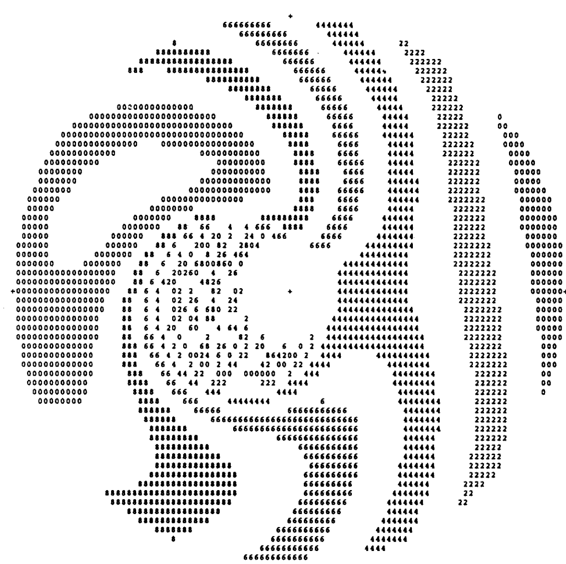
CAMRAD/JA

A COMPREHENSIVE ANALYTICAL MODEL OF ROTORCRAFT AERODYNAMICS AND DYNAMICS

Johnson Aeronautics Version

Volume II : User's Manual



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

Volume II: User's Manual

Wayne Johnson

1988

CAMRAD/JA

A COMPREHENSIVE ANALYTICAL MODEL OF ROTORCRAFT AERODYNAMICS AND DYNAMICS

Johnson Aeronautics Version

Volume II: User's Manual

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

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

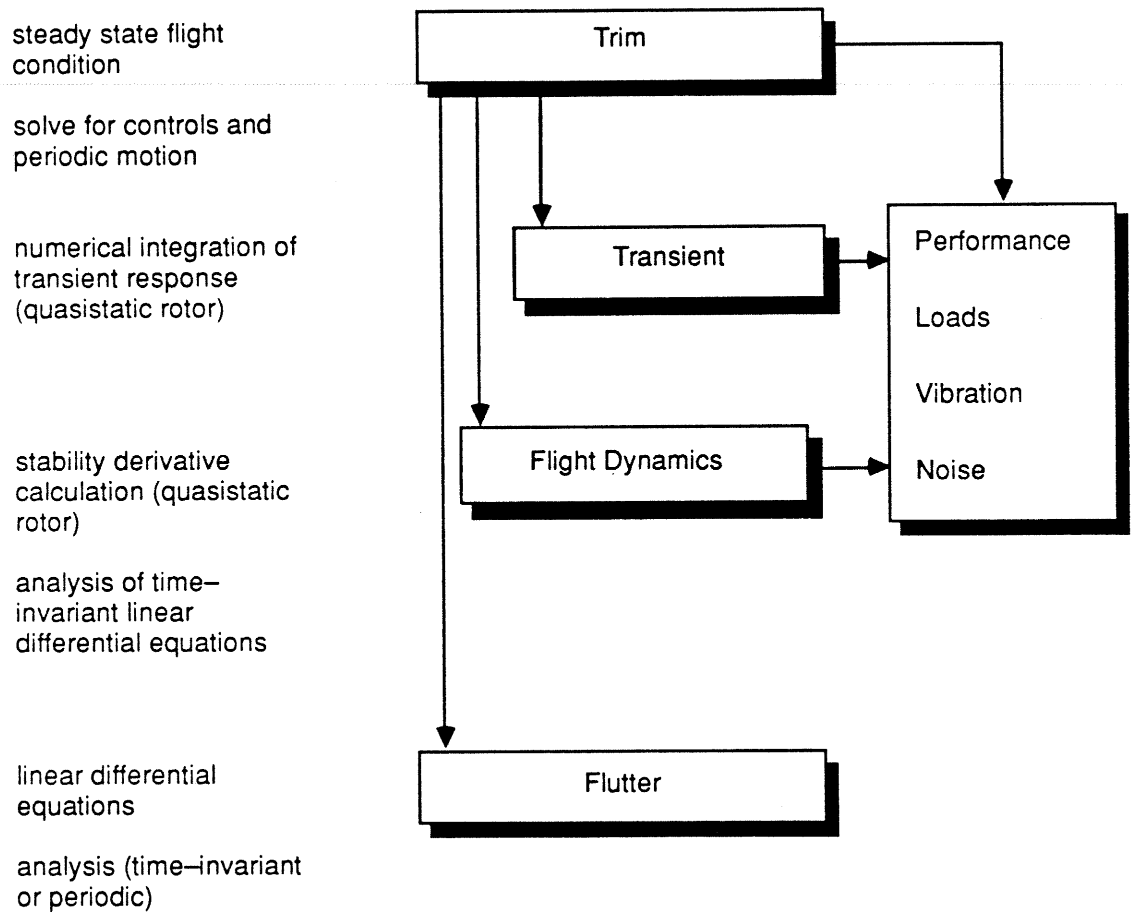


Figure 1-1. CAMRAD/JA tasks and solutions.

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

TRIM

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, gimballed, 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 modes 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 used 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

Name	Function
CAMRADJA	PRIMARY JOB AND ANALYSIS CONTROL
AEROF1	CALCULATE BLADE AERODYNAMIC FORCES
AEROS1	CALCULATE BLADE SECTION AERODYNAMIC COEFFICIENTS
AEROT1	INTERPOLATE AIRFOIL TABLES
BDIEL	PERTURBATION VELOCITIES OF ELLIPSOID
BDIMS	PERTURBATION VELOCITIES OF BODY OF REVOLUTION
BDIMSP	CALCULATE MODIFIED SLENDER BODY THEORY PARAMETERS
BDISP	PERTURBATION VELOCITIES OF SPHERE
BESSEL	CALCULATE J BESSEL FUNCTION
BODYA	CALCULATE BODY AERODYNAMIC FORCES

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
FILEI	READ OR WRITE INPUT FILE
FILEJ	READ OR WRITE JOB SCRATCH FILE

FILEO	OPEN FILE (VAX VERSION)
FILEP	READ OR WRITE PLOT DATA FILE
FILETD	CONSTRUCT TIME/DATE 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
INRTI	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

LOADAP	PRINT AND WRITE TO PLOT FILE MOTION AND AERODYNAMICS
LOADD1	CALCULATE AND PRINT ROTOR MOTION (FUNCTION OF PSI)
LOADH	CALCULATE MEAN AND HALF PEAK TO PEAK
LOADH1	PRINT AND PLOT HUB AND CONTROL LOADS
LOADHP	PRINT OR WRITE TO PLOT FILE HUB AND BLADE LOADS
LOADM	CALCULATE MEAN AND HALF PEAK TO PEAK
LOADR1	ROTOR LOADS AND NOISE
LOADS1	PRINT AND PLOT BLADE SECTION LOADS
LOADV	PRINT AND PLOT AIRFRAME VIBRATION
MINV	CALCULATE INVERSE OF MATRIX
MINVC	CALCULATE INVERSE OF COMPLEX MATRIX
MODE1	BLADE MODES
MODEA1	CALCULATE ARTICULATED BLADE FLAP AND LAG MODES
MODEB1	CALCULATE BLADE BENDING MODES
MODEC1	INITIALIZE BLADE MODE PARAMETERS
MODED1	CALCULATE BLADE ROOT GEOMETRY
MODEG	CALCULATE GALERKIN FUNCTIONS FOR BENDING MODES
MODEK1	CALCULATE KINEMATIC PITCH-BENDING COUPLING
MODEP1	PRINT BLADE MODES
MODER1	RADIAL STATIONS FOR BENDING AND TORSION MODES
MODET1	CALCULATE BLADE TORSION MODES
MOTNB1	CALCULATE BLADE AND HUB MOTION
MOTNC1	INITIALIZE ROTOR PARAMETERS AT TRIM
MOTNF1	CALCULATE ROTOR GENERALIZED FORCES
MOTNH1	CALCULATE HARMONICS OF HUB MOTION
MOTNR1	CALCULATE HARMONICS OF ROTOR MOTION
MOTNS	CALCULATE STATIC ELASTIC MOTION
NOISF1	CALCULATE FAR FIELD ROTATIONAL NOISE
NOISR1	PRINT AND PLOT FAR FIELD ROTATIONAL NOISE
NOIST	CALCULATE NOISE TIME HISTORY AND SPECTRUM
PERF	PERFORMANCE
PERFA	CALCULATE AND PRINT AIRFRAME PERFORMANCE

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
QSTRAN	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

STABL	ANALYZE FLIGHT DYNAMICS LINEAR EQUATIONS
STABLA	ACCELERATION FOR NUMERICAL INTEGRATION OF TRANSIENT
STABLI	FLIGHT DYNAMICS NUMERICAL INTEGRATION OF TRANSIENT
STABLS	CALCULATE RESPONSE FOR NUMERICAL INTEGRATION OF TRANSIENT
STABM	CALCULATE FLIGHT DYNAMICS STABILITY DERIVATIVES AND MATRICES
STABMC	PERTURB VARIABLES FOR STABILITY DERIVATIVE CALCULATION
STABMM	CONSTRUCT FLIGHT DYNAMICS MATRICES
STABMP	PRINT DURING STABILITY DERIVATIVE CALCULATION
STABP	PRINT FLIGHT DYNAMICS TRANSIENT SOLUTION
STABPI	PRINT FLIGHT DYNAMICS TRANSIENT HEADER
STRI	SELF-TUNING REGULATOR ITERATIVE SOLUTION
STRIC	CALCULATE GAIN MATRIX FOR SELF-TUNING REGULATOR
STRIJ	CALCULATE COST FUNCTION FOR SELF-TUNING REGULATOR
STRIT	CALCULATE CONTROLS FOR SELF-TUNING REGULATOR
STRIZ	CALCULATE OUTPUT FOR SELF-TUNING REGULATOR
TIMEC	CPU TIME (VAX VERSION)
TIMER	PROGRAM TIMER
TRAN	TRANSIENT
TRANC	CALCULATE TRANSIENT GUST OR CONTROL
TRANCC	CALCULATE TRANSIENT CONTROL
TRANCG	CALCULATE TRANSIENT GUST
TRANCT	CALCULATE TIME HISTORY OF TRANSIENT CONTROL/GUST
TRANI	CALCULATE TRANSIENT ACCELERATION FOR NUMERICAL INTEGRATION
TRANP	PRINT TRANSIENT SOLUTION
TRANPI	PRINT TRANSIENT HEADER AND OUTPUT DEFINITION
TRANS	CALCULATE RESPONSE FOR TRANSIENT ANALYSIS
TRCKPP	PRINTER-PLOT TIME HISTORY
TRIM	TRIM
TRIMI	CALCULATE TRIM SOLUTION BY ITERATION
TRIMIC	INCREMENT CONTROL FOR TRIM SOLUTION
TRIMIM	DEFINE CONTROLS AND TESTS FOR TRIM OPTIONS
TRIMIT	CALCULATE FORCES FOR TRIM SOLUTION

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
WKPAX1	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

Name	Function
INPUT	INPUT FILE PREPARATION
INPTJI	READ JOB NAMELIST FOR INPUT PROGRAM

Airfoil File Preparation Subprograms

Name	Function
AIRFOIL	AIRFOIL TABLE PREPARATION
AFTBPT	READ AND PRINT AIRFOIL TABLE PARAMETERS
AFTBIT	NAMELIST READ OF AIRFOIL TABLE PARAMETERS
AFTBCT	CONSTRUCT NEW AIRFOIL TABLE
AFTBPC	READ AND PRINT AIRFOIL EQUATION PARAMETERS
AFTBIC	NAMELIST READ OF AIRFOIL EQUATION PARAMETERS
AFTBOC	PRINT OF AIRFOIL EQUATION PARAMETERS
AFTBEQ	SECTION AERODYNAMIC CHARACTERISTICS FROM EQUATIONS

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

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

SDTABL	Airframe stability derivative table
UNITNO	Input/output unit numbers
CASECM	Job description
TIMECM	Timer statistics
TRIMCM	Calculated trim data
STRCM	Self-tuning regulator data
RTR1CM	Calculated rotor data
RH1CM	Transfer function matrices
BODYCM	Calculated airframe data
ENGNCM	Calculated drive train data
GUSTCM	Gust and transient control
CONTCM	Aircraft controls and motion
CONVCM	Circulation and motion convergence
MD1CM	Blade modes
INC1CM	Rotor inertial coefficients
WKV1CM	Induced velocity
MNH1CM	Hub motion
AES1CM	Blade section aerodynamics
MNR1CM	Rotor motion and airframe vibration
MNSCM	Static elastic motion
AEF1CM	Rotor forces
QR1CM	Rotor generalized forces
QBDCM	Airframe generalized forces
WG1CM	Wake geometry
WKC1CM	Wake influence coefficients
AEMNCM	Calculated motion data
LDMNCM	Calculated loads data
FLMCM	Flutter matrices
FLM1CM	Flutter rotor matrices
FLMACM	Flutter airframe matrices
FLINCM	Flutter inertial coefficients
FLAECM	Flutter aerodynamic coefficients

STDCM	Flight dynamics stability derivatives
STMCM	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: ---

A4)	File logical name:	INPUTFILE
	Function:	input data
	Unit number variable:	NFDAT (namelist NLCASE)
	Unit number default value:	40
	Subroutine using file:	INPT (through FILEI)
	Parameter controlling use:	INFILE (namelist NLCASE)
A5)	File logical name:	AFTABLE1
	Function:	rotor#1 airfoil table
	Unit number variable:	NFAF1 (namelist NLCASE)
	Unit number default value:	41
	Subroutine using file:	INPT (through FILEA1)
	Parameter controlling use:	AFFILE (namelist NLCASE)
A6)	File logical name:	AFTABLE2
	Function:	rotor#2 airfoil table
	Unit number variable:	NFAF2 (namelist NLCASE)
	Unit number default value:	42
	Subroutine using file:	INPT (through FILEA2)
	Parameter controlling use:	AFFILE (namelist NLCASE), NROTOR
A7)	File logical name:	PLOTFILE
	Function:	plot data
	Unit number variable:	NFPLT (namelist NLCASE)
	Unit number default value:	43
	Subroutine using file:	any (through FILEP)
	Parameter controlling use:	PLFILE (namelist NLCASE)
A8)	File logical name:	BENDMODE1
	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)
A9)	File logical name:	BENDMODE2
	Function:	rotor#2 bending modes
	Unit number variable:	NFBND2 (namelist NLCASE)
	Unit number default value:	62
	Subroutine using file:	INPT (through FILEB2)
	Parameter controlling use:	HINGE (namelist NLRTR)
A10)	File logical name:	AEROINT1
	Function:	rotor#1 body interference vel
	Unit number variable:	NFINT1 (namelist NLCASE)
	Unit number default value:	63
	Subroutine using file:	INPT (through FILEV1)
	Parameter controlling use:	OPINTV (namelist NLBODY)

A11)	File logical name:	AEROINT2
	Function:	rotor#2 body interference vel
	Unit number variable:	NFINT2 (namelist NLCASE)
	Unit number default value:	64
	Subroutine using file:	INPT (through FILEV2)
	Parameter controlling use:	OPINTV (namelist NLBODY)
A12)	File logical name:	BODYAERO
	Function:	airframe aero coefficients
	Unit number variable:	NFBAT (namelist NLCASE)
	Unit number default value:	65
	Subroutine using file:	INPT (through FILEF)
	Parameter controlling use:	OPBAT (namelist NLBODY)
A13)	File logical name:	STABDERIV
	Function:	airframe stability derivatives
	Unit number variable:	NFDRV (namelist NLCASE)
	Unit number default value:	66
	Subroutine using file:	INPT (through FILED)
	Parameter controlling use:	OPDRV (namelist NLBODY)
A14)	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)
A15)	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)
A16)	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)
A17)	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)

A18) File logical name: SCRATCHJ.CAMRAD
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: SCRATCHL.CAMRAD
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: SCRATCHF.CAMRAD
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: INPUTFILE
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:	NFNLIO (namelist NLJOB)
	Unit number default value:	40 (NFNLI = NFNLIO+n = 41 to 49)
	Subroutine using file:	INPUT (through INPTN)
	Parameter controlling use:	NLISTI (namelist NLJOB)
B5)	File logical name:	OUTPUTFILE
	Function:	output data file
	Unit number variable:	NFDATO (namelist NLJOB)
	Unit number default value:	50
	Subroutine using file:	INPUT (through FILEI)
	Parameter controlling use:	NFILEO (namelist NLJOB)
B6)	File logical name:	OUTPUTLIST
	Function:	output namelist data
	Unit number variable:	NFNLO (namelist NLJOB)
	Unit number default value:	51
	Subroutine using file:	INPUT (through INPTN)
	Parameter controlling use:	NLISTO (namelist NLJOB)

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)

C4)	File logical name:	AFDECKn (n = 1 to 10)
	Function:	C81 format file
	Unit number variable:	NFILEI (namelist NLTABL)
	Unit number default value:	51 (NFILEN=NFILEI-1+n= 51 to 60)
	Subroutine using file:	AFTBRD
	Parameter controlling use:	OPREAD, NRB (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.

INPUT

```
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

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 OPREAD = 0, then
         call FILEA1 (read airfoil file)
    if OPREAD = 0 and AFTOP = 1, then
         call AFTBOC
call AFTBCT
    if OPREAD ne 0, then construct new table
         for JR = 1 to NRB (radial segments)
              if OPREAD ne 2, then
                   call AFTBPC
                        call AFTBIC (read namelist NLCHAR)
                        call AFTBOC
              if OPREAD = 2, then
                   call AFTBRD
                        call FILEO
                        read C81 airfoil file
                        call FILEC
              angle of attack iteration
                   Mach number iteration
                        if OPREAD ne 2, then
                             call AFTBEQ (equations)
                        if OPREAD = 2, then
                             call AFTBIN (C81 airfoil file)
                                  call INTERP
         call FILEA1 (write airfoil file)
call AFTBPR
    if OPPRNT(1) ne 0, then interpolate and print airfoil data
         call AEROT1
    if OPPRNT(2) ne 0, then interpolate and plot airfoil data
         call AEROT1
         call AFTBPP
    if OPPRNT(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)
```

INPT

```
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 NLLOAD)

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

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
```

TRIMW

```
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 PERFM1
    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 NOISR1
```

LOADD1

```
call FILEP
call LOADAP
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 LDSI1
    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
```

STRI

```
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
```

STRIZ

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

STRIC

```
call MINV
```

RAMF

```
call TIMER
call BODYC
call BODYI1
call MOTNC1
call MODE1
call BODYI2
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 ENGNC
          call ENGNM1
          call ENGNM2

          for 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 WAKEB1
  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 LDAMP1

call TIMER
```

BODYF

```
call BODYFN

     call BODYA

          call BODYAT

               call INTERP

call BODYFL

     call BODYD

          call INTERP
```

WAKEC1

```
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

     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 NPRNTP 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 PRNTHT
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 WAKEC1
              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 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."

CAMRAD/JA Namelists

Label	Data	Commons
NLTRIM	Job and trim	TMDATA,SRDATA
NLRTR	Rotor#1	R1DATA
NLWAKE	Wake, rotor#1	G1DATA,W1DATA
NLRTR	Rotor#2	R2DATA
NLWAKE	Wake, rotor#2	G2DATA,W2DATA
NLBODY	Airframe and drive train	BDDATA,BADATA,ENDATA
NLLOAD	Loads, airframe and rotor#1	LADATA,L1DATA
NLLOAD	Loads, airframe and rotor#2	LADATA,L2DATA
NLFLUT	Flutter	FLDATA,HCDATA
NLSTAB	Flight dynamics	STDATA,GCDATA,HCDATA
NLTRAN	Transient	TNDATA,GCDATA

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 OPSTR 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.

logical name	unit number	format	use
INPUTFILE	NFDATI	input file	read
INPUTLIST1	NFNLIO+1	namelist	read
..	..	..	..
INPUTLIST9	NFNLIO+9	namelist	read
OUTPUTFILE	NFDATO	input file	written
OUTPUTLIST	NFNLO	namelist	written

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)	NLBODY	
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 hela.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.

Job description

NPRINT	integer; 1 to print all parameters; 0 to not; default 1
NFILEI	integer; 1 to read input file, 0 to not; default 0
NLISTI	integer; n to read n namelist files (maximum 9), 0 to not; default 0
NFILEO	integer; 1 to write input file, 0 to not; default 1
NLISTO	integer; 1 to write namelist file, 0 to not; default 0

NLISTC(10) integer; specification of contents of output namelist file (NLTRIM always present); 0 to suppress read; default 0

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

Input and output unit numbers

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

NFNLIO integer; unit number for read of namelist files; default 40; unit number of nth namelist file is NFNLIO+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 α, 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 α at the boundaries between the groups, and the indices of these points: N_a, α_{n_1} to $\alpha_{n_{Na}}$, and n_1 to n_{Na} (for N_a - 1 groups). The organization is illustrated below.

++++++++++++++++++++++++++++++++						angle-of-attack points
+	+	+	+	+	+	boundaries of groups
1	2	3	4	5 ...	N_a	boundary index
n_1	n_2	n_3	n_4	n_5 ...	n_{Na}	angle-of-attack index at boundary
α_1	α_2	α_3	α_4	α_5 ...	α_{Na}	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 n_{Na} is the total number of values in the table. The angle-of-attack range in the table should be from -180 to 180 deg

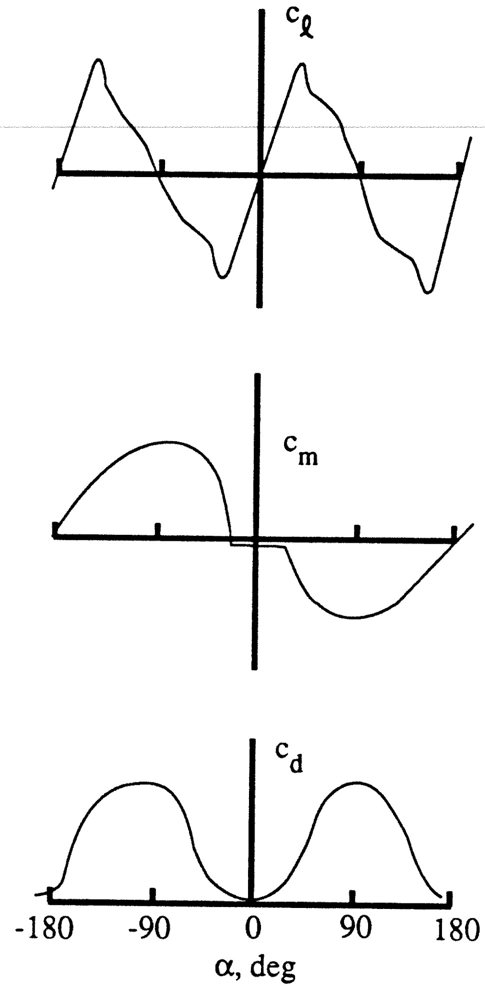


Figure 4-1. Sketch of section aerodynamic characteristics.

(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: N_r, and r_1 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 Re_{t1} is specified, corresponding to Mach number = 1. Then the Reynolds number of the airfoil data at Mach number M is $Re = M\ Re_{t1}$. This parameter will be used for the Reynolds number correction of the airfoil table by the rotorcraft analysis. The value of Re_{t1} can be superseded by input directly to the rotorcraft analysis. If $Re_{t1} = 0$, 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.

logical name	unit number	format	use
AFTABLE	NFILEO	CAMRAD/JA	read (OPREAD=0)
AFDECK1	NFILEI	C81	read (OPREAD=2)
..	..	..	..
AFDECK10	NFILEI+9	C81	read (OPREAD=2)
AFTABLE	NFILEO	CAMRAD/JA	written (OPREAD=1 or 2)

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
$DEFINE/USER_MODE SYS$OUTPUT af.out
$RUN AIRFOIL
 &NLTABL OPREAD=2,TITLE='.....',NRB=2,R=0.,.5,1.,RETAB1=...,&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
$DEFINE/USER_MODE SYS$OUTPUT af.out
$RUN AIRFOIL
 &NLTABL OPREAD=1,TITLE='.....',RETAB1=...,&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.

```
$ASSIGN af.tab AFTABLE
$DEFINE/USER_MODE SYS$OUTPUT af.out
$RUN AIRFOIL
 &NLTABL OPREAD=0,&END
```

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 tables 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; α 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_r; maximum 10
R(NRB+1)	real; boundaries of segments; R(1)=0., R(NRB+1)=1.
RETAB1(NRB)	real; Reynolds number Re_{t1} of airfoil table for M = 1; Re = M Re_{t1} for Mach number = M

Note: maximum NAB*NMB*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; $a = c_{l\alpha}$ at $M = 0$ (per rad)
MDIV	real; lift divergence Mach number M_{div}
CLMAX	real; c_{lmax} at $M = 0$
FSTALL	real; factor f_s for c_{lmax}
MSTALL	real; Mach number M_s for c_{lmax}
GSTALL	real; factor g_s for stall c_l
HSTALL	real; factor h_s for stall c_l
CLF	real; c_{lf} for stall c_l
CMAC	real; $c_{m_{ac}}$
CMS	real; c_{m_s}
DEL0	real; δ_0
DEL1	real; δ_1
DEL2	real; δ_2
DCDDM	real; $\partial c_d/\partial M$
MCRIT	real; critical Mach number at $\alpha = 0$
ACRIT	real; α_{crit} where critical Mach number is zero
ALFD	real; drag stall angle (deg)
CDF	real; c_{d_f} 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:

		Read Format
HEADER	NML,NAL,NMD,NAD,NMM,NAM	A30,6I2
	ML(1).......ML(NML)	7X,9F7.0
AL(1)	CL(1,1).....CL(1,NML)	10F7.0/(7X,9F7.0)
..	.. ..	..
AL(NAL)	CL(NAL,1)...CL(NAL,NML)	10F7.0/(7X,9F7.0)
	MD(1).......MD(NMD)	7X,9F7.0
AD(1)	CD(1,1).....CD(1,NMD)	10F7.0/(7X,9F7.0)
..	.. ..	..
AD(NAD)	CD(NAD,1)...CD(NAD,NMD)	10F7.0/(7X,9F7.0)
	MM(1).......MM(NMM)	7X,9F7.0
AM(1)	CM(1,1).....CM(1,NMM)	10F7.0/(7X,9F7.0)
..	.. ..	..
AM(NAM)	CM(NAM,1)...CM(NAM,NMM)	10F7.0/(7X,9F7.0)

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:

NAB	6
NA	1,16,28,88,100,115
A	-180.,-150.,-30.,30.,150.,180.
NMB	3
NM	1,7,21
M	0.,.6,.95
NRB	1
R	0.,1.
RETAB1	0.

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:

CLA	5.7
MDIV	0.75
CLMAX	1.2
FSTALL	0.5
MSTALL	0.4
GSTALL	1.2
HSTALL	0.6
CLF	1.12
CMAC	0.
CMS	-0.07
DEL0	0.0084
DEL1	-0.0102
DEL2	0.384
DCDDM	0.65
MCRIT	0.8
ACRIT	33.
ALFD	12.
CDF	2.05

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

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

$$c_l = c_{l\alpha}\alpha$$

$$c_m = c_{m_{ac}}$$

$$c_d = \delta_0 + \delta_1\alpha + \delta_2\alpha^2 + \Delta c_d$$

$$\Delta c_d = \max(0,\ \partial c_d/\partial M(M-M_c))$$

$$M_c = \max(0,\ M_{crit}(1-|\alpha|/\alpha_{crit}))$$

B. Stall angle

$$c_{ls} = c_{l\max} \min\left(1,\ \frac{(1-M) + f_s(M-M_s)}{1-M_s}\right)$$

$$\alpha_s = c_{ls}/c_{l\alpha}$$

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

$$c_l = \text{sign}(\alpha) \max\left[\frac{(g_s\alpha_s - |\alpha|)c_{ls} + (|\alpha| - \alpha_s)h_s c_{ls}}{g_s\alpha_s - \alpha_s},\ \max(h_s c_{ls},\ c_{lf}\sin 2|\alpha|)\right]$$

$$c_l = c_{lf}\sin 2\alpha \qquad \text{if } |\alpha| > 45^\circ$$

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

$$c_m = \begin{cases} \text{sign}(\alpha)\ \dfrac{(60 - |\alpha|)c_{m_s} + (|\alpha| - \alpha_s).75c_{m_f}}{60 - \alpha_s} & |\alpha| < 60^\circ \\ \text{sign}(\alpha)\ \dfrac{(90 - |\alpha|).75c_{m_f} + (|\alpha| - 60)c_{m_f}}{30} & |\alpha| > 60^\circ \end{cases}$$

$$c_{m_f} = -\tfrac{1}{4}c_d(\alpha=90) = -\tfrac{1}{4}(c_d(\alpha=\alpha_d) + c_{d_f})$$

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

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

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

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

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

$$c_m = c_m + (\tfrac{1}{2}\cos\alpha_e)c_l + (\tfrac{1}{2}\sin\alpha_e)c_d$$

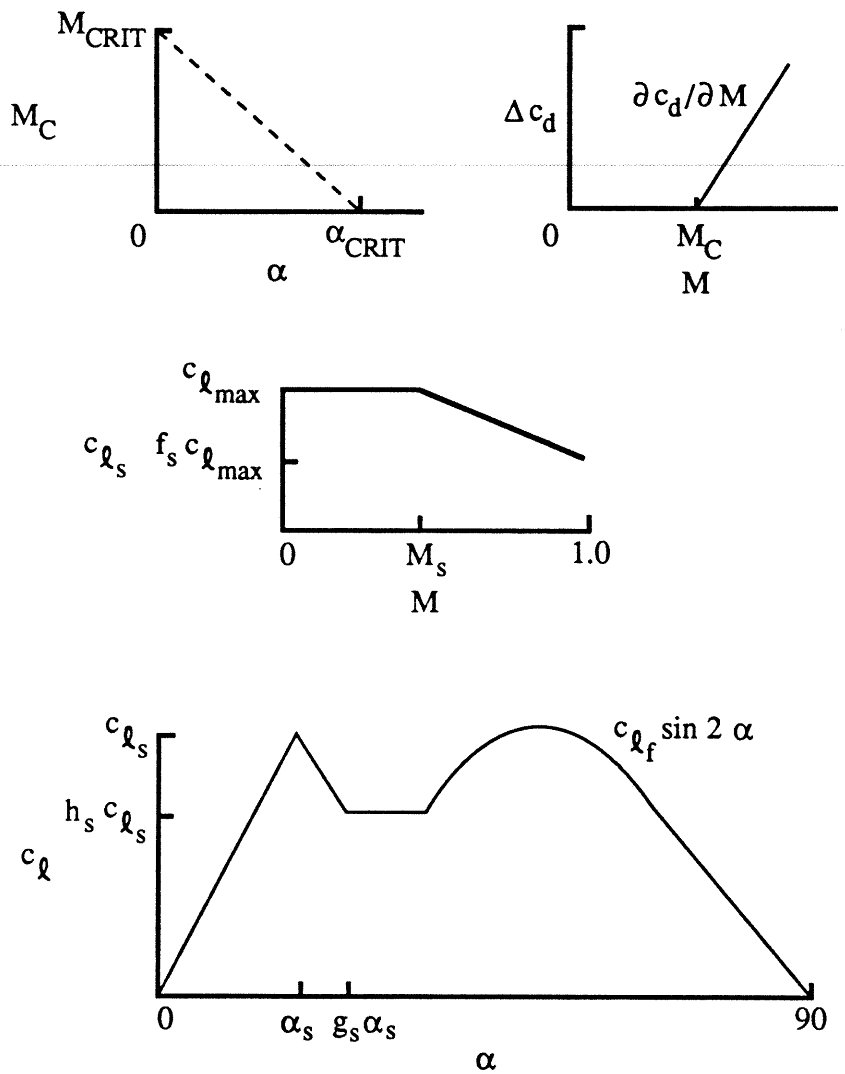


Figure 4-2. Airfoil characteristics -- lift and drag coefficients.

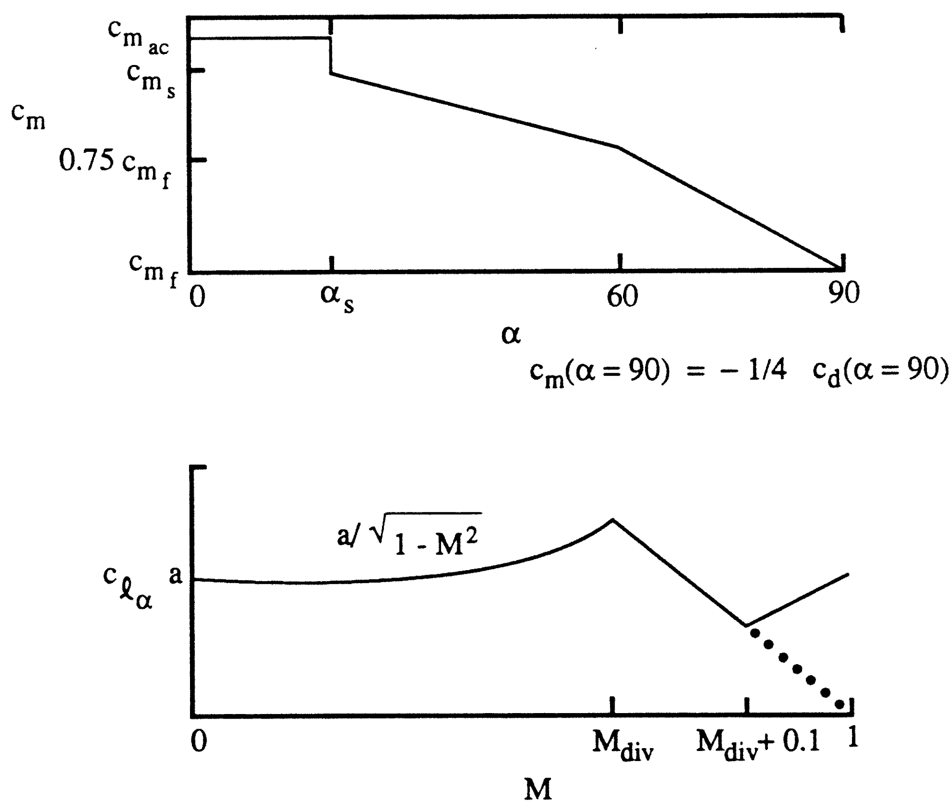


Figure 4-3. Airfoil characteristics -- moment coefficient and lift-curve slope.

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:	BENDMODE1
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=...,ETAPH=...,
&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.

IR = 1	$r = r_{FA}$
IR = 2	$r = r_{PB}$
IR = 3	$r = r_{root}$
IR = 4	$r = 1.$
IR = 5-15	$r = (j - 1)0.1$, $j = 1$ to 11
IR = 16-66	$r = (j - 1)\Delta r$, $j = 1$ to MRM+1
	(Δr = 1/MRM)
IR = 67-96	$r = r_j$, $j = 1$ to MRA (aerodynamic stations)

The aerodynamic radial stations r_j are the midponts 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 MODER1 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.

File logical name:	AEROINT1
Function:	rotor#1 body interference vel
Unit number variable:	NFINT1 (namelist NLCASE)
Unit number default value:	63
Subroutine using file:	INPT (through FILEV1)
Parameter controlling use:	OPINTV (namelist NLBODY)

File logical name:	AEROINT2
Function:	rotor#2 body interference vel
Unit number variable:	NFINT2 (namelist NLCASE)
Unit number default value:	64
Subroutine using file:	INPT (through FILEV2)
Parameter controlling use:	OPINTV (namelist NLBODY)

The file has namelist format:

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

The input parameters are as follows.

NVINT	integer; specification of coordinate axes of velocity data 1 airframe axes 2 nonrotating shaft axes 3 rotating shaft axes
	real; velocity (divided by flight speed V), for airframe axes
VINTX(MRA,MPSI)	x component, + forward
VINTY(MRA,MPSI)	y component, + right
VINTZ(MRA,MPSI)	z component, + down

	real; velocity (divided by flight speed V), for nonrotating shaft axes
VINTX(MRA,MPSI)	x component, + aft
VINTY(MRA,MPSI)	y component, + to advancing side
VINTZ(MRA,MPSI)	z component, + up
	real; velocity (divided by flight speed V), for rotating shaft axes
VINTR(MRA,MPSI)	Δu_R component, + outboard
VINTT(MRA,MPSI)	Δu_T component, + to trailing edge
VINTP(MRA,MPSI)	Δu_P component, + down

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, ψ_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 ψ_j = j(360/MPSI) degrees, measured from downstream in the direction of rotation of the rotor.

The arrays are dimensioned for 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 AES1CM).

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.

File logical name:	BODYAERO
Function:	airframe aero coefficients
Unit number variable:	NFBAT (namelist NLCASE)
Unit number default value:	65
Subroutine using file:	INPT (through FILEF)
Parameter controlling use:	OPBAT (namelist NLBODY)

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:

```
                                              Read Format

TITLE                                         A32
AREA        CHORD                             2F12.0
LABEL       ND                                A12,I2
Di          NMi       NAi                     F12.0,2I2
            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 ND
```

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.

HEADER	file header (32 characters)
AREA	reference area S for coefficients; ft^2 or m^2
CHORD	reference chord c for coefficients; ft or m
LABEL	CL, CD, or CM
ND	number of elevator angle entries; i = 1 to ND
Di	elevator angle (deg) for angle-of-attack and Mach number array to follow
NMi	number of Mach number entries
NAi	number of angle-of-attack entries
M(NMi,ND)	Mach numbers
A(NAi,ND)	angles of attack (deg); relative to the airframe (F) axes
C(NMi,NAi,ND)	coefficient

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 NAi 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.

File logical name:	STABDERIV
Function:	airframe stability derivatives
Unit number variable:	NFDRV (namelist NLCASE)
Unit number default value:	66
Subroutine using file:	INPT (through FILED)
Parameter controlling use:	OPDRV (namelist NLBODY)

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.

HEADER	file header (32 characters)
AREA	reference area S for coefficients; ft^2 or m^2
CHORD	reference chord c for coefficients; ft or m
SPAN	reference span b for coefficients; ft or m

LABELi	table identifier; i = 1 to 33
NMi	number of Mach number entries
NAi	number of angle-of-attack entries
M(NMi,33)	Mach numbers
A(NAi,33)	angles of attack (deg); relative to the airframe (F) axes
C(NMi,NAi,33)	coefficient

LABELi 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. NMi and NAi 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
	real; coefficients from airfoil tables
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, ψ_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 ψ_j = j(360/MPSI) degrees, measured from downstream in the direction of rotation of the rotor.

The arrays are dimensioned for 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.

CAMRAD/JA Namelists

Label	Data	Commons
NLTRIM	Job and trim	TMDATA,SRDATA
NLRTR	Rotor#1	R1DATA
NLWAKE	Wake, rotor#1	G1DATA,W1DATA
NLRTR	Rotor#2	R2DATA
NLWAKE	Wake, rotor#2	G2DATA,W2DATA
NLBODY	Airframe and drive train	BDDATA,BADATA,ENDATA
NLLOAD	Loads, airframe and rotor#1	LADATA,L1DATA
NLLOAD	Loads, airframe and rotor#2	LADATA,L2DATA
NLFLUT	Flutter	FLDATA,HCDATA
NLSTAB	Flight dynamics	STDATA,GCDATA,HCDATA
NLTRAN	Transient	TNDATA,GCDATA

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 NLCASE, 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 NLCASE.

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

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:

Label	Data
NLCASE	Case
NLTRIM	Job and trim
NLRTR	Rotor
NLWAKE	Wake
NLBODY	Airframe and drive train
NLLOAD	Airframe and rotor loads
NLFLUT	Flutter
NLSTAB	Flight dynamics
NLTRAN	Transient

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

NCASES	integer; number of cases; default 1
INFILE	integer; read of input file 0 never 1 first case only (default) 2 each case
AFFILE	integer; read of airfoil files (first case only) 0 neither 1 rotor#1 2 rotor#2 3 rotor#1 and rotor#2 4 rotor#2 = rotor#1 (default)
PLFILE	integer; write to plot file enabled 0 never (default) 1 one file per job 2 one file per case

Input and output unit numbers

NFDAT	integer; unit number for read of input file; default 40
NFAF1	integer; unit number for read of rotor#1 airfoil file; default 41
NFAF2	integer; unit number for read of rotor#2 airfoil file; default 42
NFPLT	integer; unit number for write of plot file; default 43
NFBND1	integer; unit number for read of rotor#1 bending mode file; default 61
NFBND2	integer; unit number for read of rotor#2 bending mode file; default 62
NFINT1	integer; unit number for read of rotor#1 airframe inteference velocity file; default 63

NFINT2	integer; unit number for read of rotor#2 airframe inteference 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
NFCI1	integer; unit number for read of rotor#1 CFD interface input file; default 71
NFCI2	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

TITLE	character; title for job and case, maximum 80 characters
CODE	character; job or case identification, maximum 20 characters
OPUNIT	integer; unit system: 1 for English units (ft-slug-sec), 2 for metric units (m-kg-sec)
ANTYPE(3)	integer; analysis tasks (0 to suppress) ANTYPE(1) flutter ANTYPE(2) flight dynamics ANTYPE(3) transient
OPREAD(10)	integer; namelists to be read for this case (0 to suppress read) OPREAD(1) NLRTR, rotor#1 OPREAD(2) NLWAKE, rotor#1 OPREAD(3) NLRTR, rotor#2 OPREAD(4) NLWAKE, rotor#2 OPREAD(5) NLBODY OPREAD(6) NLLOAD, rotor#1 OPREAD(7) NLLOAD, rotor#2 OPREAD(8) NLFLUT OPREAD(9) NLSTAB OPREAD(10) NLTRAN
NPRNTI	integer; print of input parameters 0 none 1 parameters related to current job; determined by: LEVEL, ANTYPE (NLTRIM, NLFLUT, NLSTAB), OPFDAN, MNOISE, MVIB, NRVIB, MHHC, MHHCF, MHHAF, OPASE (NLFLUT, NLSTAB), OPSTR 2 all parameters, for current configuration; determined by: NROTOR, CONFIG, MPSI, MRA, MRI, NEM, NAF, NWING, NBODY, MLOOP, NTSTR, NZSTR, NPOFF

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 DBTIME(2), stop debug print at DBTIME(3)
  - 2 start debug print at 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

DEBUG(24) 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, WKPAX, WKPFW, 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

VKTS	real; aircraft speed V (knots)
VEL	real; velocity ratio, V/ΩR; only used if VKTS not input input either VEL or VKTS by job namelist; if neither parameter is defined, V = 0 is used
VTIP	real; rotor#1 tip speed ΩR (ft/sec or m/sec)
RPM	real; rotor#1 rotational speed (rpm); only used if VTIP not input 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
OPDENS	integer; specification of aerodynamic environment 1 altitude and standard day 2 altitude and temperature 3 density and temperature
ALTMSL	real; altitude above mean sea level (ft or m); used if OPDENS = 1 or 2
TEMP	real; air temperature (deg F or deg C); used if OPDENS = 2 or 3
DENSE	real; air density (slug/ft^3 or kg/m^3); used if OPDENS = 3
OPGRND	integer; ground effect analysis: 0 for out of ground effect, ne 0 for in ground effect
HAGL	real; altitude aircraft center-of-gravity above ground (ft or m), for ground effect analysis

Aircraft description

NROTOR — integer; number of rotors

AFLAP — real; wing flap angle δ_f (deg)

OPENGN — 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	$q_1 \cdots q_{10}$	$p_0 \cdots p_4$	β_G
rotor#2	$q_1 \cdots q_{10}$ (bending)	$p_0 \cdots p_4$ (torsion)	β_G (gimbal/teeter)
airframe	ϕ_F θ_F ψ_F x_F y_F z_F (rigid body)		$q_{S7} \cdots q_{S36}$ (flexible body)
drive train	ψ_S ψ_I ψ_e (rotor/engine speed)		$\Delta\theta_t$ $\Delta\theta_{govr_1}$ $\Delta\theta_{govr_2}$ (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

Motion analysis

MPSI	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
MHARM(2)	integer; number of harmonics in rotor motion analysis; maximum 20; 0 for only mean response MHARM(1) rotor#1 MHARM(2) rotor#2
MHARMF(2)	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 MHARMF(1) rotor#1 MHARMF(2) rotor#2
MPSIR	integer; in harmonic motion solution, number of rotor azimuth steps between update of airframe vibration
MREV	integer; in harmonic motion solution, number of revolutions between tests for motion convergence
ITERM	integer; maximum number of motion iterations
EPMOTN	real; tolerance for motion convergence (deg)
ITERC	integer; maximum number of circulation iterations
EPCIRC	real; tolerance for circulation convergence ($\Delta C_T/\sigma$)

Wake analysis

LEVEL(2) — 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
- LEVEL(1) rotor#1
- LEVEL(2) rotor#2

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

ITERU — at uniform inflow stage

ITERR — at nonuniform inflow, prescribed wake geometry stage

ITERF — at nonuniform inflow, free wake geometry stage

Trim analysis

OPTRIM 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

OPTRIM = 1 trim forces and moments with
δ_o δ_c δ_s δ_p θ_{FT} ϕ_{FT}

2 trim forces and moments with
δ_o δ_c δ_s δ_p θ_{FT} ψ_{FP}

3 trim forces, moments, and power with
δ_o δ_c δ_s δ_p θ_{FT} ϕ_{FT} θ_{FP}

4 trim forces, moments, and power with
δ_o δ_c δ_s δ_p θ_{FT} ψ_{FP} θ_{FP}

5 trim forces, moments, and power with
δ_o δ_c δ_s δ_p δ_t θ_{FT} ϕ_{FT}

6 trim forces, moments, and power with
δ_o δ_c δ_s δ_p δ_t θ_{FT} ψ_{FP}

7 trim symmetric forces and moment with
δ_o δ_s θ_{FT}

8 trim symmetric forces, moment, and power with
δ_o δ_s θ_{FT} θ_{FP}

9 trim symmetric forces, moment, and power with
δ_o δ_s δ_t θ_{FT}

wind tunnel options

OPTRIM = TTC

trimmed quantities

L = rotor lift C_L/σ (wind axes)
X = rotor drag C_X/σ (wind axes)
Y = rotor side force C_Y/σ (shaft axes)
T = rotor thrust C_T/σ (shaft axes)
P = rotor power C_P/σ

TT = 11	trim	L				β_s	(MT =2)
TT = 12	trim		P			β_s	(MT =2)
TT = 13	trim			X		Y	(MT =2)
TT = 14	trim				β_c	β_s	(MT =2)
TT = 15	trim	L	P			β_s	(MT =3)
TT = 16	trim	L		X		Y	(MT =3)
TT = 17	trim	L			β_c	β_s	(MT =3)
TT = 18	trim		P	X		Y	(MT =3)
TT = 19	trim		P		β_c	β_s	(MT =3)
TT = 20	trim			X	β_c	β_s	(MT =3)
TT = 21	trim	L	P	X		Y	(MT =4)
TT = 22	trim	L	P		β_c	β_s	(MT =4)
TT = 23	trim	L		X	β_c	β_s	(MT =4)
TT = 24	trim	T				β_s	(MT =2)
TT = 25	trim	T	P			β_s	(MT =3)
TT = 26	trim	T			β_c	β_s	(MT =3)
TT = 27	trim	T	P		β_c	β_s	(MT =4)

trim variables

if MT = 2

C	= 1,4	with	δ_o			δ_c
C	= 2,5	with	δ_s			δ_c
C	= 3,6	with	θ_T			δ_c

if MT = 3

C	= 1,4	with	δ_o	δ_s	δ_c
C	= 2,5	with	δ_o	θ_T	δ_c
C	= 3,6	with	δ_s	θ_T	δ_c

if MT = 4

C = 1,4 with δ_o δ_s θ_T δ_c

only longitudinal trimmed quantities (no Y or β_s) and trim variables (no δ_c) if C = 4, 5, or 6

MTRIM	integer; maximum number of iterations on controls to achieve trim
MTRIMD	integer; number of trim iterations between perturbation identification of derivative matrix
DELTA	real; control step in perturbation identification of derivative matrix (stick displacement or aircraft attitude, deg)
OPTIDR	integer; ne 0 for recursive update of trim derivative matrix
ALPHA	real; weight in recursive update (exponential window)
FACTOR	real; factor reducing control increment in order to improve trim convergence
EPTRIM	real; tolerance on trim convergence
OPGOVT	integer; use of governor in trim 0 trim collective stick δ_0 1 trim rotor#1 governor 2 trim rotor#2 governor 3 trim both rotor governors
OPWT2T	integer; rotor forces and power evaluation for wind tunnel trim options (OPTRIM GT 100) 1 rotor#1 quantities only 2 sum of both rotors

Initial control settings

COLL — real; collective stick displacement δ_o or $\Delta\theta_{govr}$ (deg), positive up

LATCYC — real; lateral cyclic stick displacement δ_c (deg), positive right

LNGCYC — real; longitudinal cyclic stick displacement δ_s (deg), positive forward

PEDAL — real; pedal displacement δ_p (deg), positive to right

THROTL — real; throttle displacement δ_t (deg)

APITCH — real; for free flight cases, aircraft pitch angle θ_{FT} (deg), positive nose up; for wind tunnel cases, rotor shaft angle-of-attack θ_T (deg), positive nose up

AROLL — real; for free flight cases, aircraft roll angle ϕ_{FT} (deg), positive to right

ACLIMB — real; for free flight cases, aircraft climb angle θ_{FP} (deg), positive up

AYAW — real; for free flight cases, aircraft yaw angle ψ_{FP} (deg), positive to right; for wind tunnel cases, test module yaw angle ψ_T (deg), positive to right

RTURN — real; trim turn rate $\dot{\psi}_F$ (deg/sec), positive to right; only used for free flight cases

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

θ_{FT} and ϕ_{FT} define orientation of body axes relative to earth axes

θ_{FP} and ψ_{FP} define orientation of velocity axes relative to earth axes; $V_{climb} = V\sin\theta_{FP}$ and $V_{side} = V\sin\psi_{FP}\cos\theta_{FP}$

Targets for wind tunnel trim

CTTRIM	real; thrust C_T/σ (shaft axes) or lift C_L/σ (wind axes)
CPTRIM	real; power C_P/σ
CXTRIM	real; rotor drag C_X/σ (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/σ (shaft axes)
BCTRIM	real; longitudinal tip-path-plane tilt relative shaft β_c (deg), positive for forward tilt
BSTRIM	real; lateral tip-path-plane tilt relative shaft β_s (deg), positive for advancing side up

Trim output control

NPRNTT	integer; parameter n, trim/performance/loads 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

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

MSTR	integer; maximum number of iterations on control to achieve convergence; 0 to turn off regulator
PIDSTR	integer; perturbation identification of T-matrix 0 never (use input TMTRX) 1 at beginning of trim, then every MIDSTR regulator iterations 2 each trim iteration, then every MIDSTR regulator iterations
MIDSTR	integer; number of regulator iterations between perturbation identification of T-matrix; le 0 for never
RIDSTR	integer; if ne 0, use recursive update of T-matrix
EPSTR	real; tolerance on regulator convergence

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 nnth cosine harmonic
harmonic = SNnn nnth sine harmonic

category = RTR rotor primary controls

rotor = RTR1 rotor #1
rotor = RTR2 rotor #2
rotor = SYM symmetric
rotor = ANTI antisymmetric

quantity = T75 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

summary of allowed keywords:

category	rotor	quantity	harmonic
HHC	RTR1 RTR2 SYM ANTI	ROT COLL LAT LNG FRCn	CSnn SNnn
RTR	RTR1 RTR2 SYM ANTI	T75 T1C T1S	
BODY		DELF DELE DELA DELR THTT FRCn	

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

```
        quantity = MX        roll moment
        quantity = MY        pitch moment
        quantity = Q         torque

         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 = ROOT   rotating frame root loads

           rotor = RTR1      rotor #1
           rotor = RTR2      rotor #2
           rotor = SYM       symmetric
           rotor = ANTI      antisymmetric

        quantity = FX        inplane shear
        quantity = FR        radial shear
        quantity = FZ        vertical shear
        quantity = MX        flapwise moment
        quantity = MZ        lagwise moment
        quantity = MC        control 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 = BLDS   blade loads, rotating shaft axes

           rotor = RTR1      rotor #1
           rotor = RTR2      rotor #2
           rotor = SYM       symmetric
           rotor = ANTI      antisymmetric

        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

         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
```

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		POWR		
	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

TITLE	character; title for rotor data, maximum 80 characters
TYPE	character*4; label for rotor; suggest MAIN, FRNT, RGHT, or LOWR for rotor#1; and TAIL, REAR, LEFT, or UPPR for rotor#2
RADIUS	real; blade radius R (ft or m)
NBLADE	integer; number of blades
SIGMA	real; solidity ratio $\sigma = Nc_m/\pi R$ (based on mean chord)
ROTATE	integer; rotor rotation direction (viewed from above): 1 for counterclockwise, -1 for clockwise
VTIPN	real; normal stip speed ΩR_o (ft/sec or m/sec)

Higher harmonic forces and control -- rotating frame control

MHHC	integer; number of harmonics, n = 1 to MHHC; maximum 20
THHC(MHHC)	real; pitch amplitude, $\cos(n\psi)$; 1/rev not used (deg)
THHS(MHHC)	real; pitch amplitude, $\sin(n\psi)$; 1/rev not used (deg)

Higher harmonic forces and control -- nonrotating frame control

MHHCF	integer; number of harmonics, p = 1 to MHHCF; maximum 10
TOHHC(MHHCF)	real; collective pitch amplitude, $\cos(pN\psi)$ (deg)
TOHHS(MHHCF)	real; collective pitch amplitude, $\sin(pN\psi)$ (deg)
TCHHC(MHHCF)	real; lateral cyclic pitch amplitude, $\cos(pN\psi)$ (deg)
TCHHS(MHHCF)	real; lateral cyclic pitch amplitude, $\sin(pN\psi)$ (deg)
TSHHC(MHHCF)	real; longitudinal cyclic pitch amplitude, $\cos(pN\psi)$ (deg)
TSHHS(MHHCF)	real; longitudinal cyclic pitch amplitude, $\sin(pN\psi)$ (deg)

Higher harmonic forces and control -- airframe forces

MHHAF	integer; number of harmonics, p = 1 to MHHAF; maximum 10
FHHC(NAF,MHHAF)	real; force amplitude, $\cos(pN\psi)$ (lb or N)
FHHS(NAF,MHHAF)	real; force amplitude, $\sin(pN\psi)$ (lb or N)

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 θ_{tw} (deg); used to calculate TWISTA and TWISTI, if LINTW = 1
RGMAX	real; r_{Gmax}/R (maximum bound circulation for induced velocity calculation found outboard of r_{Gmax})
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

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 $c_{ext} = c_{old}$ 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 $\dot{\alpha}$ stall delay
- 3 $\dot{\alpha}$ stall delay, with dynamic stall vortex loads
- 4 $\sqrt{\dot{\alpha}}$ stall delay
- 5 $\sqrt{\dot{\alpha}}$ 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; τ 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: α_{ds} (deg)

ALFRE(3) — stall recovery angle of attack for lift, drag, and moment: α_{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)

INFLOW(1) at this rotor: 0 for uniform, 1 for nonuniform

INFLOW(2) at other rotor: 0 for zero, 1 for empirical, 2 for average at hub, 3 for nonuniform (only if $\Omega_2 = \Omega_1$)

INFLOW(3) at wing-body: 0 for zero, 1 for empirical, 2 for nonuniform

INFLOW(4) at horizontal tail: 0 for zero, 1 for empirical, 2 for nonuniform

INFLOW(5) at vertical tail: 0 for zero, 1 for empirical, 2 for nonuniform

INFLOW(6) number of points off rotor disk: 0 for none, maximum NPOFF

KHLMDA real; empirical inflow correction factor for hover, κ_h

KFLMDA real; empirical inflow correction factor for forward flight, κ_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_m on linear inflow variation produced by hub moments (typically 1.)
	real; factor for interference velocity at other rotor (κ_{21} or κ_{12}); linear variation between KINTH at μ = 0.05 and KINTF at μ = 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/σ of fixed wake geometry for hover trim near zero thrust (0.01 used if input value is 0.)

Dynamic model -- bending and torsion modes

HINGE — integer; specification of blade bending mode type
- -1 read bending modes from file
- 0 articulated (rigid flap and lag modes only)
- 1 cantilever
- 2 hinged
- 3 hinged flap and cantilever lag (coning hub)
- 4 cantilever flap and hinged lag (lagging hub)

RCPL — real; structural coupling parameter $\mathfrak{R}$ (effective pitch angle $\mathfrak{R}\theta$ used to calculate blade bending modes; normally 1.)

real; hinge offset e/R (extent of rigid hub for cantilver blade)

EFLAP — flap

ELAG — lag

KFLAP — flap hinge spring (ft-lb/rad or m-N/rad)

KLAG — lag hinge spring (ft-lb/rad or m-N/rad)

TSPRNG — real; flap and lag hinge pitch angle at zero collective (deg), θ_{ho}

RCPLS — real; flap and lag hinge coupling parameter, $\mathfrak{R}_s$

pitch angle of the flap and lag hinges: $\theta_h = \theta_{ho} + \mathfrak{R}_s\theta_{75}$

RFA — real; feathering axis radial location, r_{FA}/R

MRB — integer; number of radial stations for integration in blade modes calculation; maximum 100

MRM — integer; number of radial stations for integration of inertial coefficients; maximum 50

EPMODE — real; criterion on change of collective pitch to update blade modes, $\Delta\theta_{75}$ (deg)

NONROT — integer; ne 0 to calculate nonrotating bending frequencies

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 ν_{GC} or teeter ν_T
NUGS	lateral gimbal ν_{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)
TDAMP0	collective
TDAMPC	cyclic
TDAMPR	rotating
WTIN	integer; source of control system stiffness: 1 for stiffness, 2 for frequency
	real; control system frequency ω_θ (per rev, at normal tip speed VTIPN), used if WTIN = 2
FT0	collective
FTC	cyclic
FTR	reactionless
	real; control system stiffness K_θ (ft-lb/rad or m-N/rad), used if WTIN = 1
KT0	collective
KTC	cyclic
KTR	reactionless

Dynamic model -- lag damper

LDAMPC	real; linear lag damper coefficient C_ζ (ft-lb/rad/sec or m-N/rad/sec); estimated damping if a nonlinear damper is used (LDAMPM gt 0.); the lag mode has structural damping also (GSB)
LDAMPM	real; maximum moment of nonlinear lag damper, M_{LD} (ft-lb or m-N); linear lag damper used if LDAMPM=0.
LDAMPR	real; lag velocity $\dot{\zeta}_{LD}$ where maximum moment of lag damper occurs (rad/sec); hydraulic damping below $\dot{\zeta}_{LD}$ and friction damping above

Dynamic model -- blade properties

GSB(NBM)	real; bending mode structural damping g_s
GST(NTM)	real; torsion mode structural damping g_s
MBLADE	real; blade mass (slug or kg); if le 0., integral of section mass MASS used
MASST	real; tip mass (slug or kg); the tip mass can also be included directly in the section mass distribution
XIT	real; chordwise offset of the tip mass center of gravity aft of the elastic axis, x_I/R

Dynamic model -- pitch-bending coupling

KPIN	integer; source of pitch-bending coupling: 1 for input, 2 for calculated; negative to suppress $\cos\theta_{75}$ factor in pitch-bending and pitch-gimbal coupling
	real; root geometry to calculate pitch-bending coupling (KPIN = 2 or -2)
PHIPH	pitch horn cant angle, ϕ_{PH} (deg)
PHIPL	pitch link cant angle, ϕ_{PL} (deg)
RPB	pitch bearing radial location, r_{PB}/R
RPH	pitch horn radial location, r_{PH}/R
XPH	pitch horn length, x_{PH}/R
ATANKP(NBM)	real; pitch-bending coupling $\tan^{-1}K_P$ (deg), for pitch horn level (KPIN = 1 or -1)
DEL3G	real; pitch-gimbal coupling $\tan^{-1}K_{PG}$ (deg), for pitch horn level

Dynamic model -- root geometry

ZFA	real; gimbal undersling, z_{FA}/R
XFA	real; torque offset, x_{FA}/R
CONE	real; precone angle δ_{FA_1} (deg), positive up
DROOP	real; droop angle δ_{FA_2} (deg) at $\theta_{75} = 0$, positive down from precone
SWEEP	real; sweep angle δ_{FA_3} (deg) at $\theta_{75} = 0$, positive aft
FDROOP	real; feathering axis droop angle δ_{FA_4} (deg), positive down from precone
FSWEEP	real; feathering axis sweep angle δ_{FA_5} (deg), positive aft

Dynamics model -- analysis

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

MRA	integer; number of aerodynamic segments; maximum 30
RAE(MRA+1)	real; radial stations r/R at edges of aerodynamic segments; sequential, from root to tip
	the following quantities are specified at the midpoints of the aerodynamic segments, from root to tip
CHORD(MRA)	real; blade chord, c/R
TWISTA(MRA)	real; blade twist relative 75% radius, θ_{tw} (deg)
THETZL(MRA)	real; incremental pitch of zero lift line, θ_{ZL} (deg); θ_{ZL} is the pitch of the axis corresponding to zero angle of attack in the airfoil tables, relative to the twist angle TWISTA
XA(MRA)	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
XAC(MRA)	real; offset of aerodynamic center (for unsteady aerodynamics only) aft of elastic axis, x_{AC}/R
ASWEEP(MRA)	real; sweep angle of quarter-chord line relative to reference span line (deg), positive aft
	real; Mach number correction factor, $M_{eff} = fM$
MCORRL(MRA)	for lift coefficient
MCORRD(MRA)	for drag coefficient
MCORRM(MRA)	for moment coefficient
DELCD(MRA)	real; drag coefficient increment, Δc_d
DELCM(MRA)	real; moment coefficient increment, Δc_m
RETAB1(MRA)	real; Reynolds number Re_{t1} of airfoil table for Mach number = 1 ($Re = M\ Re_{t1}$ for Mach number M); supercedes value in airfoil table (if ne 0.)

Blade section inertial and structural characteristics

MRI	integer; number of radial stations where characteristics are defined; maximum 51
RI(MRI)	real; radial stations r/R where characteristics are defined; sequential, from root to tip, RI(1)=0. and RI(MRI)=1.
TWISTI(MRI)	real; blade twist relative 75% radius, θ_{tw} (deg)
MASS(MRI)	real; section mass, m (slug/ft or kg/m)
XI(MRI)	real; offset of center of gravity aft of elastic axis, x_I/R
XC(MRI)	real; offset of tension center (modulus-weighted centroid) aft of elastic axis, x_C/R
KP2(MRI)	real; polar radius of gyration about elastic axis, k_P^2/R^2
EIZZ(MRI)	real; flapwise bending stiffness, EI_{zz} ($lb\text{-}ft^2$ or $N\text{-}m^2$)
EIXX(MRI)	real; chordwise bending stiffness, EI_{xx} ($lb\text{-}ft^2$ or $N\text{-}m^2$)
ITHETA(MRI)	real; section moment of inertia about elastic axis, I_θ (slug-ft or kg-m)
GJ(MRI)	real; torsional stiffness, GJ ($lb\text{-}ft^2$ or $N\text{-}m^2$)

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}
	real; rolling up wake model
RRU	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 ϕ_{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_{LW}
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

WKMODL(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

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

Computational domain for CFD interface (partial angle-of-attack calculation)

DOMCFD — integer; specification of what wake elements inside the domain are to be excluded

- 0 behind reference blade, for wake age $\phi<\phi_{bound}$
- 1 behind reference blade, for wake age $\phi<\phi_{bound}$; and all tip vortices
- 2 all wake elements

PHICFD — real; limit on wake age behind reference blade, ϕ_{bound} (deg)

XCFD(6) — real; extent of domain; relative tip, rotating tip-path-plane axes, divided by rotor radius; all values positive

- XCFD(1) chordwise, back
- XCFD(2) chordwise, front
- XCFD(3) radial, inboard
- XCFD(4) radial, outboard
- XCFD(5) vertical, lower
- XCFD(6) vertical, upper

Prescribed or rigid wake geometry

KRWG	integer; extent of prescribed or rigid wake geometry, K_{RWG} (age $\phi = K\Delta\Psi$); maximum 144
OPRWG	integer; wake geometry model 1 from $K_1 = f_1\lambda$, $K_2 = f_2\lambda$, input K_3, input K_4 2 option #1, without interference velocity in λ 3 from input K_1, K_2, K_3, K_4 4 Landgrebe hover model, from thrust coefficient 5 Landgrebe hover model, from maximum bound circulation 6 Kocurek and Tangler hover model, from thrust coefficient 7 Kocurek and Tangler hover model, from maximum bound circulation
FK2TWG	real; factor on tip vortex K_2 in prescribed wake geometry models (OPRWG = 4 to 7)
FGMXWG	real; factor on equivalent C_T obtained from maximum bound circulation, for prescribed wake geometry models (OPRWG = 5 or 7)
	real; factors f_1 and f_2 for prescribed wake geometry
FWGT(2)	tip vortex
FWGSI(2)	inside sheet edge
FWGSO(2)	outside sheet edge
	real; constants K_1, K_2, K_3, K_4 for prescribed wake geometry
KWGT(4)	tip vortex
KWGSI(4)	inside sheet edge
KWGSO(4)	outside sheet edge

Free wake geometry

KFWG	integer; extent of free wake geometry distortion calculation, K_{FWG} (age $\phi = K\Delta\psi$); maximum 96, multiple MPSI
OPFWG	integer; wake geometry model 1 Scully free wake geometry 2 option#1, without interference velocity
WGMODL(2)	integer; definition of wake model: 0 to omit element, 1 for line segment, 2 for sheet element WGMODL(1) inboard wake trailed vorticity WGMODL(2) wake shed vorticity
COREWG(3)	real; vortex core radii, r_c/R COREWG(1) tip vortices COREWG(2) inboard trailed wake lines; le 0. for default = .5*(RTWG(2)-RTWG(1)) COREWG(3) inboard shed wake lines; le 0. for default = .4
RTWG(2)	real; radial station r/R of trailed vorticity RTWG(1) inside sheet edge RTWG(2) outside sheet edge, or trailed line
MRVBWG	integer; number of wake revolutions used below point where induced velocity is being calculated
LDMWG	integer; parameter l_{DM}; general update every $l_{DM}\Delta\psi$ increment in boundary age
NDMWG(MPSI)	integer; paraemter $n_{DM}(\psi_j)$, j = 1 to MPSI; boundary update every n_{DM} increment in age
DQWG(2)	real; incremental velocity criteria; if magnitude of velocity increment greater than DQWG, then: DQWG(1) near wake element definition DQWG(2) integrate bound vortex line in time
ITERWG	integer; number of wake geometry iterations
FACTWG	real; relaxation factor, introducing lag in distortion calculation to improve convergence

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 leve 3 print of wake geometry distortion: induced velocity contribution of wake element printed if greater than QWGDB

6.3.5 Namelist NLBODY

Airframe configuration

TITLE	character; title for airframe data; maximum 80 characters
CONFIG	integer; specification of rotorcraft configuration 0 one rotor 1 single main rotor and tail rotor (tail rotor is rotor#2) 2 tandem main rotors (rear rotor is rotor#2) 3 tilting proprotor (left rotor is rotor#2) 4 coaxial main rotors (upper rotor is rotor#2)
WEIGHT	real; aircraft gross weight, including rotors (lb or kg)
	real; aircraft moments of inertia, including rotors (slug-ft^2 or kg-m^2)
IXX	I_{xx}
IYY	I_{yy}
IZZ	I_{zz}
IXY	I_{xy}
IXZ	I_{xz}
IYZ	I_{yz}
TRATIO	real; ratio of rotor#2 rotational speed to rotor#1 rotational speed, Ω_2/Ω_1 (transmission gear ratio)
ASHAFT(2)	shaft angle-of-attack θ_R (deg), positive rearward ASHAFT(1) rotor#1 ASHAFT(2) rotor#2
ACANT(2)	shaft cant angle ϕ_R (deg); positive to right for main rotor; positive upward for tail rotor; positive inward in helicopter mode for tilt rotor ACANT(1) rotor#1 ACANT(2) rotor#2

ATILT	real; nacelle tilt angle α_p (deg), for tilting proprotor configuration only (CONFIG=3); 0. for airplane mode, 90. for helicopter mode
HMAST	rotor mast length from pivot to hub (ft or m), for tilting proprotor configuration only (CONFIG=3)
DPSI21	$\Delta\psi_{21}$ (deg); rotor#2 azimuth angle ψ_2 when rotor#1 azimuth angle $\psi_1 = 0$; must be 0. if $\Omega_2 \neq \Omega_1$
CANTHT	horizontal tail cant angle ϕ_{HT} (deg), positive to left
CANTVT	vertical tail cant angle ϕ_{VT} (deg), positive to right

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

NEM	integer; number of airframe modes for which data supplied; maximum 30
QFREQ(NEM)	real; frequency of modes, ω_k (Hz)
QMASS(NEM)	real; generalized mass of modes, including rotors (slug or kg)
QDAMP(NEM)	real; structural damping, g_s
DOFSYM(NEM)	integer; type of mode: 1 for symmetric, -1 for antisymmetric (only required for flutter analysis with OPSYMM ne 0, tilting proprotor configuration)
	real; pitch/mast-bending coupling (rad/ft or rad/m)
KPMC1(NEM)	rotor#1, $K_{MC} = - \partial\theta_{1C}/\partial q_S$
KPMS1(NEM)	rotor#1, $K_{MS} = - \partial\theta_{1S}/\partial q_S$
KPMC2(NEM)	rotor#2, $K_{MC} = - \partial\theta_{1C}/\partial q_S$
KPMS2(NEM)	rotor#2, $K_{MS} = - \partial\theta_{1S}/\partial q_S$
	real; linear mode shapes at hub (ft/ft or m/m)
ZETAR1(3,NEM)	rotor#1 hub
ZETAR2(3,NEM)	rotor#2 hub
	real; angular mode shapes at hub (rad/ft or rad/m)
GAMAR1(3,NEM)	rotor#1 hub
GAMAR2(3,NEM)	rotor#2 hub
QDAMPA(NEM)	real; aerodynamic damping, $F_{q\dot{q}} = \partial(Q/\frac{1}{2}\rho V^2)/\partial(\dot{q}_S/V)$ (ft^2 or m^2)
QCNTRL(4,NEM)	real; aerodynamic control derivative, for flaperon, elevator, aileron, and rudder; $F_{q\delta} = \partial(Q/\frac{1}{2}\rho V^2)/\partial\delta$ (ft^2/rad or m^2/rad)

Auxiliary forces

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

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
KOCFE	K_o, collective stick to collective pitch
KCCFE	K_c, lateral cyclic stick to cyclic pitch
KSCFE	K_s, longitudinal cyclic stick to cyclic pitch
KPCFE	K_p, pedal to tail rotor collective pitch
PCCFE	$\Delta\psi_c$, lateral cyclic stick to cyclic pitch
PSCFE	$\Delta\psi_s$, longitudinal cyclic stick to cyclic pitch
	real; tilting proprotor configuration
KOCFE	K_o, collective stick to collective pitch
KCCFE	K_c, lateral cyclic stick to differential collective pitch
KSCFE	K_s, longitudinal cyclic stick to cyclic pitch
KPCFE	K_p, pedal to differential cyclic pitch
PSCFE	$\Delta\psi_s$, longitudinal cyclic stick to cyclic pitch
PPCFE	$\Delta\psi_p$, pedal to differential cyclic pitch

	real; coaxial main rotor configuration	
KOCFE	K_o, collective stick to collective pitch	
KCCFE	K_c, lateral cyclic stick to cyclic pitch	
KSCFE	K_s, longitudinal cyclic stick to cyclic pitch	
KPCFE	K_p, pedal to differential collective pitch	
PCCFE	$\Delta\psi_c$, lateral cyclic stick to cyclic pitch	
PSCFE	$\Delta\psi_s$, longitudinal cyclic stick to cyclic pitch	
	real; tandem main rotor configuration	
KFOCFE	K_{Fo}, collective stick to front collective pitch	
KROCFE	K_{Ro}, collective stick to rear collective pitch	
KFCCFE	K_{Fc}, lateral cyclic stick to front cyclic pitch	
KRCCFE	K_{Rc}, lateral cyclic stick to rear cyclic pitch	
KFSCFE	K_{Fs}, longitudinal cyclic stick to front collective pitch	
KRSCFE	K_{Rs}, longitudinal cyclic stick to rear collective pitch	
KFPCFE	K_{Fp}, pedal to front cyclic pitch	
KRPCFE	K_{Rp}, pedal to rear cyclic pitch	
PFCCFE	$\Delta\psi_{Fc}$, lateral cyclic stick to front cyclic pitch	
PRCCFE	$\Delta\psi_{Rc}$, lateral cyclic stick to rear cyclic pitch	
PFPCFE	$\Delta\psi_{Fp}$, pedal to front cyclic pitch	
PRPCFE	$\Delta\psi_{Rp}$, pedal to rear cyclic pitch	

	real; aircraft controls (all configurations)
KFCFE	K_f, collective stick to flaperon
KECFE	K_e, longitudinal cyclic stick to elevator
KACFE	K_a, lateral cyclic tick to ailerons
KRCFE	K_r, pedal to rudder
KTCFE	K_t, collective stick to engine
KTTCFE	K_{tt}, throttle to engine
KATCFE	K_{at}, throttle to auxiliary thrust force
KAPCFE	K_{ap}, pedal to auxiliary torque force
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

OPBAT — integer; airframe aerodynamic coefficients table
- 0 not use
- 1 read file

OPDRV — integer; airframe stability derivatives (flutter, flight dynamics, and transient analyses)
- 0 not use
- 1 use namelist parameters
- 2 read file

OPDRVU — integer; units of angle-of-attack, sideslip, and control derivatives (namelist or file); not used if OPDRV=0
- 0 per-radian
- 1 per-degree

Nonlinear aerodynamic model (equations)

	real; wing-body angles
IWB	incidence angle, i_{WB} (deg)
IWBD	incidence angle for drag, i_{WBd} (deg)
AMAXW	maximum angle-of-attack (deg)
	real; wing-body lift
LFTAW	derivative with angle-of-attack, L_α/q (ft^2/rad or m^2/rad)
LFTDW	derivative with flap, L_{δ_F}/q (ft^2/rad or m^2/rad)
LFTFW	derivative with flaperon, L_{δ_f}/q (ft^2/rad or m^2/rad)
	real; wing-body drag
DRGOW	base value, $f_{WB} = D_o/q$ (ft^2 or m^2)
DRGVW	vertical drag, f_{vert} (ft^2 or m^2)
DRGIW	induced drag, $\partial(D_i/q)/\partial(L/q)^2$ (ft^{-2} or m^{-2})
DRGDW	derivative with flap, $D_{o_{\delta F}}/q$ (ft^2/rad or m^2/rad)
DRGFW	derivative with flaperon, $D_{o_{\delta f}}/q$ (ft^2/rad or m^2/rad)
	real; wing-body pitch moment
MOMOW	base value, M_o/q (ft^3 or m^3)
MOMAW	derivative with angle-of-attack, M_α/q (ft^3/rad or m^3/rad)
MOMDW	derivative with flap, $M_{\delta F}/q$ (ft^3/rad or m^3/rad)
MOMFW	derivative with flaperon, $M_{\delta f}/q$ (ft^3/rad or m^3/rad)

SIDEB	real; wing-body side force derivative with sideslip, Y_β/q (ft^2/rad or m^2/rad)
SIDEP	derivative with rolling, VY_p/q (ft^3/rad or m^3/rad)
SIDER	derivative with yawing, VY_r/q (ft^3/rad or m^3/rad)
SIDEA	derivative with aileron, $Y_{\delta a}/q$ (ft^2/rad or m^2/rad)
ROLLB	real; wing-body roll moment derivative with sideslip, $N_{x\beta}/q$ (ft^3/rad or m^3/rad)
ROLLP	derivative with rolling, VN_{xp}/q (ft^4/rad or m^4/rad)
ROLLR	derivative with yawing, VN_{xr}/q (ft^4/rad or m^4/rad)
ROLLA	derivative with aileron, $N_{x\delta a}/q$ (ft^3/rad or m^3/rad)
YAWB	real; wing-body yaw moment derivative with sideslip, $N_{z\beta}/q$ (ft^3/rad or m^3/rad)
YAWP	derivative with rolling, VN_{zp}/q (ft^4/rad or m^4/rad)
YAWR	derivative with yawing, VN_{zr}/q (ft^4/rad or m^4/rad)
YAWA	derivative with aileron, $N_{z\delta a}/q$ (ft^3/rad or m^3/rad)

	real; horizontal tail lift	
LFTAH	derivative with angle of attack, L_α/q (ft^2/rad or m^2/rad)	
LFTEH	derivative with elevator, $L_{\delta e}/q$ (ft^2/rad or m^2/rad)	
AMAXH	maximum angle of attack (deg)	
IHT	incidence angle, i_{HT} (deg)	
	real; vertical tail lift	
LFTAV	derivative with angle of attack, L_α/q (ft^2/rad or m^2/rad)	
LFTRV	derivative with rudder, $L_{\delta r}/q$ (ft^2/rad or m^2/rad)	
AMAXV	maximum angle of attack (deg)	
IVT	incidence angle, i_{VT} (deg)	
OPTINT	integer; 0 to suppress airframe/tail aerodynamic interference ($\epsilon = 0$ and $\sigma = 0$)	
	real; airframe/tail interference	
FETAIL	tail angle of attack change, $(\partial\epsilon/(\partial(L/q))^{-1}$ (ft^2 or m^2)	
LHTAIL	horizontal tail length l_{HT} (ft or m)	
HVTAIL	vertical tail height h_{VT}, positive up (ft or m)	

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).

real; airframe lift coefficient

CLA	derivative with angle of attack, $C_{L\alpha}$
CLM	derivative with Mach number, C_{L_M}
CLAD	derivative with $\dot{w}$, $C_{L\dot{\alpha}}$
CLQ	derivative with pitch rate, C_{Lq}
CLDE	derivative with elevator, $C_{L\delta_e}$
CLDF	derivative with flaperon, $C_{L\delta_f}$

real; airframe drag coefficient

CDA	derivative with angle of attack, $C_{D\alpha}$
CDM	derivative with Mach number, C_{D_M}
CDAD	derivative with $\dot{w}$, $C_{D\dot{\alpha}}$
CDQ	derivative with pitch rate, C_{Dq}
CDDE	derivative with elevator, $C_{D\delta_e}$
CDDF	derivative with flaperon, $C_{D\delta_f}$

real; airframe pitch moment coefficient

CMA derivative with angle of attack, $C_{M\alpha}$

CMM derivative with Mach number, C_{M_M}

CMAD derivative with $\dot{w}$, $C_{M_{\dot{\alpha}}}$

CMQ derivative with pitch rate, C_{Mq}

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

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

$$C_L = (\text{lift})/(qS)$$

$$C_D = (\text{drag})/(qS)$$

$$C_M = (\text{pitch moment})/(qSc)$$

$$(C)_{\dot{\alpha}} = \partial C/\partial(\dot{\alpha}c/2V)$$

$$(C)_q = \partial C/\partial(qc/2V)$$

real; airframe side force coefficient

CYB derivative with sideslip, C_{y_β}

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, $C_{y_{\delta r}}$

real; airframe roll moment coefficient

CLB derivative with sideslip, C_{l_β}

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

CNB derivative with sideslip, $C_{n\beta}$

CNP derivative with roll rate, C_{np}

CNR derivative with yaw rate, C_{nr}

CNDA derivative with aileron, $C_{n\delta a}$

CNDR derivative with rudder, $C_{n\delta r}$

C_y = (side force)/(qS)
C_l = (roll moment)/(qSb)
C_n = (yaw moment)/(qSb)
$(C)_p = \partial C/\partial(pb/2V)$
$(C)_r = \partial C/\partial(rb/2V)$

Airframe/rotor aerodynamic interference

OPINTV(2) integer; analysis control: 0 for no interference; 1 for velocities read from file; 2 for calculated velocities
- OPINTV(1) rotor#1
- OPINTV(2) rotor#2

OPI1BP(4) integer; blade position model for interference calculation, rotor#1
- OPI1BP(1) eq 0 to suppress inplane motion
- OPI1BP(2) eq 0 to suppress all harmonics except mean
- OPI1BP(3) eq 0 for linear from root to tip
- OPI1BP(4) eq 0 to suppress twist in aerodynamic center offset

OPI2BP(4) integer; blade position model for interference calculation, rotor#2
- OPI2BP(1) eq 0 to suppress inplane motion
- OPI2BP(2) eq 0 to suppress all harmonics except mean
- OPI2BP(3) eq 0 for linear from root to tip
- 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)	buttline, positive right
WLWING(3,NWING)	waterline, positive up
WCIRC(NWING)	real; baseline wing bound circulation, $\Gamma_o/V = (L/q)/(2b_w) = c_w C_L/2$; 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 Γ/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; le 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

NBODY	integer; number of bodies; maximum 5
	real; position of body center; ft or m
FSBODY(NBODY)	fuselage station, positive aft
BLBODY(NBODY)	buttline, positive right
WLBODY(NBODY)	waterline, positive up
	real; orientation of body axis relative to fuselage coordinates; deg
BYAW(NBODY)	yaw angle ψ_A, positive right
BPITCH(NBODY)	pitch angle θ_A, positive up
BLNGTH(NBODY)	real; body length; ft or m
BTHICK(NBODY)	real; body thickness ratio (maximum thickness divided by length)
BSHAPE(NBODY)	integer; body shape: 1 for ellipsoid (sphere if thickness ratio ge 1.); 2 for sphere; 3 for airfoil-shaped body of revolution

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 $\partial P_E/\partial\theta_t$ (dimensional); for both engines if ENGPOS=3

if the throttle variable θ_t is only used for the governor, then just the products

$$K_P \partial P_E/\partial\theta_t = -\partial P/\partial\dot{\psi}_s$$

$$K_I \partial P_E/\partial\theta_t = -\partial P/\psi_s$$

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 κ; 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 (ψ_e degree of freedom)

GSI — real; interconnect shaft structural damping g_s (ψ_I degree of freedom)

Governor parameters

	real; governor proportional feedback gains (sec)
KPGOVE	throttle, $K_p = -\partial\theta_t/\partial\dot{\psi}_s$
KPGOV1	rotor#1 collective, $K_p = \partial\theta/\partial\dot{\psi}_s$
KPGOV2	rotor#2 collective, $K_p = \partial\theta/\partial\dot{\psi}_s$
	real; governor integral feedback gains
KIGOVE	throttle, $K_I = -\partial\theta/\partial\psi_s$
KIGOV1	rotor#1 collective, $K_I = \partial\theta/\partial\psi_s$
KIGOV2	rotor#2 collective, $K_I = \partial\theta/\partial\psi_s$
	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 proprotor 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 (ψ = JWAKE * $\Delta\psi$)
NGPLOT	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)
NSPAN	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)

MDLOAD integer; output control; le 0 to suppress

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

NDPRNT(5) integer; print
- 0 for none
- 1 for time history
- 2 for harmonics
- 3 for both

NDPLOT(5) integer; printer-plot
- 0 for none
- 1 for time history

NDFILE(5) 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) 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 ψ and r)

MALOAD integer; output control; le 0 to suppress

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

DAPLOT(86) real; polar printer-plot increment; plot last digit of integer part of value/increment (if multiple of NPOLAR); 0. for default value

NAPRNT(86) integer; print
- 0 for none
- 1 for time history
- 2 for harmonics
- 3 for both

NAPLOT(86) integer; printer-plot; 0 for none; sum:
- 1 for time history
- \+ 2 for spanwise
- \+ 4 for polar

NAFILE(86) 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
(1) angle of attack, α
(2) Mach number, M
(3) yaw angle, Λ
section coefficients
(4) lift, c_l
(5) drag, c_d
(6) moment, c_m
(7) radial drag, c_{dr}
(8) bound circulation, Γ
section velocity
(9) perpendicular, u_P
(10) tangential, u_T
(11) radial, u_R
(12) resultant, U
(13) pitch angle, θ

(14) inflow angle, ϕ
displacement
(15) lag
(16) flap
effective angle of attack, α_{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) λ_x, longitudinal
(35) λ_y, lateral
(36) λ_z, vertical
rotor interference induced velocity
(37) λ_x, longitudinal
(38) λ_y, lateral
(39) λ_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

(49) normal (thrust) $F_z/c = d(C_T/\sigma)/dr$
(50) pitch moment, M_a/c
(51) radial, $\bar{F}_r/c$ (shaft axes)
power coefficient
(52) total, $d(C_P/\sigma)/dr$
(53) induced, $d(C_{Pi}/\sigma)/dr$
(54) rotor interference, $d(C_{Pint}/\sigma)/dr$
(55) body interference, $d(C_{Pint}/\sigma)/dr$
(56) profile power, $d(C_{Po}/\sigma)/dr$
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), $F_z = dT/dr$
(64) pitch moment, M_a
(65) radial, $\bar{F}_r$ (shaft axes)
power
(66) total, dP/dr
(67) induced, dP_i/dr
(68) rotor interference, dP_{int}/dr
(69) body interference, dP_{int}/dr
(70) profile, dP_o/dr
input airframe interference velocity
(71) VINTX or VINTR
(72) VINTY or VINTT
(73) VINTZ or VINTP
airframe interference velocity
(74) radial, Δu_R
(75) tangential, Δu_T
(76) perpendicular, Δu_P
CFD interface output
(77) partial α
(78) c_{ltab}
(79) c_{dtab}
(80) c_{mtab}
CFD interface input
(81) c_{lold}
(82) c_{dold}
(83) c_{mold}
(84) c_{lext}
(85) c_{dext}
(86) c_{mext}

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

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

NHPLOT(4) — integer; printer-plot
- 0 for none
- 1 for time history

NHFILE(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

NRFILE(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

Far field rotational noise

MNOISE	integer; number of microphones; maximum 10; le 0 for no noise analysis
	real; position of microphone relative to rotor hub
RANGE(MNOISE)	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_1/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

NNPRNT	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

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

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	$\beta_0^{(i)}\ \beta_{1C}^{(i)}\ \beta_{1S}^{(i)} \cdot\cdot \beta_{N/2}^{(i)}$	$\theta_0^{(i)}\ \theta_{1C}^{(i)}\ \theta_{1S}^{(i)} \cdot\cdot \theta_{N/2}^{(i)}$	$\beta_{GC}\ \beta_{GS}$	ψ_S	$\lambda_u\ \lambda_x\ \lambda_y$
rotor#2	$\beta_0^{(i)}\ \beta_{1C}^{(i)}\ \beta_{1S}^{(i)} \cdot\cdot \beta_{N/2}^{(i)}$	$\theta_0^{(i)}\ \theta_{1C}^{(i)}\ \theta_{1S}^{(i)} \cdot\cdot \theta_{N/2}^{(i)}$	$\beta_{GC}\ \beta_{GS}$	ψ_I	$\lambda_u\ \lambda_x\ \lambda_y$
	bending (15)	pitch/torsion (9)	gimbal teeter	rotor speed	inflow

airframe	$\phi_F\ \theta_F\ \psi_F\ x_F\ y_F\ z_Y$	$q_{S7} \cdots q_{S36}$	ψ_e	$\Delta\theta_e\ \Delta\theta_{govr1}\ \Delta\theta_{govr2}$
	rigid body	flexible body (30)	engine speed	governor

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, β_{GC} is replaced by β_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 θ_0, θ_{1C}, θ_{1S}, θ_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 NLLOAD

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 (F0PLT,F1PLT,NF0PLT,NF1PLT,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,F0PLT,F1PLT,SCALE

Frequency response and gust response

F0PLT	real; beginning frequency, for linear scale
F1PLT	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)

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, NFOPLT, 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

MHACT(MLOOP)	integer; number of elements
IDHACT(MHACT,MLOOP)	integer; element identification
HREACT(MHACT,MLOOP)	real; real part parameter
HIMACT(MHACT,MLOOP)	real; imaginary part parameter

Transfer functions -- feedback

MHOUT(NYCL,MLOOP)	integer; number of elements
IDHOUT(MHOUT,NYCL,MLOOP)	integer; element identification
HREOUT(MHOUT,NYCL,MLOOP)	real; real part parameter
HIMOUT(MHOUT,NYCL,MLOOP)	real; imaginary part parameter

Transfer functions -- feedforward (3 paths)

MHDEL(3,NFFDCL,MLOOP)	integer; number of elements
IDHDEL(MHDEL,3,NFFDCL,MLOOP)	integer; element identification
HREDEL(MHDEL,3,NFFDCL,MLOOP)	real; real part parameter
HIMDEL(MHDEL,3,NFFDCL,MLOOP)	real; imaginary part parameter

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

element	IDHxxx	HRExxx	HIMxxx	factor	pole/zero
null	0	not used	not used	none	
gain	1	K	not used	K	
lag	2	τ, sec	not used	$e^{-\tau s}$	
real pole	10	τ, sec	not used	$(s - p)^{-1}$	$p = -1/\tau$
real pole	11	Re, sec^{-1}	0.	$(s - p)^{-1}$	$p = (Re, 0.)$
complex pole	11	Re, sec^{-1}	Im, sec^{-1}	$(s - p_1)^{-1}(s - p_2)^{-1}$	$p = (Re, \pm Im)$
complex pole	12	ζ, mag ≤ 1	ω_n, sec^{-1}	$(s - p_1)^{-1}(s - p_2)^{-1}$	$p = (-\omega_n\zeta, \pm\omega_n(1 - \zeta^2)^{1/2})$
complex pole	12	ζ, mag > 1	ω_n, sec^{-1}	$(s - p_1)^{-1}(s - p_2)^{-1}$	$p = (-\omega_n\zeta \pm \omega_n(\zeta^2 - 1)^{1/2}, 0.)$
real zero	20	τ, sec	not used	$(s - z)$	$z = -1/\tau$
real zero	21	Re, sec^{-1}	0.	$(s - z)$	$z = (Re, 0.)$
complex zero	21	Re, sec^{-1}	Im, sec^{-1}	$(s - z_1)(s - z_2)$	$z = (Re, \pm Im)$
complex zero	22	ζ, mag ≤ 1	ω_n, sec^{-1}	$(s - z_1)(s - z_2)$	$z = (-\omega_n\zeta, \pm\omega_n(1 - \zeta^2)^{1/2})$
complex zero	22	ζ, mag > 1	ω_n, sec^{-1}	$(s - z_1)(s - z_2)$	$z = (-\omega_n\zeta \pm \omega_n(\zeta^2 - 1)^{1/2}, 0.)$

6.3.8 Namelist NLSTAB

Flight dynamics analysis

NPRNTP	integer; le 0 to suppress performance print during stability derivative calculation
NPRNTL	integer; le 0 to suppress loads print during stability derivative calculation
ITERS	integer; number of iterations between wake geometry and motion/force calculations
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

Degrees of freedom, controls, and gust

DOF(7) integer; vector defining degrees of freedom, ne 0 if used; order: ϕ_F, θ_F, ψ_F, x_F, y_F, z_F, ψ_S

CON(21) integer; vector defining control variables; ne 0 if used; order:

rotor#1	θ_o	θ_{1C}	θ_{1S}		
rotor#2	θ_o	θ_{1C}	θ_{1S}		
airframe	δ_f	δ_e	δ_a	δ_r	θ_t
aux forces	f_1	f_2	f_3	f_4	f_5
pilot	δ_o	δ_c	δ_s	δ_p	δ_t

GUS(3) integer; vector defiing 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

MSTR — integer; maximum number of iterations on control to achieve convergence; 0 to turn off regulator

PIDSTR — integer; perturbation identification of T-matrix
- 0 never (use T-matrix from trim)
- 1 at start of each stability derivative iteration, then every MIDSTR regulator iterations

MIDSTR — integer; number of regulator iterations between perturbation identification of T-matrix; le 0 for never

RIDSTR — integer; recursive update of T-matrix, if ne 0

EPSTR — real; tolerance on regulator convergence

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,NF0PLT,NF1PLT,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, F0PLT, F1PLT, SCALE

Frequency response and gust response

F0PLT	real; beginning frequency, for linear scale
F1PLT	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

TSTEP	real; time step (sec)
TMAX	real; maximum time (sec)
NPRNTT	integer; print of transient data every NPRNTT-th time step; le 0 to suppress
NTPLOT	integer; if ne 0, printer-plot time history of system response

parameters defining prescribed controls and gust; see namelist NLTRAN for description

OPTRAN
OPHIST
CTIME
CMAG(5)
GTIME
GMAG(3)
GDIST(2)
VELG
PSIG
OPGUST(3)

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, 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

NPRNTP	integer; le 0 to suppress performance print during transient
NPRNTL	integer; le 0 to suppress loads print during transient
ITERT	integer; number of iterations between wake geometry and motion/force calculations
OPLMDA	integer; induced velocity calculation 0 update influence coefficients and inflow 1 only update inflow 2 update neither influence coeffcieints nor inflow
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

DOF(7)	integer; vector defining degrees of freedom in numerical integration; 0 to suppress acceleration; order: ϕ_F, θ_F, ψ_F, x_F, y_F, z_F, ψ_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: δ_o, δ_c, δ_s, δ_p, δ_t, u_G, v_G, w_G

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; le 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: δ_o, δ_c, δ_s, δ_p, δ_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_o (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 ψ_g (deg)

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

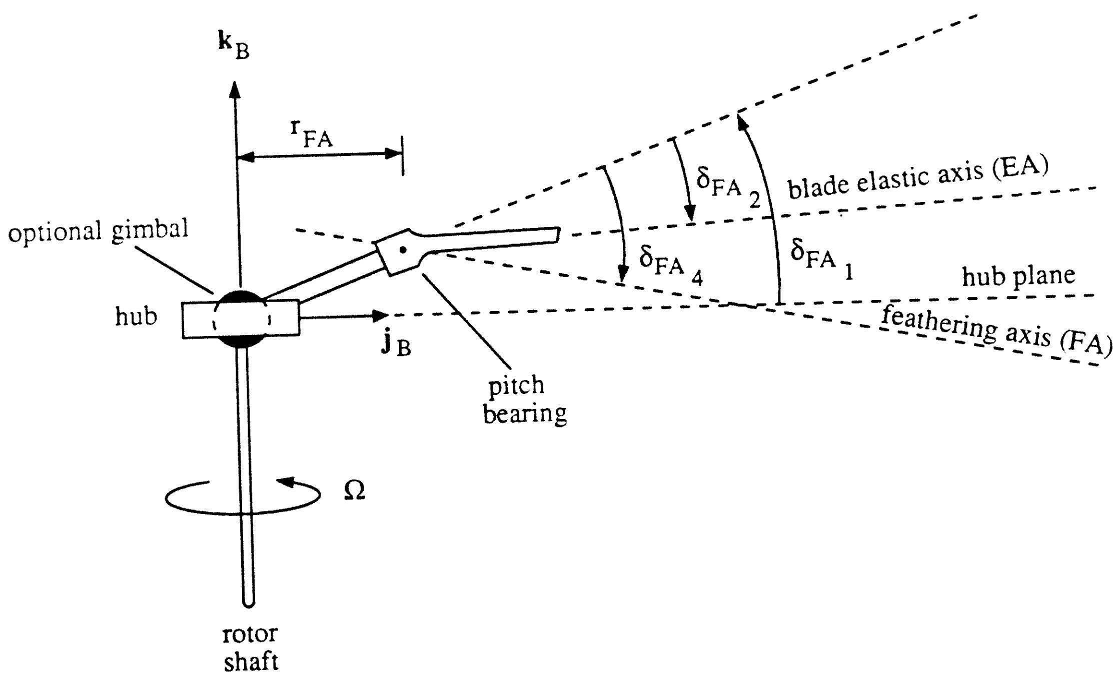


Figure 6-1. Schematic of the rotor hub and root geometry (side view). Only a single, undistorted blade is shown, without the gimbal undersling. The gimbal is omitted from the model for articulated and hingeless rotors.

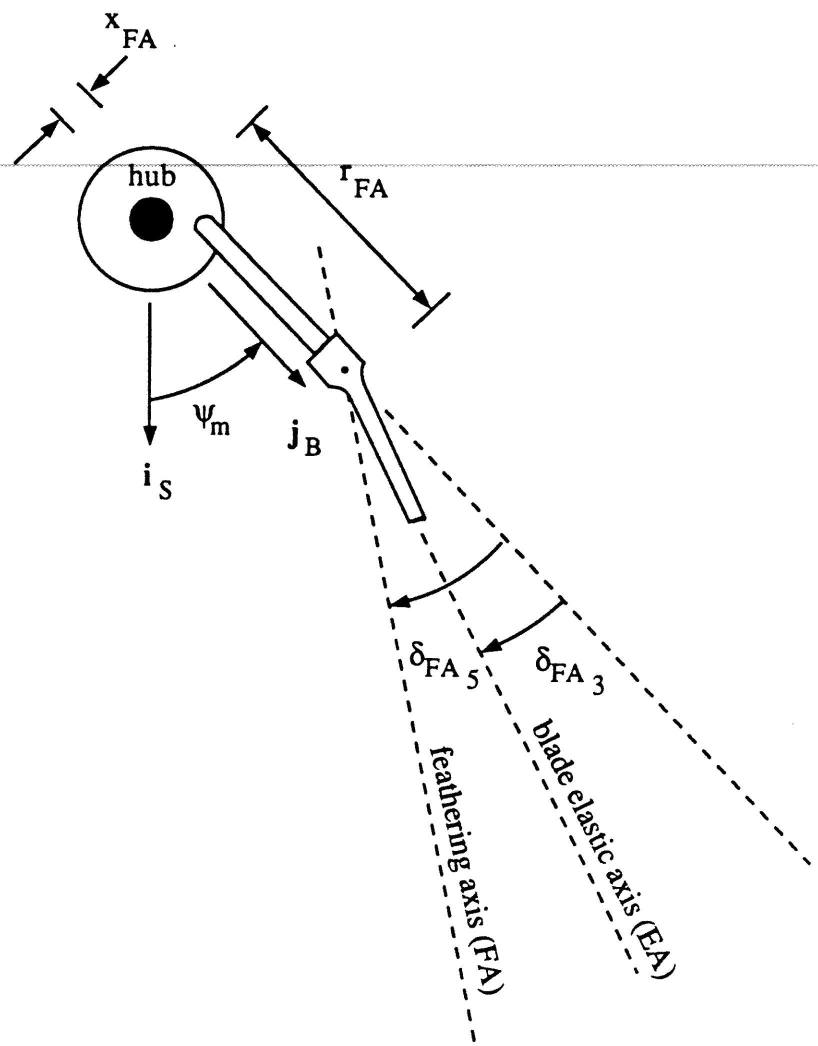


Figure 6-2. Schematic of the rotor hub and root geometry (top view). Only a single, undistorted blade is shown, without the gimbal undersling. The gimbal is omitted from the model for articulated and hingeless rotors.

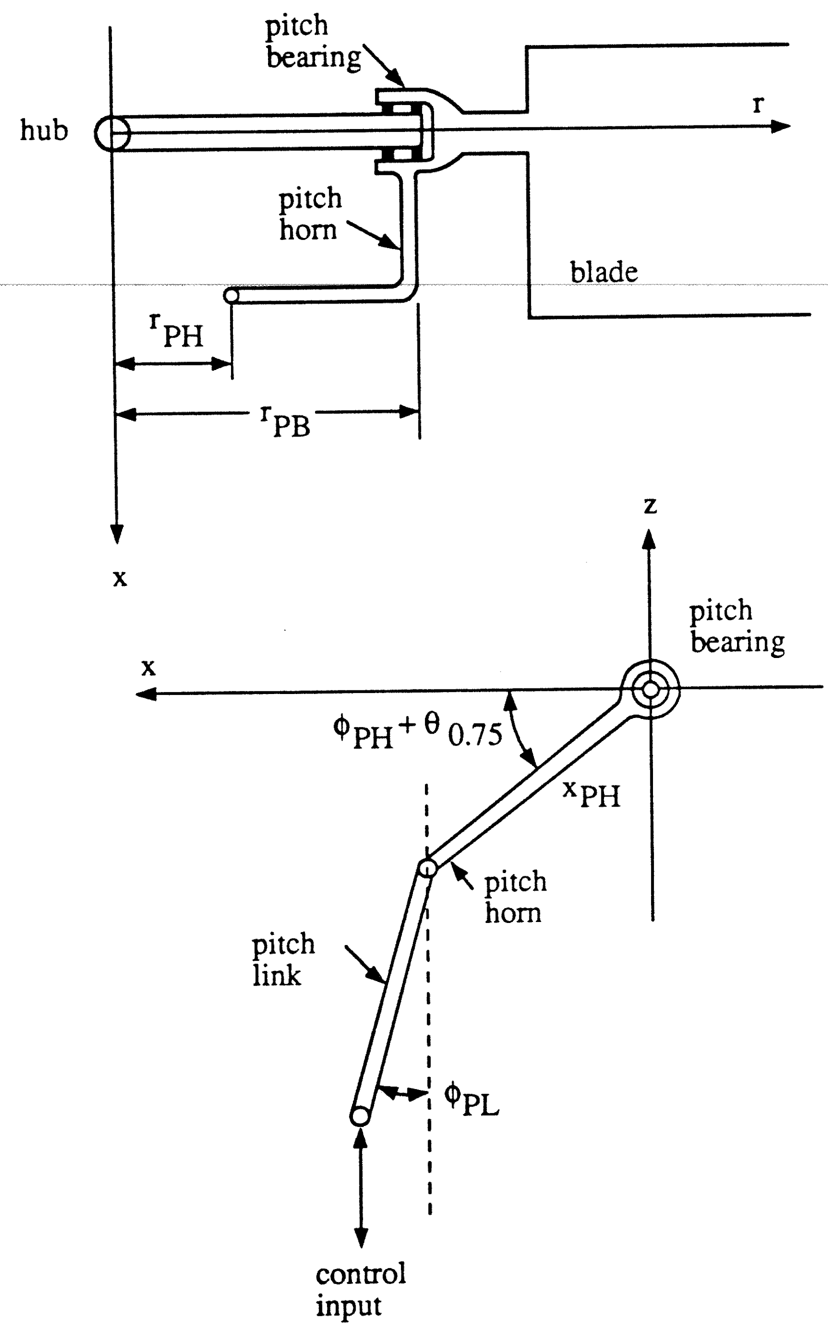


Figure 6-3. Schematic of blade root and control system geometry for calculation the kinematic pitch-bending coupling.

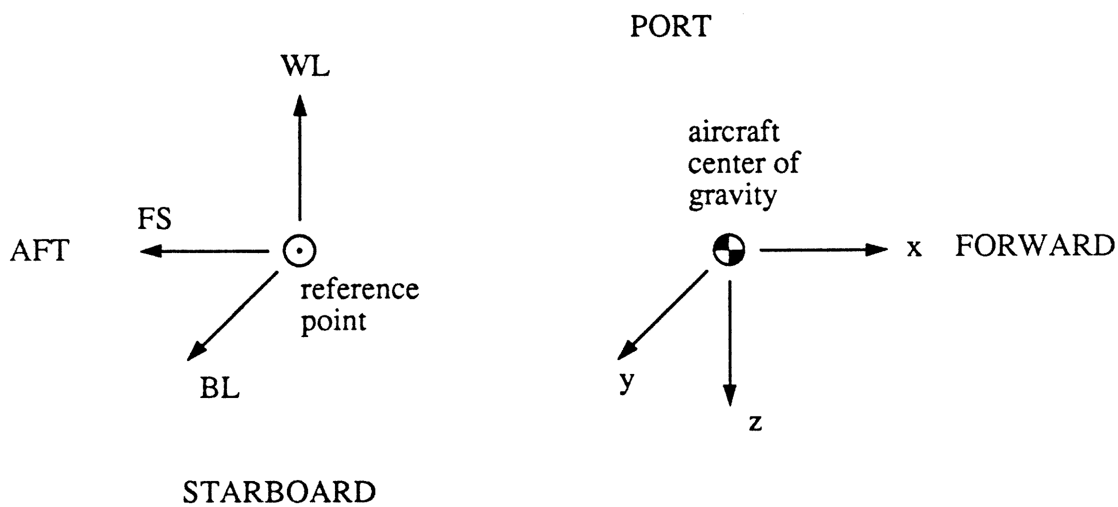


Figure 6-4. Definition of aircraft geometry.

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: ξ_k and γ_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 ξ_k is dimensionless (i.e., m/m or ft/ft); and that the hub angular motion γ_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 ξ and γ_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, γ_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, ξ_k and γ_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_I, 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 ψ_f and elevation angle θ_f. These angles are measured relative to the fuselage axis system (F frame): ψ_f is in the x-y plane, positive counterclockwise from the negative x axis; θ_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 ξ_k is required at the point of application of the auxiliary force. The following input parameters are involved: NAF, FSAF, BLAF, WLAF, AZAF, ELAF, ZETAAF, 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

$$v^T = [(\theta_o\ \theta_{1C}\ \theta_{1S})_1\ (\theta_o\ \theta_{1C}\ \theta_{1S})_2\ \delta_f\ \delta_e\ \delta_a\ \delta_r\ \theta_t\ (f_l)]$$

Except for the auxiliary forces, the units of v are radians internally, and degrees for input/output. The units of the auxiliary forces f_l 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 = (CNTRLZ(11), 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.

$$\begin{bmatrix} M_y/q \\ D/q \\ L/q \end{bmatrix} = \begin{bmatrix} M_o/q \\ f_{WB} + f_{vert}\sin^2(\alpha_{WB} + i_D) + (L/q)^2/(\pi e l_W^2) \\ 0 \end{bmatrix} + \begin{bmatrix} ScC_M \\ SC_D \\ SC_L \end{bmatrix}$$

$$+ \begin{bmatrix} M_\alpha/q & M_{\delta f}/q & M_{\delta F}/q \\ 0 & D_{o\delta f}/q & D_{o\delta F}/q \\ L_\alpha/q & L_{\delta f}/q & L_{\delta F}/q \end{bmatrix} \begin{bmatrix} \alpha_{WB} + i_{WB} \\ \delta_f \\ \delta_F \end{bmatrix}$$

$$\begin{bmatrix} M_x/q \\ M_z/q \\ Y/q \end{bmatrix} = \begin{bmatrix} N_{x\beta}/q & VN_{xp}/q & VN_{xr}/q & N_{x\delta a}/q \\ N_{z\beta}/q & VN_{zp}/q & VN_{zr}/q & N_{z\delta a}/q \\ Y_\beta/q & VY_p/q & VY_r/q & Y_{\delta a}/q \end{bmatrix} \begin{bmatrix} \beta_{WB} \\ p/V \\ r/V \\ \delta_a \end{bmatrix}$$

and for the horizontal tail and vertical tail loads:

$$L_{HT}/q = (L_\alpha/q)(\alpha_{HT} + i_{HT}) + (L_{\delta e}/q)\delta_e$$

$$L_{VT}/q = (L_\alpha/q)(\alpha_{VT} + i_{VT}) + (L_{\delta r}/q)\delta_r$$

$$\epsilon = \frac{(L/q)_{WB}}{f_\epsilon} - \frac{(L_\alpha/q)_{WB}}{f_\epsilon}\frac{l_{HT}}{V}\dot{\alpha}_{WB}$$

$$\sigma = \frac{z_{VT}}{V} p$$

Here δ_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 α^2 dependence. The zero lift angles relative to the fuselage axis system are $-i_{WB}$, $-i_{HT}$, 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, 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:

$$\begin{pmatrix} M/q \\ -X/q \\ -Z/q \end{pmatrix} = \begin{bmatrix} VM_{\dot\alpha}/q & 2M/q+VM_u/q & M_\alpha/q & VM_q/q & M_\delta/q & M_\delta/q \\ VD_{\dot\alpha}/q & 2D/q+VD_u/q & D_\alpha/q-L/q & VD_q/q & D_\delta/q & D_\delta/q \\ VL_{\dot\alpha}/q & 2L/q+VL_u/q & L_\alpha/q+D/q & VL_q/q & L_\delta/q & L_\delta/q \end{bmatrix} \begin{pmatrix} \ddot z/V^2 \\ \dot x/V \\ \dot z/V \\ q/V \\ \delta_e \\ \delta_f \end{pmatrix}$$

$$= \begin{bmatrix} \tfrac{1}{2}Sc^2C_{M\dot\alpha} & Sc(2C_M+MC_{MM}) & ScC_{M\alpha} & \tfrac{1}{2}Sc^2C_{Mq} & ScC_{M\delta} & ScC_{M\delta} \\ \tfrac{1}{2}ScC_{D\dot\alpha} & S(2C_D+MC_{DM}) & S(C_{D\alpha}-C_L) & \tfrac{1}{2}ScC_{Dq} & SC_{D\delta} & SC_{D\delta} \\ \tfrac{1}{2}ScC_{L\dot\alpha} & S(2C_L+MC_{LM}) & S(C_{L\alpha}+C_D) & \tfrac{1}{2}ScC_{Lq} & SC_{L\delta} & SC_{L\delta} \end{bmatrix} \begin{pmatrix} \ddot z/V^2 \\ \dot x/V \\ \dot z/V \\ q/V \\ \delta_e \\ \delta_f \end{pmatrix}$$

$$\begin{pmatrix} L/q \\ N/q \\ Y/q \end{pmatrix} = \begin{bmatrix} N_{x\beta}/q & VN_{xp}/q & VN_{xr}/q & N_{x\delta}/q & N_{x\delta}/q \\ N_{z\beta}/q & VN_{zp}/q & VN_{zr}/q & N_{z\delta}/q & N_{z\delta}/q \\ Y_\beta/q & VY_p/q & VY_r/q & Y_\delta/q & Y_\delta/q \end{bmatrix} \begin{pmatrix} \dot y/V \\ p/V \\ r/V \\ \delta_a \\ \delta_r \end{pmatrix}$$

$$= \begin{bmatrix} SbC_{l\beta} & \tfrac{1}{2}Sb^2C_{lp} & \tfrac{1}{2}Sb^2C_{lr} & SbC_{l\delta} & SbC_{l\delta} \\ SbC_{n\beta} & \tfrac{1}{2}Sb^2C_{np} & \tfrac{1}{2}Sb^2C_{nr} & SbC_{n\delta} & SbC_{n\delta} \\ SC_{y\beta} & \tfrac{1}{2}SbC_{yp} & \tfrac{1}{2}SbC_{yr} & SC_{y\delta} & SC_{y\delta} \end{bmatrix} \begin{pmatrix} \dot y/V \\ p/V \\ r/V \\ \delta_a \\ \delta_r \end{pmatrix}$$

The coefficients are based on a reference area S, chord c, and span b: C_L = lift/qS, C_D = drag/qS, C_M = pitch-moment/qSc; C_y = side-force/qS, C_l = roll-moment/qSb, C_n = yaw-moment/qSb. The dimensionless rate derivatives are $(C)_{\dot\alpha} = \partial C/\partial(\dot\alpha c/2V)$, $(C)_q = \partial C/\partial(qc/2V)$; $(C)_p = \partial C/\partial(pb/2V)$, $(C)_r = \partial C/\partial(rb/2V)$. The stability derivative coefficients 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.

	quantity	+ direction
wing-body loads		
	lift	up
	drag	aft
	side force	right
	pitch moment	nose up
	roll moment	right
	yaw moment	nose right
tail forces		
	horizontal tail lift	up
	vertical tail lift	left
tail cant angle		
	horizontal tail	left
	vertical tail	right
motion		
	sideslip	aircraft to right
	roll rate	to right
	pitch rate	nose up
	yaw rate	nose right
controls (down, lift increase)		
	flaperon	lift up
	elevator	pitch down
	aileron	roll left
	rudder	yaw right

Hence the following signs are expected for the control derivatives.

nonlinear equations

SIDEB = Y_β/q negative

YAWB = $N_{z\beta}/q$ negative

ROLLA = $N_{x_{\delta a}}/q$ negative

LFTEH = L_{HT_δ}/q positive

LFTRV = L_{VT_δ}/q positive

stability derivatives

CYB = $C_{y\beta}$ negative

CNB = $C_{n\beta}$ negative

CLDA = $C_{l_{\delta a}}$ negative

CMDE = $C_{m_{\delta e}}$ negative

CNDR = $C_{n_{\delta r}}$ 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:

$$M_k(\ddot{q}_{sk} + g_s\omega_k\dot{q}_{sk} + \omega_k{}^2q_{sk}) = (Q_k)_{rotors}$$

$$+ \tfrac{1}{2}\rho V^2\left[- F_{q_k\dot{q}_k}\frac{\dot{q}_{sk}}{V} + F_{q_k\delta}\begin{pmatrix}\delta f\\ \delta e\\ \delta a\\ \delta r\end{pmatrix}\right] + (Q_k)_f$$

The corresponding input variables, in namelist NLBODY, are:

QMASS =	M_k	(slug or kg)
QDAMP =	g_s	
QFREQ =	ω_k	(input in Hz, used in rad/sec)
QDAMPA =	$F_{q_k\dot{q}_k}$	(ft^2 or m^2)
QCNTRL =	$F_{q_k\delta}$	(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

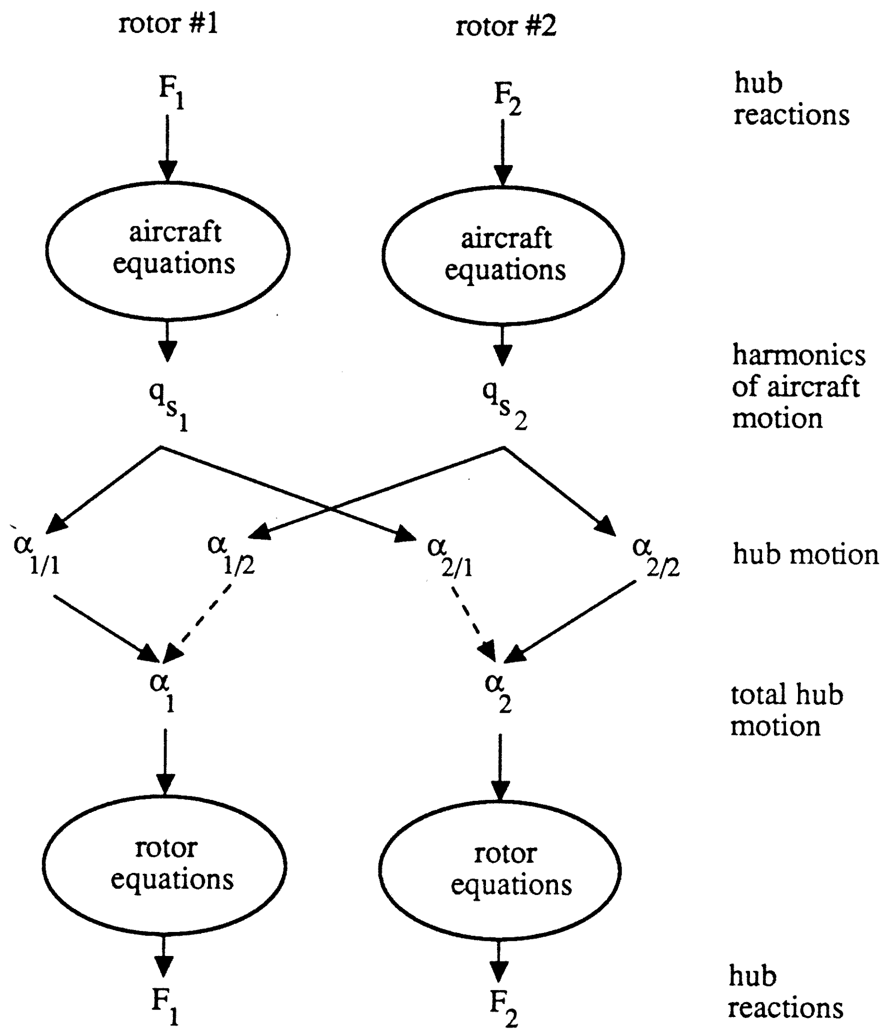


Figure 6-5. Outline of dynamic interaction of the two rotors.

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(1), 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, TEMP, 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, AYAW. (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 AYAW), and turn rate (RTURN) are used. For wind tunnel cases, the test module pitch and yaw angles (APITCH and AYAW) 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 CPTRIM. The wind tunnel trim options use the appropriate targets. The rotor drag force (CXTRIM or XTRIM) is positive aft; the rotor side force is positive toward the advancing side. The longitudinal flapping angle (BCTRIM) 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 (LNGCYC and LATCYC) produce negative main rotor cyclic pitch changes (T1S 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, EPTRIM = 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/m^2 or dB
d) airframe vibratory response: dimensionless or dimensional, as selected by OPDRES (namelist NLLOAD)

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 NLLOAD. 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 NLRTR). 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','1   ','    ','COS ','1   ',
OUTSTR(1,2)='VIB ','RTR1','1   ','    ','SIN ','1   ',
CONSTR(1,1)='HHC ','RTR1','COLL','CS1 ',
CONSTR(1,2)='HHC ','RTR1','COLL','SN1 ',
```

(assuming that the higher harmonic control and the airframe vibratory response are appropriately defined).

6.5.3 <u>Namelist NLRTR</u>

(1) INFLOW. This parameter must be set appropriately when using LEVEL = 1 or 2.

(2) OPSTLL. The intended dynamic stall models are OPSTLL = 3 ($\dot{\alpha}$ stall delay with dynamic stall vortex loads) and OPSTLL = 4 ($(\dot{\alpha})^{1/2}$ 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 $Re_t = M\ Re_{t1}$ is the Reynolds number of the airfoil table data. The parameter Re_{t1} is specified as part of the airfoil file; if RETAB1 in namelist NLRTR is nonzero, it supersedes the value in the airfoil file. If $Re_{t1} = 0$ (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.

$$c/R = \pi\sigma/N$$

$$m = 3I_b/R^3 = 3\rho acR/\gamma = 0.1278\sigma R^2/\gamma N$$

$$I_\theta = (\sigma/N)^2 I_b/R = (\sigma/N)^2 \rho acR^3/\gamma = 0.0426(\sigma/N)^3 R^4/\gamma$$

$$GJ = (0.6366\omega_\theta \Omega R)^2 I_\theta$$

$$EI_{zz} = m\Omega^2 R^4/k_z \quad (k_z = 40 \text{ to } 1000,\ 300 \text{ typical})$$

$$EI_{xx} = m\Omega^2 R^4/k_x \quad (k_x = 2 \text{ to } 30,\ 10 \text{ typical})$$

Here N is the number of blades; R is the rotor radius; ΩR is the rotor tip speed; and ω_θ 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*MPSI*(KNW+1). With MPSI = 24 and NCNW = 45000, the maximum number of radial stations (MRA) for a given KNW are:

KNW	MRA
2	25
3	21
4	19
5	17
6	16
7	15

The required size of the far wake influence coefficient matrix is NC = MPSI**2*MR. With MPSI = 24 and NC = 30000, the maximum number of radial stations (MR, both on and off the rotor disk) is 52. The rotorcraft 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, chose 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*MPSI. At low speeds, more revolutions are required: typically about .4/(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) OPIVTL = 0 (inboard trailed wake not rolled up)
(g) OPRGI = 0 (use the calculated inboard peak radial location)
(h) OPWKBP = 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 FGMXWG = 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/(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) LDMWG = $180^o/\Delta\psi$ = MPSI/2
(h) NDMWG = $90^o/\Delta\psi$ = MPSI/4 fore and aft, $45^o/\Delta\psi$ = MPSI/8 on the sides of the rotor disk; so for MPSI = 24, NDMWG = 3*6, 6*3,6*6, 6*3, 3*6
(i) DQWG = $0.04\lambda_i$ to $0.08\lambda_i$ (large for efficiency, small for accuracy)
(j) QWGDB = $.5\lambda_i$ to λ_i (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 (OPSYMM 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 $0.68\tau_W c_W^2$, where τ_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 (OPWKBP in namelist NLWAKE). See volume I, section 4.4.

6.5.6 Namelist NLLOAD.

(1) MVIB, NRVIB, MVLOAD. The airframe sensors are defined by the parameters in namelist NLLOAD. 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

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 NFREQ and FREQ) with a smaller frequency resolution around the phase = -180 and magnitude = 1 points; typically $\Delta\omega/\omega = 0.005$ 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 MPSIR=MPSI, MREV=1 with only rotor degrees of freedom.
!  Use MPSIR=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,
       MPSIR=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.
       NPRNTT=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 cyclc 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.,WTDELT=25*.1,
       FACTS=.8,EPSTR=.1,
$END
$NLRTR
!  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,KFLMDA=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.,OPHVIB=3*1,
       NUGC=1.,NUGS=1.,LDAMPR=1.,GSB=10*.01,GST=5*.01,
       RCPL=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=2,0,2,0,2,0,2,0,2,2,3,0,3   ! hover
      WKMODL=10*2,3*3                    ! forward flight
      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.
!     OPRWG=6      ! hover
      OPRWG=1      ! forward flight
      KRWG=96,OPWGT=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,MRVBWG=2,LDMWG=12,NDMWG=3*6,6*3,6*6,6*3,3*6,
        DQWG=2*.0005,ITERWG=2,FACTWG=.5,IPWGDB=2*6,QWGDB=.1,
$END
$NLRTR
!  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,.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,KFLMDA=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.,OPHVIB=3*1,
        NUGC=1.,NUGS=1.,LDAMPR=1.,GSB=10*.01,GST=5*.01,
        RCPL=1.,MBLADE=-1.,MRB=50,MRM=50,
        HINGE=2,EPMODE=.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 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=2,0,2,0,2,0,2,0,2,2,3,0,3    !  hover
       WKMODL=10*2,3*3                     !  forward flight
       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.
!     OPRWG=6      !  hover
      OPRWG=1      !  forward flight
      KRWG=96,OPWGT=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,MRVBWG=2,LDMWG=12,NDMWG=3*6,6*3,6*6,6*3,3*6,
      DQWG=2*.0005,ITERWG=2,FACTWG=.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.,ENGPOS=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.
      KOCFE=1.,KCCFE=1.,KSCFE=1.,KPCFE=1.,
      KFOCFE=1.,KFCCFE=1.,KFSCFE=1.,KFPCFE=1.,
      KROCFE=1.,KRCCFE=1.,KRSCFE=1.,KRPCFE=1.,
!
!  Rotor/airframe aerodynamic interference model.
!  WXCIRC and WXTHCK given far field values.
      OPI1BP=4*1,OPI2BP=4*1,WXCIRC=5*.25,WXTHCK=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,NSPAN=25,NPSI=24,     !  hover
       NPOLAR=2,NSPAN=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 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,NSPAN=25,NPSI=24,     !  hover
       NPOLAR=2,NSPAN=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 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
$NLFLUT
!  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,OPSYMM=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,KASGE=1.,OPRINT=1,
!
!  Evaluation of differential equations.
       DALPHA=1.,DMACH=.05,DELTA=.01,
       MPSICC=12,MPSIPC=720,NINTPC=1,
!
```

```
!  Analysis definition.
       ANTYPE=1,3*0,NSYSAN=0,
       OPBODE=1,MBODE=1,
       OPTIME=3,
       LGUST=3*0.,GRMS=3*1.,MGUST=0,OPSPEC=1,
       F0PLT=0.,F1PLT=2.,NF0PLT=-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,DELTA=.01,
       OPPRNT=3*0,1,NPRNTP=0,NPRNTL=0,
!
!  Analysis definition.
       EQTYPE=3*1,ANTYPE=1,4*0,NSYSAN=0,
       OPBODE=1,MBODE=1,
       OPTIME=3,
       LGUST=3*0.,GRMS=3*1.,MGUST=0,OPSPEC=1,
       F0PLT=0.,F1PLT=2.,NF0PLT=-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
```

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.

Namelist	Parameter	Subroutine	Contents
NLTRIM	NTFILE	PERFR1,PERFR2	motion harmonics
NLTRIM	NEFILE	MODEP1,MODEP2	blade modes
NLLOAD	NVFILE	LOADV	vibration
NLLOAD	NGFILE	GEOMP1,GEOMP2	wake geometry
NLLOAD	NDFILE	LOADD1,LOADD2	motion and aero
NLLOAD	NAFILE	LOADA1,LOADA2	aerodynamics
NLLOAD	NHFILE	LOADH1,LOADH2	hub loads
NLLOAD	NRFILE	LOADS1,LOADS2	blade loads
NLLOAD	NNFILE	NOISR1,NOISR2	noise
NLFLUT	NLFILE	FLUT	Floquet eigenvalues
NLFLUT	NLFILE	FLUTL	eigenvalues
NLFLUT	NBFILE	FLUTL	frequency response
NLFLUT	NTFILE	FLUTL	linear time history
NLFLUT	NMFILE	FLUTL	matrices
NLSTAB	NLFILE	STABL	eigenvalues
NLSTAB	NBFILE	STABL	frequency response
NLSTAB	NTFILE	STABL	linear time history
NLSTAB	NMFILE	STABL	matrices
NLSTAB	NTFILE	STABLI	time history
NLTRAN	NTFILE	TRAN	time history

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

C

F

O

P