Enhanced Modeling of First-Order Plant Equations of Motion for Aeroelastic and Aeroservoelastic Applications

Anthony S. Pototzky[1]

NASA Langley Research Center, Hampton Virginia 23681

A methodology is described for generating first-order plant equations of motion for aeroelastic and aeroservoelastic applications. The description begins with the process of generating data files representing specialized mode-shapes, such as rigid-body and control surface modes, using both PATRAN and NASTRAN analysis. NASTRAN executes the 146 solution sequence using numerous Direct Matrix Abstraction Program (DMAP) calls to import the mode-shape files and to perform the aeroelastic response analysis. The aeroelastic response analysis calculates and extracts structural frequencies, generalized masses, frequency-dependent generalized aerodynamic force (GAF) coefficients, sensor deflections and load coefficients data as text-formatted data files. The data files are then re-sequenced and re-formatted using a custom written FORTRAN program. The text-formatted data files are stored and coefficients for s-plane equations are fitted to the frequency-dependent GAF coefficients using two Interactions of Structures, Aerodynamics and Controls (ISAC) programs. With tabular files from stored data created by ISAC, MATLAB generates the first-order aeroservoelastic plant equations of motion. These equations include control-surface actuator, turbulence, sensor and load modeling. Altitude varying root-locus plot and PSD plot results for a model of the F-18 aircraft are presented to demonstrate the capability.

Nomenclature

$A_0 \ldots A_6$	=	s-plane fit coefficients
A-bar	=	RMS with the gust intensity factor, $\sigma = 1$ in/sec.
ASE	=	aeroservoelastic
$\bar{\boldsymbol{C}}$	=	generalized damping matrix
$\widetilde{\boldsymbol{D}}$	=	generalized damping matrix including aerodynamic effects
DCM	=	data complex manager of the ISAC code
DMAP	=	direct matrix abstraction program-NASTRAN programming language
DMIG	=	direct matrix input at points-type of gridpoint data entry for NASTRAN
DOF	=	degree of freedom
DYNARES	=	dynamic response program (within which s-splane fits are performed) of the ISAC code
EOM	=	equations of motion
f	=	frequency, Hz.
FE	=	finite element
FEM	=	finite element model
GAF	=	generalized aerodynamic force coefficients
GLA	=	gust load alleviation
ISAC	=	Interaction of Structures, Aerodynamics and Controls
k	=	reduced frequency
$\boldsymbol{K}$	=	generalized stiffness matrix
$\widetilde{\boldsymbol{K}}$	=	generalized stiffness matrix including aerodynamic effects
$\boldsymbol{M}$	=	generalized mass matrix
Lanczos	=	method of NASTRAN EIGRL eigenvalue analysis
LMAT	=	NASTRAN internally stored matrix data

[1]Aerospace Engineer, Aeroelasticity Branch, M. S. 340/ Anthony.S.Pototzky@nasa.gov, Senior Member AIAA.

$\tilde{\boldsymbol{M}}$	=	generalized mass matrix including aerodynamic effects
MFT	=	matched filter theory
N_0	=	mean frequency of zero crossings
PHIGR	=	NASTRAN internally stored data block for storing control mode shapes
PHIGRLM	=	NASTRAN internally stored data block for storing load mode shapes
PHRBNEW	=	NASTRAN internally stored data block
PSD	=	power spectral density
$\bar{\boldsymbol{q}}$	=	dynamic pressure
RMS	=	root mean square
s	=	Laplace variable
σ	=	gust intensity factor
$\bar{s}$	=	Laplace variable
SINV	=	option in EIGR eigenvalue analysis to use the enhanced inverse power method of NASTRAN
v	=	flight velocity
v_{gust}	=	gust penetration velocity

I. Introduction

Over the years NASTRAN (Ref. 1) has been a versatile tool to perform many types of aeroelastic-related analyses. It is used to perform static aeroelastic analysis, such as trim, control reversal or divergence. NASTRAN flutter analysis has been well utilized by the aircraft industry as the "work horse" for design and certification processes to clear the aircraft flight envelope. Many types of frequency response analysis can be performed, such as power spectral density (PSD), random response, RMS and mean frequency of zero crossings. With frequency analysis, users can input their own atmospheric turbulence spectra greatly expanding the versatility of NASTRAN. The more advanced capability of NASTRAN includes performing transient aeroelastic response analysis where user-supplied time-history excitations can be very general, such as those representing atmospheric turbulence. This transient capability may be coupled to simple feedback control systems to evaluate controller stability and performance with aeroelastic systems.

As many capabilities as NASTRAN has, there are many analyses with its present suite of solution capabilities that are extremely difficult or impossible to perform. Examples of analyses impossible to perform in NASTRAN are real-time simulation or nonlinear time-domain analysis of aeroelastic aircraft systems or applying "modern techniques" for designing control systems. Other examples, where NASTRAN cannot be used directly, are in matched filter theory (MFT) analysis for obtaining time-correlated gust load results; for Lyapunov equation analysis for computing RMS and N_o values of state variables and output quantities; or for performing aeroelastic simulation analysis where Mach number, dynamic pressure and velocity are continuously varied throughout the analysis.

Over the last few years, modern techniques of designing control systems and the fact that aircraft have become much more flexible required enhanced techniques of representing aircraft flight dynamics. Using "6 DOF" representations of aircraft to design control systems many times has insufficient fidelity to capture the necessary physics of the problem. As employed in this paper, the term "aeroservoelasticity" includes all the typical components (rigid-body translations and rotations of the airframe, rotations of the control surfaces about their hingelines, the flexible modes of the aircraft, unsteady aerodynamics, sensor equations, and actuator dynamics), but does not include control laws.

Two software tools have been developed to generate aeroservoelastic (ASE) plant equations of motion (EOM). One is the NASA developed Interaction of Structures, Aerodynamics and Controls (ISAC) suite of modules (Ref. 2) for generating first-order aeroservoelastic plant equations of motion, which can be used in conjunction with feedback controller designs, to analyze the stability and performance of closed-loop control systems. Until the development of ISAC, all aeroelastic analyses were performed in the frequency domain. ISAC permitted aeroelastic analysis in the time domain by transforming the unsteady aerodynamics into the s-plane. Unfortunately, with the improvements of computer hardware and changes in FORTRAN compilers, over the years several of ISAC's modules, have become victims of neglect with lack of consistent maintenance and have been rendered unusable.

Another, more recently developed software tool, which generates ASE plant EOM is available commercially from ZONA Technology (Ref. 3). ZONA's aeroservoelastic capability employs techniques similar to those in ISAC for generating first-order ASE plant equations of motion. The software incorporating modern algorithms and numerical techniques forms state-space ASE plant equations of motion, which can be combined with control laws, and performs analyses such as gust response, control-system stability and aircraft maneuvers.

This paper describes the development of a capability to generate first-order linear-time-invariant ASE plant EOM. The capability includes: (1) NASTRAN DMAP (Ref. 4) codes; (2) a custom-written FORTRAN code; (3) modules from ISAC; and (4) MATLAB (Ref. 5) scripts. The capability was initially developed as a way to extend and supplement the capabilities resident in NASTRAN's aeroelastic response analysis. However, it is possible to attain some of the components, such as unsteady aerodynamics, from other sources, such as the ZONA aerodynamics codes.

The functions of the four elements are:

1. NASTRAN DMAP codes -
   Modify NASSTRAN solution sequence 146, dynamic aeroelastic analysis, so that necessary matrices are output from NASTRAN.
2. FORTRAN code -
   Re-format, re-sequence and re-select the NASTRAN output matrices.
3. ISAC modules -
   Create s-plane fits of the unsteady aerodynamics.
4. MATLAB scripts -
   Assemble the final ASE plant EOM.

This paper also outlines two additional capabilities not directly related to generating first-order ASE plant EOM. The first of these is the generation of shear forces, bending moments and torsion moments using the concept of the "load mode shapes." The second addresses a method of scaling that easily converts the aeroelastic equations of motion of a full-sized aircraft into ones of a wind-tunnel model, as described in reference 6.

A full-sized, half-model of an F-18 is used to demonstrate the first-order ASE plant EOM capabilities. With the first-order ASE plant EOM, flutter analysis in the form of root-locus results is presented with matched-point altitude variation. Additional analysis results presented in this paper are PSD and RMS of loads and accelerations from atmospheric turbulence models using first-order ASE plant EOM.

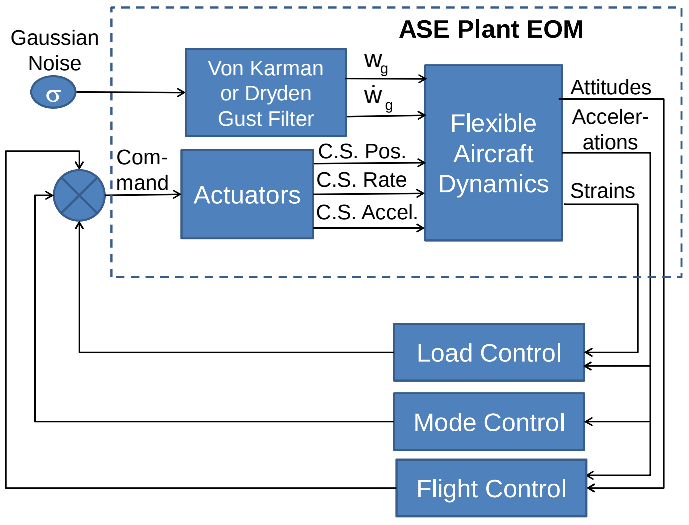
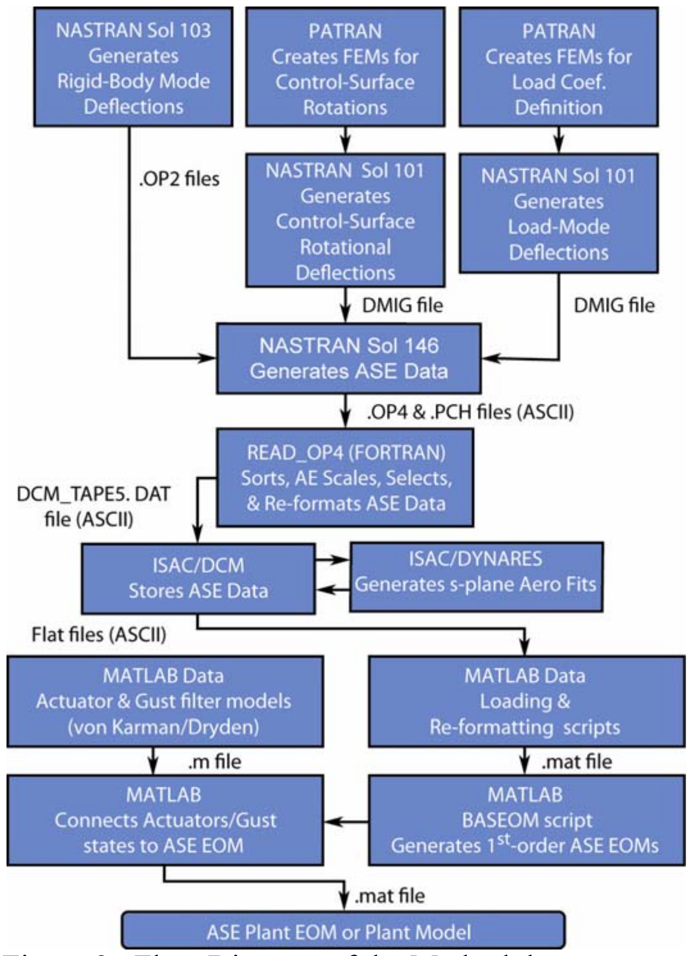


Figure 1. Aeroservoelastic system diagram.

II. Aeroservoelastic System Representation

In modeling aeroservoelastic systems, there are three major components, which involve generally the aerodynamics, the aircraft structure and the control system. However, in a more detailed study, which this paper covers, it can include many more subsystem components as summarized in figure 1. The part the diagram enclosed by the dashed line can be identified as the ASE "plant" and is the part that is being addressed by the methodology described in the paper.

Some of the subsystem components like the flight control may be associated more directly with rigid-body dynamics of the aircraft and others like the load control (such as a gust load alleviation-or, GLA-system) more directly to the aircraft flexibility, but for highly flexible aircraft, all the subsystems of the aircraft affect to some degree all the other subsystem parts such as rigid-body and flexible parts. However, with extremely flexible aircraft such as the Helios, the rigid-body and elastic parts are highly coupled to the extent that one or the other cannot be modeled by itself



NASTRAN Sol 103 Generates Rigid-Body Mode Deflections
PATRAN Creates FEMs for Control-Surface Rotations
PATRAN Creates FEMs for Load Coef. Definition
.OP2 files
NASTRAN Sol 101 Generates Control-Surface Rotational Deflections
NASTRAN Sol 101 Generates Load-Mode Deflections
DMIG file
DMIG file
NASTRAN Sol 146 Generates ASE Data
.OP4 & .PCH files (ASCII)
READ_OP4 (FORTRAN) Sorts, AE Scales, Selects, & Re-formats ASE Data
DCM_TAPE5. DAT file (ASCII)
ISAC/DCM Stores ASE Data
ISAC/DYNARES Generates s-plane Aero Fits
Flat files (ASCII)
MATLAB Data Actuator & Gust filter models (von Karman/Dryden)
MATLAB Data Loading & Re-formatting scripts
.m file
.mat file
MATLAB Connects Actuators/Gust states to ASE EOM
MATLAB BASEOM script Generates 1st-order ASE EOMs
.mat file
ASE Plant EOM or Plant Model

Figure 2. Flow Diagram of the Methodology.

without the considerable loss of modeling fidelity. The paper will examine the methods to construct many of the subsystem components comprising the aeroservoelastic plant of the aeroservoelastic system.

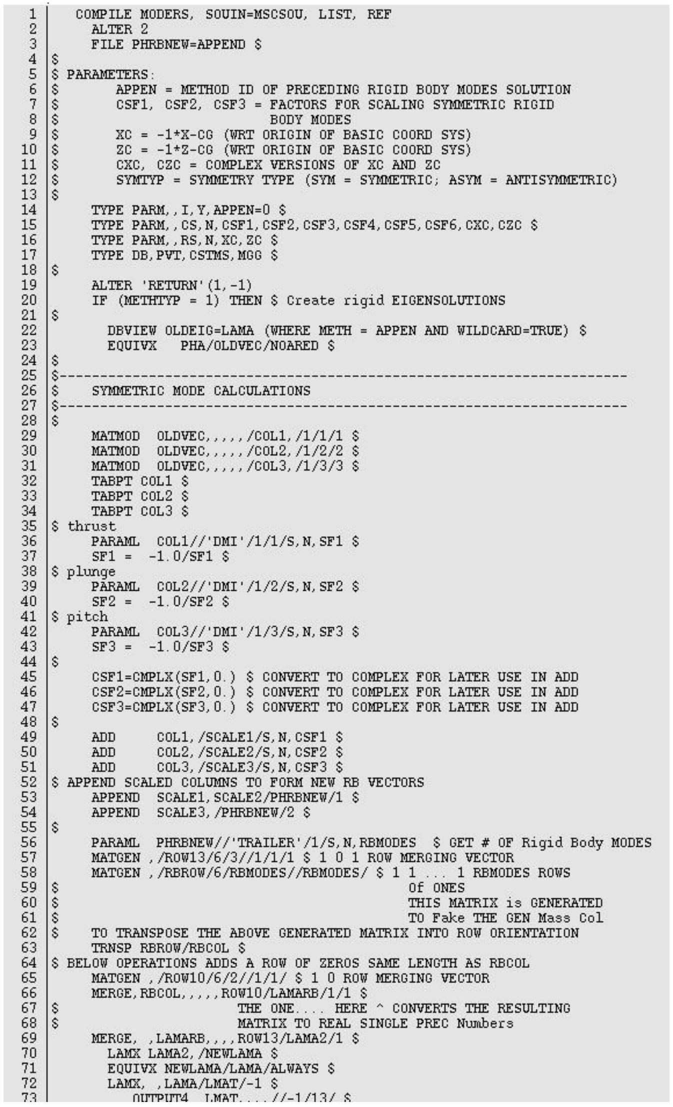
III. **Methodology**

A description is provided of the methodology developed to generate aeroservoelastic first-order equations. The flow diagram in figure 2 shows tasks, data paths and computational software used to create, gather and store the intermediate data to produce the desired results.

It can be observed from the flow diagram that, in this methodology, several diverse software capabilities are combined to accomplish the ASE plant EOM generation task. If the data structure is known, the simplest way to connect these capabilities is by transferring the appropriate data in an ASCII format. This approach was used to avoid, or at least to minimize, data format incompatibility problems between various computational platforms. A lesser reason to use ASCII was to aid the initial development of the methodology by being able to observe data structures, types and formats directly and very easily with a simple text editor. There are exceptions to this ASCII-only philosophy, for example, when .OP2 and .mat binary files are reused by the same software capability that originally generated them.

In the following sections of the paper, salient details of the various tasks are given in order to describe the methodology including details of codes used in performing the task as well as intermediate results obtained.

Table 1. DMAP to normalize rigid-body deflections and set frequencies to zero.



```
   COMPILE MODERS, SOUIN=MSCSOU, LIST, REF
     ALTER 2
     FILE PHRBNEW=APPEND $
$
$ PARAMETERS:
$       APPEN = METHOD ID OF PRECEDING RIGID BODY MODES SOLUTION
$       CSF1, CSF2, CSF3 = FACTORS FOR SCALING SYMMETRIC RIGID
$                          BODY MODES
$       XC = -1*X-CG (WRT ORIGIN OF BASIC COORD SYS)
$       ZC = -1*Z-CG (WRT ORIGIN OF BASIC COORD SYS)
$       CXC, CZC = COMPLEX VERSIONS OF XC AND ZC
$       SYMTYP = SYMMETRY TYPE (SYM = SYMMETRIC; ASYM = ANTISYMMETRIC)
$
     TYPE PARM,,I,Y,APPEN=0 $
     TYPE PARM,,CS,N,CSF1,CSF2,CSF3,CSF4,CSF5,CSF6,CXC,CZC $
     TYPE PARM,,RS,N,XC,ZC $
     TYPE DB,PVT,CSTMS,MGG $
$
     ALTER 'RETURN'(1,-1)
     IF (METHTYP = 1) THEN $ Create rigid EIGENSOLUTIONS
$
       DBVIEW OLDEIG=LAMA (WHERE METH = APPEN AND WILDCARD=TRUE) $
       EQUIVX   PHA/OLDVEC/NOARED $
$
$---------------------------------------------------------------------
$    SYMMETRIC MODE CALCULATIONS
$---------------------------------------------------------------------
$
     MATMOD  OLDVEC,,,,,/COL1,/1/1/1 $
     MATMOD  OLDVEC,,,,,/COL2,/1/2/2 $
     MATMOD  OLDVEC,,,,,/COL3,/1/3/3 $
     TABPT COL1 $
     TABPT COL2 $
     TABPT COL3 $
$ thrust
     PARAML  COL1//'DMI'/1/1/S,N,SF1 $
     SF1 =  -1.0/SF1 $
$ plunge
     PARAML  COL2//'DMI'/1/2/S,N,SF2 $
     SF2 =  -1.0/SF2 $
$ pitch
     PARAML  COL3//'DMI'/1/3/S,N,SF3 $
     SF3 =  -1.0/SF3 $
$
     CSF1=CMPLX(SF1,0.) $ CONVERT TO COMPLEX FOR LATER USE IN ADD
     CSF2=CMPLX(SF2,0.) $ CONVERT TO COMPLEX FOR LATER USE IN ADD
     CSF3=CMPLX(SF3,0.) $ CONVERT TO COMPLEX FOR LATER USE IN ADD
$
     ADD     COL1,/SCALE1/S,N,CSF1 $
     ADD     COL2,/SCALE2/S,N,CSF2 $
     ADD     COL3,/SCALE3/S,N,CSF3 $
$ APPEND SCALED COLUMNS TO FORM NEW RB VECTORS
     APPEND  SCALE1,SCALE2/PHRBNEW/1 $
     APPEND  SCALE3,/PHRBNEW/2 $
$
     PARAML  PHRBNEW//'TRAILER'/1/S,N,RBMODES  $ GET # OF Rigid Body MODES
     MATGEN ,/ROW13/6/3//1/1/1 $ 1 0 1 ROW MERGING VECTOR
     MATGEN ,/RBROW/6/RBMODES//RBMODES/ $ 1 1 ... 1 RBMODES ROWS
$                                             Of ONES
$                                             THIS MATRIX is GENERATED
$                                             TO Fake THE GEN Mass Col
$    TO TRANSPOSE THE ABOVE GENERATED MATRIX INTO ROW ORIENTATION
     TRNSP RBROW/RBCOL $
$ BELOW OPERATIONS ADDS A ROW OF ZEROS SAME LENGTH AS RBCOL
     MATGEN ,/ROW10/6/2//1/1/ $ 1 0 ROW MERGING VECTOR
     MERGE,RBCOL,,,,,ROW10/LAMARB/1/1 $
$                     THE ONE.... HERE ^ CONVERTS THE RESULTING
$                     MATRIX TO REAL SINGLE PREC Numbers
     MERGE, ,LAMARB,,,,ROW13/LAMA2/1 $
       LAMX LAMA2,/NEWLAMA $
       EQUIVX NEWLAMA/LAMA/ALWAYS $
       LAMX, ,LAMA/LMAT/-1 $
          OUTPUT4  LMAT,,,,//-1/13/ $
```

A. **Generating Prescribed Rigid-Body Mode Shapes**

The most convenient way for generating rigid-body mode shapes would be to use the more advanced Lanczos eigenvalue analysis method of solution 146. Using this direct approach, these rigid-body mode shapes could be produced in conjunction with the flexible modes in one eigenvalue extraction process within the same NASTRAN run used to produce all the other ASE data. Unfortunately, this approach produces a set of rigid-body mode-shape deflections that does not give the desired results for ASE analysis. In the early development of the current methodology, it was found that the Lanczos method produces rigid-body mode-shape deflections in arbitrary directions unsuitable for designing and analyzing conventional flight control systems, even though the mode shapes produced were orthogonal. With this shortfall with Lanczos method coupled with SINV's known inconsistency of finding flexible modes, the process of generating rigid-body modes requires a separate run of the NASTRAN 103 solution using the SINV option of the EIGR method to generate only rigid-body mode shapes prior to running the solution 146. For a free-free finite-element model, the SINV option has the capability of generating the "pure" pitch, plunge and fore-aft rigid-body mode shapes about the finite element model's center of mass location. In addition, for a full span FEM configuration, the SINV option can also generate the roll, the side translation and the yaw rigid-body mode shapes. Even by using the SINV option, the rigid-body results produced by NASTRAN are not completely satisfactory. Adjustments to the rigid-

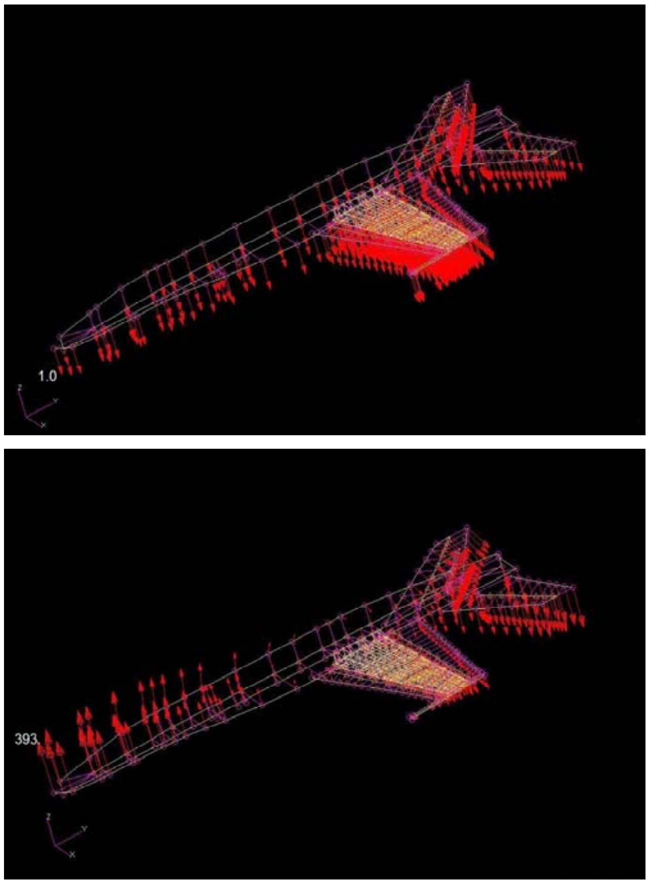


Figure 3. PATRAN plots of plunge (top) and pitch (bottom) rigid-body mode deflections and the F-18 finite-element model.

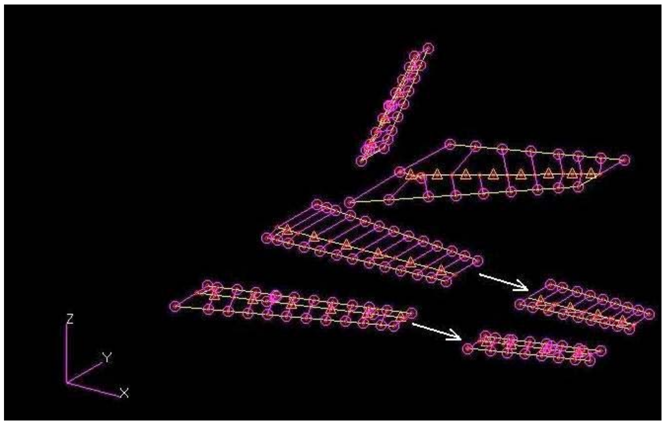


Figure 4. PATRAN plot of all the F-18 control-surface FE models.

body deflections are still required because these deflections are not normalized to unity using the standard NASTRAN 103 solution resources. For the current methodology, a DMAP was developed and used to set the pitch mode to uniform one radian deflections and the plunge mode to uniform unity negative downward deflections. This DMAP alter is given in table 1.

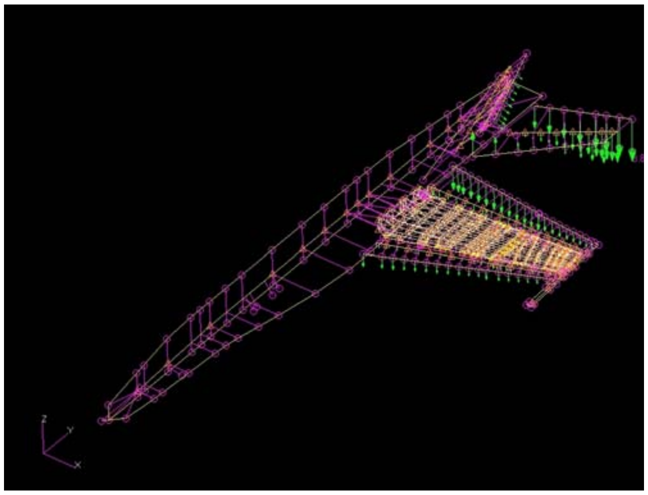


Figure 5. PATRAN plot of the F-18 with all the control-surfaces deflected one radian.

Besides the comment statements, most of the DMAP listed in table 1 sets up data blocks or parameter names and extracts data block parameter values to be used in executing the DMAP task. The DMAP on lines 29 to 34 outputs each column of the rigid-body gridpoint deflections to determine which of the gridpoint degrees of freedom is needed to scale each of the respective mode shapes. The DMAP from lines 36 to 43 calculates the scale factor for each mode and from lines 49 to 51 scales each rigid-body mode shape. The APPEND commands of lines 53 and 54 restack the column of normalized rigid-body mode-shape deflections into the PHRBNEW data block which is output to a .op2 binary data file using the OUTPUT statement of line 74. The PHRBNEW data block and as well as other data blocks mentioned in the paper, which can be user defined, are variables in NASTRAN to internally store data. In a similar data manipulation operation, the function of the DMAP statements from lines 57 to 73 sets the nearly zero eigenvalue frequencies of the rigid-body vibration modes to exactly zero. The NASTRAN variable LMAT contains the modified eigen-frequencies as well as some additional data and is output onto another .op2 binary data file as shown on line 73.

Along with outputting the deflections to the .op2 file, the NASTRAN solution 103 outputs plot information of the plunge and pitch modes to a separate .xdb file which was rendered for viewing with PATRAN as shown in figure 3. For the plunge mode, all the deflections shown by the red arrows are 1.0 inch corresponding to the length unit used in the F-18 FE modeling. For the pitch mode, the largest deflection is 393 inches represented by the arrow at the nose of the FE model, which is the deflection obtained rotating the vehicle one radian about its center of mass.

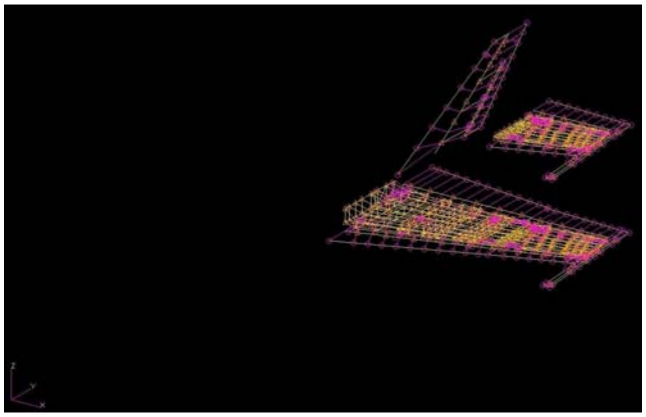
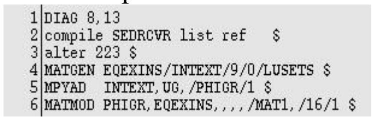


Figure 6. PATRAN plot of the three F-18 load-mode FE models.

B. Generating Control-Surface Mode Shapes

To generate control-surface mode shapes for the solution 146, only the gridpoint deflections are needed from a one-radian rotation of each control surface. The initial task in this process is to tailor the full FE model with PATRAN to construct the individual finite-element structure for each control surface. The PATRAN rendered plot of figure 4 displays the six control-surface FE models, which are comprised of two leading and two trailing edge surfaces on the wing, horizontal tail and rudder surfaces on the F-18 empennage. For clarity, the two outboard control-surface FE models in the plot were moved slightly outboard as indicated by the white arrows. To enable NASTRAN to perform the deflection calculation, the next step of the process is to apply with PATRAN a one-radian forced-rotation boundary condition to a gridpoint anywhere along the hingeline. At this level of control-surface FE model refinement, NASTRAN solution 101 (static analyses) will generate gridpoint deflections onto .f06 and .xdb files. Now, by adding the

Table 2. DMAP to output DMIG formatted punch file.



```
1 DIAG 8,13
2 compile SEDRCVR list ref   $
3 alter 223 $
4 MATGEN EQEXINS/INTEXT/9/0/LUSETS $
5 MPYAD  INTEXT,UG,/PHIGR/1 $
6 MATMOD PHIGR,EQEXINS,,,,/MAT1,/16/1 $
```

DMAP shown in table 2 to this NASTRAN input data, the control-surface deflections in a DMIG format will also be output onto a punch file. Specifically, MATMOD, which is a DMAP utility module (Ref. 4), of statement 6 generates the DMIG entries using option 16. Since each 101 solution generates a different set of deflections in a DMIG entry form, some text editing is necessary to combine these separate DMIG entries into a single ASCII file. The single file, which contains all the DMIG entries, will eventually be input by a NASTRAN solution 146 run to form PHIGR,

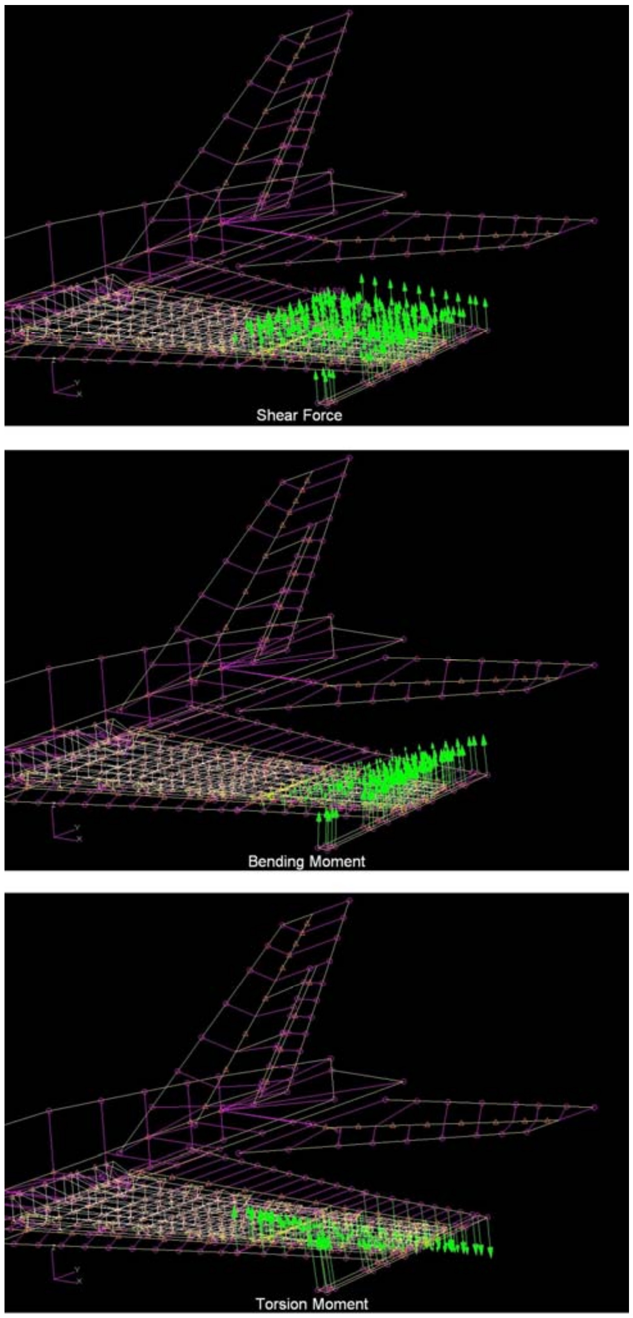


Figure 7. PATRAN plots of the F-18 shear force, bending and torsion-moment load-mode shapes.

which is the internal variable name for the direct input matrix. This part of the computations will be explained later in more detial, but suffice to to say now, in the PHIGR matrix, the control surface deflections are stored in separate columns, each representing one of the control-surface modes.

One of the main attractions of using the DMIG form of data entry is that only the subset of gridpoint deflections as defined by the smaller control-surface FE models need to be input into the full F-18 FE model during the execution of NASTRAN solution 146 and all the remaining gridpoint deflections of the full model are automatically set to zero values. In figure 5, the arrows represent gridpoint deflections of all the control surfaces when each is rotated one radian in the positive direction or the direction of arrows pointing downward.

C. Generating Load Mode Shapes

The process for generating the load modes was inspired by the process used to generate control-surface modes to the extent that the same steps apply in generating the deflections for either set of DMIG entries. Load mode shapes are not really mode shapes as produced by an eigenvalue analysis, but they are used the same way as mode shapes. Their function is to produce one unit gridpoint defections for shear forces and produce one radian gridpoint deflections representing their respective moment arms for the bending and torsion moments computations. Instead of defining FE structures outboard of a control-surface hingeline, the FE models used to generate load modes are defined outboard of a "load station." These load stations can be placed across any part of the structure where loads are calculated or, using NASTRAN terminology, where loads need to be "monitored." For the F-18 case, the load stations were placed at

Table 3. The DMAP to import /combine various mode shapes and output ASE data.

```
$ DMAP to generate combo of RB, FLEX, CS modes, GMass, Freqs. & Load Coefs.
$ Written by A.S. Pototzky, 03-30-2010
compile phase0 list ref   $
    alter 'gp4'  $
    mtrxin  ,,matpool,eqexins,sils,/phigr,phigrlm,/lusets////   $
    call dbstore phigr,phigrlm,,,//0/0/'dball'/0   $
COMPILE MODERS, SOUIN=MSCSOU, LIST, REF
     ALTER 2
     FILE LAMA1=APPEND/PHIA1=APPEND/PHRBNEW=APPEND $
     TYPE PARM,,I,Y,APPEN=0 $
     type parm,,i,n,nophigr  $
     TYPE DB,PVT,CSTMS,MGG $
     ALTER 'RETURN'(1,-1)
     IF (METHTYP = 1) THEN $ APPENDING EIGENSOLUTIONS
     MESSAGE //' ' $
     MESSAGE //' APPENDING FLEXIBLE MODES ONTO THE RIGID BODY' $
     MESSAGE //' MODES MODES CALCULATED USING METHOD'/APPEN $
     MESSAGE //' ' $
$ GET OLD EIGENVECTORS AND EIGENVALUES
       INPUTT4  /PHRBNEW,,,,/1/11/-1/1 $
       LAMX, ,LAMA/LMAT/-1 $
       INPUTT4 /OLDLMAT,,,,/1/13/-1/1   $
       MATGEN ,/COL34/6/5/2/2/1 $ 0 0 1 1 0 COLUMN PARTITIONING VECTOR
       MATGEN ,/ROW13/6/3//1/1/1 $ 1 0 1 ROW MERGING VECTOR
       PARTN OLDLMAT,COL34,/01,02,OLDMAT1,04/1/1 $
$                      THE ONE........... HERE ^ CONVERTS
$                      THE RESULTING
$                      MATRIX TO REAL SINGLE PREC Numbers
       PARTN LMAT,COL34,/M1,M2,MAT1,M4/1 $
       TRNSP OLDMAT1/OLDMAT $
       TRNSP MAT1/MAT $
     call dbfetch /phigr,phigrlm,,,/0/0/0/0/nophigr    $
     PARAML  PHIGR//'TRAILER'/1/S,N,NCMODES  $ GET # OF CONTROL MODES
     PARAML  PHIGRLM//'TRAILER'/1/S,N,LDMODES  $ GET # OF Load MODES
     MATGEN ,/CNRLROW/6/NCMODES//NCMODES/ $ 1 1 ... 1 NCMODES ROWS
$                                         Of ONES
$                                         THIS MATRIX is GENERATED
$                                         to fake the GEN Mass Col
$    TO TRANSPOSE THE ABOVE GENERATED MATRIX INTO ROW ORIENTATION
     TRNSP CNRLROW/CNRLCOL $
$    BELOW OPERATIONS ADDS A ROW OF ZEROS SAME LENGTH AS CNRLCOL
     MATGEN ,/ROW10/6/2//1/1/ $ 1 0 ROW MERGING VECTOR
     MERGE,CNRLCOL,,,,,ROW10/LAMACNRL/1/1 $
$                      THE ONE.... HERE ^ CONVERTS THE RESULTING
$                      MATRIX TO REAL SINGLE PREC Numbers
       APPEND OLDMAT,MAT/LAMAI $
       APPEND LAMAI,LAMACNRL/LAMA1 $
$ THE MERGE OPERATION BELOW ADDS ROW OF ZEROS BETWEEN THE ABOVE FIRST
$ AND SECOND ROWS
       MERGE, ,LAMA1,,,,ROW13/LAMA2/1 $
$  OUTPUTS THE VIBRATION ANALYSIS FREQUENCIES TO ASCII OUTPUT4 FORMAT TO UNIT 23
$  THESE INCLUDES THE ZERO FREQ'S FOR THE RIGID BODY MODES AND THE CONTROL SURFA
$  MODES.  EXCEPT FOR THE FLEX MODES ALL OTHER GEN. MASS TERMS ARE BOGUS
       OUTPUT4  LAMA2,,,,//-1/23/99//9 $
       LAMX LAMA2,/NEWLAMA $
     APPEND PHRBNEW,PHA/PHIA1 $
     ADD PHIGR,/UGB/(1.0,0.0)//3  $
     ADD PHIGRLM,/UGBLM/(1.0,0.0)//3  $
       PARAML PHIA1//'TRAILER'/1/S,N,NEIGV $ GET CORRECT NUMBER OF EIGENVECTORS
       MESSAGE //' NUMBER OF EIGENVECTORS AFTER APPEND ='/NEIGV $
       MESSAGE //' FOLLOWING EIGENVALUE SUMMARY CONTAINS COMBINED MODES' $
       EQUIVX NEWLAMA/LAMA/ALWAYS $
       OFP LAMA// $
$-----------------------------------------------------------------------
$ TEMPORARILY APPEND CONTROL MODES TO COMPUTE
$ GENERALIZED (COUPLED) MASS MATRIX
     PARAML  PHA//'TRAILER'/2/S,N,ASIZE $ GET SIZE OF A-SET
     PARAML  UGB//'TRAILER'/2/S,N,GSIZE  $ GET SIZE OF G-SET
     PARAML  UGB//'TRAILER'/1/S,N,NCMODES  $ GET # OF CONTROL MODES
     MESSAGE //'NUMBER OF CONTROL MODES READ = '/NCMODES $
     MESSAGE //'G-SET SIZE OF CONTROL MODES = '/GSIZE $
     MESSAGE //'A-SET SIZE (FROM CURRENT RUN) = '/ASIZE $
     IF (GSIZE > ASIZE) THEN $ SEE IF ASIZE = GSIZE
          MESSAGE //'PARITIONING OLD EIGENVECTORS TO A-SIZE' $
          UPARTN     USET,UGB/UA,,,/'G'/'A'/'O'/1 $ PARTITION TO A-SET SIZE
       ELSE
          MESSAGE //'NO A-SET REDUCTION: '/
                  'EQUIVALENCING OLD G-SIZE EIGENVECTORS TO A-SIZE' $
          EQUIVX     UGB/UA/ALWAYS $ IF NO REDUCTION THEN EQUIV
     ENDIF $
$ Mode displacement related calutions below.
     PARAML  UGBLM//'TRAILER'/2/S,N,GSIZE  $ GET SIZE OF G-SET
     PARAML  UGBLM//'TRAILER'/1/S,N,LDMODES  $ GET # OF CONTROL MODES
     MESSAGE //'NUMBER OF LOAD MODES READ = '/LDMODES $
     MESSAGE //'G-SET SIZE OF LOAD MODES = '/GSIZE $
     MESSAGE //'A-SET SIZE (FROM CURRENT RUN) = '/ASIZE $
     IF (GSIZE > ASIZE) THEN $ SEE IF ASIZE = GSIZE
          MESSAGE //'PARITIONING OLD EIGENVECTORS TO A-SIZE' $
          UPARTN     USET,UGBLM/UALM,,,/'G'/'A'/'O'/1 $ PARTITION TO A-SET SIZE
       ELSE
          MESSAGE //'NO A-SET REDUCTION: '/
                  'EQUIVALENCING OLD G-SIZE EIGENVECTORS TO A-SIZE' $
          EQUIVX     UGBLM/UALM/ALWAYS $ IF NO REDUCTION THEN EQUIV
     ENDIF $
$
     APPEND  PHIA1,UA/PHIA2 $ APPEND CONTROL MODES TO RB+FLEX MODES
     EQUIVX PHIA2/PHA/ALWAYS $
     SMPYAD  PHIA2,MMAA,PHIA2,,,/MCOUP/3////1////6 $ COMPUTES the GEN. MASS
     EQUIVX   MCOUP/MI/-1 $
     OUTPUT4  MCOUP,,,,//0/23/99//9
$
     SMPYAD  PHIA2,KXX,PHIA2,,,/EIGVMAT2/3////1////6 $ COMPUTES the Gen. Stiffne
     EQUIVX   EIGVMAT2/LAMMAT/-1 $
     TABPT EIGVMAT2/    $
     SMPYAD  UALM,KXX,PHIA2,,,/LDCOEF/3////1////6 $ COMPUTES the MODE DISPL LOAD
     TABPT LDCOEF/    $
     TRNSP LDCOEF/LDCOEFT $
     TABPT LDCOEFT/    $
```

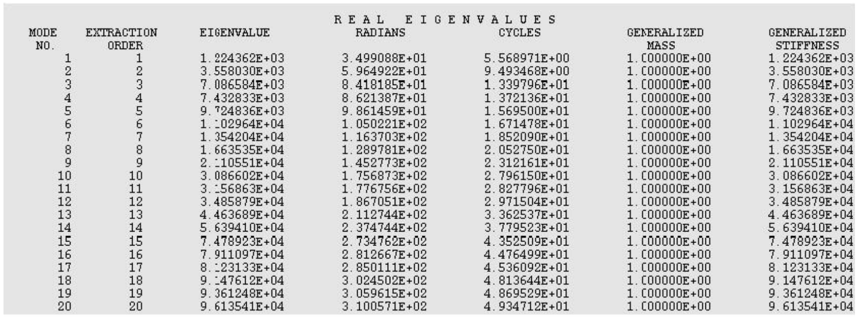
locations where strain-gauge sensors monitor loads, such as the location where the wing joins the fuselage and at the wing-fold location. To demonstrate the ease with which load monitoring can be setup analytically, a third load station location was also chosen in the area where the fin joins the fuselage. Figure 6 shows the three FE models which contribute to the load mode deflections and the areas of the FE models of figure 6 severed the full FE model of figure 5 roughly define the approximate lines of the various load-stations. Again to enable NASTRAN to perform the deflection calculation, it is necessary to use PATRAN to apply a one-inch forced translational boundary conditions to produce the deflections for shear forces and a one-radian forced rotational boundary conditions to produce the deflections for both bending and torsion moments. To illustrate the results, figure 7 shows three PATRAN plots of load mode shapes for calculating shear force, bending moment and torsion moment coefficients at the wing-fold location.

D. Generating NASTRAN ASE Data

As indicated in figure 2, NASTRAN solution 146, or Aeroelastic Response Analysis, is the main solution sequence used to generate all the data necessary to construct the first-order ASE EOM, which includes DMAP alters shown in table 3 for the structural part and table 4 for the aerodynamic part. In the discussion, there will be several variable names mentioned, some of these are used to store data on the .DBALL database such as LAMA, PHA, MMAA or KXX and other variables are in matrix data form that are more amenable in performing computations with DMAP commands, some of these are LMAT, OLDLMAT, MAT, LAMAI, LAMACNRL, NEWLAMA and LAMA2.

For the first part of table 3, in statements 3 through 6, the PHASE0 subDMAP of NASTRAN reads in the DMIG data for both the control-surface mode shapes and the load-mode shapes, into the variables PHIGR and PHIGRLM respectively and then immediately stores the variable data on the .dball file as data blocks. This stored data is used much later in the solution sequence after a number of other subDMAP calls have been completed.

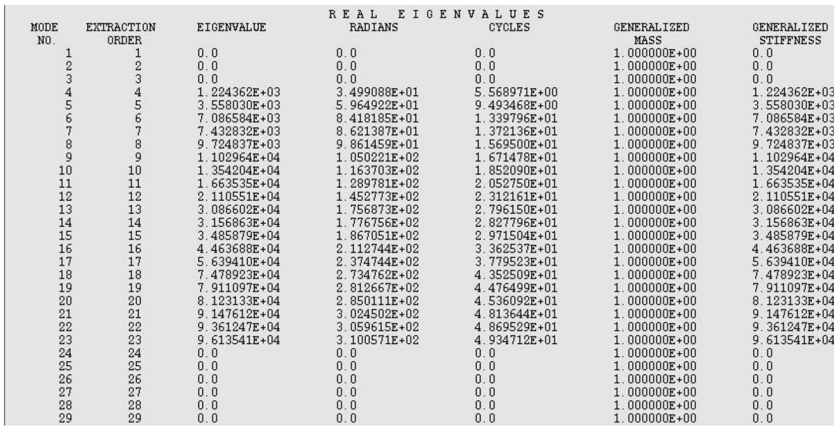
However, before the DMAP execution begins at statement 7 of table 3, the MODERS subDMAP, which is part of the standard solution sequence, first processes only the flexible-mode eigenvalue results, which are stored on a newly generated LAMA data block. This LAMA data block contains in a table form the following information as shown in Figure 8: the flexible-mode eigenvalues and their associated frequencies, as well as the generalized mass and stiffness results. The listing in figure 8 is a standard NASTRAN output, which was extracted from the .f06 file. The DMAP in table 3 uses this new LAMA data-block consisting of only the flexible modes to build a larger LAMA data block which includes the rigid-body and the control surface information needed for the ASE modeling.



```
                                    R E A L   E I G E N V A L U E S
 MODE    EXTRACTION      EIGENVALUE          RADIANS            CYCLES            GENERALIZED         GENERALIZED
  NO.       ORDER                                                                    MASS              STIFFNESS
     1          1        1.224362E+03      3.499088E+01       5.568971E+00       1.000000E+00       1.224362E+03
     2          2        3.558030E+03      5.964922E+01       9.493468E+00       1.000000E+00       3.558030E+03
     3          3        7.086584E+03      8.418185E+01       1.339796E+01       1.000000E+00       7.086584E+03
     4          4        7.432833E+03      8.621387E+01       1.372136E+01       1.000000E+00       7.432833E+03
     5          5        9.724836E+03      9.861459E+01       1.569500E+01       1.000000E+00       9.724836E+03
     6          6        1.102964E+04      1.050221E+02       1.671478E+01       1.000000E+00       1.102964E+04
     7          7        1.354204E+04      1.163703E+02       1.852090E+01       1.000000E+00       1.354204E+04
     8          8        1.663535E+04      1.289781E+02       2.052750E+01       1.000000E+00       1.663535E+04
     9          9        2.110551E+04      1.452773E+02       2.312161E+01       1.000000E+00       2.110551E+04
    10         10        3.086602E+04      1.756873E+02       2.796150E+01       1.000000E+00       3.086602E+04
    11         11        3.156863E+04      1.776756E+02       2.827796E+01       1.000000E+00       3.156863E+04
    12         12        3.485879E+04      1.867051E+02       2.971504E+01       1.000000E+00       3.485879E+04
    13         13        4.463689E+04      2.112744E+02       3.362537E+01       1.000000E+00       4.463689E+04
    14         14        5.639410E+04      2.374744E+02       3.779523E+01       1.000000E+00       5.639410E+04
    15         15        7.478923E+04      2.734762E+02       4.352509E+01       1.000000E+00       7.478923E+04
    16         16        7.911097E+04      2.812667E+02       4.476499E+01       1.000000E+00       7.911097E+04
    17         17        8.123133E+04      2.850111E+02       4.536092E+01       1.000000E+00       8.123133E+04
    18         18        9.147612E+04      3.024502E+02       4.813644E+01       1.000000E+00       9.147612E+04
    19         19        9.361248E+04      3.059615E+02       4.869529E+01       1.000000E+00       9.361248E+04
    20         20        9.613541E+04      3.100571E+02       4.934712E+01       1.000000E+00       9.613541E+04
```

Figure 8. The flexible-mode eigenvalue related information output by the OFP module.

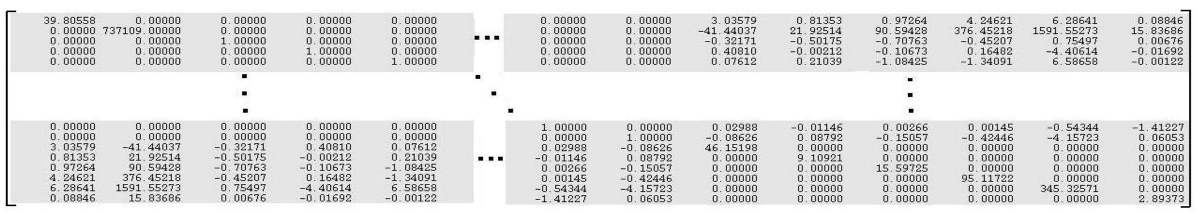
The tasks at the beginning of the MODERS part of the DMAP listing in table 3 perform the similar functions as the beginning of the DMAP in table 1. To orient the reader, the DMAP in table 3, to start with, performs two simultaneous procedures, one reassembling the eigenvalue related information that is the LAMA data and the other related to reassembling the eigenvector information that is the PHA data. We now begin our discussion with reassembling the LAMA information. The nearly half the DMAP statements in table 3 are devoted to generating the eigenvalue results shown in Figure 9, which was generated by the OFP module in statement 63. The INPUT4 of statements 22 in table 3 reads LMAT .op2 binary data file output by the NASTRAN solution 103 run as described section A of the paper and renames this old LMAT matrix data to OLDLMAT. The LMAX module in statement 21 transforms the newly generated LAMA data block of the flexible modes to a matrix data form, again called LMAT. In statements 23-31, the OLDLMAT matrix data and the new LMAT matrix data are partitioned and transposed to form the matrices OLDMAT and MAT, respectively. The matrix OLDMAT is appended to MAT in statement 46 to form the LAMAI matrix. Data blocks PHIGR and PHIGRLM are retrieved from the .dball database in statement 32. PARAML of statement 33 determines the number of control surface modes that the PHIGR data block contains and the DMAP in statements 35 through 45 builds the data for the last five columns for mode numbers 24 to 29 as shown in figure 9. Finally, in statement 47, the LAMAI matrix data, which contains the rigid-body and the flexible mode information, is appended to LAMACNRL matrix data containing control-surface mode information to form the LAMA1 matrix data. After inserting an additional row of zeros into LAMA1 matrix data with statements 42 and 43, LAMA2 matrix data is formed, which is output to an .opt4 file in a text format. With the DMAP of statement 55, the LAMA2 matrix data is converted back into a data block form, now temporarily called NEWLAMA. With statement 62, the original LAMA data block created by solution 146 for the flexible modes is replaced by NEWLAMA data by the EQUIVX command. Now the LAMA block data contains the desired results and the contents are seen in figure 9. One reason for developing this set of DMAP alters was to have the ability to output appropriately arranged frequency data for the ASE EOM modeling, but the second reason was to form an appropriately sized LAMA data block. Without the modified LAMA data block that matches the combined size of the flexible mode shapes generated by MODERS and imported mode-shapes, which include both rigid-body and the control-surface parts, solution 146 of NASTRAN cannot produce a .xdb file that PATRAN uses to plot mode-shape deflection results. This file is crucial in performing some of the subsequent tasks with PATRAN for the ASE modeling, such as plotting and checking the rigid-body, flexible and control-surface mode-shape deflections and verifying the accuracy of their splines used for the aerodynamic modeling.


```
                                            R E A L   E I G E N V A L U E S
   MODE    EXTRACTION      EIGENVALUE            RADIANS             CYCLES            GENERALIZED         GENERALIZED
    NO.       ORDER                                                                       MASS              STIFFNESS
        1         1        0.0                 0.0                 0.0                 1.000000E+00        0.0
        2         2        0.0                 0.0                 0.0                 1.000000E+00        0.0
        3         3        0.0                 0.0                 0.0                 1.000000E+00        0.0
        4         4        1.224362E+03        3.499088E+01        5.568971E+00        1.000000E+00        1.224362E+03
        5         5        3.558030E+03        5.964922E+01        9.493468E+00        1.000000E+00        3.558030E+03
        6         6        7.086584E+03        8.418185E+01        1.339796E+01        1.000000E+00        7.086584E+03
        7         7        7.432832E+03        8.621387E+01        1.372136E+01        1.000000E+00        7.432832E+03
        8         8        9.724837E+03        9.861459E+01        1.569500E+01        1.000000E+00        9.724837E+03
        9         9        1.102964E+04        1.050221E+02        1.671478E+01        1.000000E+00        1.102964E+04
       10        10        1.354204E+04        1.163703E+02        1.852090E+01        1.000000E+00        1.354204E+04
       11        11        1.663535E+04        1.289781E+02        2.052750E+01        1.000000E+00        1.663535E+04
       12        12        2.110551E+04        1.452773E+02        2.312161E+01        1.000000E+00        2.110551E+04
       13        13        3.086602E+04        1.756873E+02        2.796150E+01        1.000000E+00        3.086602E+04
       14        14        3.156863E+04        1.776756E+02        2.827796E+01        1.000000E+00        3.156863E+04
       15        15        3.485879E+04        1.867051E+02        2.971504E+01        1.000000E+00        3.485879E+04
       16        16        4.463688E+04        2.112744E+02        3.362537E+01        1.000000E+00        4.463688E+04
       17        17        5.639410E+04        2.374744E+02        3.779523E+01        1.000000E+00        5.639410E+04
       18        18        7.478923E+04        2.734762E+02        4.352509E+01        1.000000E+00        7.478923E+04
       19        19        7.911097E+04        2.812667E+02        4.476499E+01        1.000000E+00        7.911097E+04
       20        20        8.123133E+04        2.850111E+02        4.536092E+01        1.000000E+00        8.123133E+04
       21        21        9.147612E+04        3.024502E+02        4.813644E+01        1.000000E+00        9.147612E+04
       22        22        9.361247E+04        3.059615E+02        4.869529E+01        1.000000E+00        9.361247E+04
       23        23        9.613541E+04        3.100571E+02        4.934712E+01        1.000000E+00        9.613541E+04
       24        24        0.0                 0.0                 0.0                 1.000000E+00        0.0
       25        25        0.0                 0.0                 0.0                 1.000000E+00        0.0
       26        26        0.0                 0.0                 0.0                 1.000000E+00        0.0
       27        27        0.0                 0.0                 0.0                 1.000000E+00        0.0
       28        28        0.0                 0.0                 0.0                 1.000000E+00        0.0
       29        29        0.0                 0.0                 0.0                 1.000000E+00        0.0
```
Figure 9. The DMAP modified eigenvalue related information output by the OPF module.

Fortunately, the remaining part of the DMAP of table 3 used to assemble the mode-shape deflections is much more straight-forward than reassembling the eigenvalue related information. Nevertheless, it parallels the procedure for importing and assembling the modified LAMA data block. The commands in statements 20 and 32 import the rigid-body and the control-surface deflections as the PHRBNEW and PHIGR matrices, respectively. For the rigid-body and flexible mode vectors, the APPEND of statement 56 stacks them and calls the result, PHIA1. In statements 67 to 80, the DMAP processing determines the sizes of the imported control-surface vectors, and then converts them to A-set size, if necessary. The final A-set sized result is called the UA matrix. The previously formed PHIA1 matrix is appended to UA to form PHIA2. With the PHIA2 matrix containing the rigid-body, flexible and the control-surface mode shapes and with the MMAA matrix, which is the FEM mass matrix, the SMPYAD processor of statement 98 performs the matrix operation expressed in equation (1)

$$[\mathrm{MCOUP}] = [\mathrm{PHIA2}]^{T}\,[\mathrm{MMAA}][\mathrm{PHIA2}] \tag{1}$$

This operation computes the generalized mass matrix. The MCOUP matrix is now suitable for ASE EOM modeling and is placed on the .op4 file with the OUTPUT4 command of statement 100.



$$[\mathrm{MCOUP}] = \begin{bmatrix}
39.80558 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & & 0.00000 & 0.00000 & 3.03579 & 0.81353 & 0.97264 & 4.24621 & 6.28641 & 0.08846 \\
0.00000 & 737109.00000 & 0.00000 & 0.00000 & 0.00000 & & 0.00000 & 0.00000 & -41.44037 & 21.92514 & 90.59428 & 376.45218 & 1591.55273 & 15.83686 \\
0.00000 & 0.00000 & 1.00000 & 0.00000 & 0.00000 & \cdots & 0.00000 & 0.00000 & -0.32171 & -0.50175 & -0.70763 & -0.45207 & 0.75497 & 0.00676 \\
0.00000 & 0.00000 & 0.00000 & 1.00000 & 0.00000 & & 0.00000 & 0.00000 & 0.40810 & -0.00212 & -0.10673 & 0.16482 & -4.40614 & -0.01692 \\
0.00000 & 0.00000 & 0.00000 & 0.00000 & 1.00000 & & 0.00000 & 0.00000 & 0.07612 & 0.21039 & -1.08425 & -1.34091 & 6.58658 & -0.00122 \\
 & & \vdots & & & \ddots & & & & & \vdots & & & \\
0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & & 1.00000 & 0.00000 & 0.02988 & -0.01146 & 0.00266 & 0.00145 & -0.54344 & -1.41227 \\
0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & & 0.00000 & 1.00000 & -0.08626 & 0.08792 & -0.15057 & -0.42446 & -4.15723 & 0.06053 \\
3.03579 & -41.44037 & -0.32171 & 0.40810 & 0.07612 & & 0.02988 & -0.08626 & 46.15198 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 \\
0.81353 & 21.92514 & -0.50175 & -0.00212 & 0.21039 & \cdots & -0.01146 & 0.08792 & 0.00000 & 9.10921 & 0.00000 & 0.00000 & 0.00000 & 0.00000 \\
0.97264 & 90.59428 & -0.70763 & -0.10673 & -1.08425 & & 0.00266 & -0.15057 & 0.00000 & 0.00000 & 15.59725 & 0.00000 & 0.00000 & 0.00000 \\
4.24621 & 376.45218 & -0.45207 & 0.16482 & -1.34091 & & 0.00145 & -0.42446 & 0.00000 & 0.00000 & 0.00000 & 95.11722 & 0.00000 & 0.00000 \\
6.28641 & 1591.55273 & 0.75497 & -4.40614 & 6.58658 & & -0.54344 & -4.15723 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 345.32571 & 0.00000 \\
0.08846 & 15.83686 & 0.00676 & -0.01692 & -0.00122 & & -1.41227 & 0.06053 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 2.89373
\end{bmatrix}$$

Figure 10. The generalized mass matrix.

Figure 10 shows the four corners of the generalized mass matrix, [MCOUP]. The upper left corner of this matrix contains rigid-body mass and pitch-inertia terms and the first three unity-mass terms of the flexible modes. The last six columns of the upper right corner contain the control surface mass-coupling terms. The last six columns of the bottom right corner contain six diagonal elements representing the rotational inertia terms of the control surfaces. The bottom left corner mass-coupling terms are the transposes of the terms of upper right corner. Finally, the EQUIVX command of statement 97 replaces the PHA data block, which contains only the flexible mode shapes originally generated by MODERS subDMAP, with the new PHIA2 mode shapes. The modified PHA will be used later in performing the spline and aerodynamic calculations.

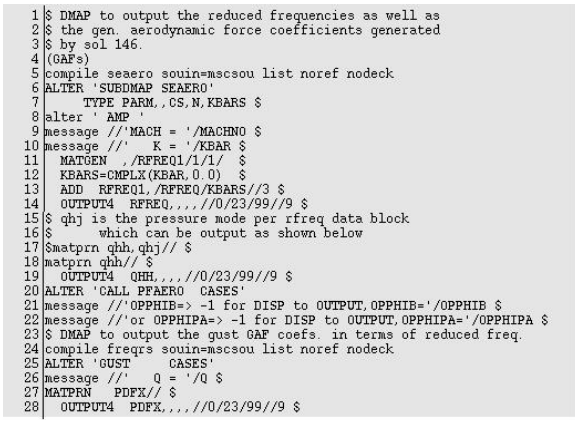
Similar to statement 98, the statement 105 of table 3 performs a triple matrix multiplication to generate the mode-displacement coefficients, as expressed in equation (2)

$$[\mathrm{LDCOEF}] = [\mathrm{UALM}]^{T}\,[\mathrm{KXX}][\mathrm{PHIA\,2}] \tag{2}$$

where UALM contains the load mode vectors (also refer to as direction vectors), KXX contains the FE model stiffness matrix, and PHIA2 contains the rigid-body, flexible and the control-surface mode shapes. Despite the simplicity of this triple matrix multiplication task, it is instructive to understand each step taking place in this operation. In the first step, the stiffness matrix multiplies each flexible-mode-shape vector of PHIA2 to generate the applied forces at each of the gridpoints. These applied forces can be interpreted as the force vectors acting on gridpoints required to deform the structure into its mode-shape deflection. In the second step, the direction vector, which is composed of the load-mode deflections in the six degrees-of-freedom of the gridpoint using a dot-product operation, multiplies the force vector acting on a gridpoint. The third step performed by the SMPYAD processor sums the dot products at each gridpoint over of the entire structure to compute the load coefficient for a particular flexible mode. The mode-displacement load coefficients are functions of the generalized coordinate displacements, which are available in the ASE EOM formulation, making it a convenient way to formulate structural loads equations various locations on the aircraft.

The first part of table 4 contains a DMAP to output the reduced frequency and its corresponding set of motion GAF coefficients, internally called QHH in NASTRAN and the second part the gust GAF coefficients, internally called PDFX. The DMAP statements 11 through 19 write the motion GAF coefficients to the .op4 file. An important and somewhat of a nuisance outcome of this part of the DMAP process is that motion GAF coefficients are *not* output in the ascending sequence of reduced frequency as specified on the MKAERO entry. NASTRAN's internal processing logic dictates this seemingly random sequence.

Table 4. DMAP to output the reduced frequencies and both the motion and gust GAF coefficients.



```
 1 $ DMAP to output the reduced frequencies as well as
 2 $ the gen. aerodynamic force coefficients generated
 3 $ by sol 146.
 4 (GAFs)
 5 compile seaero souin=mscsou list noref nodeck
 6 ALTER 'SUBDMAP SEAERO'
 7      TYPE PARM,,CS,N,KBARS $
 8 alter ' AMP '
 9 message //'MACH = '/MACHNO $
10 message //'   K = '/KBAR $
11   MATGEN  ,/RFREQ1/1/1/  $
12   KBARS=CMPLX(KBAR,0.0)  $
13   ADD  RFREQ1,/RFREQ/KBARS//3 $
14   OUTPUT4  RFREQ,,,,//0/23/99//9 $
15 $ qhj is the pressure mode per rfreq data block
16 $       which can be output as shown below
17 $matprn qhh,qhj// $
18 matprn qhh// $
19   OUTPUT4  QHH,,,,//0/23/99//9 $
20 ALTER 'CALL PFAERO  CASES'
21 message //'OPPHIB=> -1 for DISP to OUTPUT,OPPHIB='/OPPHIB $
22 message //'or OPPHIPA=> -1 for DISP to OUTPUT,OPPHIPA='/OPPHIPA $
23 $ DMAP to output the gust GAF coefs. in terms of reduced freq.
24 compile freqrs souin=mscsou list noref nodeck
25 ALTER 'GUST     CASES'
26 message //'   Q = '/Q $
27 MATPRN   PDFX// $
28   OUTPUT4  PDFX,,,,//0/23/99//9 $
```

For the gust GAF coefficients, on the other hand, the output is in an ascending order of the reduced frequencies to the .op4 file, as specified on the FREQ entry (see Ref. 1). The DMAP following statement 24 of table 4 performs the gust GAF output processing in the FREQRS subDMAP module. In this module, the FREQ entry is ordinarily used to specify Hertz-unit frequencies needed in performing the frequency-domain aeroelastic response analyses. However, for our purposes, if we specify on the GUST entry (see Ref. 1) the gust penetration velocity, V_{gust} equal to $\pi * \overline{c}$, then, on the FREQ entry, the desired reduced-frequency units can be used instead of the Hertz units. For this re-definition procedure to be correctly utilized, the same reference length, $\overline{c}$, must be used here as used on the AERO entry (again see Ref. 1). To understand how this re-definition of gust penetration velocity works, it is necessary to present the equation NASTRAN uses to internally convert the frequency in Hertz, f, to reduced-frequency, k, for retrieving unsteady-aerodynamics data to perform gust computations,

$$k = f * (\pi * \overline{c}) / v_{gust} \tag{3}$$

If now the gust penetration velocity, v_{gust} is set to $\pi * \overline{c}$, then equation (3) simplifies to,

$$k = f \tag{4}$$

With this gust penetration velocity substitution, the frequencies specified in the FREQ entry are now used as reduced frequencies by NASTRAN.

E. READ_OP4 Fortran Program

The READ_OP4 program's major function is to read in three files: (1) a text-formatted .op4 file; (2) a punch file, which contains the selected six-degree-of-freedom gridpoint deflection data for the acceleration sensors; and (3) if load coefficients in solution 146 are computed, another text-formatted .op4 file. The program has numerous other functions and is outlined in the flow diagram of figure 11.

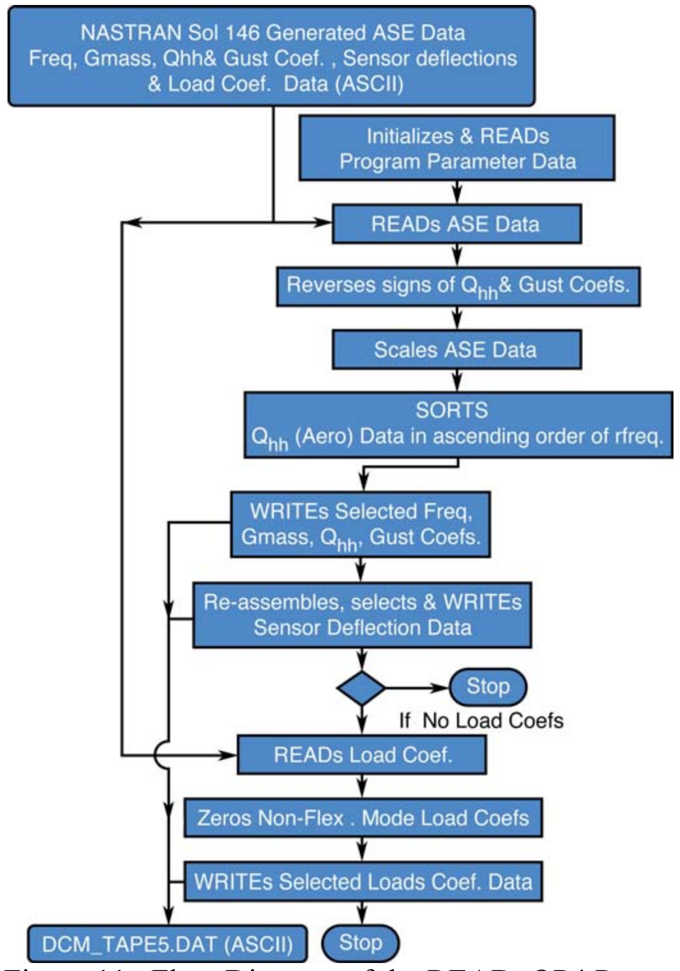

NASTRAN Sol 146 Generated ASE Data
Freq, Gmass, Qhh& Gust Coef. , Sensor deflections & Load Coef. Data (ASCII)
Initializes & READs Program Parameter Data
READs ASE Data
Reverses signs of Q_{hh}& Gust Coefs.
Scales ASE Data
SORTS Q_{hh} (Aero) Data in ascending order of rfreq.
WRITEs Selected Freq, Gmass, Q_{hh}, Gust Coefs.
Re-assembles, selects & WRITEs Sensor Deflection Data
Stop
If No Load Coefs
READs Load Coef.
Zeros Non-Flex . Mode Load Coefs
WRITEs Selected Loads Coef. Data
DCM_TAPE5.DAT (ASCII)
Stop

Figure 11. Flow Diagram of the READ_OP4 Program.

NASTRAN computes the motion and gust GAF coefficients as required for its own internal aeroelastic response computations. However, for the ASE EOM presented here, these GAF coefficients are used on the left-hand side of EOM. To accommodate this formulation, READ_OP4 program reverses the signs of all the NASTRAN generated GAFs.

Reference 6 describes a previously developed code that was incorporated into the READ_OP4 FORTRAN program in order to scale the data generated by NASTRAN. The code has the capability to scale the ASE EOM for a full-sized aircraft to the ASE EOM for a wind-tunnel model by applying aeroelastic scaling principles on both the structural and the unsteady aerodynamic data, which was the original intent of the code. Notwithstanding, this code is general enough to be used to scale aircraft of any size. The process involves using the frequency ratio, size ratio and mass ratio to scale masses, dynamic pressures, unsteady aerodynamics, structural frequencies, control surfaces and sensors.

As mentioned previously the motion GAF coefficients are not output by NASTRAN in ascending order of reduced frequencies. The *isort* subroutine in table 5 re-orders the frequency-dependent motion GAF data with an indexed array to an ascending order of reduced frequencies. The ISAC/DYNARES

Table 5. Subroutine isort.

```
subroutine isort(idr,id,rfreq,nk,nnk)
real*8 rfreq(nnk)
integer*4 id(nnk),idr(nnk),ichose,i,j,k,idx
do i=1,nk
id(i)=i
end do
do i=nk,1,-1
ichose=id(i)
do j=1,i
if (rfreq(id(j)).le.rfreq(ichose))then
ichose=id(j)
idx=j
end if
end do
k=id(i)
id(i)=ichose
id(idx)=k
end do
do i=1,nk
idr(nk+1-i)=id(i)
end do
return
end
```

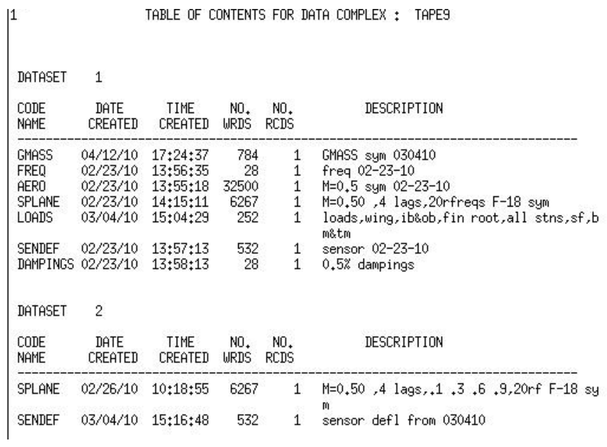
program that generates the s-plane coefficients by using a least-square fitting procedure requires the frequency-dependent motion GAF coefficients to be in ascending order, and is the reason that the re-ordering task is performed here prior to the fitting procedure.

The WRITE statements in the READ_OP4 program can be used to select specific frequencies and modes to output onto a DCM_TAPE5.DAT formatted text file for use in building the first-order ASE EOM. With this capability, the modal data related to specific vibration analysis frequencies can be included or ignored for any ASE analysis.

The READ_OP4 program can also select specific sensor locations or load monitoring points of the aircraft for performing a particular ASE analysis. Of course, this capability is tied to the selection process performed first by NASTRAN to write the gridpoint deflection data to a punch file. On the punch file data, the deflections of all six gridpoint directions are ordinarily provided for all the gridpoints selected and for all the vibration modes of the NASTRAN analysis. The READ_OP4 program, however, is capable of re-arranging the punch file format data to a new file format where all the modal deflections are output for each of gridpoint direction that the user specifies. This selection, however, is made with input data to the READ_OP4 program. The program also performs the same selection process for the load coefficients. The program also lists the load or gridpoint numbers and directions of the data stored on the new file. With this newly formatted file, the sensor deflection and the load coefficient data are in a form needed as input to the ISAC DCM program and ultimately for the first-order ASE EOM.

F. ISAC Programs to Store ASE Data and to Perform s-plane Fits

The DCM_TAPE5.DAT file contains the following sets of ASE data as written by the READ_OP4 program: reduced frequencies, generalized aerodynamic force coefficients, generalized masses, vibration frequencies, sensor deflections and load coefficients. With a text editor, the structural damping associated with the modal system is also appended to the DCM_TAPE5.DAT file. Interactively, the long columns of text data sets residing on the DCM_TAPE5.DAT file are stored by the ISAC/DCM program on TAPE9.RAF, a binary database file (see table of contents of TAPE9.RAF in Figure 12). The ISAC/DYNARES program then uses the 'AERO' data stored on TAPE9.RAF to perform an s-plane fit using the following equation to produce fit coefficients, A_0 thru A_6, for each reduced frequency-dependent motion or gust GAF:



```
1                 TABLE OF CONTENTS FOR DATA COMPLEX :  TAPE9



 DATASET    1

 CODE      DATE      TIME     NO.  NO.          DESCRIPTION
 NAME     CREATED   CREATED  WRDS  RCDS
 ---------------------------------------------------------------------------------
 GMASS    04/12/10  17:24:37   784    1    GMASS sym 030410
 FREQ     02/23/10  13:56:35    28    1    freq 02-23-10
 AERO     02/23/10  13:55:18 32500    1    M=0.5 sym 02-23-10
 SPLANE   02/23/10  14:15:11  6267    1    M=0.50 ,4 lags,20rfreqs F-18 sym
 LOADS    03/04/10  15:04:29   252    1    loads,wing,ib&ob,fin root,all stns,sf,b
                                           m&tm
 SENDEF   02/23/10  13:57:13   532    1    sensor 02-23-10
 DAMPINGS 02/23/10  13:58:13    28    1    0.5% dampings


 DATASET    2

 CODE      DATE      TIME     NO.  NO.          DESCRIPTION
 NAME     CREATED   CREATED  WRDS  RCDS
 ---------------------------------------------------------------------------------
 SPLANE   02/26/10  10:18:55  6267    1    M=0.50 ,4 lags,.1 .3 .6 .9,20rf F-18 sy
                                           m
 SENDEF   03/04/10  15:16:48   532    1    sensor defl from 030410
```

Figure 12. Table of contents of TAPE9.RAF file.

$$\mathrm{GAF}_{jk}(s) = A_{0,j,k} + A_{1,j,k}\,\bar{s} + A_{2,j,k}\,\bar{s}^2 + \sum_{i=1}^{i=4} \bar{s}A_{i+2,j,k}/(\bar{s}+\beta_i) \qquad (5)$$

$$\bar{s} = \bar{c}/(2 * v)s \qquad (6)$$

where s is the Laplace variable and β_i are the lags. Although there are four lag terms indicated in equation (5), i.e. $\bar{s}A_3/(\bar{s}+\beta_1)$ thru $\bar{s}A_6/(\bar{s}+\beta_4)$, a lesser number of lag terms can be used, especially if the unsteady aerodynamics modeling is warranted to be sufficiently accurate by doing so. The lags, β_1 thru β_4, are user specified. Common practice is to use values in the lower to mid range of the reduced frequencies specified to generate the frequency-dependent GAF coefficients.

DYNARES stores the s-plane coefficients on the TAPE9.RAF as the 'SPLANE' data (see again, figure 12). Further details for the operation of both programs are available in reference 2 and a description of the fitting procedure is provided in reference 7.

Finally ISAC/DCM is needed again to interactively generate ASCII files of all data listed in figure 12 that come from the binary file, TAPE9.RAF. The data is output on DCM_TAPE6.DAT, which is a long contiguous ASCII file with text headers describing the data types. Since the numerical data is already in a tabular form, with a text editor, the DCM_TAPE6.DAT file can easily be partitioned into "flat" files for each data type. Figure 10 is an example of a flat file format, which are text files containing a rectangular table of numbers. These smaller flat files can then be seamlessly imported by MATLAB using the LOAD command.

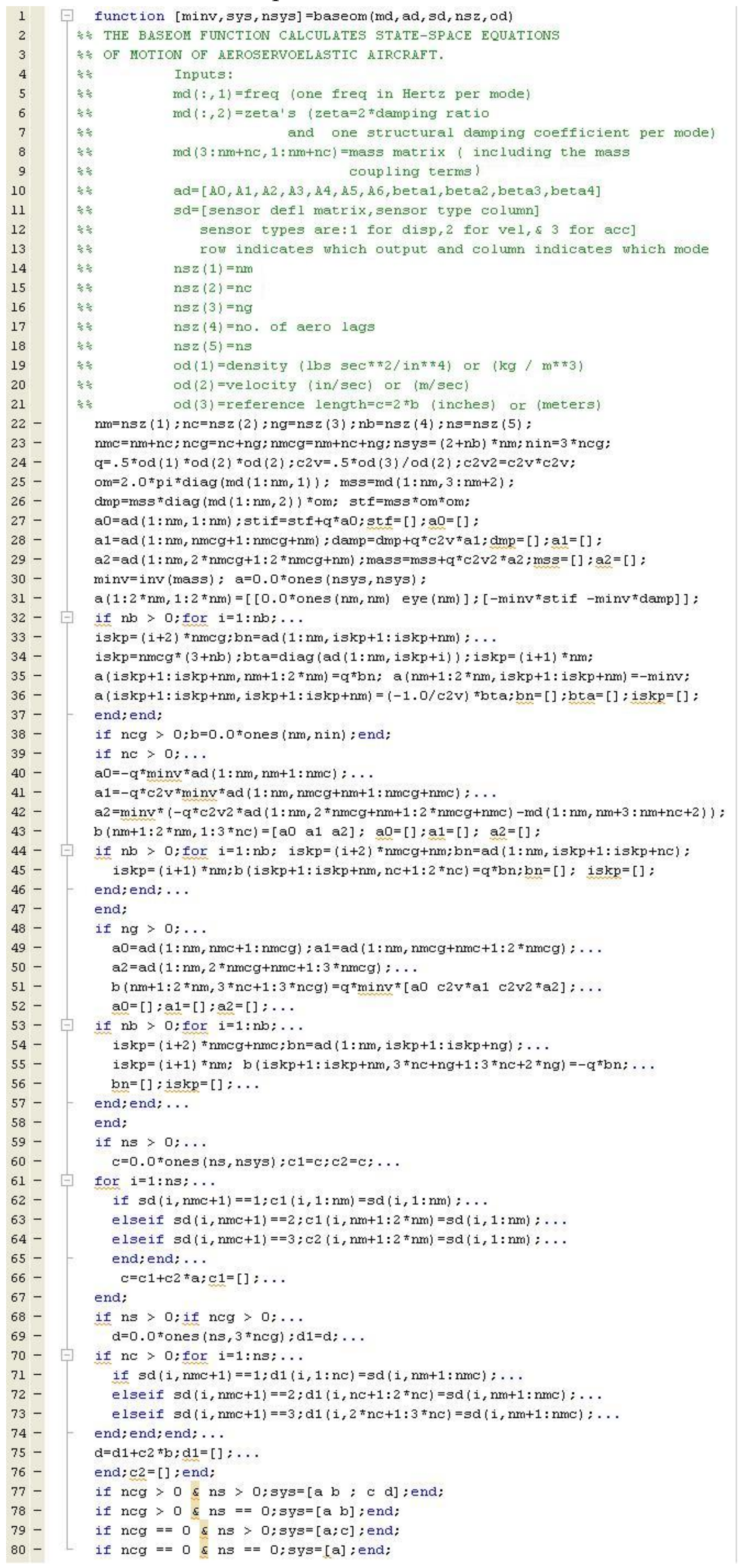
G. Generating the ASE EOM with MATLAB

The form of the aeroservoelastic equations of motion used here is available from reference 8. This matrix equation is given as follows:

$$\begin{Bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \dot{\boldsymbol{x}}_3 \\ \vdots \\ \dot{\boldsymbol{x}}_6 \end{Bmatrix} = \begin{bmatrix} 0 & [\boldsymbol{I}] & 0 & \cdots & 0 \\ -[\widetilde{\boldsymbol{M}}]^{-1}([\widetilde{\boldsymbol{K}}] & [\widetilde{\boldsymbol{D}}] & \bar{\boldsymbol{q}}[\boldsymbol{I}] & \cdots & \bar{\boldsymbol{q}}[\boldsymbol{I}]) \\ 0 & [\boldsymbol{A}_3] & -\beta_1(2\mathrm{v}/\bar{\boldsymbol{c}})[\boldsymbol{I}] & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & \vdots \\ 0 & [\boldsymbol{A}_6] & 0 & 0 & -\beta_4(2\mathrm{v}/\bar{\boldsymbol{c}})[\boldsymbol{I}] \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \vdots \\ \boldsymbol{x}_6 \end{Bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ -[\widetilde{\boldsymbol{M}}]^{-1}([\widetilde{\boldsymbol{K}}_c] & [\widetilde{\boldsymbol{D}}_c] & [\widetilde{\boldsymbol{M}}_c]) \\ 0 & [\boldsymbol{A}_{c1}] & 0 \\ \vdots & \vdots & \vdots \\ 0 & [\boldsymbol{A}_{c4}] & 0 \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_c \\ \dot{\boldsymbol{u}}_c \\ \ddot{\boldsymbol{u}}_c \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ -[\widetilde{\boldsymbol{M}}]^{-1}([\widetilde{\boldsymbol{K}}_g] & [\widetilde{\boldsymbol{D}}_g]) \\ 0 & [\boldsymbol{A}_{g1}] \\ \vdots & \vdots \\ 0 & [\boldsymbol{A}_{g4}] \end{bmatrix} \begin{Bmatrix} \boldsymbol{w}_g \\ \dot{\boldsymbol{w}}_g \end{Bmatrix} \qquad (7)$$

where $\widetilde{\boldsymbol{M}} = \boldsymbol{M} + \bar{\boldsymbol{q}}(\bar{\boldsymbol{c}}/(2\mathrm{v}))^2 \boldsymbol{A}_2$,

Table 6. MATLAB script, baseom.m.



```
  function [minv,sys,nsys]=baseom(md,ad,sd,nsz,od)
%% THE BASEOM FUNCTION CALCULATES STATE-SPACE EQUATIONS
%% OF MOTION OF AEROSERVOELASTIC AIRCRAFT.
%%          Inputs:
%%          md(:,1)=freq (one freq in Hertz per mode)
%%          md(:,2)=zeta's (zeta=2*damping ratio
%%                       and  one structural damping coefficient per mode)
%%          md(3:nm+nc,1:nm+nc)=mass matrix ( including the mass
%%                              coupling terms)
%%          ad=[A0,A1,A2,A3,A4,A5,A6,beta1,beta2,beta3,beta4]
%%          sd=[sensor defl matrix,sensor type column]
%%             sensor types are:1 for disp,2 for vel,& 3 for acc]
%%             row indicates which output and column indicates which mode
%%          nsz(1)=nm
%%          nsz(2)=nc
%%          nsz(3)=ng
%%          nsz(4)=no. of aero lags
%%          nsz(5)=ns
%%          od(1)=density (lbs sec**2/in**4) or (kg / m**3)
%%          od(2)=velocity (in/sec) or (m/sec)
%%          od(3)=reference length=c=2*b (inches) or (meters)
  nm=nsz(1);nc=nsz(2);ng=nsz(3);nb=nsz(4);ns=nsz(5);
  nmc=nm+nc;ncg=nc+ng;nmcg=nm+nc+ng;nsys=(2+nb)*nm;nin=3*ncg;
  q=.5*od(1)*od(2)*od(2);c2v=.5*od(3)/od(2);c2v2=c2v*c2v;
  om=2.0*pi*diag(md(1:nm,1)); mss=md(1:nm,3:nm+2);
  dmp=mss*diag(md(1:nm,2))*om; stf=mss*om*om;
  a0=ad(1:nm,1:nm);stif=stf+q*a0;stf=[];a0=[];
  a1=ad(1:nm,nmcg+1:nmcg+nm);damp=dmp+q*c2v*a1;dmp=[];a1=[];
  a2=ad(1:nm,2*nmcg+1:2*nmcg+nm);mass=mss+q*c2v2*a2;mss=[];a2=[];
  minv=inv(mass); a=0.0*ones(nsys,nsys);
  a(1:2*nm,1:2*nm)=[[0.0*ones(nm,nm) eye(nm)];[-minv*stif -minv*damp]];
  if nb > 0;for i=1:nb;...
  iskp=(i+2)*nmcg;bn=ad(1:nm,iskp+1:iskp+nm);...
  iskp=nmcg*(3+nb);bta=diag(ad(1:nm,iskp+i));iskp=(i+1)*nm;
  a(iskp+1:iskp+nm,nm+1:2*nm)=q*bn; a(nm+1:2*nm,iskp+1:iskp+nm)=-minv;
  a(iskp+1:iskp+nm,iskp+1:iskp+nm)=(-1.0/c2v)*bta;bn=[];bta=[];iskp=[];
  end;end;
  if ncg > 0;b=0.0*ones(nm,nin);end;
  if nc > 0;...
  a0=-q*minv*ad(1:nm,nm+1:nmc);...
  a1=-q*c2v*minv*ad(1:nm,nmcg+nm+1:nmcg+nmc);...
  a2=minv*(-q*c2v2*ad(1:nm,2*nmcg+nm+1:2*nmcg+nmc)-md(1:nm,nm+3:nm+nc+2));
  b(nm+1:2*nm,1:3*nc)=[a0 a1 a2]; a0=[];a1=[]; a2=[];
  if nb > 0;for i=1:nb; iskp=(i+2)*nmcg+nm;bn=ad(1:nm,iskp+1:iskp+nc);
    iskp=(i+1)*nm;b(iskp+1:iskp+nm,nc+1:2*nc)=q*bn;bn=[]; iskp=[];
  end;end;...
  end;
  if ng > 0;...
    a0=ad(1:nm,nmc+1:nmcg);a1=ad(1:nm,nmcg+nmc+1:2*nmcg);...
    a2=ad(1:nm,2*nmcg+nmc+1:3*nmcg);...
    b(nm+1:2*nm,3*nc+1:3*ncg)=q*minv*[a0 c2v*a1 c2v2*a2];...
    a0=[];a1=[];a2=[];...
  if nb > 0;for i=1:nb;...
    iskp=(i+2)*nmcg+nmc;bn=ad(1:nm,iskp+1:iskp+ng);...
    iskp=(i+1)*nm; b(iskp+1:iskp+nm,3*nc+ng+1:3*nc+2*ng)=-q*bn;...
    bn=[];iskp=[];...
  end;end;...
  end;
  if ns > 0;...
    c=0.0*ones(ns,nsys);c1=c;c2=c;...
  for i=1:ns;...
    if sd(i,nmc+1)==1;c1(i,1:nm)=sd(i,1:nm);...
    elseif sd(i,nmc+1)==2;c1(i,nm+1:2*nm)=sd(i,1:nm);...
    elseif sd(i,nmc+1)==3;c2(i,nm+1:2*nm)=sd(i,1:nm);...
    end;end;...
     c=c1+c2*a;c1=[];...
  end;
  if ns > 0;if ncg > 0;...
    d=0.0*ones(ns,3*ncg);d1=d;...
  if nc > 0;for i=1:ns;...
    if sd(i,nmc+1)==1;d1(i,1:nc)=sd(i,nm+1:nmc);...
    elseif sd(i,nmc+1)==2;d1(i,nc+1:2*nc)=sd(i,nm+1:nmc);...
    elseif sd(i,nmc+1)==3;d1(i,2*nc+1:3*nc)=sd(i,nm+1:nmc);...
  end;end;end;...
  d=d1+c2*b;d1=[];...
  end;c2=[];end;
  if ncg > 0 & ns > 0;sys=[a b ; c d];end;
  if ncg > 0 & ns == 0;sys=[a b];end;
  if ncg == 0 & ns > 0;sys=[a;c];end;
  if ncg == 0 & ns == 0;sys=[a];end;
```

$$\widetilde{\boldsymbol{K}} = \boldsymbol{K} + \bar{\boldsymbol{q}}\boldsymbol{A}_0\,,$$

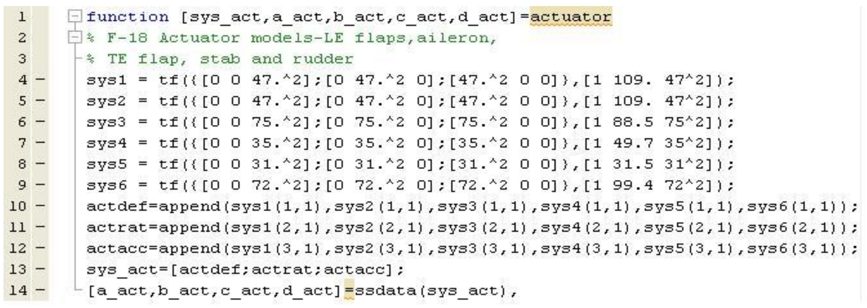
$$\widetilde{\boldsymbol{D}} = \boldsymbol{D} + \bar{\boldsymbol{q}}\left(\bar{\boldsymbol{c}}/(2\mathrm{v})\right)\boldsymbol{A}_1$$

and $\bar{\boldsymbol{q}} = \tfrac{1}{2}\rho v^2$. The $[\boldsymbol{A}_0]$ thru $[\boldsymbol{A}_6]$ matrices contain the coefficients obtained by fitting equation (5), and their row/column dimensions are the same as those of the motion and the gust GAF coefficient matrices. Generally four normalized aerodynamic lags are used and are depicted as, β_1 thru β_4, in equation (7). The state variables, $\{\boldsymbol{x}_1\}$ and $\{\boldsymbol{x}_2\}$ are the generalized displacements and velocities for all the modes, respectively. The variables $\{\boldsymbol{x}_3\}$ thru $\{\boldsymbol{x}_6\}$ are the states associated with aerodynamic lags. The $\{\boldsymbol{u}\}$, $\{\dot{\boldsymbol{u}}\}$ and $\{\ddot{\boldsymbol{u}}\}$ are the displacements, rates and accelerations of the control surfaces produced by the actuator models. The $\boldsymbol{w_g}$ and $\dot{\boldsymbol{w}}_{\boldsymbol{g}}$ inputs representing the vertical turbulence velocity and acceleration, respectively.

The mass coupling terms together with accelerations caused by the control-surface motion produce in ASE equation (7) inertial force effects on the airframe or, stated more graphically, produce a "dog wagging tail" effect. This force is present even when the aerodynamic forces are zero. Including this effect is an important part of ASE modeling, especially when considering that controlling this force can be used to reduce buffet loads (Ref. 9).

Table 6 gives the MATLAB script used to generate both state-space equations and the output equations. With the large array of input parameters available to the user, the baseom.m script is very adaptable in controlling the number of state-space and output equations or whether controls and turbulence inputs are included. Furthermore, these parameters permit adjustments of analysis conditions. To ease the task of generating the input data for the baseom.m script, a number of preprocessor scripts were written which perform the relatively simple function of re-arranging or re-formatting the different data types and which are not presented here.

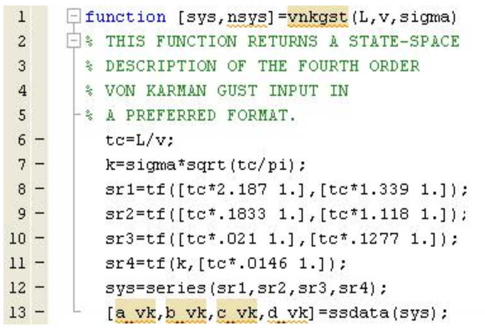
Table 7. MATLAB script, actuator.m.



```
1   function [sys_act,a_act,b_act,c_act,d_act]=actuator
2   % F-18 Actuator models-LE flaps,aileron,
3   % TE flap, stab and rudder
4   sys1 = tf({[0 0 47.^2];[0 47.^2 0];[47.^2 0 0]},[1 109. 47^2]);
5   sys2 = tf({[0 0 47.^2];[0 47.^2 0];[47.^2 0 0]},[1 109. 47^2]);
6   sys3 = tf({[0 0 75.^2];[0 75.^2 0];[75.^2 0 0]},[1 88.5 75^2]);
7   sys4 = tf({[0 0 35.^2];[0 35.^2 0];[35.^2 0 0]},[1 49.7 35^2]);
8   sys5 = tf({[0 0 31.^2];[0 31.^2 0];[31.^2 0 0]},[1 31.5 31^2]);
9   sys6 = tf({[0 0 72.^2];[0 72.^2 0];[72.^2 0 0]},[1 99.4 72^2]);
10  actdef=append(sys1(1,1),sys2(1,1),sys3(1,1),sys4(1,1),sys5(1,1),sys6(1,1));
11  actrat=append(sys1(2,1),sys2(2,1),sys3(2,1),sys4(2,1),sys5(2,1),sys6(2,1));
12  actacc=append(sys1(3,1),sys2(3,1),sys3(3,1),sys4(3,1),sys5(3,1),sys6(3,1));
13  sys_act=[actdef;actrat;actacc];
14  [a_act,b_act,c_act,d_act]=ssdata(sys_act),
```

The $\{u\}, \{\dot{u}\}$ and $\{\ddot{u}\}$ inputs of equation (7) require in general the displacement, the rate and the acceleration outputs from a first order representation of each actuator. The actuator.m script presented in table 7 will generate the appropriate state-space EOM and output equations. As shown in statements 4 thru 9, the actuators are represented as second-order transfer functions. Using MATLAB's Linear Time Invariant (LTI) commands in statements 10 thru 13, these transfer functions are combined and their outputs are rearranged. Finally, the LTI command, ssdata, generates the first-order state and output equation representation. To numerically illustrate the LTI capability performed, the following leading-edge inboard-actuator transfer function is given as the starting point equation:

Table 8. MATLAB script, vnkgst.m.



```
function [sys,nsys]=vnkgst(L,v,sigma)
% THIS FUNCTION RETURNS A STATE-SPACE
% DESCRIPTION OF THE FOURTH ORDER
% VON KARMAN GUST INPUT IN
% A PREFERRED FORMAT.
  tc=L/v;
  k=sigma*sqrt(tc/pi);
  sr1=tf([tc*2.187 1.],[tc*1.339 1.]);
  sr2=tf([tc*.1833 1.],[tc*1.118 1.]);
  sr3=tf([tc*.021 1.],[tc*.1277 1.]);
  sr4=tf(k,[tc*.0146 1.]);
  sys=series(sr1,sr2,sr3,sr4);
  [a_vk,b_vk,c_vk,d_vk]=ssdata(sys);
```

$$\frac{\boldsymbol{u_c}}{\boldsymbol{u_{com}}} = \frac{2209}{\boldsymbol{s}^2 + 109\boldsymbol{s} + 2209} \tag{8}$$

This transfer function is then transformed with the LTI capability to the following first-order state and output equation form:

$$\begin{aligned}
\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} &= \begin{bmatrix} -109 & -2209 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} 2209 \\ 0 \end{bmatrix} u_{com} \\
\begin{Bmatrix} u_c \\ \dot{u}_c \\ \ddot{u}_c \end{Bmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -109 & -2209 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 2209 \end{bmatrix} u_{com}
\end{aligned} \tag{9}$$

Table 8 gives the MATLAB script for the von Karman gust filter used to characterize atmospheric turbulence (reference 10). The filter is derived through an approximation procedure and is based on the von Karman spectrum that is generally used in frequency-domain aeroelastic response analysis as in NASTRAN solution 146. The script generates both w_g and $\dot{w}_g$ characterizing atmospheric turbulence. The MATLAB script vnkgst.m performs with LTI the same three tasks of representing, combining the gust-filter transfer-function equation and generating the state and output equations for w_g and $\dot{w}_g$ as the actuator.m script.

IV. Results

Using the example F-18 aircraft, the ASE EOM developed with the methodology described in the paper will be employed to perform an open-loop flutter analysis and a turbulence analysis. The flight analysis conditions chosen are 0.5 Mach number at several altitudes.

The first result is a flutter root-locus with altitude variation at matched point conditions from 30,000 feet to sea level. The data were obtained from the standard-atmospheric tables of reference 11, which was tabulated into MATLAB script form. Structural damping of 2% was chosen. Figure 13 shows the flutter root-locus results of an open-loop, free-free symmetrical configuration, which includes the plunge and pitch rigid-body modes. The roots generally follow a stabilizing direction with increasing dynamic pressure, except for the 13th structural mode.

The second set of results is from a turbulence analysis at 15,000 feet. The power-spectral-density function of the wing root and tip vertical accelerations are shown in Figure 14. The gust intensity was set at a nominal value of 1 inch/sec in the application of the von Karman spectrum and the gust is applied in

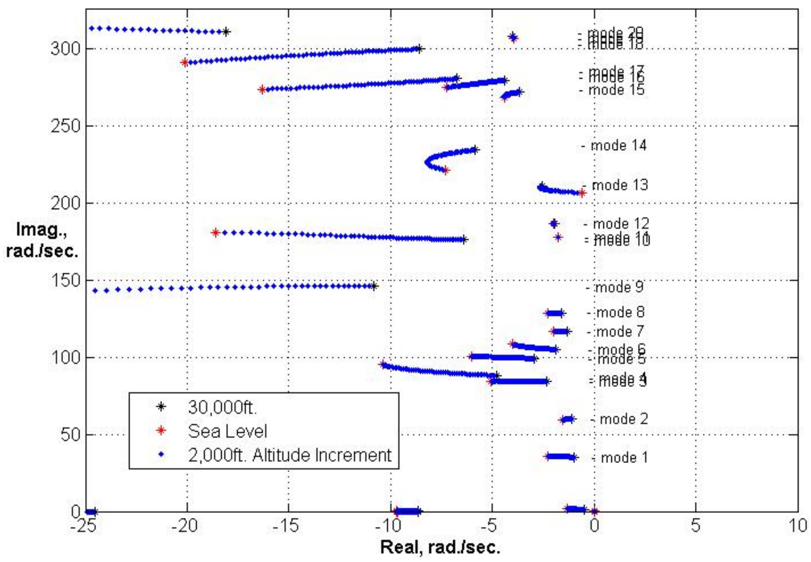


Figure 13. Open-loop root-locus flutter analysis performed from 30,000 feet to sea level at 0.5 Mach on an F-18 FEM using the symmetrical modes.

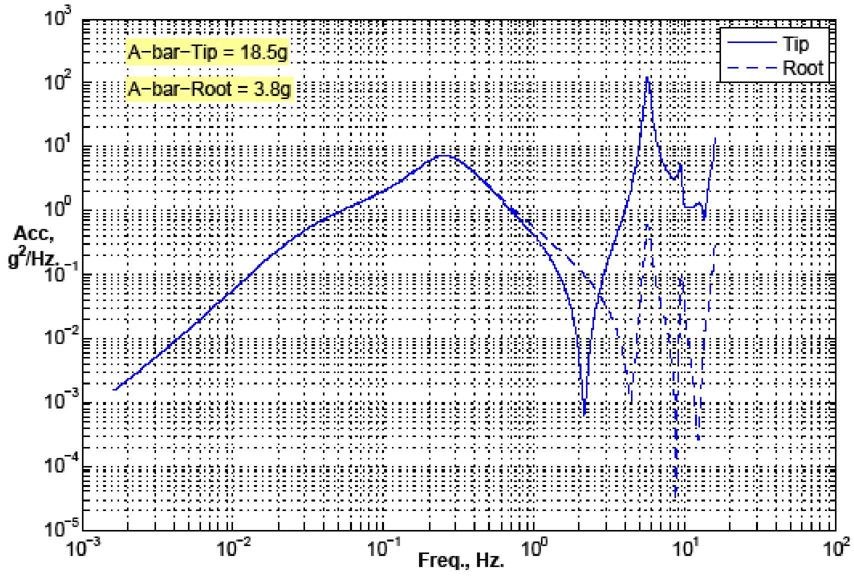


Figure 14. PSD of wing-tip and root acceleration at 15,000 ft. altitude and 0.50 Mach.

the vertical direction for all the results. The respective responses in the wing first bending mode (approx. 5.5 Hz.) are very different in the two plots. The response in this mode at the wing tip is two and a half orders of magnitude higher than the corresponding response at the wing root. The respective response magnitudes in the short-period mode (approx. 0.25 Hz) are nearly the same. Thus the large difference in the RMS values between the two responses is due almost entirely to the differences in the wing-first-bending and the higher modes.

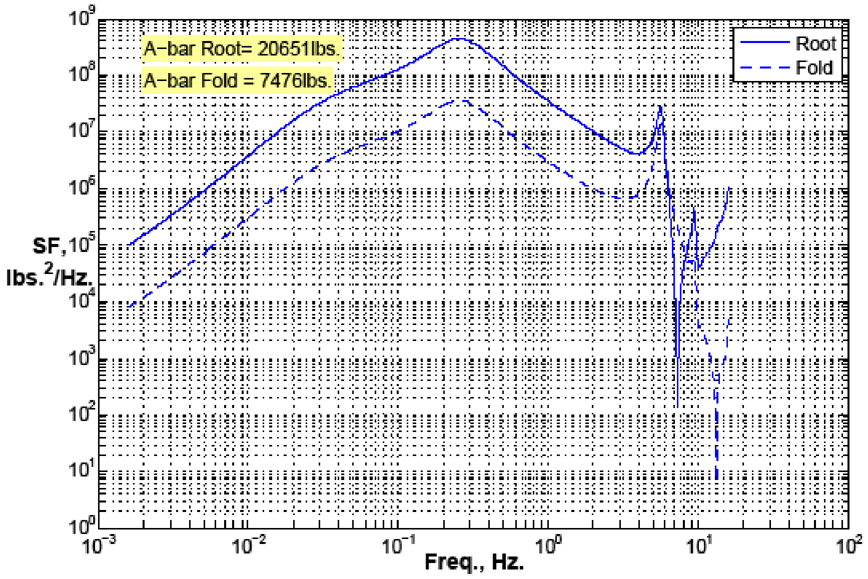


Figure 15. PSDs and A-bars of wing-root and the wing-fold shear force.

At this point, it is worthwhile to expound on some of the merits of using the approach described in the paper. The RMS values presented here were not computed by integrating the PSDs, which is the method used by NASTRAN, but computed by the Lyapunov equation method, which is used for all the plot results. Using the Lyapunov method to perform the RMS computation is not only much simpler to code and execute, much faster, and more exact but, furthermore, it can only be produced with a state-space formulation of the EOM as developed here. The integration method developed for NASTRAN is made up of hundreds of lines of code and for large problems, is very slow; for very low damped modes, is not exact; solves stable systems; and also erroneously solves unstable (fluttering) systems. On the other hand, the Lyapunov method determines and assesses the stability disposition of the system and only works for stable systems. It is, in fact, one of the more desirable methods available to control system designers for determining the stability of controllers, since, at the same time, it provides controller performance information similar to a cost

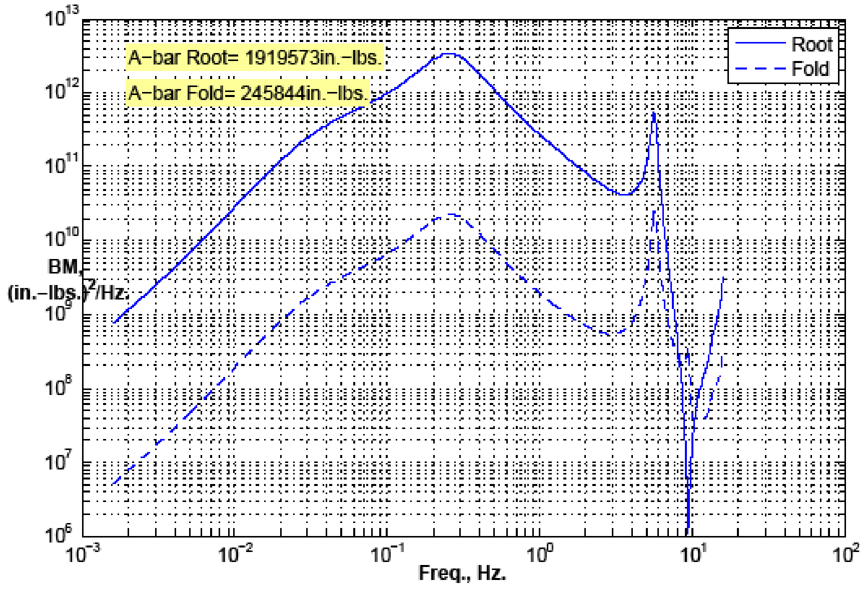


Figure 16. PSDs and A-bars of wing-root and the wing-fold bending moment.

function.

The third set of results is loads at the wing-root and the wing-fold locations. Both the root and fold results are plotted for the shear-force, bending- and torsion-moment loads on figures 15, 16, and 17, respectively. For the wing-root loads, the major contributions to the RMSs come from the short period mode similar to the wing-root acceleration RMSs, which also comes from the short-period mode contributions. However, at the wing fold location, the shear-force and bending-moment ratios of the square-root of short-period values to their respective RMSs is much less than one, indicating a greater contribution from the structural modes. Strengthening this argument are the PSD plots at the wing-fold location that show the peaks of the short-period and the first bending modes are at more comparable levels. The torsion moment loads stand out as an exception. At both the wing-root and wing-fold locations, the square-root of the peak and RMS ratios for both cases is nearly one. In figure 17 of the torsion moment result, the short-period PSD peaks substantially overwhelm all the other peaks making the short-period mode the dominant player for the torsion moment load in terms of RMS.

A final note on the load calculation methodology developed and presented here in the paper. Although, the mode-displacement coefficients were computed using NASTRAN's aeroelastic response analyses and DMAP capabilities; it is interesting to note that the author found no simple or easy way to use these load coefficients to perform aeroelastic gust load response analysis in NASTRAN. The aeroelastic response analysis does not appear to be set up or able to perform load response analysis. These coefficients were, however, incorporated into the state-space formulation developed here with almost no effort to obtain the load results presented here.

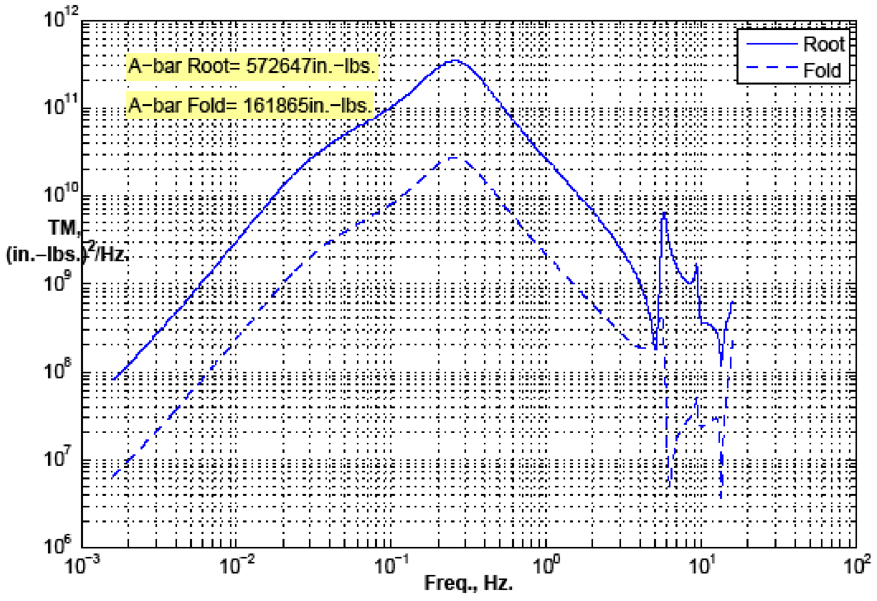


Figure 17. PSDs and A-bars of wing-root and the wing-fold torsion moment.

V. Conclusions

A methodology for generating enhanced first-order aeroservoelastic plant equations of motion was presented. The paper gives first, a flow diagram of the aeroservoelastic system, then a flow chart of the methodology. Intermediate results of each step of using the methodology are provided as a guide to implementing the procedures. Numerous DMAP alters were also given in tables throughout the paper which were incorporated in NASTRAN to perform the many non-standard tasks. A flow chart of the

READ_OP4 code also is provided, which reads formatted text and punch files, re-formats and re-sequences NASTRAN generated output data into a form usable in ISAC and ultimately in MATLAB. The paper outlined the data storing and s-plane fitting capabilities of the ISAC programs. The form of the first-order ASE EOM and the MATLAB script used to generate them were presented. An altitude-varying root-locus result, PSD of wing-tip and root accelerations and PSDs of loads at wing root and fold locations along with their respective RMSs were presented to demonstrate the enhanced ASE capability.

References

1. Rodden, William R. and Johnson, Erwin H.: MSC/NASTRAN Aeroelastic Analysis, USER'S GUIDE, V68, The MacNeal-Schwendler Corporation, 1994.
2. Peel, Ellwood L. and Adams, William M.: "A Digital Program For Calculating the Interaction Between Flexible Structures, Unsteady Aerodynamics and Active Controls," NASA TM-80040, Jan. 1979.
3. No authors listed: ZAERO Version 6.0; User's Manual, Zona Technologies, Scottsdale, AZ, May 2002.
4. Reymond, Michael: MD NASTRAN 2006 DMAP Programmer's Guide, The MacNeal-Schwendler Corporation, 2006.
5. Using MATLAB Version 6, The MathWorks, Inc., Natick, MA, 2000.
6. Pototzky, Anthony S.:"Scaling Laws Applied to a Modal Formulation of the Aeroservoelastic Equations," 43rd AIAA Structure, Structural Dynamics, and Materials Conference, Denver, Colorado, April 2002.
7. Adams, Jr., William M.; Tiffany, Sherwood H.; "Newsom, Jerry R. and Peele, Ellwood L.: STABCAR- A Program for Finding Characteristic Roots of Systems Having Transcendental Stability Matrices," NASA TP-2165, June 1984.
8. Mukhopadhyay, Vivek; Newsom, Jerry R.; and Abel, Irving: "A Method for Obtaining Reduced-Order Control Laws for High-Order Systems Using Optimization Techniques," NASA Tech. Paper 1876, August 1981.
9. Moses, Robert W., Pototzky, Anthony S. et al: "Controlling buffeting loads by rudder and piezo-actuation," International Forum for Aeroelasticity and Dynamics, Munich, Germany, 2005.
10. Pototzky, A. S., Zeiler, T. A and Perry III, B.: "Calculating Time-Correlated Gust Loads Using Matched-Filter and Random Process Theories," Journal of Aircraft, Vol. 28, No. 5, 1991, p 346-352.
11. U. S. Standard Atmosphere, Washington, DC, 1962.