CHAPTER 17

SISO DISK DRIVE ACTUATOR MODEL

17.1 Introduction

This chapter will use an ANSYS model of a complete disk drive actuator/suspension system to expand on the methods and examples of the last two chapters.

While simple in appearance, a disk drive actuator/suspension system must fulfill a number of exacting requirements. The suspension system is required to provide a stiff connection between the actuator and the head in the seeking/track-following direction, while providing a compliant system in a direction perpendicular to the plane of the disk. This allows the air bearing supported head to comply to the shape and vibration of the disk. The actuator is designed with low mass to allow fast seeking. It must have resonant characteristics which provide small residual vibration following a seek from one track to another. Since the entire disk drive is subject to various shock and vibration events, the actuator dynamics must aid in preventing the head from unloading from the disk during the event.

The actuator/suspension system used as the example for this and the next chapter is a single disk actuator, with two arms and two suspensions. It is purposely designed with poor resonance characteristics (different thickness arms, coil positioned off the mass center of the system, etc.) in order to provide a richer resonance picture for analysis.

We will assume that the servo system used with the actuator is a sampled system with a 20khz sample rate, meaning that the Nyquist frequency is 10khz. We need to understand all the modes of vibration of the system up to at least 20khz because the sampled system will alias frequencies that are higher than 10khz back into the 0 to 10khz range.

We will find that **the dynamics of this ANSYS model with approximately 21000 degrees of freedom can be described well using between 8 and 20 modes of vibration** (16 to 40 states), depending on what measure of "goodness" is used. If we are interested in impulse response, we will see in the next chapter that using only eight modes results in a system with approximately a 5% error. For a good fit in the frequency domain through 10 khz only 8 modes are required, while a good fit through 20 khz requires 20 modes. In a well-designed actuator (this example is poorly designed as mentioned earlier) fewer than 20 modes are required since symmetry will couple in fewer modes.

This actuator/suspension model is a good example of what the book is all about: generating low order models of complicated systems, in this case a model which is approximately 1000 times smaller than the original model.

Once the ANSYS model results are available, a MATLAB model will be created. Then we will analyze several methods of reducing the size of the model. In the previous chapters, we used dc gains of the individual modes of vibration to rank the most important modes to keep. If we use uniform damping (the same zeta value for all modes) we will reach the same ranking conclusion using either dc gain or peak gain. However, if we use non-uniform damping, peak gain ranking is required. The MATLAB code will prompt for whether uniform or non-uniform damping is being used and will choose the appropriate ranking, dc gain or peak gain. The next chapter will introduce another, more elegant method of ranking modes to be eliminated, balanced reduction.

17.2 Actuator Description

Figure 17.1 shows top and cross-sectioned side views of the actuator used for the analysis. The global XYZ coordinate system for the model is indicated.

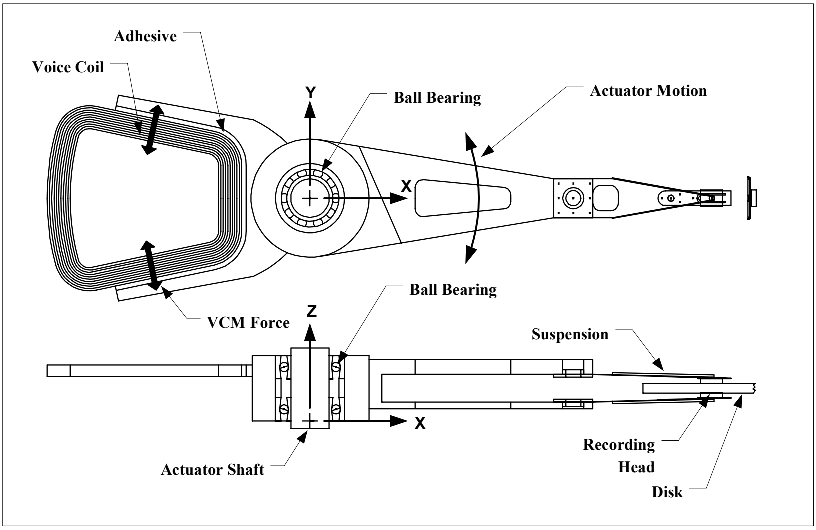

Figure 17.1: Drawing of actuator/suspension system.

The shaft is constrained in all directions, providing a fixed reference about which the actuator rotates on two axially preloaded ball bearings. This actuator is purposely designed to have poor dynamic characteristics, as seen in the side view. The coil, to which the Voice Coil Motor (VCM) forces are applied, is not centered between the two bearings and the two arms are of unequal thickness. Both the coil force mispositioning and the unequal arm thickness inertial effects will tend to excite rotations about the x axis.

The coil is bonded to the aluminum actuator body. During operation, current passes through the coil windings. The current interacts with the magnetic field from pairs of magnets above and below the straight legs of the coil (not shown), creating forces on the straight legs. The direction of the force is dependent on the direction of the current in the coil, clockwise or counterclockwise. The motion of the actuator due to the coil force is indicated by “Actuator Motion.”

The suspensions are designed to provide a preload of several grams force onto the disk surface. During operation the preload is counterbalanced by the air bearing lifting force, controlling the flying height spacing between the head and disk to less than several microinches. During shipment, the preload tends to hold the head down on the disk surface in the event of shock and vibration events, preventing potential damage caused by the head lifting off and striking the disk.

17.3 ANSYS Suspension Model Description

Before analyzing the complete actuator/suspension system, we will analyze only the suspension system. Understanding the dynamics of sensitive components of larger assemblies as components can add considerable insight to interpretation of the dynamics of the overall system.

The suspension portion of the actuator/suspension model is shown in Figures 17.2 and 17.3. The complete suspension is depicted in Figure 17.2, and the “flexure” portion of the suspension is shown in Figure 17.3.

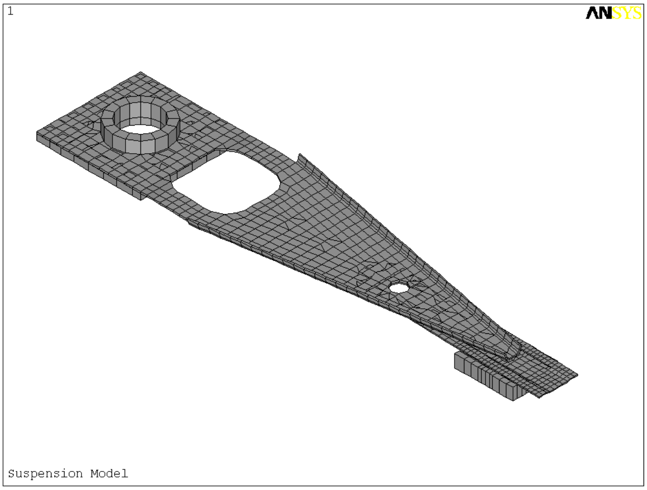

Figure 17.2: Suspension model.

The recording head (slider) is bonded to the center section of the flexure. The "dimple" at the center of the slider tongue provides a point contact about which the slider can rotate in the pitch and roll directions. The tip of the dimple and the contact point on the underside of the loadbeam are constrained to move together in translation. The flexure body is laser welded to the loadbeam (the triangular section), which is itself laser welded to the swage plate at the left-hand end.

The boundary conditions for the suspension model are: the swage plate is constrained in the x and z directions and the four slider corners are constrained in the z direction. A large mass is attached at the swage plate to allow for y direction ground acceleration forcing function. Because there is no constraint in the y direction there will be a zero-frequency, rigid body mode in that direction.

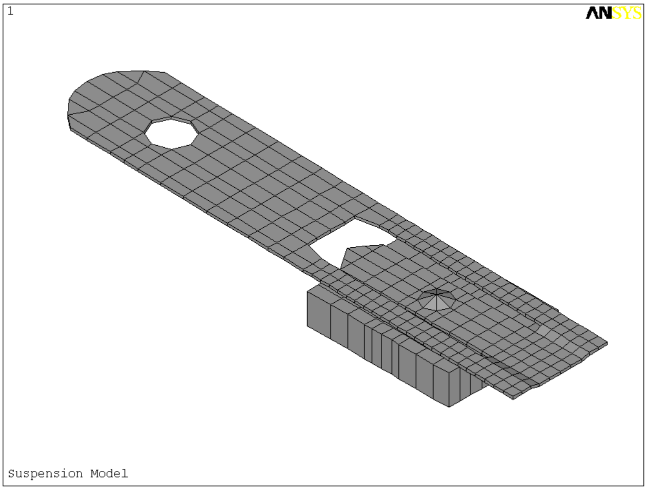

Figure 17.3: Flexure and recording head (slider) portion of suspension. Note the "dimple" at the center of the slider, a point about which the slider rotates to comply with the disk topology.

The model is built with the ability to easily change the critical flatness and forming parameters because the dynamics of the suspension are so dependent on the geometry. Small (0.025 mm, 0.001 inch) defects in critical forming and flatness parameters can drastically change the resonance characteristics,

The suspension model is made completely of eight-node brick elements. Laser welds and bonded joints are simulated by "merging" the nodes being welded or bonded, essentially creating a rigid joint at that connection.

The ANSYS suspension-only model, **srun.inp**, is included in the available downloads but will not be discussed. Running the model with different values for the three input parameters "zht," "bump" and "offset" will show the extreme sensitivity of the first torsion mode (described below) to these parameters.

17.4 ANSYS Suspension Model Results

The suspension has six modes of vibration in the 0 to 10 khz frequency range. The ANSYS frequency response plot for the suspension is shown in Figure 17.4. The six modes in the 0 to 10 khz will be plotted and described below.

17.4.1 Frequency Response

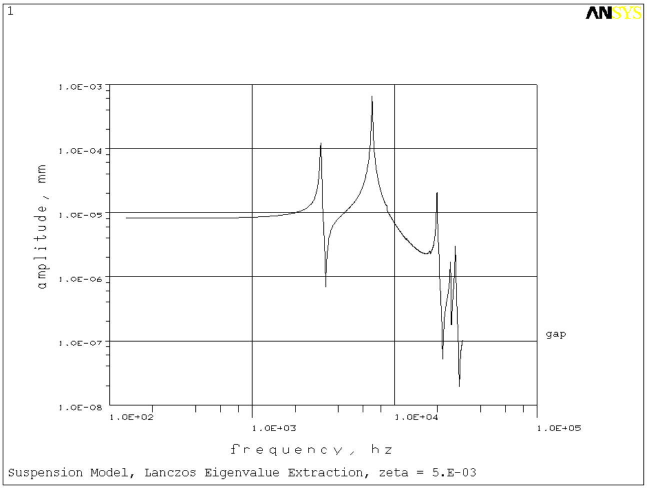

Figure 17.4: Suspension frequency response for a y direction forcing function.

17.4.2 Mode Shape Plots

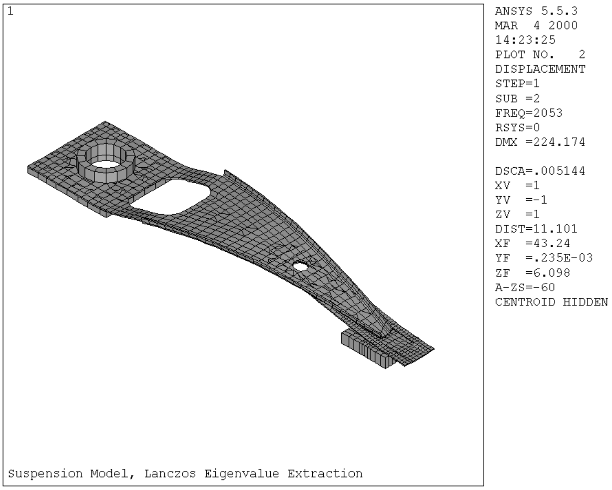

Figure 17.5: Mode 2, 2053 hz, first bending mode.

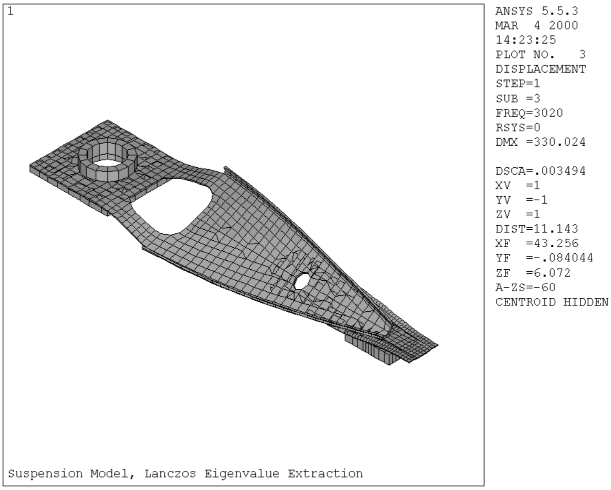

Figure 17.6: Mode 3, 3020 hz, first torsion mode.

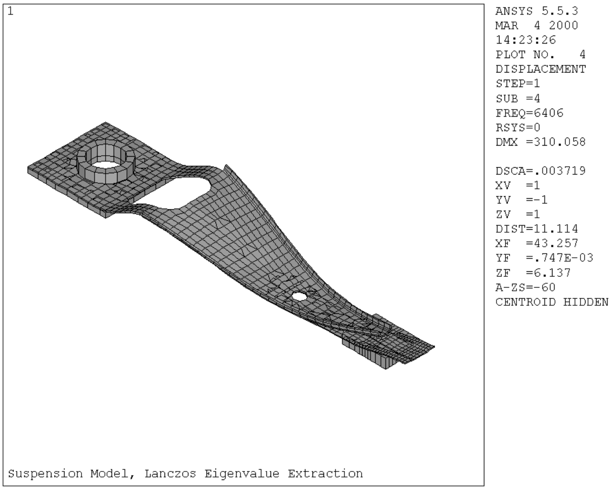

Figure 17.7: Mode 4, 6406 hz, second bending mode.

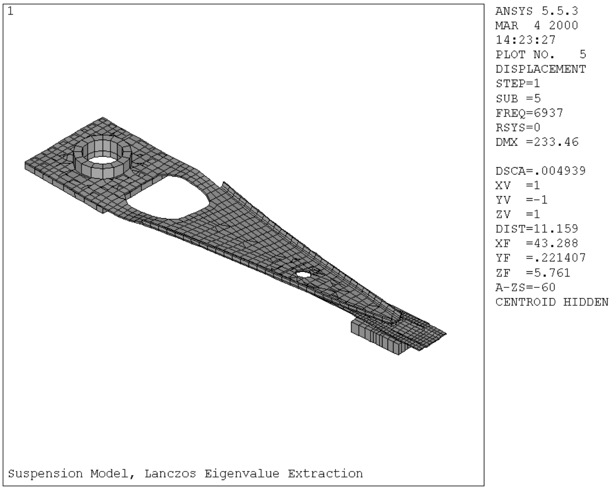

Figure 17.8: Mode 5, 6937 hz, sway or lateral mode.

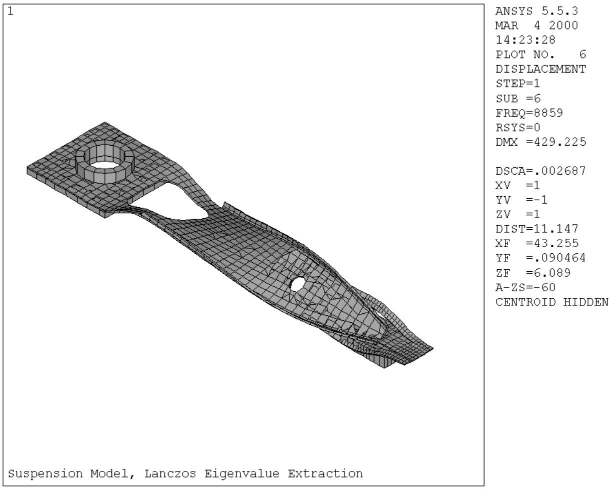

Figure 17.9: Mode 6, 8859 hz, second torsion mode.

The suspension frequency response plot and mode shape plots complement each other and help to develop a visual, intuitive understanding of modal coupling. The only modes that have y direction motion of the slider relative to the swage plate are the first torsion and sway modes as can be seen in the frequency response plot of Figure 17.4. All the other modes have motions which are orthogonal to the motion of interest. The first bending mode is the most obvious example. Since its motion in only in the z direction, it cannot be excited by a y direction forcing function, and thus, does not couple into the frequency response.

17.5 ANSYS Actuator/Suspension Model Description

The complete actuator/suspension model is shown in Figure 17.10. It also is made of eight-node brick elements except for the inclusion of spring elements which are used to simulate the ball bearings' individual ball stiffnesses.

The shaft and inner radii of the two ball bearing inner rings are fully constrained. The four corners of each of the sliders are constrained for zero motion in the z direction, essentially creating an infinitely stiff air bearing.

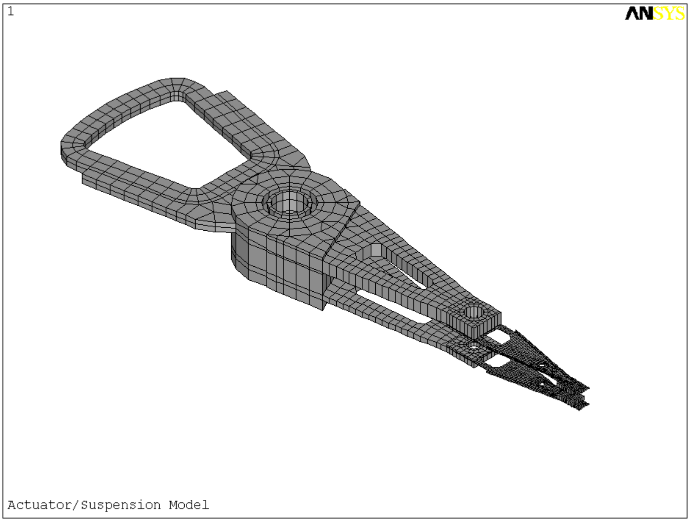

Figure 17.10: Complete actuator/suspension model.

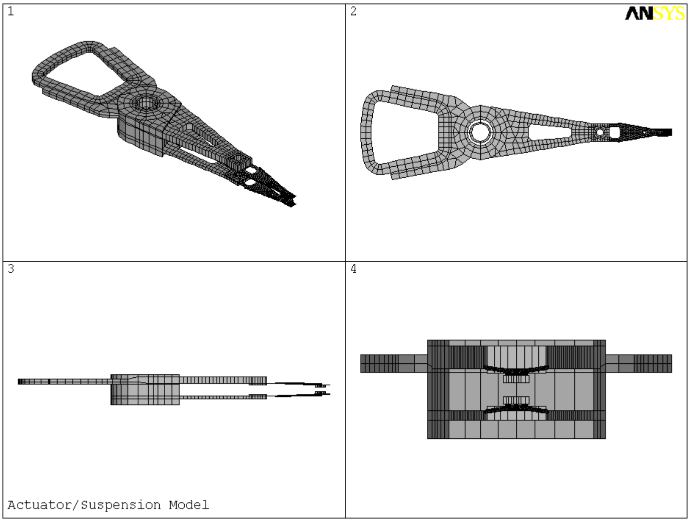

Figure 17.11: Actuator / suspension model, four views.

The primary motion of the actuator is rotation about the pivot bearing, therefore the final model has the coordinate system transformed from a Cartesian x,y,z coordinate system to a Cylindrical, r, θ and z system, with the two origins coincident.

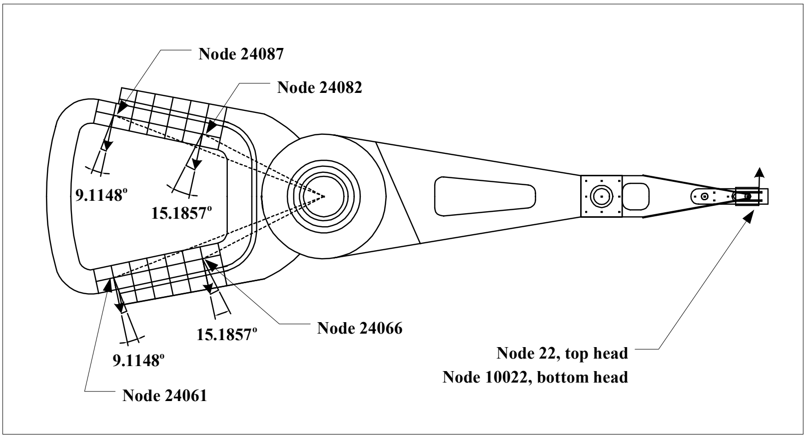

Figure 17.12: Nodes used for reduced MATLAB model. Shown with partial finite element mesh at coil.

For reduced models we only require eigenvector information for degrees of freedom where forces are applied and where displacements are required. Figure 17.12 shows the nodes used for the reduced MATLAB model. The four nodes 24061, 24066, 24082 and 24087 are located in the center of the coil in the z direction and are used for simulating the VCM force. The forces created by the interactions between the current in the straight legs of the coil and the magnetic field are perpendicular to the straight leg sections. Since the coordinate system is cylindrical, the forces are decomposed into radial and circumferential components as shown in Figure 17.12. Nodes 22 and 10022 are the nodes for the top and bottom heads (heads 1 and 0), respectively. The arrows at the nodes indicate the direction of forces, and the angles show the directions of the force, measured from the circumferential direction. The components in the radial and circumferential directions are taken using the angles.

The model uses only the circumferential motion of the heads, which, if divided by the radius from the pivot to the head, will give output in radians.

The actuator/suspension ANSYS code, **arun.inp,** is too large to be listed here but is available for downloading.

17.6 ANSYS Actuator/Suspension Model Results

A recommended sequence for analyzing dynamic finite element models is:

1) Plot resonant frequencies versus mode numbers to get a feel for the frequency range. See if there are any significant jumps in frequency between modes which can indicate the system transitioning from one type of characteristic motion to another. For example, a sequence of bending modes transitioning into a sequence of torsional modes.

2) Plot frequency responses to define which modes couple into the response.

3) Plot and animate the mode shapes that contribute to the response, identifying modes that couple into motions in directions of interest and those that do not. Visually get a sense of how the geometry of the structure affects the modes.

4) Run parameter studies to understand the sensitivity of critical modes to design variables: dimensions, tolerances, material properties, etc.

17.6.1 Eigenvalues, Frequency Responses

The actuator/suspension model was run using the Block Lanczos method to extract the first 50 eigenvalues and eigenvectors. The plot of frequency versus number of modes is shown in Figure 17.13. The first mode, the rigid body mode, was calculated to be 0.0101 hz, with the first oscillatory mode frequency at 785 hz.

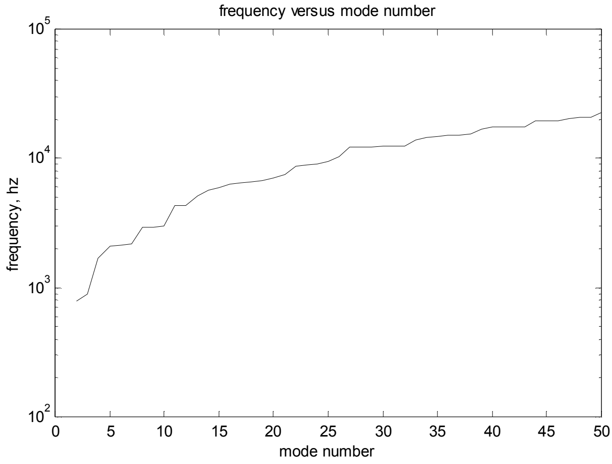

Figure 17.13: Frequencies versus mode number.

Mode 50 is at 22350 hz, which is slightly higher than our objective of including all the modes through 20 khz.

Frequency responses for the displacements of heads 0 and 1 (bottom and top heads) for coil input force can be seen in Figures 17.14 and 17.15. Mode shape plots, with undeformed and deformed shapes, are then shown for the modes which are evident in the frequency response plots. In addition, some typical modes that do not couple into the frequency response are shown.

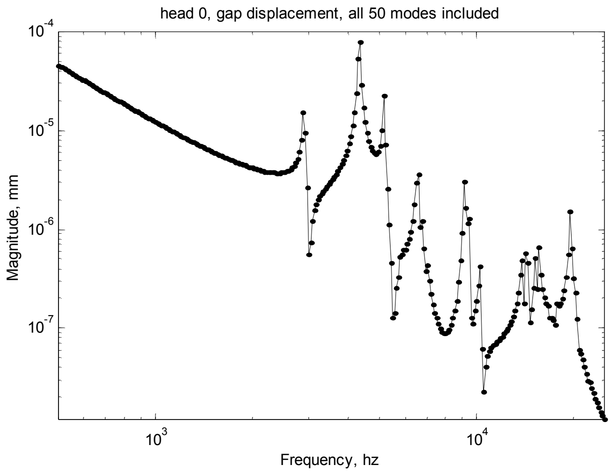

Figure 17.14: Frequency response for head 0 for coil input.

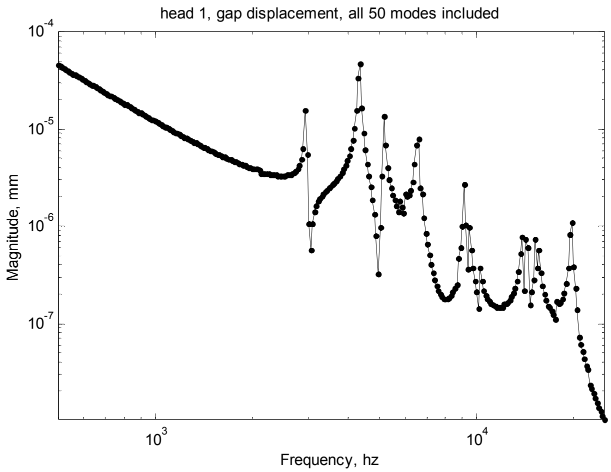

Figure 17.15: Frequency response for head 1 for coil input.

17.6.2 Mode Shape Plots

In this section we will plot overlaid undeformed and deformed modes shapes for selected modes, which will then be described and discussed in the next section.

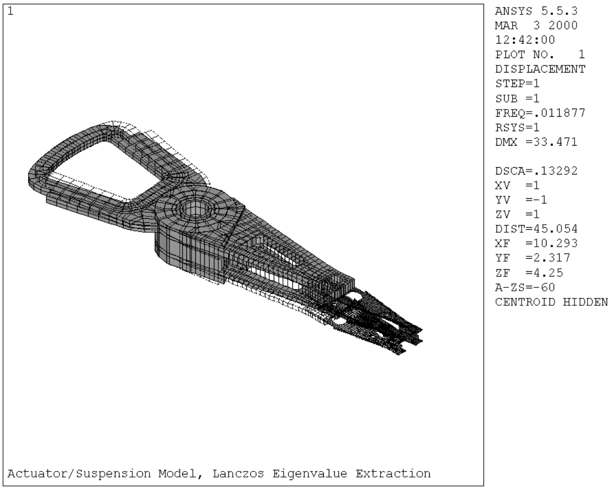

Figure 17.16: Mode 1 undeformed/deformed mode shape plot, 0.012 hz rigid body rotation.

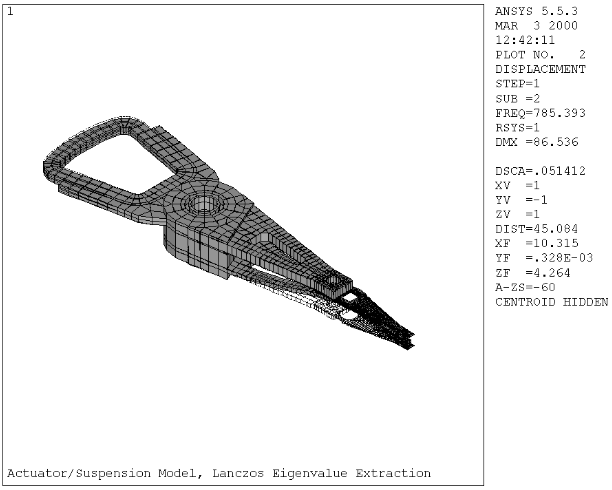

Figure 17.17: Mode 2 mode shape plot, 785 hz. Bending of bottom arm.

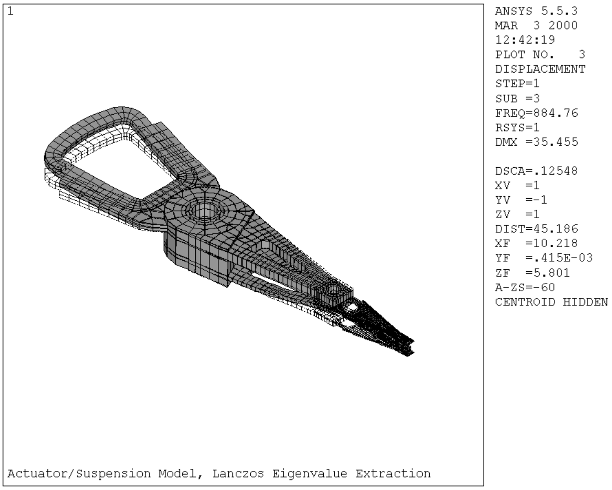

Figure 17.18: Mode 3 mode shape plot, 885 hz, coil and bottom arm bending.

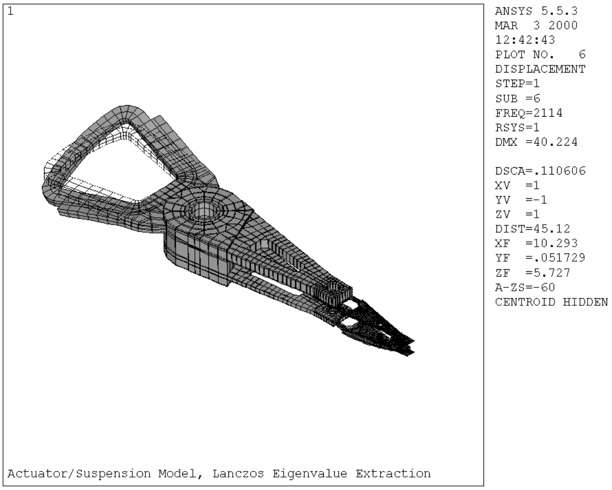

Figure 17.19: Mode 6 mode shape plot, 2114 hz, coil torsion.

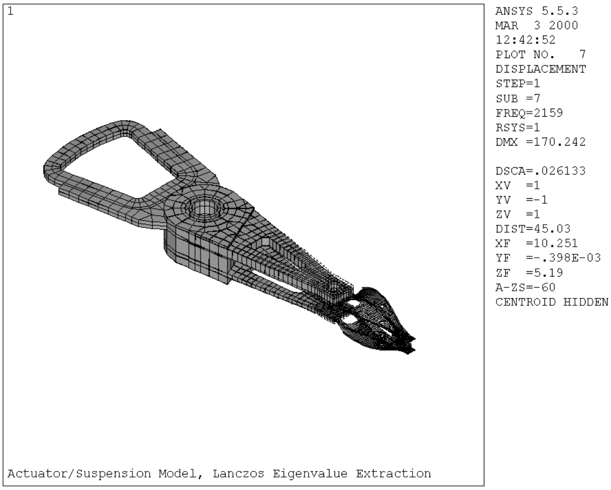

Figure 17.20: Mode 7 mode shape plot, 2159 hz, suspension bending modes.

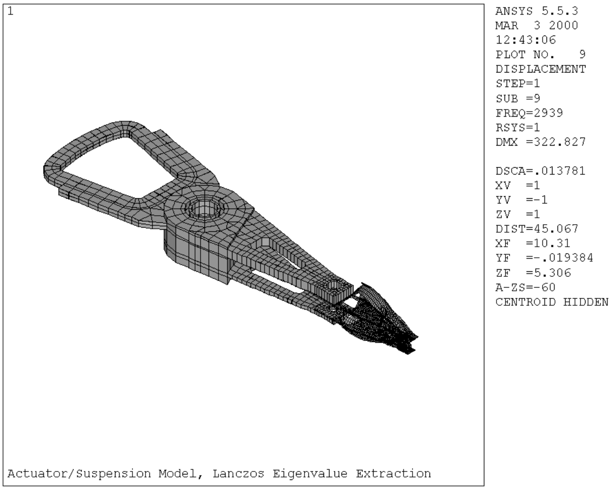

Figure 17.21: Mode 9 mode shape plot, 2939 hz, suspension torsion mode.

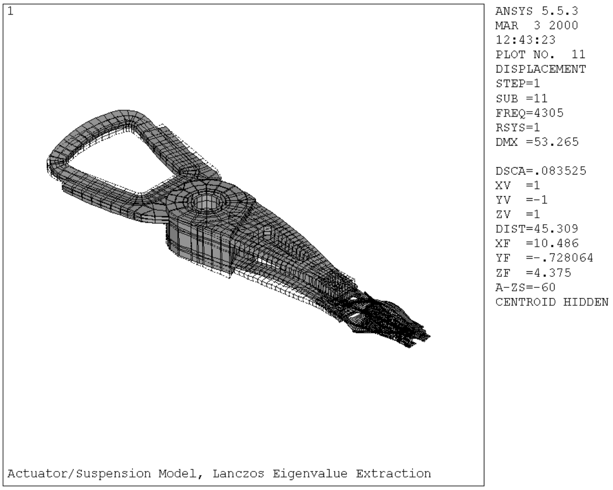

Figure 17.22: Mode 11 mode shape plot, 4305 hz, system mode.

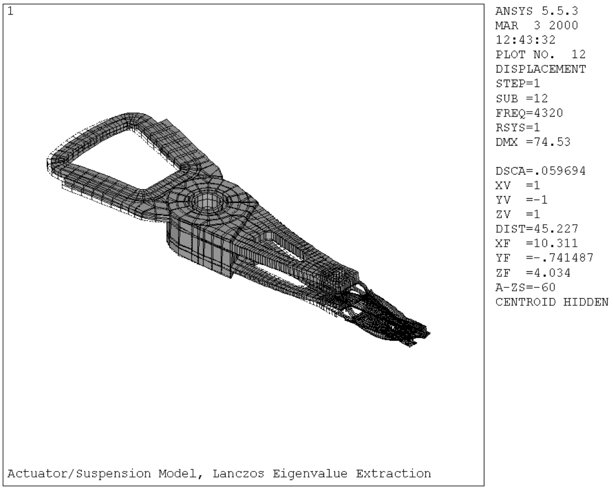

Figure 17.23: Mode 12 mode shape plot, 4320 hz, radial mode.

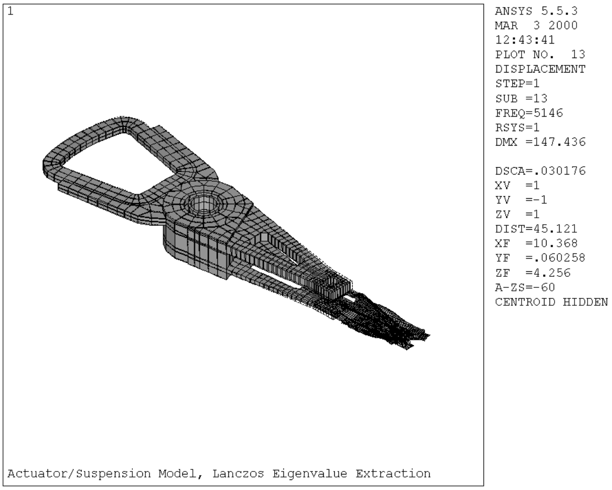

Figure 17.24: Mode 13 mode shape plot, 5146 hz.

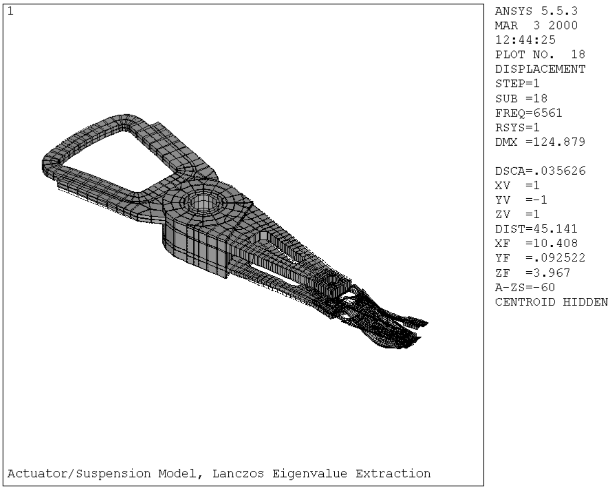

Figure 17.25: Mode 18 mode shape plot, 6561 hz.

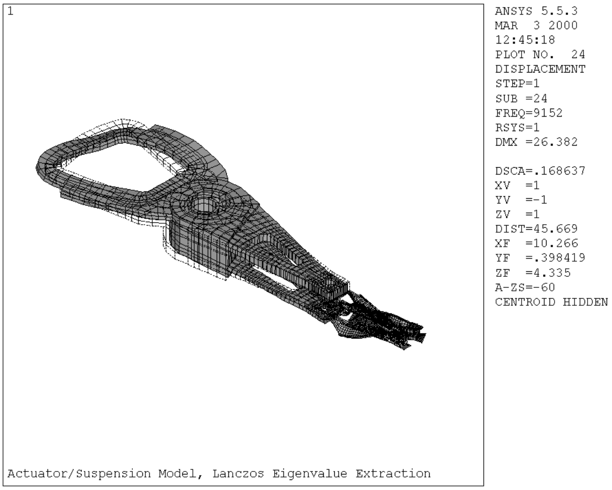

Figure 17.26: Mode 24 mode shape plot, 9152 hz.

17.6.3 Mode Shape Discussion

We will now correlate the two frequency response plots, Figures 17.14 and 17.15, with the mode shape plots above to start getting an intuitive feel for which modes couple into the response plots and which modes do not.

Mode 1, the rigid body mode, shows up as the 40db/decade low frequency slopes on both frequency responses, head 0 and head 1.

Modes 2 and 3, at 785 and 884 hz, are representative of modes that do not couple because of the direction of the motion. Both modes involve only bending motions of arms and/or coil in the x-z plane. Since the motions are perpendicular (orthogonal) to the direction of force and to the direction of the head in the circumferential direction, the modes should not couple into the frequency response plots. Therefore we see no resonance peaks at these two frequencies.

Mode 6 at 2114 hz is a coil/actuator torsion mode that shows up as the small pole/zero pair in the head 1 frequency response.

Mode 7 at 2159 hz is a suspension bending mode that does not couple into the response.

Mode 9 at 2939 hz is a suspension torsion mode that interacts with the rigid body mode to create the significant pole/zero pair at 2939 hz.

Modes 11 and 12 at 4305 hz and 4320 hz are the major system modes with significant y direction motion of the coil, bearings, arms and suspensions. These are the two modes associated with the highest resonant peak in the frequency response. What appears to be a single peak is actually two peaks.

Mode 13 at 5146 hz is a mode which involves torsion of the coil and actuator body about the x axis with the suspensions moving torsionally and laterally.

Mode 18 at 6561 hz is a suspension sway mode, where the suspension-only mode at 6937 hz (Figure 17.8) is reduced to 6561 hz because it is attached to the flexible actuator.

Mode 24 at 9152 hz is a highly deformed actuator mode, in which the actuator hub moves significantly about the ball bearing, the coil deforms and suspensions and arms deflect.

17.6.4 ANSYS Output Example Listing

A partial listing of the eigenvector output (actrl.eig) for modes 1, 2, 11 and 12 is shown below. These four modes were chosen for listing and discussion because they illustrate some key points about interpreting ANSYS eigenvector output. The important information in each of the eigenvector sections is highlighted in bold type. The "SUBSTEP" is the mode number, and "FREQ" is the eigenvalue in hz. Since the output is in cylindrical coordinates, UX, UY and UZ refer to radial, circumferential and z axis coordinates, respectively. Since all the elements attached to the six nodes listed are eight-node brick elements, with only translational degrees of freedom, all the rotation eigenvector values are zero. The six nodes listed correspond to the two heads, 22 and 10022 and the four coil forcing function nodes, 24061, 24066, 24082 and 24087. See Figure 17.12 for node locations. We need both radial (UX) and circumferential (UY) directions because the forces applied by the VCM to the coil are perpendicular to the straight legs of the coil, and have both radial and circumferential components.

```
PRINT DOF  NODAL SOLUTION PER NODE

 ***** POST1 NODAL DEGREE OF FREEDOM LISTING *****

 LOAD STEP=     1  SUBSTEP=     1
  FREQ=  0.11877E-01  LOAD CASE=   0

 THE FOLLOWING DEGREE OF FREEDOM RESULTS ARE IN COORDINATE SYSTEM
1

 NODE       UX          UY           UZ         ROTX      ROTY      ROTZ
      22  0.30718E-06  32.772    0.85804E-12  0.0000    0.0000    0.0000
```

10022	0.30759E-06	32.772	-0.49994E-10	0.0000	0.0000	0.0000
24061	0.11969E-06	16.968	-0.17668E-08	0.0000	0.0000	0.0000
24066	0.77415E-07	10.274	-0.15751E-08	0.0000	0.0000	0.0000
24082	0.68508E-07	10.274	-0.15395E-08	0.0000	0.0000	0.0000
24087	0.10089E-06	16.968	-0.16990E-08	0.0000	0.0000	0.0000

MAXIMUM ABSOLUTE VALUES

NODE	10022	22	24061	0	0	0
VALUE	0.30759E-06	32.772	-0.17668E-08	0.0000	0.0000	0.0000

*ENDDO INDEX= I

***** POST1 NODAL DEGREE OF FREEDOM LISTING *****

LOAD STEP= 1 **SUBSTEP= 2**
FREQ= 785.39 LOAD CASE= 0

THE FOLLOWING DEGREE OF FREEDOM RESULTS ARE IN COORDINATE SYSTEM
1

NODE	**UX**	**UY**	UZ	ROTX	ROTY	ROTZ
22	-0.25631	-0.19637E-01	0.15936E-04	0.0000	0.0000	0.0000
10022	0.92764	-0.10736	0 .29519E-02	0.0000	0.0000	0.0000
24061	0.18573	-0.67085E-01	-5.7724	0.0000	0.0000	0.0000
24066	0.17688	-0.88331E-01	-2.1255	0.0000	0.0000	0.0000
24082	0.17616	0.95885E-01	-2.1213	0.0000	0.0000	0.0000
24087	0.18506	0.79278E-01	-5.7661	0.0000	0.0000	0.0000

MAXIMUM ABSOLUTE VALUES

NODE	10022	10022	24061	0	0	0
VALUE	0.92764	-0.10736	-5.7724	0.0000	0.0000	0.0000

***** POST1 NODAL DEGREE OF FREEDOM LISTING *****

LOAD STEP= 1 **SUBSTEP= 11**
FREQ= 4305.3 LOAD CASE= 0

THE FOLLOWING DEGREE OF FREEDOM RESULTS ARE IN COORDINATE SYSTEM
1

NODE	**UX**	**UY**	UZ	ROTX	ROTY	ROTZ
22	-4.4488	27.588	-0.66528E-04	0.0000	0.0000	0.0000
10022	3.9832	41.657	0.44809E-01	0.0000	0.0000	0.0000
24061	-0.43605	-10.023	-8.7664	0.0000	0.0000	0.0000
24066	0.35112	-3.5631	-11.532	0.0000	0.0000	0.0000
24082	3.9625	-1.1137	-14.210	0.0000	0.0000	0.0000
24087	5.0136	-7.8562	-6.0297	0.0000	0.0000	0.0000

MAXIMUM ABSOLUTE VALUES

NODE	24087	10022	24082	0	0	0
VALUE	5.0136	41.657	-14.210	0.0000	0.0000	0.0000

***** POST1 NODAL DEGREE OF FREEDOM LISTING *****

LOAD STEP= 1 **SUBSTEP= 12**
FREQ= 4320.1 LOAD CASE= 0

THE FOLLOWING DEGREE OF FREEDOM RESULTS ARE IN COORDINATE SYSTEM 1

NODE	**UX**	**UY**	UZ	ROTX	ROTY	ROTZ
22	4.3947	36.811	-0.25761E-02	0.0000	0.0000	0.0000
10022	-0.88223	62.097	0.34209E-01	0.0000	0.0000	0.0000
24061	-5.3622	-11.584	3.9397	0.0000	0.0000	0.0000
24066	-3.9590	-2.2258	10.513	0.0000	0.0000	0.0000
24082	0.81662	-4.0070	7.7931	0.0000	0.0000	0.0000
24087	2.0281	-13.160	6.6813	0.0000	0.0000	0.0000

MAXIMUM ABSOLUTE VALUES

NODE	24061	10022	24066	0	0	0
VALUE	-5.3622	62.097	10.513	0.0000	0.0000	0.0000

We will now discuss the eigenvector listings above in light of the frequency response and mode shape plots reviewed earlier. Once again, we will make the connection between modes that contribute to frequency responses and those that do not.

Mode 1 shows that all the UX and UZ entries are essentially zero, which is appropriate for a rigid body mode where the actuator is rotating about the shaft, with only circumferential, UY, displacements. The relative amplitudes of each UY entry are related by their radial distances from the shaft. The frequency calculated is not exactly zero because of rounding and slight geometric errors which create small stiffnesses in rotation about the shaft.

Mode 2 is the first oscillatory mode, the arm bending mode. A mode which involves only UZ motion will have no cross-coupling in the y direction since the actuator system is symmetrical about the x axis. In a typical disk drive, the actuator is not perfectly symmetrical, and modes whose motions are primarily in the vertical direction will couple in the y direction. All of the UY entries for this mode are very small relative to the UZ entries, indicating that the contribution of this mode to the y direction motion of the head should be small.

Modes 12 and 13 are the major system modes, those modes with the highest amplitude motion on the frequency response plot. The entries in the UY column are significant relative to the entries for mode 2 and are of the same order of magnitude as those in mode 1. This indicates that this mode is relatively important for our desired frequency response.

The eigenvalues and UX and UY eigenvector entries are stripped out of the actrl.eig file and stored in the MATLAB .mat file actrl_eig.mat (Appendix 1). Now we are ready to read the ANSYS results into MATLAB and start developing the reduced model.

17.7 MATLAB Model, MATLAB Code act8.m Listing and Results

17.7.1 Code Description

The code starts by reading in the ANSYS model eigenvalue and eigenvector results for all 50 modes from actrl_eig.mat. The VCM force components in the radial and circumferential directions are then defined using the angles shown in Figure 17.12.

The user is prompted to specify whether the same zeta value is to be used for all modes (uniform damping), or whether each mode can have different values, non-uniform damping. If uniform damping is specified, the user is prompted to enter a value for zeta, a vector of uniform damping values is created and dc gains are calculated. If non-uniform damping is chosen, a damping vector is read in from **zetain.m** and peak gains are calculated. The appropriate gains are then sorted and plotted, indicating the most important modes to retain. Typically uniform damping is taken in the range of 0.005 (0.5% of critical damping) to 0.02 (2% of critical damping). If experimental data is available, the damping values for each mode in **zetain.m** can be matched to its experimentally determined value.

Once the user defines the number of modes to be retained, two state space systems are automatically built. The first includes all 50 modes and the second includes the sorted, reduced number of modes. The 50-mode response is plotted for either head 0 or head 1 with individual mode contributions overlaid.

Since the servo system postulated for the actuator has a 20 khz sample frequency, the Nyquist frequency is half that, or 10 khz. This means that resonances higher in frequency than the Nyquist frequency will be aliased back to the 0 to 10 khz range. The user is prompted for the sample frequency to be used (default 20 khz). The MATLAB “c2d” command is used to create a discrete model of the original continuous system. A discrete frequency response, with upper limit of the Nyquist frequency, is created and plotted, overlaying the original continuous frequency response. If the sample rate is high enough, this overlay allows one to see that it will not alias critical modes of vibration. Experimentally, the only information available from a discrete servo system frequency response is up to the Nyquist frequency. Measurements which are independent of the servo system (such as from an external laser measurement system) are required to identify modes higher than the Nyquist frequency. An example of using a very low sampling frequency with this actuator system will be shown.

Frequency responses are calculated using the reduced, sorted modes, truncating the less important modes and using the "modred" "mdc" option. Truncating is the same as using the "del" option on the MATLAB "modred" command.

17.7.2 Input, dof Definition

The first section of code reads in the eigenvalue/eigenvector data from **actrl_eig.mat** and defines explicitly the degrees of freedom used. The original ANSYS model has approximately 21000 degrees of freedom. By defining only the degrees of freedom required for the desired frequency response, we can reduce the number of degrees required for the MATLAB model to 12: the radial and circumferential components of the two head nodes and the four coil forcing function nodes.

```
%	act8.m

	clear all;

	hold off;

	clf;

%	load the Block Lanczos .mat file actrl_eig.mat, containing evr - the modal matrix,
%	freqvec - the frequency vector and node_numbers - the vector of node numbers
%	for the modal matrix

%	the output for the ANSYS run is the following dof's

%  dof  node        dir       where
%   1      22       ux - radial, top head gap
%   2   10022       ux - radial, bottom head gap
%   3   24061       ux - radial, coil
%   4   24066       ux - radial, coil
%   5   24082       ux - radial, coil
%   6   24087       ux - radial, coil
%   7      22       uy - circumferential, top head gap
%   8   10022       uy - circumferential, bottom head gap
%   9   24061       uy - circumferential, coil
%  10   24066       uy - circumferential, coil
%  11   24082       uy - circumferential, coil
%  12   24087       uy - circumferential, coil

	load actrl_eig;
```

```
[numdof,num_modes_total] = size(evr);

freqvec(1) = 0;          % set frequency of rigid body mode to zero

xn = evr;
```

17.7.3 Forcing Function Definition, dc Gain Calculation

A vector of the squares of the eigenvalues, in rad/sec units, for use in the gain calculations is generated. Like the dc gain calculation with a rigid body mode discussed in the last chapter, we will again calculate the low frequency gain of the rigid body mode using the lowest frequency defined in the frequency response calculation.

The forcing function components for the four coil nodes are defined, again using Figure 17.12 as the reference. A unity force is applied at the coil, and evenly distributed among the four nodes. The force at each coil node is decomposed into its components in the radial and circumferential (x and y) directions. The coil forces in physical coordinates are then defined for each coil node and where the ux and uy force entries for the head nodes, dof 1, 2, 7 and 8 are all zero.

A discussion of what is meant by "Single Input Single Output" (SISO) is appropriate here. This model is a "SI" or Single Input model because the same force is applied to all four coil nodes, requiring only a single column vector for the input matrix "b." The fact that forces are applied to multiple nodes has no significance relative to the "SI" definition.

In Chapter 15, (15.2) and (15.3), we found that the dc gain and peak gain of for the i^{th} mode are given by the expressions:

$$\frac{z_{ji}}{F_{ki}} = \frac{z_{nji} z_{nki}}{\omega_i^2}, \tag{17.1}$$

$$\frac{z_{ji}}{F_{ki}} = \frac{-j}{2\zeta_i}(\text{dc gain}) \tag{17.2}$$

where $z_{nji} z_{nki}$, the residue, is the product of the jth (output) row and kth (force applied) row terms of the ith eigenvector divided by the square of the eigenvalue for the i^{th} mode and ζ_i is the damping for the i^{th} mode. For all the models so far in the book, forces have been applied at a single node and displacements have been taken at a single node, making the above definitions clear. Here we are applying the same force to four coil nodes, so we will define a composite forcing function which will consist of the force applied to each node times the eigenvector value for that node, f_physical'*xn. The dimensions of this operation are (1 x ndof) x (ndof x nmodes) = (1 x nmodes), so we have a **composite** force vector for each mode.

This composite force vector is then multiplied element by element by the rows of the eigenvector matrix corresponding to the uy direction displacements of the two heads.

We will calculate and plot the gains for both head 0 and head 1 but will only calculate frequency response results for one or the other (user defined). Thus there is no ambiguity about whether to rank modes based on the gains of head 0 or head 1, only the one chosen for frequency response calculations is used for ranking.

```
%   calculate the dc amplitude of the displacement of each mode by
%   multiplying the composite forcing function by the output row

        omega2 = (2*pi*freqvec)'.^2;     %  convert to radians and square

%       define frequency range for frequency response

        freqlo = 501;

        freqhi = 25000;

        flo=log10(freqlo) ;
        fhi=log10(freqhi) ;

        f=logspace(flo,fhi,300) ;
        frad=f*2*pi ;

%       define radial and circumferential forces applied at four coil force nodes
%       "x" is radial, "y" is circumferential, total force is unity

        n24061fx = 0.25*sin(9.1148*pi/180);
        n24061fy = 0.25*cos(9.1148*pi/180);

        n24066fx = 0.25*sin(15.1657*pi/180);
        n24066fy = 0.25*cos(15.1657*pi/180);

        n24082fx = -0.25*sin(15.1657*pi/180);
        n24082fy = 0.25*cos(15.1657*pi/180);

        n24087fx = -0.25*sin(9.1148*pi/180);
        n24087fy = 0.25*cos(9.1148*pi/180);

%       f_physical is the vector of physical force
%       zeros at each output dof and input force at the input dof
```

```
            f_physical = [    0
                              0
                    n24061fx
                    n24066fx
                    n24082fx
                    n24087fx
                          0
                          0
                    n24061fy
                    n24066fy
                    n24082fy
                    n24087fy  ];

%           define composite forcing function, force applied to each node times
            eigenvector value
%           for that node

            force = f_physical'*xn;

%           choose which head to use for frequency responses

            head = input('enter "0" default for head 0 or "1" for head 1 ...  ');

            if  isempty(head)
                    head = 0;
            end

%           prompt for uniform or variable zeta

            zeta_type = input('enter "1" to read in damping vector (zetain.m)  …
                                  or "enter" for uniform damping ...  ');

            if  (isempty(zeta_type))

                    zeta_type = 0;

                    zeta_uniform = input('enter value for uniform damping, …
                                                .005 is 0.5% of critical (default) ... ');

                    if (isempty(zeta_uniform))
                            zeta_uniform = 0.005;
                    end

                    zeta_unsort = zeta_uniform*ones(num_modes_total,1);

                    gainstr = ‘dc gain’;

            else

                    zetain;       % read in zeta_unsort damping vector from zetain.m file

                    gainstr = ‘peak gain’;

            end
```

```
        if  length(zeta_unsort) ~= num_modes_total

                error(['error - zetain vector has ',num2str(length(zeta_unsort)), …
                        ' entries instead of ',num2str(num_modes_total)]);

        end

%       calculate dc gains if uniform damping, peak gains if non-uniform

        if  zeta_type == 0              % dc gain

                gain_h0 = abs([force(1)*xn(8,1)/frad(1) ...
                        force(2:num_modes_total).*xn(8,2:num_modes_total) ...
                        ./omega2(2:num_modes_total)]);

                gain_h1 = abs([force(1)*xn(7,1)/frad(1) ...
                        force(2:num_modes_total).*xn(7,2:num_modes_total) ...
                        ./omega2(2:num_modes_total)]);

        elseif  zeta_type == 1          % peak gain

                gain_h0 = abs([force(1)*xn(8,1)/frad(1) ...
                force(2:num_modes_total).*xn(8,2:num_modes_total) ...
                ./((2*zeta_unsort(2:num_modes_total))'.*omega2(2:num_modes_total))]);

                gain_h1 = abs([force(1)*xn(7,1)/frad(1) ...
                force(2:num_modes_total).*xn(7,2:num_modes_total) ...
                ./((2*zeta_unsort(2:num_modes_total))'.*omega2(2:num_modes_total))]);

        end

%       sort gains, keeping track of original and new indices so can rearrange
%       eigenvalues and eigenvectors

        [gain_h0_sort,index_h0_sort] = sort(gain_h0);

        [gain_h1_sort,index_h1_sort] = sort(gain_h1);

        gain_h0_sort = fliplr(gain_h0_sort);                % max to min

        gain_h1_sort = fliplr(gain_h1_sort);                % max to min

        index_h0_sort = fliplr(index_h0_sort)               % max to min indices

        index_h1_sort = fliplr(index_h1_sort)               % max to min indices

        index_orig = 1:num_modes_total;

        if  head == 0

                index_sort = index_h0_sort;

                headstr = 'head 0';
```

```
                    index_out = 2;

          elseif  head == 1

                    index_sort = index_h1_sort;

                    headstr = 'head 1';

                    index_out = 1;

          end

%         plot results

          semilogy(index_orig(2:num_modes_total),freqvec(2:num_modes_total),'k-');
          title('frequency versus mode number')
          xlabel('mode number')
          ylabel('frequency, hz')
          grid off
          disp('execution paused to display figure, "enter" to continue'); pause

          semilogy(index_orig,gain_h0,'k-',index_orig,gain_h1,'k.-')
          title('dc value of each mode contribution versus mode number')
          xlabel('mode number')
          ylabel('dc value')
          legend('head 0','head 1')
          grid off
          disp('execution paused to display figure, "enter" to continue'); pause

          loglog(freqvec(2:num_modes_total),gain_h0(2:num_modes_total),'k-', ...
                     freqvec(2:num_modes_total),gain_h1(2:num_modes_total),'k.-')
          title('dc value of each mode contribution versus frequency')
          xlabel('frequency, hz')
          ylabel('dc value')
          legend('head 0','head 1')
          axis([500 25000 -inf 1e-4])
          grid off
          disp('execution paused to display figure, "enter" to continue'); pause

          semilogy(index_orig,gain_h0_sort,'k-',index_orig,gain_h1_sort,'k.-')
          title('sorted dc value of each mode versus number of modes included')
          xlabel('modes included')
          ylabel('sorted dc value')
          legend('head 0','head 1')
          grid off

%         choose number of modes to use based on ranking of dc gain values

          num_modes_used = input(['enter how many modes (including rigid body) …
                to include, 'num2str(num_modes_total),' max, 8 default ...  ']);

          if (isempty(num_modes_used))
                    num_modes_used = 8;
          end
```

```
num_states_used = 2*num_modes_used;
```

17.7.4 Ranking Results

Here, we will begin by reviewing the frequency versus mode number plot to get a feel for the frequency range of the model.

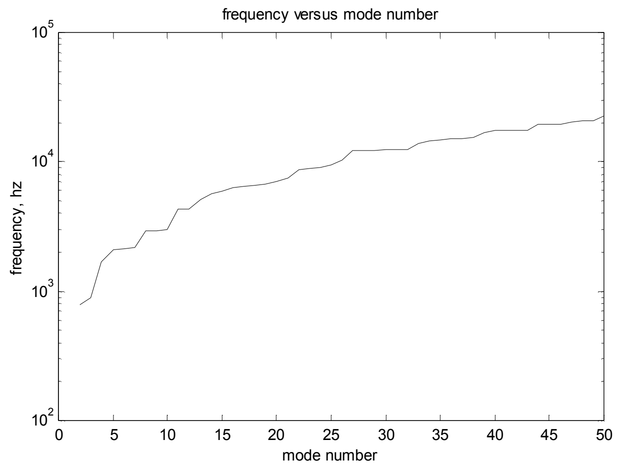

Figure 17.27: Frequency versus mode number.

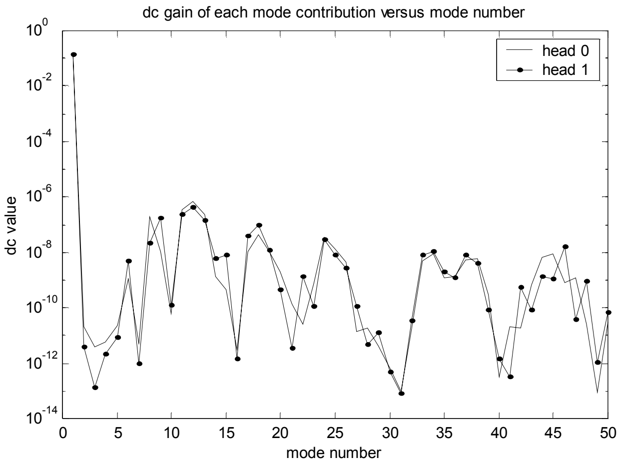

Figure 17.28: dc gain versus mode number, uniform damping zeta 0.005 (0.5% of critical damping) for all modes.

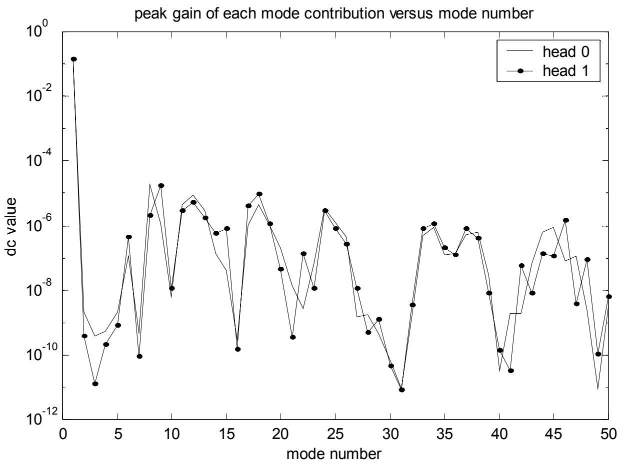

Figure 17.29: Peak gain versus mode number, non-uniform damping, zeta = 0.04 (4% of critical damping) for modes 11, 12 and 13.

The dc and peak gain plots for both head 0 and head 1 are shown above. Note the relative heights of the dc and peak gains for modes 11, 12 and 13. In the peak gain plot, those three gains are lower than the two gains immediately to the left. Conversely, in the dc gain plot the three modes are the highest gains with the exception of the rigid body mode.

The same two plots versus frequency, instead of mode number:

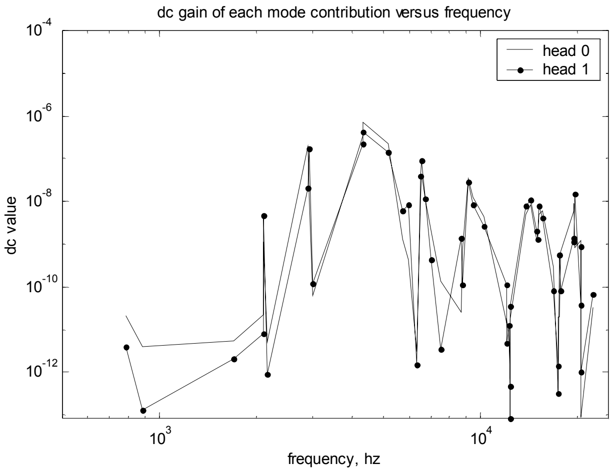

Figure 17.30: dc gain versus frequency.

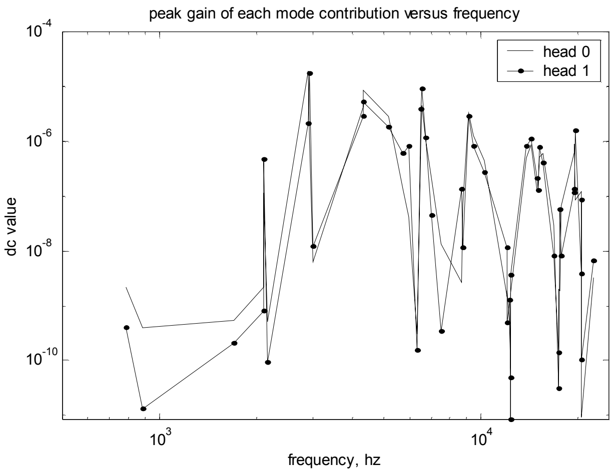

Figure 17.31: Peak gain versus frequency.

The gain plots versus mode number include the rigid body mode low frequency gain, while the gain plots versus frequency do not include the rigid body mode.

Figure 17.32 shows the modes ranked from most to least significant for the uniform damping (dc gain) case and includes the low frequency (500 hz) dc gain of the rigid body mode.

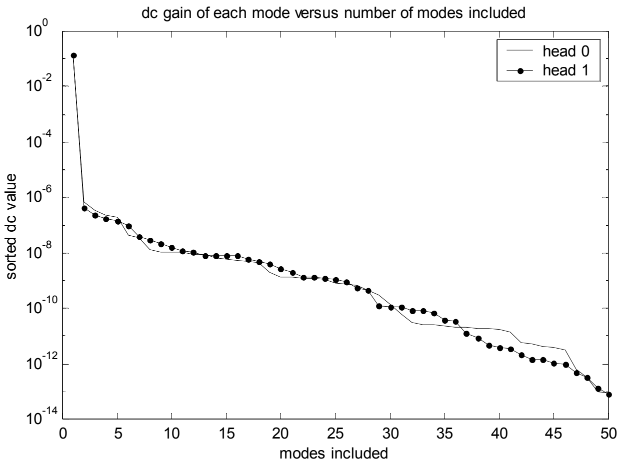

Figure 17.32: Sorted dc gain versus number of modes included.

Relative to the 500 hz low frequency gain of the rigid body mode, the next most significant mode is lower by almost six orders of magnitude. Note that both head 0 and head 1 have similar magnitude curves, although the ordering of individual ranked modes are different. Furthermore, after the drop in dc gain from the rigid body mode to the second mode, there are no other significant drops. Gain is changing gradually, so there is no clear demarcation indicating the number of modes needed to be included. Picking the number of modes to use will be quite subjective, with each additional mode improving the model only slightly.

17.7.5 Building State Space Matrices

To prepare for building the system matrices, two sets of eigenvalue vectors and eigenvector matrices are defined. The first set is the original, unsorted eigenvalues and eigenvectors. The second set consists of the rearranged eigenvalues, eigenvectors and the damping vector, sorted by dc or peak gain. Using the same techniques defined in earlier chapters, the a, b and c matrices are formed.

```
%       define eigenvalues and eigenvectors for unsorted and sorted modes

%       all modes included model, use original order

        xnnew = xn(:,(1:num_modes_total));

        freqnew = freqvec((1:num_modes_total));

        zeta = zeta_unsort;

%       all modes included, sorted

        xnnew_sort = xn(:,index_sort(1:num_modes_total));

        freqnew_sort = freqvec(index_sort(1:num_modes_total));

        zeta_sort = zeta_unsort(index_sort(1:num_modes_total));

%       define variables for all modes included system matrix, a

        w = freqnew*2*pi;               % frequencies in rad/sec

        w2 = w.^2;

        zw = 2*zeta_unsort.*w;

%       define variables for all modes included sorted system matrix, a_sort

        w_sort = freqnew_sort*2*pi;     % frequencies in rad/sec
```

```
        w2_sort = w_sort.^2;

        zw_sort = 2*zeta_sort.*w_sort;

%       define size of system matrix

        asize = 2*num_modes_total;

        disp('   ');
        disp('   ');
        disp(['size of system matrix a is ',num2str(asize)]);

%       setup system matrix for all modes included model

        a = zeros(asize);

        for  col = 2:2:asize

        row = col-1;

        a(row,col) = 1;

        end

        for  col = 1:2:asize

        row = col+1;

        a(row,col) = -w2((col+1)/2);

        end

        for  col = 2:2:asize

        row = col;

        a(row,col) = -zw(col/2);

        end

%       setup system matrix for sorted all modes included model

        a_sort = zeros(asize);

        for  col = 2:2:asize

        row = col-1;

        a_sort(row,col) = 1;

        end

        for  col = 1:2:asize
```

```
            row = col+1;
            a_sort(row,col) = -w2_sort((col+1)/2);
            end
            for  col = 2:2:asize
            row = col;
            a_sort(row,col) = -zw_sort(col/2);
            end
%           setup input matrix b, state space forcing function in principal coordinates
%           now setup the principal force vector for the three cases, all modes, sort
%           f_principal is the vector of forces in principal coordinates
            f_principal = xnnew'*f_physical;
%           b is the vector of forces in principal coordinates, state space form
            b = zeros(2*num_modes_total,1);
            for  cnt = 1:num_modes_total
                        b(2*cnt) = f_principal(cnt);
            end
%           f_principal_sort is the vector of forces in principal coordinates
            f_principal_sort = xnnew_sort'*f_physical;
%           b_sort is the vector of forces in principal coordinates, state space form
            b_sort = zeros(2*num_modes_total,1);
            for  cnt = 1:num_modes_used
                        b_sort(2*cnt) = f_principal_sort(cnt);
            end
%           setup cdisp and cvel, padded xn matrices to give the displacement and velocity
%           vectors in physical coordinates
%           cdisp and cvel each have numdof rows and alternating columns
%           consisting of columns  of xnnew and zeros to give total columns equal
%           to the number of states
%           all modes included cdisp and cvel
            for  col = 1:2:2*length(freqnew)
```

```
        for  row = 1:numdof
            c_disp(row,col) = xnnew(row,ceil(col/2));
            cvel(row,col) = 0;
        end
    end
    for  col = 2:2:2*length(freqnew)
        for  row = 1:numdof
            c_disp(row,col) = 0;
            cvel(row,col) = xnnew(row,col/2);
        end
    end
%   all modes included sorted cdisp and cvel
    for  col = 1:2:2*length(freqnew_sort)
        for  row = 1:numdof
            cdisp_sort(row,col) = xnnew_sort(row,ceil(col/2));
            cvel_sort(row,col) = 0;
        end
    end
    for  col = 2:2:2*length(freqnew_sort)
        for  row = 1:numdof
            cdisp_sort(row,col) = 0;
            cvel_sort(row,col) = xnnew_sort(row,col/2);
        end
    end
%   define output
    d = [0];  %
```

17.7.6 Define State Space Systems, Original and Reduced

Now that the original and sorted state space matrices are available, we can use the "ss" command to define the systems for analysis. The following systems are set up:

1) unsorted model with all modes included

2) sorted model with all modes included

3) sorted, truncated reduced model using the sorted model from 2) above (same as the "modred" "del" option)

4) sorted, "modred" "mdc" option reduction using the sorted model from 2) above

The bode command is used to define magnitude and phase vectors for (1), (3) and (4) above.

In order to see the effects of different servo sample rates on aliasing of high frequency modes, the user is prompted to enter a sample frequency, which defaults to 20 khz. Examples of several sample rates are shown below. A discussion of aliasing is outside the scope of the book but several references are recommended (Franklin 1994 and Franklin 1998).

```
%       define state space systems with the "ss" command, outputs are the
%       two gap displacements

%       define unsorted all modes included system

        sys = ss(a,b,c_disp(7:8,:),d);

%       define sorted all modes included system

        sys_sort = ss(a_sort,b_sort,cdisp_sort(7:8,:),d);

%       define sorted reduced system

        a_sort_red = a_sort(1:num_states_used,1:num_states_used);

        b_sort_red = b_sort(1:num_states_used);

        cdisp_sort_red = cdisp_sort(7:8,1:num_states_used);

        sys_sort_red = ss(a_sort_red,b_sort_red,cdisp_sort_red,d);

%       define modred "mdc" reduced system, modred "del" option same as sorted reduced
               above
```

```
            states_del = (2*num_modes_used+1):2*num_modes_total;

            sys_mdc = modred(sys_sort,states_del,'mdc');

            sys_mdc_nosort = modred(sys,[17:100],'mdc');

%           use "bode" command to generate magnitude/phase vectors

            [mag,phs] = bode(sys,frad);

            [mag_sort_red,phs_sort_red] = bode(sys_sort_red,frad);

            [mag_mdc,phs_mdc]=bode(sys_mdc,frad)  ;

            [mag_mdc_nosort,phs_mdc_nosort]=bode(sys_mdc_nosort,frad)  ;

%           convert magnitude to db

            magdb = 20*log10(mag);

            mag_sort_reddb = 20*log10(mag_sort_red);

            mag_mdcdb = 20*log10(mag_mdc);

%           check on discretized system aliasing

            sample_freq = input('enter sample frequency, khz, default 20 khz ...  ');

            if  isempty(sample_freq)

                        sample_freq = 20;

            end

            nyquist_freq = sample_freq/2;

            disp(['Nyquist frequency is ',num2str(nyquist_freq),' khz']);

            ts = 1/(1000*sample_freq);

            freqdlo = 500;

            freqdhi = 1000*nyquist_freq;       % only take frequency response to nyquist_freq

            fdlo=log10(freqdlo) ;
            fdhi=log10(freqdhi) ;

            fd=logspace(fdlo,fdhi,400) ;
            fdrad=fd*2*pi ;

            sysd = c2d(sys,ts);

            [magd,phsd] = bode(sysd,fdrad);
```

```
magddb = 20*log10(magd);
```

17.7.7 Plotting of Results

The code section below plots the frequency response for the model including all 50 modes and overlaying the individual mode contributions. The sampled frequency response is also plotted, with an overlay of the original 50-mode model response for comparison.

The two reduced models are then plotted, including the individual mode contributions.

The workspace in saved in **act8_data.mat** for use in the **balreal.m** code in Chapter 18.

```
%       start plotting

%       plot all modes included response

        loglog(f,mag(index_out,:),'k.-')
        title([headstr ', gap displacement, all ',num2str(num_modes_total),' modes included'])
        xlabel('Frequency, hz')
        ylabel('Magnitude, mm')
        axis([500 25000 -inf 1e-4])
        grid off
        disp('execution paused to display figure, "enter" to continue'); pause

        hold on

        max_modes_plot = num_modes_total;

        for  pcnt = 1:max_modes_plot

                index = 2*pcnt;

                amode = a(index-1:index,index-1:index);

                bmode = b(index-1:index);

                cmode = c_disp(7:8,index-1:index);

                dmode = [0];

                sys_mode = ss(amode,bmode,cmode,dmode);

                [mag_mode,phs_mode]=bode(sys_mode,frad)  ;

                mag_modedb = 20*log10(mag_mode);
```

```
			loglog(f,mag_mode(index_out,:),'k-')

		end

		axis([500 25000 -inf 1e-4])

		disp('execution paused to display figure, "enter" to continue'); pause

		hold off

		loglog(f,mag(index_out,:),'k-',fd,magd(index_out,:),'k.-')
		title([headstr ', gap displacement, all ',num2str(num_modes_total), ...
			' modes included, Nyquist frequency ',num2str(nyquist_freq),' hz'])
		xlabel('Frequency, hz')
		ylabel('Magnitude, mm')
		legend('continuous','discrete')
		axis([500 25000 1e-8 1e-4])
		grid off

		disp('execution paused to display figure, "enter" to continue'); pause

		if  num_modes_used < num_modes_total     % calculate and plot reduced models

%		sorted modal truncation

		loglog(f,mag(index_out,:),'k-',f,mag_sort_red(index_out,:),'k.-')
		title([headstr ', sorted modal truncation:  gap displacement, first ', ...
			num2str(num_modes_used),' modes included'])
		legend('all modes','sorted partial modes',3)
		xlabel('Frequency, hz')
		ylabel('Magnitude, mm')
		axis([500 25000 1e-8 1e-4])
		grid off

		disp('execution paused to display figure, "enter" to continue'); pause

		hold on

		for  pcnt = 1:max_modes_plot

			index = 2*pcnt;

			amode = a_sort(index-1:index,index-1:index);

			bmode = b_sort(index-1:index);

			cmode = cdisp_sort(7:8,index-1:index);

			dmode = [0];

			sys_mode = ss(amode,bmode,cmode,dmode);

			[mag_mode,phs_mode]=bode(sys_mode,frad)  ;
```

```
			loglog(f,mag_mode(index_out,:),'k-')

		end

		axis([500 25000 -inf 1e-4])

		disp('execution paused to display figure, "enter" to continue'); pause

		hold off

%		modred using 'mdc'

		loglog(f,mag(index_out,:),'k-',f,mag_mdc(index_out,:),'k.-')
		title([headstr ', reduced matched dc gain:  gap displacement, first ', ...
			num2str(num_modes_used),' sorted modes included'])
		legend('all modes','reduced mdc',3)
		xlabel('Frequency, hz')
		ylabel('Magnitude, mm')
		axis([500 25000 1e-8 1e-4])
		grid off

		disp('execution paused to display figure, "enter" to continue'); pause

		hold on

		for  pcnt = 1:max_modes_plot

			index = 2*pcnt;

			amode = a_sort(index-1:index,index-1:index);

			bmode = b_sort(index-1:index);

			cmode = cdisp_sort(7:8,index-1:index);

			dmode = [0];

			sys_mode = ss(amode,bmode,cmode,dmode);

			[mag_mode,phs_mode]=bode(sys_mode,frad)  ;

			loglog(f,mag_mode(index_out,:),'k-')

		end

		axis([500 25000 -inf 1e-4])

		disp('execution paused to display figure, "enter" to continue'); pause

		hold off

%		modred using 'mdc' with unsorted modes

		loglog(f,mag(index_out,:),'k-',f,mag_mdc_nosort(index_out,:),'k.-')
		title([headstr ', reduced unsorted matched dc gain:  gap displacement, first ', ...
```

```
                    num2str(num_modes_used),' sorted modes included'])
          legend('all modes','reduced mdc',3)
          xlabel('Frequency, hz')
          ylabel('Magnitude, mm')
          axis([500 25000 1e-8 1e-4])
          grid off

          disp('execution paused to display figure, "enter" to continue'); pause

          hold on

          for  pcnt = 1:num_modes_used

                    index = 2*pcnt;

                    amode = a(index-1:index,index-1:index);

                    bmode = b(index-1:index);

                    cmode = c_disp(7:8,index-1:index);

                    dmode = [0];

                    sys_mode = ss(amode,bmode,cmode,dmode);

                    [mag_mode,phs_mode]=bode(sys_mode,frad)  ;

                    loglog(f,mag_mode(index_out,:),'k-')

          end

          axis([500 25000 -inf 1e-4])

          disp('execution paused to display figure, "enter" to continue'); pause

          hold off

          end

%         save the workspace for use in balred.m

          save act8_data
```

Plots using the code above are discussed in the following sections.

17.8 Uniform and Non-Uniform Damping Comparison

The four figures below show a comparison between the uniform and non-uniform damping cases. The first two depict uniform damping, while the second two show non-uniform damping, with higher damping for modes 11, 12 and 13.

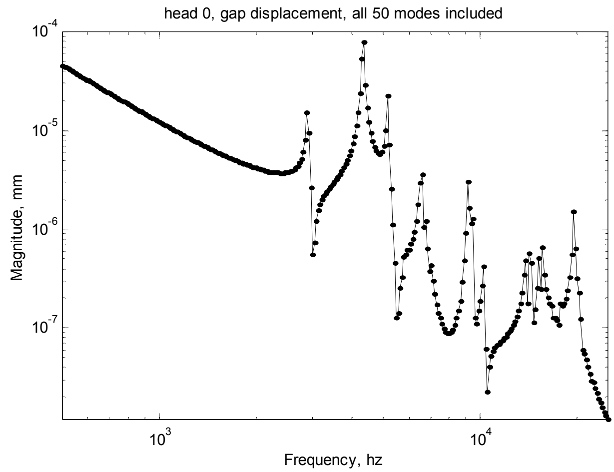

Figure 17.33: Head 0 frequency response, all 50 modes included, uniform damping with zeta = 0.005.

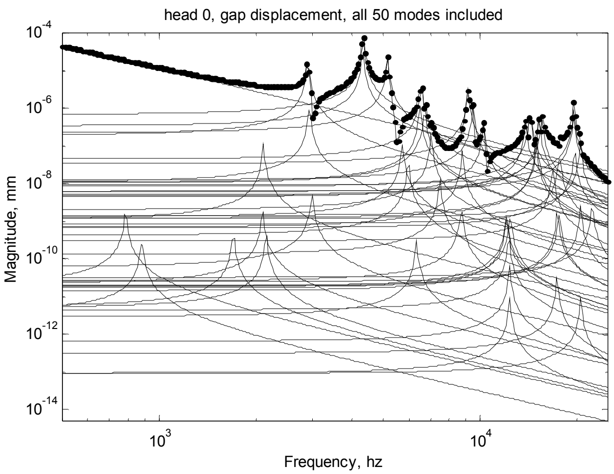

Figure 17.34: Head 0 frequency response, overlay of individual mode contributions, 50 modes included, uniform damping with zeta = 0.005.

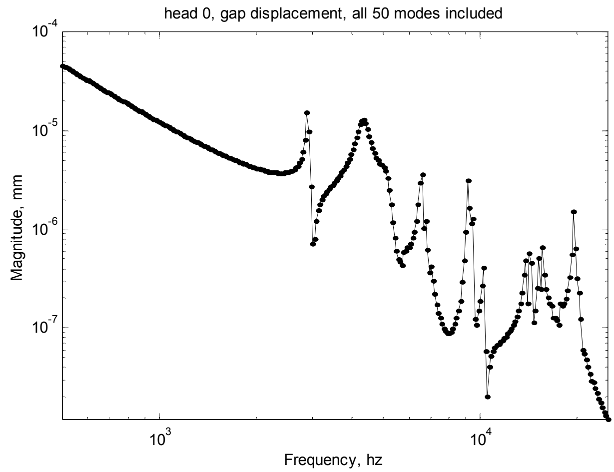

Figure 17.35: Head 0 frequency response, all 50 modes included, non-uniform damping with zeta = 0.005 for all modes except modes 11, 12 and 13, which have zeta = 0.04.

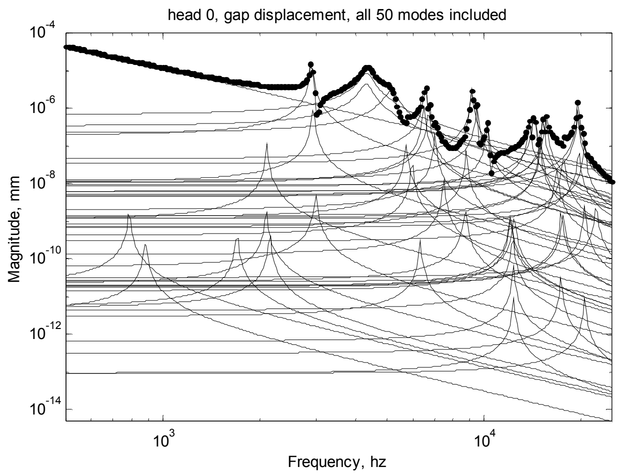

Figure 17.36: Head 0 frequency response, overlay of individual mode contributions, 50 modes included, non-uniform damping with zeta = 0.005 for all modes except modes 11, 12 and 13, which have zeta = 0.04.

Note the lower gain of the three modes in the 4 to 5.5 khz range for the non-uniform damping case.

17.9 Sample Rate and Aliasing Effects

In the two figures below we can see the effects of aliasing for two different servo system sample rates.

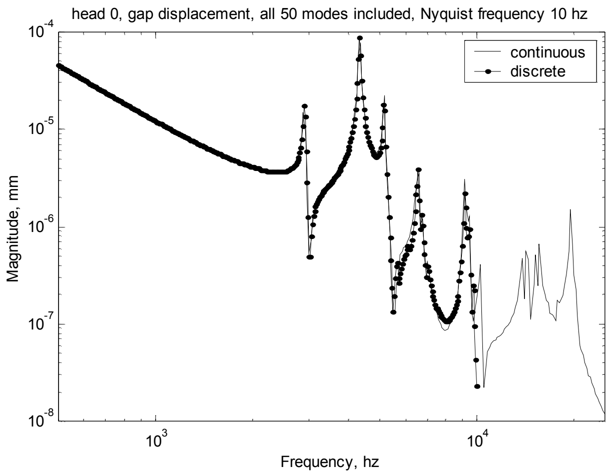

Figure 17.37: Discrete system frequency response overlaid on continuous system, sample rate 20 khz, Nyquist frequency 10 khz.

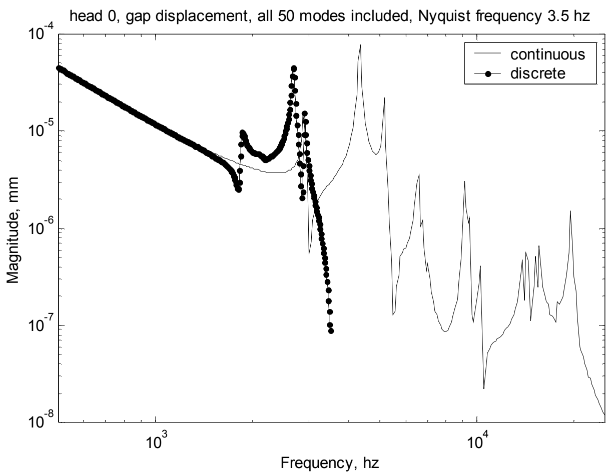

Figure 17.38: Discrete system frequency response overlaid on continuous system, sample rate 7 khz, Nyquist frequency 3.5 khz, showing aliasing effects.

The discrete system frequency response in Figure 17.37, which has a sample frequency of 20 khz, shows only small differences from the original continuous system response. The discrete system response stops at the Nyquist frequency, 10 khz.

Unlike Figure 17.37, Figure 17.38, which has a much lower sample rate of 7 khz, shows a significant difference from the original continuous system. If one uses the sampled system to experimentally measure the frequency response, it can only measure the response in the 0-Nyquist frequency range. If the discrete system shown in Figure 17.33 were measured, there would be no way to know that the peak at 2.68 khz is not an actual mechanical resonance at 2.68 khz but is the system mode at 4.32 khz which is aliased. As mentioned earlier, only a measurement using a separate system, such as a laser measurement system, will reveal the actual mechanical system response.

17.10 Reduced Truncation and Matched dc Gain Results

This section compares sorted reduced truncation and sorted match dc gain (mdc) methods, both using eight modes.

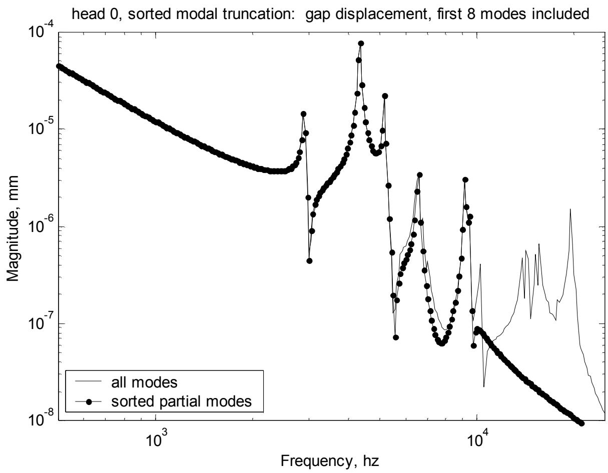

Figure 17.39: Reduced sorted modal truncation frequency response, eight modes included.

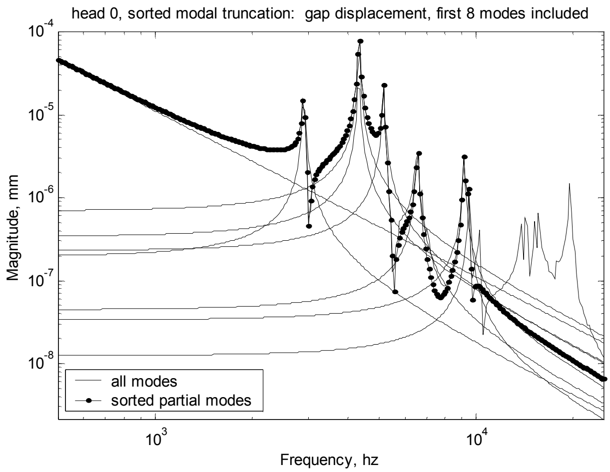

Figure 17.40: Reduced sorted modal truncation frequency response, eight modes included, showing overlay of eight individual modes.

The reduced sorted truncated system shown in Figures 17.37 and 17.38 matches the original 50-mode system frequency response quite well in the 0 to 10 khz range, but misses four modes between 10 and 20 khz.

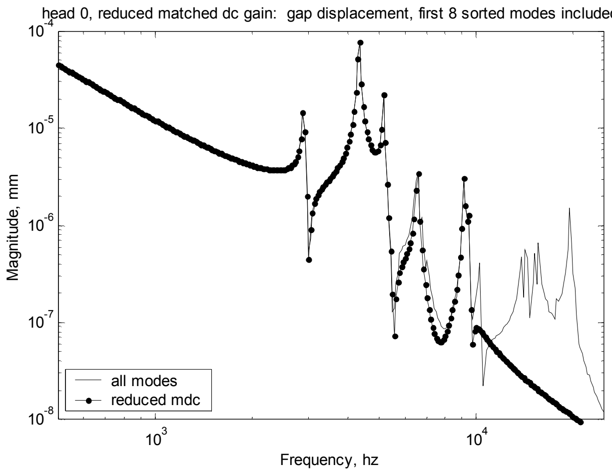

Figure 17.41: Reduced "modred" matched dc gain frequency response, eight modes included.

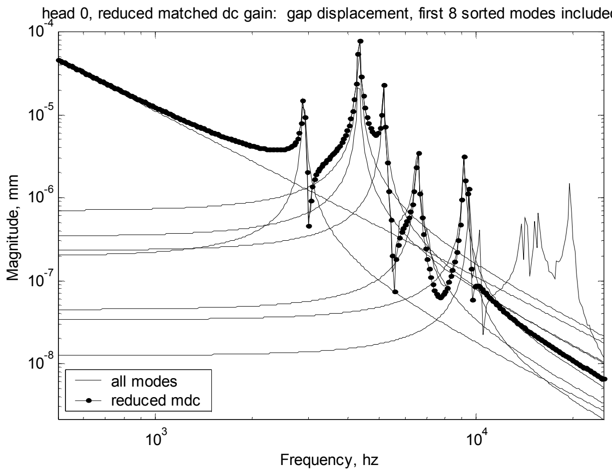

Figure 17.42: Reduced "modred" matched dc gain frequency response, eight modes included, showing overlay of eight individual modes.

The reduced "modred" matched dc (mdc) gain frequency response is virtually identical to the reduced sorted modal truncation response because the modes were sorted prior to using the matched method and the modes which were eliminated have low dc gain relative to the rigid body gain. Also, since the eliminated modes have such a small contribution to the overall response, the "flat" high frequency portion of the curve (highlighted in Figures 15.15 and 16.17) is not seen. To be sure that this was the case, the "modred" matched dc gain reduction was run on the system with unsorted modes, using the first eight modes. The results are shown below and show that the "flat" high frequency portion of the frequency response has returned.

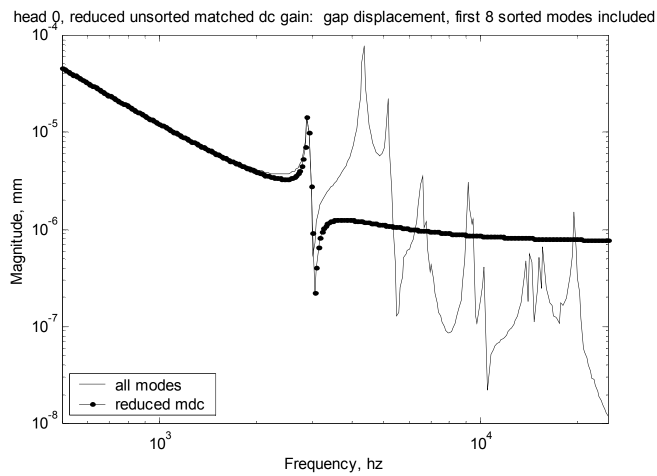

Figure 17.43: Unsorted Reduced "modred" matched dc gain frequency response, first eight unsorted modes included.

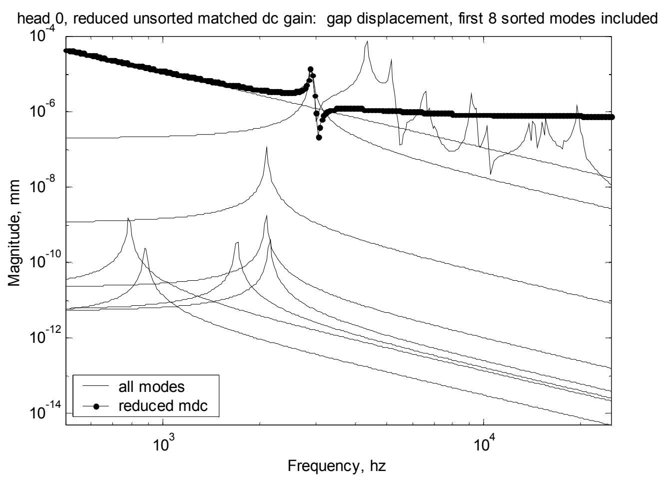

Figure 17.43: Unsorted Reduced "modred" matched dc gain frequency response, first eight unsorted modes included, showing overlay of eight individual modes.

Only eight modes were used for the reduced frequency responses in this chapter. In Chapter 18 we will compare responses for different number of reduced modes to get a sense for how many modes are required to define the pertinent dynamics.