

A DMAP Alter to Allow Amplitude-dependent Modal Damping in a Transient Solution

by

Ted Rose

Manager, MSC/NASTRAN Technical Support and Training

and

Martin McNamee

Senior Support Engineer

The MacNeal-Schwendler Corporation

Presented at the 1996 MacNeal-Schwendler World Users' Conference

Abstract:

Most structural systems display some amount of nonlinear behavior. In most analyses, we assume that the nonlinear affects are not important and use a linear approximation. Often the nonlinear affects cannot be ignored and must be included in the solution of the problem. This paper addresses a special case where the modal damping may not be a constant fraction of the critical damping ratio, rather the damping ratio is a function of the magnitude of the response.

A DMAP alter is presented which allows the user to include modal damping which is a function of the displacement (or velocity) measured at selected points.

Introduction:

MSC/NASTRAN contains many tools for dynamic analysis and design. This paper is intended to demonstrate application of one of these tools to a specific problem, that of a structure with nonlinear modal damping.

Often, when a modal survey is performed on a structure, the measured modal damping is a function of the amplitude of the vibrations. If this is the case, the engineer is forced to make a decision as to which value to use. Often, a best guess value is chosen, which (hopefully) represents the value the structure will see when in the actual design environment. The DMAP alter presented in this paper is intended to allow the engineer to enter the modal damping as a function of the amplitude of the response, and (hopefully) obtain more accurate results in the solution.

1. Basic concept

In a modal transient response, the equations of motion are transformed into modal coordinates and, in the case of modal damping, a set of uncoupled dynamic equations¹ results:

$$m_i \ddot{\xi}_i(t) + b_i \dot{\xi}_i(t) + k_i \xi_i(t) = f_i(t) \quad \text{Equation 1}$$

Where m_i is the modal mass, b_i is the modal damping, k_i is the modal stiffness, $f_i(t)$ is the modal force, and $\xi_i(t)$ is the modal displacement as a function of time. In this equation, the modal mass, damping, and stiffness are assumed to be constant with time. Let us look at the modal damping.

Modal damping is often expressed using a fraction of the critical damping for the mode, or;

$$b_i = 2 * m_i * \omega_{ni} * \zeta_i \quad \text{Equation 2}$$

where ζ_i is the fraction of critical damping for mode i .

For purposes of the DMAP alter, ζ_i is divided into a linear (ζ_l) and a nonlinear (ζ_n) term. At this point, the dynamic equation for the mode (dropping the “i” subscript) becomes;

$$m \ddot{\xi}(t) + 2 * m * \omega_n * (\zeta_l + \zeta_n) \dot{\xi}(t) + k \xi(t) = f(t) \quad \text{Equation 3}$$

or (to prepare it for the alter);

$$m \ddot{\xi}(t) + 2 * m * \omega_n * \zeta_l \dot{\xi}(t) + k \xi(t) = f(t) - 2 * m * \omega_n * \zeta_n \dot{\xi}(t) \quad \text{Equation 4}$$

The left side of the equation is the standard dynamic equation of the mode using the linear modal damping. The right side contains the original dynamic load, plus a nonlinear dynamic load, which represents the nonlinear modal damping force.

At this point, if NOLINi entries could apply forces on modal coordinates, the problem would be solved. Since NOLINi entries may not apply loads directly to modal dof, a problem arises. The following is the method used in the alter to avoid this limitation.

2. Implementation

Equation 4 looks ready for use in a transient solution using NOLIN’s. The only problems are that there is no method to directly attach a NOLIN to a modal coordinate and that physical displacements are not known until after the modal equations are integrated. The first limit can be solved by attaching the modal degrees of freedom (dof) to conventional (SPOINT or GRID related) dof in the model. The approach used by the alter is to place the entire model into a superelement (you all knew this was coming) and perform

component modal synthesis on it to obtain the modes. These modes can be attached to either SPOINTs or GRID points which are passed to the residual structure.

At this point, the model would appear to be ready to solve, except that the “measurement” dof must be in the residual structure in order to calculate a nonlinear force as a function of it’s displacement. At first, this would seem easy, but if the GRID point is placed in the residual, when the superelement modes are calculated, the dof associated with the measurement point would be held constrained by default. An avoidance to this would be to place the measurement point in the SECSET for the superelement, however, this would result in modifying the modal coordinates when the superelement matrices were processed (see section 9.4.8 of reference 2). This means the dof selected to represent the system modes would not be correct at the final solution.

This problem is addressed by the DMAP alter in the following way. MPCs are used to attach any measurement dof to SPOINTs (in this way, only the measurement dof and the modes are external to the superelement). The SPOINTs representing the measurement dof are placed in the SECSET for the superelement and therefore are not constrained during the eigenvalue solution. The alter disables the INREL module (which would remove any part of the modes which the static transformation vectors could represent), and uses the modes as calculated by the program, but replaces the static transformation matrix by a null matrix. In this way, the modes are passed down with no modification. However, the relation needed to calculate the displacement of the measurement dof is not available, since they are not included in the reduction. The alter handles this by storing the eigenvectors for the measurement dof in the database in MATPOOL (input table for the MTRXIN module - which reads DMIG entries) for use later.

Since the modes are being passed down with no modification, it is possible to apply the linear (ζ_i) modal damping at the superelement. This alter is a modified form of “sedampa.v682”³, which allows modal damping to be applied to superelement modes.

Since there is no static transformation and the modes are not modified, the reduced matrices for the superelement are the correct matrices to be used in a modal solution. There is still the problem of calculating the displacement of the measurement dof during the transient solution..

At the residual structure, the alter requires that the measurement dof be defined as belonging to the “U1” set. The alter will then perform the following operations to “replace” a modal coordinate by the physical dof without changing the dynamic properties of the solution.

First, the user has to select which modal coordinates will be replaced by the measurement dof. This is done by the use of MPC equations. The modes which will be replaced by measurement dof are defined as dependent on MPC equation (to place them in the “M” set). The form of the MPC equations does not matter - the alter does not actually apply the MPC’s from the residual structure input, rather it simply uses them to determine the modes to be replaced.

The alter uses a modification on the method from Section 14.1 of the Theoretical Manual⁴ to attach the independent modes to the measurement dof. In this method, MPC equations are written using the eigenvectors to attach a mode to physical dof. This is done as follows:

For a single measurement dof, whose displacement is U_i , it’s displacement may be represented as a linear combination of the eigenvectors for that dof:

$$U_i = \Phi_{1i}\xi_1 + \Phi_{2i}\xi_2 + \Phi_{3i}\xi_3 + \dots + \Phi_{ni}\xi_n \quad \text{Equation 5}$$

where Φ_{ji} = eigenvector j dof I.

Equation 5 may be re-written as:

$$U_i = \{\Phi\}[\xi] \quad \text{Equation 6}$$

This may be rewritten in the form of an MPC equation with the physical (measurement) dof as the dependent dof as follows:

$$0 = -U_i + \{\Phi\}[\xi] \quad \text{Equation 7}$$

Using the MPC equation to define the mode which will replace the measurement dof as dependent, the mode shapes may be written as:

$$\{\Phi\} = [\Phi_M | \Phi_N] \quad \text{Equation 8}$$

where Φ_M is the dependent mode and Φ_N are the independent modes. The DMAP alter re-orders the eigenvector equations as follows and uses them as MPC equations to generate the “GM” matrix (the MPC transformation matrix).

$$0 = [\Phi_M | \Phi_N | -I] \begin{pmatrix} \xi_m \\ \xi_n \\ U_i \end{pmatrix} \quad \text{Equation 9}$$

where ξ_M is the displacement of the M^{th} (dependent) modal coordinate and ξ_N are the displacements of the N^{th} (independent) modal coordinates.

Once the alter has applied the above MPC relationship, the resulting equations will contain only the modal coordinates which were not specified as dependent on an MPC and also the measurement dof. At this point, the resulting equations have the same properties as the original equations, but contain all the necessary terms to apply equation 4 using NOLINS.

3. User Interface

- A) Place the entire model into a superelement.
- B) For each "measurement dof", create an SPOINT and write an MPC equation tying the SOPOINT to it. The SPOINT MUST be independent on the MPC - for example - to measure Grid 3, dof 1, use SPOINT 10003 as follows:


```
SPOINT,10003
MPC,100,3,1,1,.,10003,0,-1.
```
- C) Use additional SPOINTS for the SEQSET.
- D) Apply any constraints to the model (in the superelement).
- E) Define any loads to be applied using LOADSET-LSEQ in the superelement
- F) Perform CMS on the superelement.(The superelement modes should be the system modes)
- G) Apply the linear modal damping to the modes using the SDAMP case control command and the TABDMP1 for the superelement.

AT THE RESIDUAL STRUCTURE LEVEL -

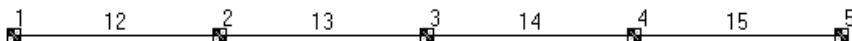
- H) Apply the dynamic load - DLOAD
- I) Define the integration time step - TSTEP (be sure to use one sufficiently small)
- J) FOR EACH "measured dof"
- 1) define the measured dof as belonging to the "U1" set by using USET or USET1
 - 2) write an MPC equation with one modal dof as the dependent dof (use a mode which will not have nonlinear damping)
- K) FOR EACH MODE WHICH MAY HAVE NONLINEAR DAMPING:
- 1) Create a NOLIN1 which applies a force on an EPOINT based on the following relationship:
$$F_{\text{epoint}} = -2 * m * \omega_n * T_{xx} (U_{\text{measured}})$$

where
m = modal mass (normally 1.0), ω_n = resonant frequency (radians/time)
 T_{xx} = look-up table for modal damping ratio based on displacement of U_{measured}
 U_{measured} = measured displacement
 - 2) define a transfer function which sets the displacement at the EPOINT equal to the force applied at the EPOINT (see sample problem below)
 - 3) define a NOLIN2 which multiplies the displacement of the EPOINT by the modal velocity and applies it at the modal dof

4. Sample Problems

The alter is written to work in SOL 109 (direct transient). This may seem like a conflict, since the approach is using modal coordinates, but even if SOL 112 were used, the final solution would use the coupled equation solver, since the replacement of a modal coordinate with a physical (measurement) dof results in coupled equations.

The model used is a simple cantilever beam as shown in the following illustration:



GRID point 1 is constrained, and all other points are constrained so that x (axial) translation is the only dof allowed.

EXAMPLE 1: Apply “nonlinear” modal damping on the first mode. The linear damping will be 1% of critical for all modes, and the “nonlinear” damping will be an additional 1% of critical damping on the first mode. A reference run is made using 2% of critical damping on the first mode and 1% of critical on the other modes and the results are compared.

In order to prepare for the second sample, we will set the problem up to apply nonlinear damping to mode 1 based on the motion of Grid Point 3, dof 1

Superelement input - using superlement 1 :

```

SESET,1,1,thru,5          $ put the entire model in the superelement
SPOINT,1001,thru,1010    $ allows ten modes - you pick the number
SEQSET1,1,0,1001,thru,1010 $ use SPOINTS to represent modes
SPOINT,10003              $ point to measure displacement
CSUPEXT,1,10003          $ SPOINT 10003 is external to superelement 1
MPC,100,3,1,1.,10003,0,-1. $ attach measurement dof 10003 to grid 3, dof 1
tabdmp1,11,crit,,,,,,+dmp11
+dmp11,0.,.01,100.,.01,endt $ modal damping table - use 1% of critical for all modes
$ don't forget the EIGR or EIGRL

```

Residual input:

```

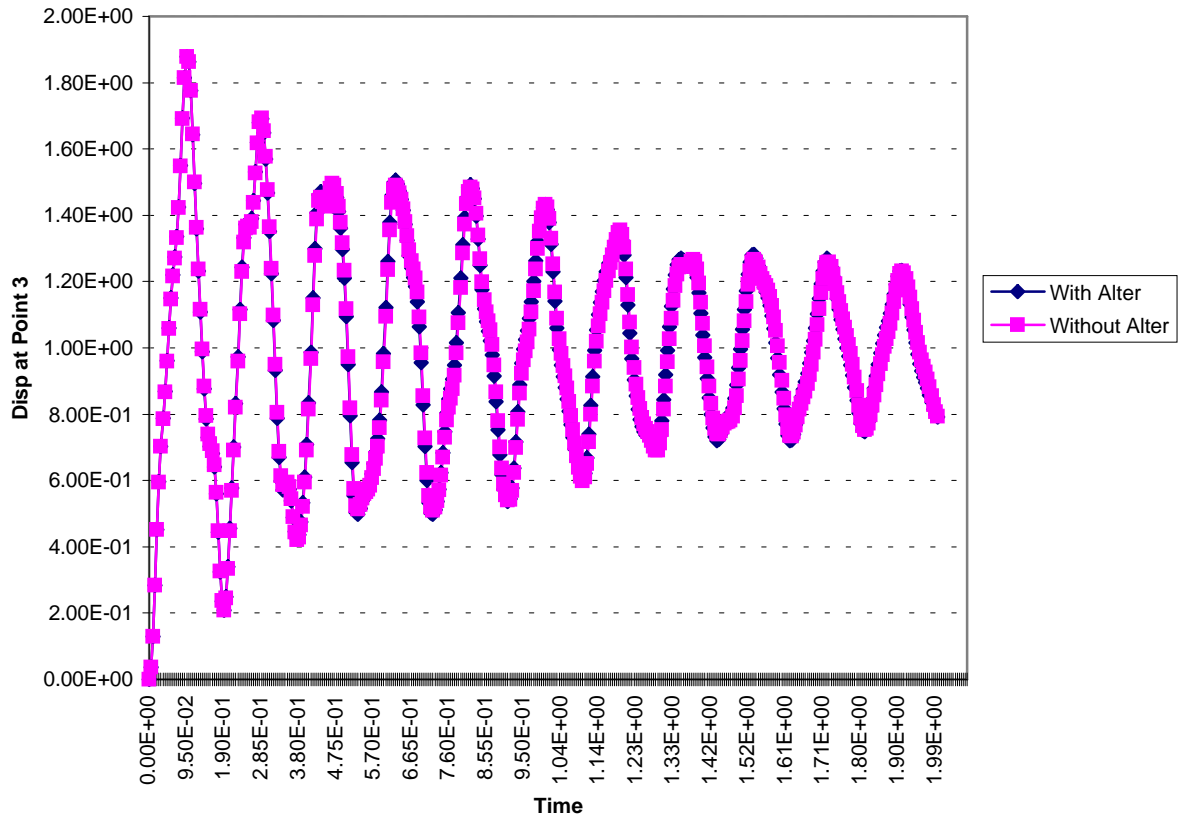
EPOINT,100001            $ point to measure  $2*m*\omega_n*\zeta$ 
NOLIN1,101,100001,0,-69.44,10003,10,401 $ calculates the  $2*m*W_n*zeta$ 
$ (where zeta is the damping ratio
$ provided by the table -
$ in this case table 401)
$  $-2.*m*\omega_n = 2.*1.0*34.72 = -69.44$ 
TABLED1,401,,,,,,+AB11 $ table of critical damping ration as a function of
+AB11,-.1,0.0,0.0,0.0,.1,0.0,endt $ disp of GRID point3, dof 1 for mode 1001
TF,100,100001,0,1.      $ diagonal term in stiffness - sets disp = applied load
NOLIN2,101,1001,0,1.,1001,10,100001,1 $ disp(100001) x vel (1001)
mpc,104,1004,0,1.,10003,0,1. $ identify mode 4 to be replaced by
$ measurement dof - remember this
$ MPC is not actually applied
uset1,u1,0,10003        $ identify SPOINT 10003 as the
$ measurement dof
$ don't forget the dynamic loading

```

RESULTS:

The following graph shows the results for the displacement of GRID 3, comparing the results using the alter to the results without the alter for the same problem.

Run 1 - Try 2% of Critical on Mode 1, 1% on the Other Modes

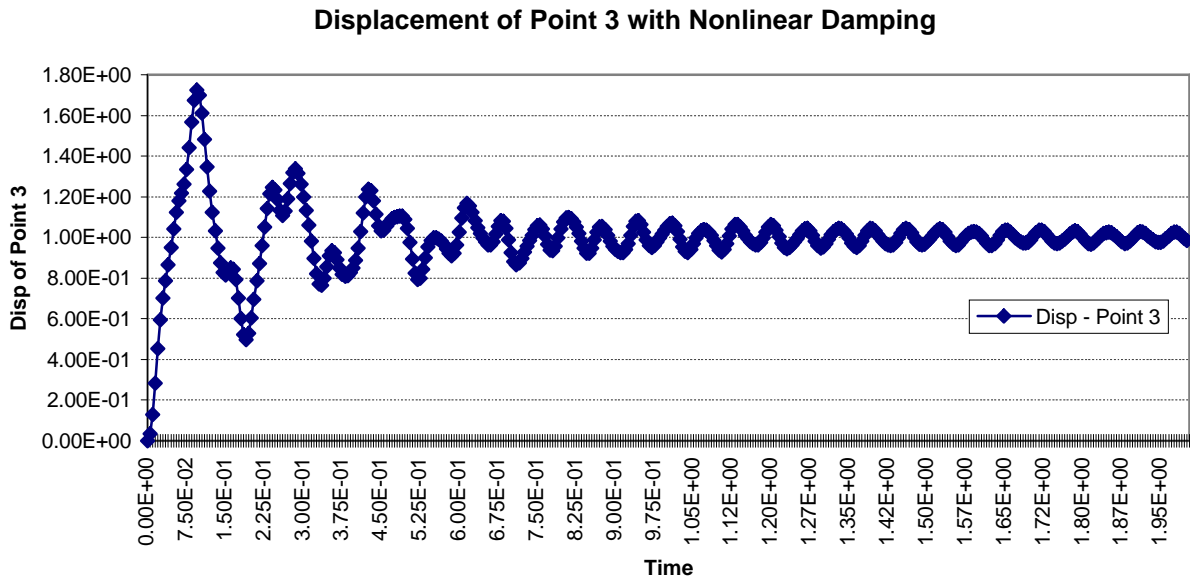


As the chart indicates, the results agree between the two runs.

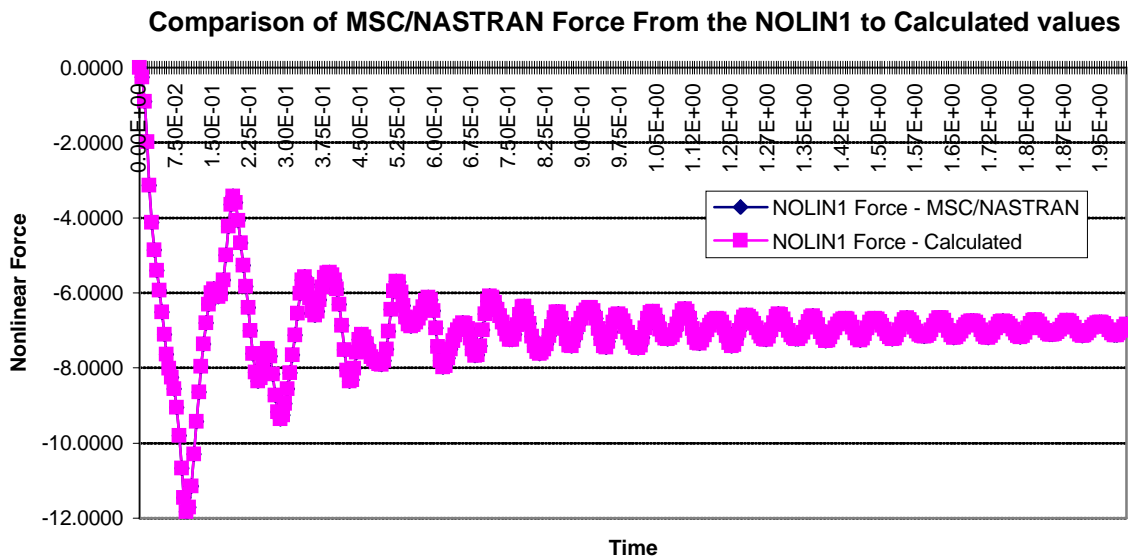
EXAMPLE 2 - Now apply nonlinear damping to mode 1. Use a function which is linear with the displacement of point 3. The following table was substituted:

```
TABLED1,401,,,,,,,,,+AB11
+AB11, -.1, 0.01, 0.00, 0.00, .1, 0.01, endt
```

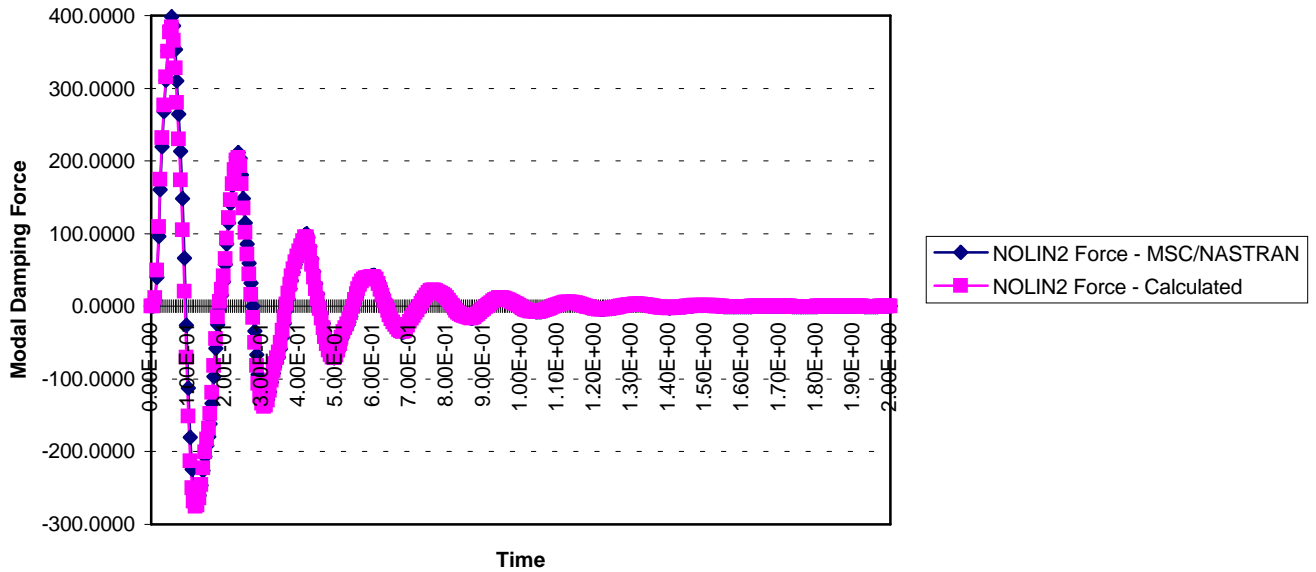
The following displacement history resulted for point 3.



Which is quite different from the first solution. In order to verify that the nonlinear damping forces were applied correctly, the displacement of Point 3 and the modal velocity of the first mode were loaded into a spreadsheet and the nonlinear forces were calculated and then compared to the MSC/NASTRAN nonlinear forces. The following two charts are the results of those comparisons:



NOLIN2 Force vs Calculated Value



As can be seen from these charts, the results from MSC/NASTRAN agree well with the calculated values.

5. CONCLUSIONS:

An approach is presented which allows the inclusion of nonlinear effects in a transient solution using modal coordinates. The alter uses MPC equations to include measurement dof with the modal coordinates and allow forces to be applied to the modal coordinates which are a function of either displacement or velocity. For this paper, the alter was used to include nonlinear modal damping as a function of the amplitude of a selected displacement. In the two samples shown, the results of the alter agree well with calculated values and a solution obtained without the alter. Thus far, the alter has only been used with a single measurement dof applying forces to a single modal coordinate. In principle, the alter allows for forces to be based on more than a single dof and for forces to be applied to more than one mode. Since those cases have not been tried yet, caution is recommended when using this alter. The alter is also intended to work with a single superelement and the residual structure, although it could be extended to allow a multi-level tree with a single (assembly) collector as the final superelement above the residual structure.

REFERENCES:

1. MSC/NASTRAN Version 68 Basic Dynamic Analysis User's Guide, Ken Blakely, The MacNeal-Schwendler Corporation.
2. MSC/NASTRAN Version 68 Reference Manual, R. Lahey, M. Miller, and M. Reymond, editors, The MacNeal-Schwendler Corporation.
3. Rose, Ted L. DMAP Alters to Apply Modal Damping and Obtain Dynamic Loading Output for Superelements, The 1993 MSC World Users' Conference, Paper 24.
4. The NASTRAN Theoretical Manual

Appendix - The DMAP Alter

```
$
$ 3-7-96 VERSION 68.2
$
$ filename - nldampa.v682 - alter for nonlinear modal damping
$ written by Ted Rose
$
$ DMAP alter to include modal damping for superelements
$
$ updated to allow modal damping on se modes, then to remove any part
$ of the static vectors which can be represented by the component modes
$
$ NOTE: this update is experimental and may result in wrong answers -
$ the matrix GOA (the static transformation) is updated, therefore
$ the statically reduced stiffness and mass will be modified. These
$ new matrices should not be used for a static solution.
$ This alter replaces the INREL module if C- or R-set dof exist for
$ the superelement. The INREL module removes any part of the modes
$ which the static transformation (GOAT) can represent. This alter
$ works in the opposite direction, the static vectors are removed.
$
$
$ based on original sedampa alter dated 6/25/1992
$
$
$ -----
compile semodes souin=mcsou nolist noref $ If used on SOL 103
alter 1 $
putsys(0,192) $
$
$
compile sedtran souin=mcsou nolist noref $
alter 1 $
putsys(0,192) $
$
compile phase1dr souin=mcsou list noref $
$alter 102,102 $
alter 'call phase1b','call phase1b,'
IF ( NOT SKIPSE ) CALL PHASE1B,
    GEOM4S,CASEW,GPLS,EQEXINS,CSTMS,BGPDTS,SILS,MATPOOL,
    DYNAMICS,FORE,EST,USSET0,RMG,KGG,MGG,BGG,K4GG,YG,PG,
    MEDGE,dit/
    GM,USSET,KSF,KFS,KSS,KVV,KAA,KOO,LOO,LAO,
    GOAT,MAA,MFF,CMPHA,CMPHO,CMLAMA,KLAA,GOAQ,MLAA1,MAR,
    MEA,MLAA,BAA,K4AA,PA,YS,UOO,PSS,UOX,
    KAAV,KLAAV,PAV,MAAV,BAAV,MLAAV,K4AAV,
    KMG,MMG,MSF,BMG,BSF,K4MG,K4SF/
    STATICS/ERROR/SEID/PEID/ALTRED/CYCLIC/NLHEAT/SEKR/
    SELR/SEMR/IMAGE/LUSETS/AUTOSPC/PRGPST/SPCGEN/
    USETPRT/USETSEL/EPZERO/EPPT/FIXEDB/IRES/PRPA/
    BAILOUT/MAXRATIO/ASING/EPSPMALU/EPSPMALC/
```

```

    EPSBIG/EPSRC/PRPHIVZ/INRLM/INREL/descycle/
    WTMASS/NSKIPL/NLOADS $
    COMPILE PHASE1B SOUIN=MSCSOU NOREF LIST
    alter 1,1 $
    SUBDMAP PHASE1B GEOM4S,CASES,GPLS,EQEXINS,CSTMS,BGPDTS,SILS,MATPOOL,
        DYNAMICS,FORE,EST,USE0,RMG,KGG,MGG,BGG,K4GG,YG,PG,
        MEDGE,dit/
        GM,USE0,KSF,KFS,KSS,KVV,KAA,KOO,LOO,LAO,
        GOAT,MAA,MFF,CMPHA,CMPHO,CMLAMA,KLAA,GOAQ,MLAA1,MAR,
        MEA,MLAA,BAA,K4AA,PA,YS,UOO,PSS,UOX,
        KAAV,KLAAV,PAV,MAAV,BAAV,MLAAV,K4AAV,
        KMG,MMG,MSF,BMG,BSF,K4MG,K4SF/
        STATICS/ERROR/SEID/PEID/ALTRED/CYCLIC/NLHEAT/SEKR/
        SELR/SEMR/IMAGE/LUSETS/AUTOSPC/PRGPST/SPCGEN/
        USETPRT/USETSEL/EPZERO/EPPRT/FIXEDB/IRES/PRPA/
        BAILOUT/MAXRATIO/ASING/EPSMALU/EPSMALC/
        EPSBIG/EPSRC/PRPHIVZ/INRLM/INREL/descycle/
        WTMASS/NSKIP/NLOADS $
    $
    ALTER 2
    TYPE PARM,,I,N,NEED,BEXISTS,NMODES $
    $
    $ Add modal damping if requested
    $
    $ALTER 54 $ AFTER CALL TO SEMRB
    alter 'call.*semrb'
    $
    If (SEID<> 0) THEN $
        DPD  ,,GPLS,SILS,USE0,/
            GPLD,SILD,USE0D,TFPOOL,DLT,PSDL,FRL,NLFT,
            TRL,EED,EQDYN/-1/S,N,LUSE0D/0/0/0/0/0/
            V,N,DYNSTAT=0/V,Y,NEWDYN=1/123/S,N,NOUE $
        GKAM  USE0D,cmpho,,CMLAMA,DIT,,,,CASES/
            MHHS,BHHS,KHHS,PHIDHS/NOUE/LMODES/LFREQ/
            99999./-1/-1/-1/0/S,N,FMODE/KDAMP $
        PARAML BHHS/'PRESENCE'///S,N,BEXISTS $
        IF (BEXISTS<>-1) THEN $ MODAL DAMPING
        PARAML  CASES/'DTI'/1/149/S,N,SDAMP $
        MESSAGE //' '$
        MESSAGE //'**** Modal damping set'/sdamp/
            'requested by user for superelement'/seid $
        MESSAGE //' '$
    PARAML BHHS/'TRAILER'/1/S,N,NMODES $
    IF (NOQSET = NMODES) THEN $
        UMERGE1 USE0,BHHS,,/BHHA/'A'/'Q'/'T' $
    ELSE $
        NEED = NOQSET-NMODES $
        MATGEN ,/PARTNA/6/NOASET/NOTSET/NMODES/NEED $
        MERGE  ,,,,BHHS,PARTNA,/BHHA/-1 $
    ENDIF $
        ADD BAA,BHHA/BAAH $
        PURGEX /BAA,,,,/ALWAYS $
        EQUIVX BAAH/BAA/ALWAYS $
    ENDIF $ MODAL DAMPING

```

```

$
ENDIF $
$ -----
COMPILE SEMR3, SOUIN=MSCSOU, LIST, NOREF
$ALTER 103 $ REMOVE INREL IF SDAMPING IS USED
alter 'vecplot.*vgt'(1,-1)
PARAML CASES/'DTI'/1/149//S,N,SDAMP $
IF (SDAMP<=0) THEN $
$ALTER 106 $ after inrel
alter 'inrel.*uset'
ELSE $
  MESSAGE //'**** DMAP INFORMATION MESSAGE ****' $
  MESSAGE //' ' $
  MESSAGE //' ' $
  MESSAGE //'**** USER HAS SPECIFIED THE USE OF MODAL DAMPING/'
  ' FOR THE CURRENT SUPERELEMENT' $
  MESSAGE //'**** THEREFORE THE INREL MODULE IS DISABLED FOR THE/'
  ' CURRENT SUPERELEMENT****' $
  MESSAGE //' ' $
TYPE PARM,,I,N,NEED $
  PARAML PHIVZ/'TRAILER'/1/S,N,NMODES $
$ replace INREL module
$
$ expand eivengector matrix to be "Q"-size
IF (NOQSET = NMODES) THEN $
  EQUIVX PHIVZ/phivzz/ALWAYS $
ELSE $
  NEED = NOQSET-NMODES $
  MATGEN ,/PARTNA/6/NOQSET/0/NMODES/NEED $
  MERGE ,,PHIVZ,,PARTNA,/phivzz/1 $
ENDIF $
  if(norc>0)then $
$ points are in the superelement R or C set - use the calculated
$ modes to produce MQQ, BQQ, and KQQ - these will be used in
$ the residual structure
$
  upartn uset,phivzz/goq,phivcr,/'v''o''c'/1 $
$ $
$ get mode shapes at R- and C- dof
$
$ upartn uset,phivzz/,,phivcr/'v''o''c'/1 $
  umerge uset,,phivcr/phicr/'v''o''c' $
  umerge uset,phicr,/phigcr/'g''v''s' $ G-size
  type parm,,i,n,luset1=0,temp1a=0 $
  paraml phigcr/'trailer'/2/s,n,luset1 $
  matgen eqexins/intext/9/0/luset1 $
  mpyad intext,phigcr,/phigcrex $
  matmod phigcrex,eqexins,,,/phcrpool,/16/1 $ create MATPOOL datablock
$
$ get ids of dof used for modes
$
  vec uset/modedof/'bitid'///10 $
  mpyad intext,modedof,/modeext $
  matmod modeext,eqexins,,,/modepool,/16/1 $ create mode identification

```

```

call dbstore phcrpool,modepool,,/0/0/'DBALL'/s,temp1a $
$
else $
  message //'DMAP alter warning message - the current superlement'/
  ' has no C- or R- set points. The use of this alter' $
  message //' depends on the use of these points to measure '/
  'displacements at the selected locations' $
  message //' It is possible that the alter may fail further'
  ' in the run.' $
  equivx phivzz/goq/always
endif $
ENDIF $
if(norc<0)then $
alter 'goq,koo,goq'
else $
$
$ R- or C- set exists - generate KQQ using V-set matrices
smpyad phivzz,kvv,phivzz,,/kqq/3///1///6 $ modal stiffness matrix
endif $
$
alter 'GOAQ,MOO,GOAQ','GOAQ,MOO,GOAQ'
smpyad phivzz,mvv1,phivzz,,/mqq/3///1///6 $ modal mass matrix
equivx mqq/mlaa/notset $
if(notset>-1)then $
UMERGE1 USET,MQQ,,/MLAA1/'A'/'Q'/'T'/0 $
endif $
$
$
compile gma , souin=mscsou, list, noref
alter 199 $ after gkam
PARAML CASES//'DTI'/1/149//S,N,SDAMP $
$
compile sekr, souin=mscsou, list, noref
$
$ replace standard MPC processing for residual structure with one which
$ attaches the modes to the measurement points
$
alter 'IF (NOMSET >=0)(,-1)
if(seid=0 and nomset<0)then $
  message //'dmap alter fatal error' $
  message //'there must be an mpc set selected for '/
  'the residual structure' $
  message //'this is how the alter determines the '/
  'attachment between the modes and the '/
  'measurement dof' $
  exit $
endif $
alter 'IF (NOMSET >=0'
if(seid=0)then $
$
$ residual structure - attach modes to measurement dof
$
type parm,,i,n,gotit=0 $
call dbfetch /phcrpool,modepool,,/0/0/0/s,gotit $

```

```

select=(name,size,cdate,ctime,zuzr01,zname) $
if(gotit<>-1)then $

    message //'dmap alter message - the identification matrices '/'
        'from the superelement do not exist' $
    message //'check your input for the superelement and for any '/'
        'other messages from the alter'
    exit $
else $
$ seid=0 and identification matrices found
PARAML  USET0//USET'/////G/S,N,NOGSET $
MTRXIN  ,,phcrPOOL,EQEXINS,SILS,/phigcrex,,/NOGSET/
        S,N,NOPhi/// $
MTRXIN  ,,modePOOL,EQEXINS,SILS,/modeext,,/NOGSET/
        S,N,NOMod/// $
$ phigcrex = eigenvectors for the C- and R- sets
$ modeext = 1.0 for columns containing the component modes
PARAML  USET0//USET'///S,N,NOUSETX//U1/S,N,NOu1SET $
if(nou1set<0)then $
    message //'dmap alter fatal - there is no U1-set defined '/'
        'for the residual structure' $
    message //'all measurement dof need to be identified '/'
        'in the U1 set' $
    message //'these dof should be defined using SPOINTs and '/'
        'connected to the physical dof in the superlement' $
    message //'by mpc equations' $
    exit $
endif $
vec  uset0/meas/'bitid'///1 $ partitioning vector for U1 set
$ get rows for measured dof only
partn  phigcrex,,meas/,phicru1,,/1 $
vec  uset0/nset/'bitid'///32 $
partn  nset,,modeext/,npart,,/1 $
partn  phicru1,npart/,phin,,phim,/1 $
merge  phin,,,npart/,phinall/1 $
    merge  ,,phinall,,modeext/,phing/1 $ mode shapes for measured dof
upartn  uset0,phing/phim1,phimn,phinm,phinm/'G/'M/'N'/2 $
    matgen  /measid/1/nou1set $
umerge1  uset0,measid,,/measid/'G/'U1/'O'/2 $
upartn  uset0,measid/,measnm,'G/'M/'N'/2 $
add  measnm,phinm/combin/(-1.,0.) $
if(nomset=1)then $ SOLVE will not invert a 1x1 matrix
    matgen  /i1/1/1 $
    add i1,phim/phiminv///2 $
else $
    solve  phim,/phiminv/3 $ invert phim -
$       if there are a lot of ID grids, this may be expensive
endif $
mpyad  phiminv,combin,/gm $
endif $
else $
$
$ regular mpc processing
$

```

```

alter 'MCE2  USET0,GM,KGG'(-,1)
endif $
$
$ replace goat and kaa by null matrices
$
alter 'LOO,UOO,KOA','KAA1/KAA'
matgen ./goat/7/nooset/noaset $
matgen ./kaa/7/noaset/noaset $
$
compile semr2, souin=mscsou, list, noref
alter 'semr2','semr2'
SUBDMAP SEMR2  MGG,GOAT,USET,GM/
  MAA,MFF,MMG,MSF/
  ERROR/NOMSET/NOSSET/NOOSET/NORC/NOQSET/NOTSET/seid $
type parm,nddl,i,n,seid $
$
$ replace maa by null matrix
$
alter 'MFF/MAA/NO'(-,1)
if(seid>0)then $
PARAML  USET/"USET"////S,N,NOUSETX/"A"/S,N,NOASET $
matgen ./maa/7/noaset/noaset $
else $
$
alter 'MAA1,MAA4'(+,1)
endif $
$
compile semrm, souin=mscsou, nolist, noref
alter 'call semr2','call semr2'
IF ( NOMAA=-1 ) CALL SEMR2  MGG,GOAT,USET,GM/
  MAA,MFF,MMG,MSF/
  ERROR/NOMSET/NOSSET/NOOSET/NORC/NOQSET/NOTSET/seid $
$
$ END OF nonlinear MODAL DAMPING alter
$

```