DATA, 21, 57, 93
R2DATA, 57, 93
RADIUS, 121
RAE, 80, 82, 91, 133, 213, 230, 245, 247
RAMF, 18, 42, 43, 102
RANGE, 170
RCPL, 128, 245, 247
RCPLS, 128, 224
RETA, 69, 72, 133, 223
RFA, 80, 128, 196, 225
RGE, 137, 227
RCMAM, 123, 245, 247
RH1CM, 22
R1, 134, 227
RIDSTR, 42, 113, 185, 193, 212, 222
242, 244
RLOAD, 169
ROLLA, 152, 208
ROLLB, 152
ROLLP, 152
ROLLR, 152
ROTATE, 121, 244, 247
RPB, 80, 131, 196
RPH, 131, 196
RPM, 96, 103, 217
RROOT, 80
RROOTN, 170, 250, 251
RRU, 135, 230, 246, 248
RSF, 234
RT1CM, 22
RTURN, 110, 217
RTVIC, 138, 231, 246, 248
RTVXX, 135, 230, 246, 248, 256
RTWG, 141, 233, 247, 249
S
SCALE, 177, 178, 179, 180, 187, 188, 189, 191, 252
SCRATCHF.CAMRAD, 26, 95
SCRATCHJ.CAMRAD, 26, 95
SCRATCHL.CAMRAD, 26, 95
SDAREA, 154
SDCONE, 154
SDSPAN, 154
269
| SDTABL, 22, 88   | TAU, 125, 245, 247 |
| SEN, 175, 184, 192, 236, 238, 239, 258 | TCHHC, 122 |
| SENSEM, 161 | TCHHS, 122 |
| SIDEA, 152 | TCIN, 148, 204, 249 |
| SIDEB, 152, 208 | TCNTRL, 148, 204, 214 |
| SIDEP, 152 | TDAMPO, 129 |
| SIDER, 152 | TDAMPC, 129 |
| SIGMA, 121 | TDAMPR, 129 |
| SPANPP, 18, 40 | TEMP, 103, 217, 243 |
| SRDATA, 21, 57, 58, 93 | THETZL, 133 |
| STAB, 18, 31, 50 | THHC, 122 |
| STABD, 18, 47, 50 | THHS, 122 |
| STABDERIV, 25, 86, 95 | THROTL, 110, 217 |
| STABDREV, 18, 47, 49, 50 | THR DLC, 159 |
| STABE, 18, 47, 50, 51, 102 | TIMEC, 19, 253 |
| STABL, 19, 51, 257 | TIMECM, 22 |
| STABLA, 19, 51 | TIMER, 19, 31, 35, 38, 39, 43, 44, 45, 46, 47, 50, 51, 52, 102 |
| STABLI, 19, 51, 257 | TITLE, 68, 100, 121, 143, 243, 244, 247, 249 |
| STABLS, 19 | TMAX, 51, 52, 19C, 194 |
| STABM, 19, 50, 102 | TMCDATA, 21, 57, 53 |
| STABMC, 19, 50, 102 | TMMTRX, 112, 113, 221 |
| STABMM, 19, 50, 102 | TNDATA, 21, 57, 53 |
| STABMP, 19, 50 | TRACE, 101, 211, 215, 222, 243 |
| STABP, 19, 51 | TRAN, 19, 31, 52, 257 |
| STABPI, 19, 51 | TRANC, 19, 51, 52 |
| STABS, 51 | TRANCC, 19, 52, 241 |
| STDATA, 21, 57, 93 | TRANCG, 19, 52, 241 |
| STDCM, 23 | TRANCM, 23 |
| STMCM, 23 | TRANCT, 19, 52, 241 |
| STRCM, 22 | TRANI, 19, 52, 1C2 |
| STRI, 19, 37, 42, 50, 52, 102, 211 | TRANP, 19, 52 |
| STRIC, 19, 34, 42 | TRANPI, 19, 52 |
| STRIJ, 19, 42 | TRANS, 19, 52 |
| STRIT, 19, 34, 42 | TRATIO, 103, 143, 249 |
| STRIZ, 19, 42 | TRCKPP, 19, 51, 52 |
| SWEEP, 132, 196, 225 | TRIM, 19, 25, 31, 35, 89 |
| T | TRIMC, 22 |
| TOHHC, 122 | TRIMI, 19, 36, 37, 102, 211 |
| TOHHS, 122 | TRIMIC, 19, 37 |
| TIC, 218 | TRIMIM, 19, 35, 36, 37 |
| T1GOV1, 160 | TRIMIT, 19, 35, 36, 37 |
| T1GOV2, 160 | TRIMIV, 20, 37 |
| T1GOVE, 160 | TRIMP, 20, 35, 36 |
| T1S, 218 | TRIMW, 20, 35, 36 |
| T2GOV1, 160 | TSHHC, 122 |
| T2GOV2, 160 | TSHHS, 122 |
| T2GOVE, 160 | TSPRNG, 128, 224 |
| TSTEP, 51, 52, 190, 194, 240, 243 | WG1CM, 22 |
| TVCL, 180, 191 | WGMODL, 141, 233, 247, 249 |
| TWISTA, 123, 133 | WICORD, 157 |
| TWISTI, 123, 134 | WISPAN, 157 |
| TWISTL, 123 | WK1CM, 22 |
| TYPE, 121, 244, 247 | WKMODL, 137, 227, 231, 246, 247, 248, 249 |
| TYPEV, 161 | WKPAX, 102 |
| TZERO, 112, 221 | WKPAx1, 20, 46 |
| U | WKPFW, 102 |
| UNITNO, 22 | WKPFW1, 20, 46 |
| | WKNW1, 20, 46 |
| V | WKPW, 102 |
| VEL, 96, 97, 103, 217 | WKPW1, 20, 46 |
| VELG, 190, 195 | WKPRU, 102 |
| VINTP, 81, 82, 167 | WKPRU1, 20, 46 |
| VINTR, 81, 82, 167 | WKV1CM, 22 |
| VINTT, 81, 82, 167 | WLAF, 147, 202 |
| VINTX, 81, 82, 167 | WLBODY, 158 |
| VINTY, 81, 82, 167 | WLCG, 145 |
| VINTZ, 81, 82, 167 |WLHT, 145 |
| VKTS, 96, 97, 103, 217 | WLOFF, 145 |
| VSAS, 241 | WLR1, 145, 234 |
| VTIP, 96, 103, 217 | WLR2, 145 |
| VTIPN, 103, 121, 129, 217 | WLVB, 162 |
| VTXL, 20, 44, 46, 47, 102 | WLVT, 145 |
| VTXS, 20, 46, 47, 102 | WLWB, 145 |
| W | WLWING, 157, 235 |
| W1DATA, 21, 57, 93 | WRCORE, 157 |
| W2DATA, 57, 93 | WTDILT, 112, 212, 221, 222, 244 |
| WAKEBI, 20, 44, 46 | WTMN, 129 |
| WAKEB2, 46 | WTZ, 112, 221, 222, 244 |
| WAKEC, 102 | WXGIRC, 157, 249 |
| WAKECl, 20, 35, 36, 46, 50, 52 | WXTHCK, 157, 249 |
| WAKEC2, 35, 36, 50, 52 | X |
| WAKEN, 102 | XA, 133, 135, 225, 227, 256 |
| WAKEN1, 20, 43 | XAC, 133, 225 |
| WAKEN2, 43 | XC, 134, 225 |
| WAKEU, 102 | XCFD, 139 |
| WAKEU1, 20, 43 | XFA, 132, 196, 225 |
| WAKEU2, 43 | XI, 134, 225 |
| WAKEX, 102 | XIT, 130 |
| WAKEXI, 20, 35 | XPH, 131, 196 |
| WAKEX2, 35 | XTRIM, 111, 218 |
| WAXS, 157, 235 | Y |
| WCIRC, 157, 235 | YAWA, 152 |
| WCIRC, 157, 235 | YAWB, 152, 208 |
| WEIGHT, 97, 143 | YAWP, 152 |

271
YAWR, 152

Z
ZEROS.LIST, 59, 61
ZETAAF, 147, 202
ZETAR, 146
ZETAR1, 146, 201
ZETAR2, 201
ZETAV, 162
ZFA, 132, 196, 225
ZTARG, 112, 221