ME 610 – HW3

Effective Mass and Effective Independence

Victor Cavalcanti

OBJECTIVE

The objective of this assignment is to perform a pre-test analysis of a 156 DOF General Purpose Spacecraft. The goal is to use techniques such as *Effective Mass, Modal Kinetic Energy and Effective Independence to* provide a test engineer with enough information on where to place sensors, and which modes they strive to observe.

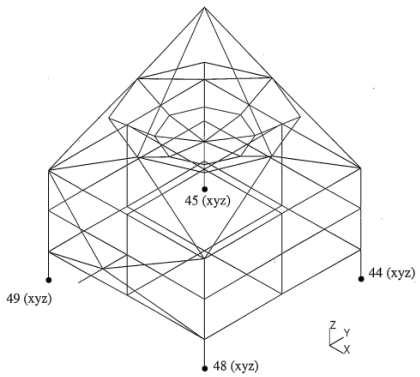

Figure 1- Simple rendering of 156-DOF GPSC. The fixed DOF are labeled.

PROCEDURE

The three main tools used in this analysis are:

Effective Mass – Used to rank and select a set of fixed-interface modes that should be dynamically complete. These modes are usually the ones sought after in vibration tests.

Modal Kinetic Energy and *Effective Independence – Used to* select a reduced set of DOF that can accurately represent the target modes. One of the criteria being that these reduced modes should be as linearly independent as possible and that they will provide the best input to noise ratio.

The analysis will begin by using *Effective Mass* to narrow the fixed interface modes to a smaller set. The requirement for this set is that each mode contributes at least **4.8%**[1] of the effective mass in any rigid body direction.

[1] Since in class you mentioned the limit of 5% to be flexible, there was a single mode at 4.81% mass in I_{xx} which I chose to include.

Effective Mass

The procedure for *Effective Mass* begins by calculating the rigid body modes and the rigid body mass matrix. The modes should be calculated about a node that contains all 6 degrees of freedom (3 translations and 3 rotations). Ideally the node will be located as close as possible to the center of mass or at the interface of the structure.

To calculate the rigid body modes, I reordered the DOF as shown below, where u_a are the DOF about which the modes will be calculated, and u_d are all other DOF. The mass and stiffness matrices are also partitioned accordingly.

$$u = \begin{Bmatrix} u_d \\ u_a \end{Bmatrix} \qquad K = \begin{bmatrix} K_{dd} & K_{da} \\ K_{ad} & K_{aa} \end{bmatrix} \qquad M = \begin{bmatrix} M_{dd} & M_{da} \\ M_{ad} & M_{aa} \end{bmatrix}$$

The rigid body modes are then simply:

$$\phi_R = \begin{bmatrix} -\bar{K_{dd}} K_{da} \\ I_{N_a x N_a} \end{bmatrix} = \begin{bmatrix} \phi_d \\ \phi_a \end{bmatrix} \tag{1}$$

And the rigid body mass matrix is (the form of the matrix assumes calculation about COM):

$$M_R = {\phi_R}^T M \phi_R = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & I_{xy} & I_{xz} \\ 0 & 0 & 0 & I_{yx} & I_{yy} & I_{yz} \\ 0 & 0 & 0 & I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{2}$$

For this particular case, I statically reduced out the rotational DOF from the mass and stiffness matrices after calculating the rigid body modes. The procedure for static reduction can be found in **APPENDIX A.[2] I also crossed out the rows of the rotational DOF in the rigid body modes. I then repartitioned my DOF, M and K and ϕ_R. Where $u_o - Interior\ DOF, u_a - Interface\ DOF$**

$$u = \begin{Bmatrix} u_a \\ u_o \end{Bmatrix} \qquad K = \begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \qquad M = \begin{bmatrix} M_{aa} & M_{ao} \\ M_{oa} & M_{oo} \end{bmatrix}$$

The o-set partition of the rigid body matrix is simply:

$$M_{Ro} = \phi_{Ro}^T M_{oo} \phi_{Ro} \tag{3}$$

Effective mass only works for fixed structures, so we must calculate the fixed interface modes ϕ. Those stem from the eigenvalue problem, where the rows and columns of the DOF which are fixed have been crossed out. The effective mass is then simply:

$$E = [\phi^T M_{oo} \phi_{Ro}].^2 \tag{4}$$

[2] Although you have noted previously that procedures should be included in the body of the report, for the sake of brevity and clarity, I choose to append procedures that have been routinely covered/used.

And the fractional contribution is then:

$$\bar{E} = [\phi^T M_{oo} \phi_{Ro}].^2 \, [diag(M_{Ro})]^{-1} \tag{5}$$

Where ϕ are the fixed-interface modes, and $.^2$ means a term-by-term squaring. The columns of E sum to the diagonal terms of the rigid body mass matrix, M_{Ro}, hence the rows of $\bar{E}$ gives the fractional contribution of that specific mode to each of the rigid body mass directions.

Modal Kinetic Energy

The procedure for Modal Kinetic begins with the general expression for KE:

$$2KE = \dot{u}^T M \dot{u} \tag{6}$$
$$2KE = \dot{q}^T \phi^T M \phi \dot{q} \tag{7}$$

If we are only interested in the dependence of the kinetic energy on the modes shapes, we must normalize the contributions of our modal velocities ($\dot{q}_i^2 = 1$).

We can then represent the fractional contribution of all DOFs for each mode as:

$$KE_{ij} = \phi_{ij} M_j \phi_i \tag{8}$$

Where:

$\phi_{ij} = jth\ row\ of\ ith\ mode$

$M_j = jth\ row\ of\ mass\ matrix$

$\phi_i = ith\ mode$

A check for the correctness of this computation would require that all the entries of each kinetic energy vector ($\boldsymbol{KE_i}$) sum up to 1. A final measure of the fractional importance of each DOF to the system would be to average the Kinetic Energy over all modes used (n_m):

$$KE = \frac{1}{n_m} \sum_{i=1}^{n_m} KE_i \tag{9}$$

Effective Independence

This method is used to generate a reduced set of DOF which can be used as an indication of proper sensor placement. This process will output a set of target DOF, whose reduced eigenproblem will yield mode shapes that are as linearly independent as possible. The idea stems from maximizing the fisher information matrix Q.

$$Q = \sum {\phi_{fs}^{i}}^{T} \phi_{fs}^{i} = \sum Q^{i} \tag{10}$$

Where here, ϕ_{fs}^{i} is the ith row of the target fixed-interface mode partition (associated to the ith candidate dof). The next step is to solve the eigenproblem:

$$|Q - \lambda I|\boldsymbol{\psi} = 0 \tag{11}$$

The eigenvectors $\boldsymbol{\psi}$ should yield these relations:

$$\boldsymbol{\psi}^{T} Q \boldsymbol{\psi} = \lambda \qquad \& \qquad \boldsymbol{\psi}^{T} \boldsymbol{\psi} = I \tag{12}$$

We then pre-multiply our eigenvectors by the kept fixed-interface modes and take the term-by-term square of the matrix to yield the absolute identification space:

$$G = (\phi_{fs}\psi).^{2} \tag{13}$$

Finally, we post-multiply by the inverse of the eigenvalue matrix λ, and sum up all the terms in the rows of the matrix (by post multiplying by a column vector of 1s, to yield the effective independence distribution. This is a measure of how necessary each of the DOF are to the linear independence of the target modes ϕ_{fs}. The sum of all the terms of this vector should add up to the rank of ϕ_{fs}.

$$E_{D=(\phi_{fs}\psi).^{2}\lambda^{-1}\{1\}_{n_m}} \tag{14}$$

RESULTS

The first part of the analysis was to use *Effective Mass* to identify a set of target modes. The following function was used to select the target modes (**APPENDIX C)**:

```
function [E,MR, MRo, M, K, dofA, dofO] = getEffectiveMass(M, K, PHI, DOFint,
DOF, PHIR)
```

The 30 Fixed-Interface modes provided the following total percentage of the rigid body mass matrix:

Rigid Body Direction	M_x	M_y	M_z	I_{xx}	I_{yy}	I_{zz}
% Total Mass	99.99%	99.99%	83.4%	83.87%	80.69%	99.99%

Table 1- Percentage the Fixed-Interface Modes contribute to the rigid body mass matrix

Effective mass identified the following **target modes** based on the criteria that the **mode should contribute at least 4.8% of the rigid body mass in any direction.** There were a total of 14 modes**:**

Mode #	1	3	4	5	6	8	9	12	13	18	19	20	21	22

Table 2- Set of target Fixed-Interface modes.

These **fixed-interface target modes** added up to the following percentage of the rigid body mass matrix:

Rigid body DIrection	M_x	M_y	M_z	I_{xx}	I_{yy}	I_{zz}
% Total Mass	99.93%	96.47%	80.41%	73%	74.92%	99.99%
% Difference	0.06%	3.52%	3.59%	12.96%	7.15%	0.00%

Table 3- Percentage the *TARGET* Fixed-Interface Modes contribute to the rigid body mass matrix

The reduced mode set is dynamically complete for motion in the x-y plane. Motions in the other planes should still be complete enough for analysis. The reduced set seems to capture most of the mass, as can be seen from above chart. The % differences between the original provided fixed-interface modes and the target set of fixed-interface modes is <5% for every category except for the inertias about the x and y axis.

The second part of this analysis was to use Modal Kinetic Energy and Effective Independence to select 19 DOF (**# target modes+5)** for sensor placement. **This process of selecting the candidate DOF set began with 57 of the most dynamically important DOF (3*# target modes+5).** The following list shows the final set of 19 DOF selected by EFI, as well as the 19 DOF selected purely by Modal Kinetic Energy. The following function was used (**APPENDIX C)**:

```
function [Efi,DOF,xtraDOF] = getEffectiveIndependence(PHI,M,K,ntargetdof,DOF)
```

EFI-DOF	1.1	11.3	12.3	13.2	13.3	14.2	17.3	18.1	18.2	18.3	35.1	40.1	40.3	42.1	42.3	43.3	46.3	47.1	50.3
KE-DOF	11.3	12.3	13.3	17.3	18.1	18.2	18.3	40.1	40.2	40.3	42.1	42.3	43.3	46.2	46.3	47.1	47.3	50.1	50.3

Table 4- List of DOF selected by EFI and Modal Kinetic Energy.

The third part of this analysis involved statically reducing the fixed system to the selected DOF and computing their respective modes. Correlation techniques were used to compare the computed modes to the partitioned fixed-interface modes provided. The modes were matched using the Modal Assurance Criterion (MAC) and Cross-Orthogonality (CO). These procedures can be found in **APPENDIX A.** The numerical values can be found in **APPENDIX B.**

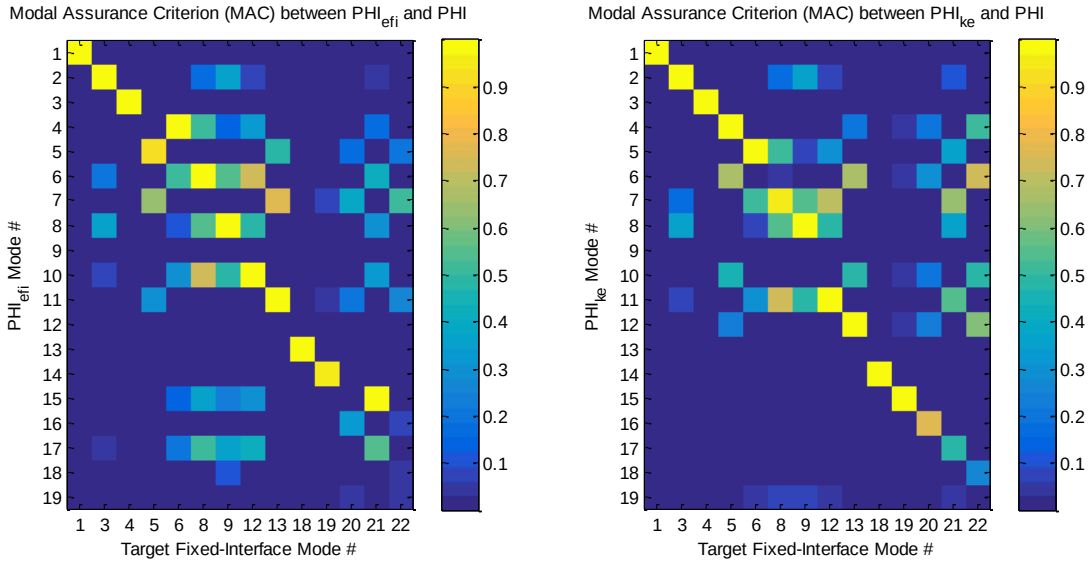

Figure 2-Mac values for the Effective Independence mode set (left) and the Modal Kinetic Energy set (right).

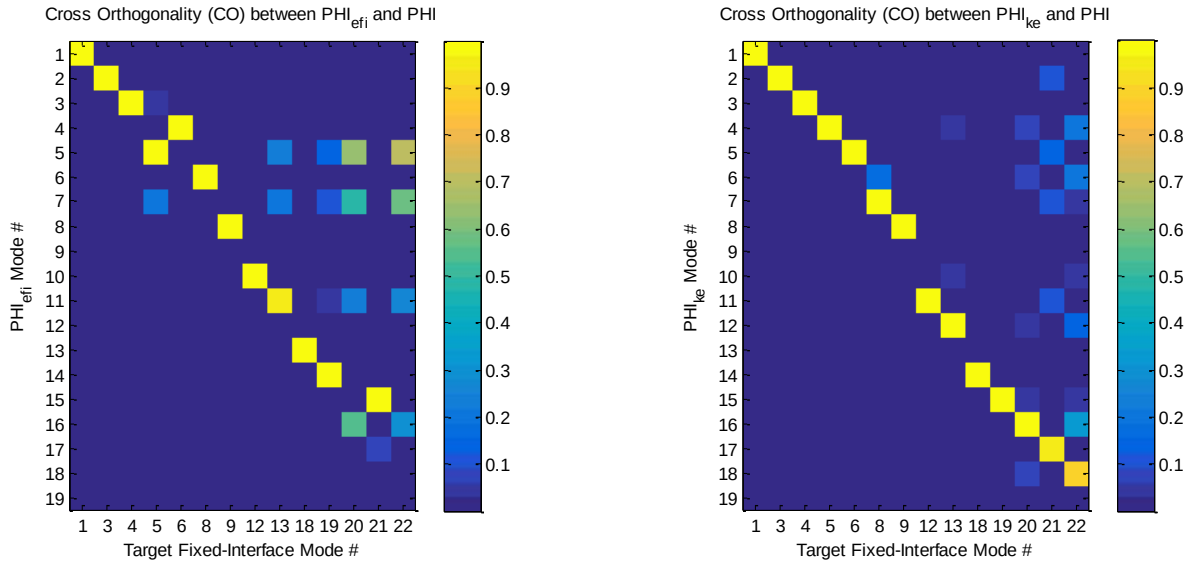

Figure 3-Cross-Orthogonality values values for the Effective Independence mode set (left) and the Modal Kinetic Energy set (right).

The following table relates how I believe the mode sets should be matched:

Test Mode #	Efi Mode #	KE Mode #
1	1	1
3	2	2
4	3	3
5	5	4
6	4	5
8	6	7
9	8	8
12	10	11
13	11	12
18	13	14
19	14	15
20	--	16
21	15	17
22	--	18

Table 5-Table showing how the modes between Effective Independence and Modal Kinetic Energy should be matched to the Test modes.

And the frequencies (in Hertz) along with their errors for these matched modes are shown below:

Test Mode #	$\omega_n - Test$	$\omega_n - Efi$	% Error	$\omega_n - KE$	% Error
1	13.96	13.96	-0.01	13.96	0.00
3	17.59	17.61	-0.11	17.60	-0.08
4	17.65	17.65	-0.01	17.65	0.00
5	18.71	20.83	-11.33	18.88	-0.91
6	20.80	21.34	-2.59	20.84	-0.15
8	25.70	25.72	-0.08	25.74	-0.14
9	30.03	30.19	-0.55	30.20	-0.59
12	43.97	44.04	-0.16	44.11	-0.32
13	44.72	45.84	-2.49	44.93	-0.47
18	78.34	78.71	-0.47	78.72	-0.49
19	102.51	102.47	0.04	102.42	0.09
20	131.89	--	--	132.41	-0.40
21	134.62	135.59	-0.72	137.98	-2.50
22	141.51	--	--	144.02	-1.78

Table 6-Tabulated Frequencies and Frequency errors.

The self-MAC values for the two sets of modes are as follows. The numerical values can be found in **APPENDIX B**:

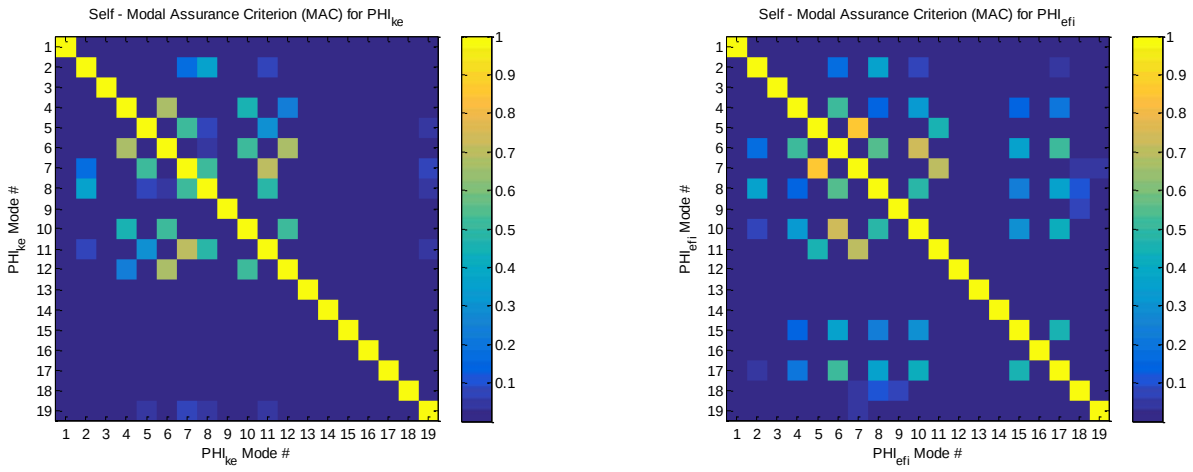

Figure 4-Self-MAC values for the Effective Independence mode set (left) and the Modal Kinetic Energy set (right).

The self-MAC shows a measure of the spatial orthogonality of the modes. The Self-MAC for **the Effective Independence has 40 off-diagonal** entries which are >0.1, while the **Modal Kinetic Energy has 24.**

The signal-noise ratio is related to the determinant of the Fisher Information Matrix. A plot of this measure for both Effective Independence and Kinetic Energy is shown below. The x-axis represents the iteration number as DOF were being removed from the final sensor set.

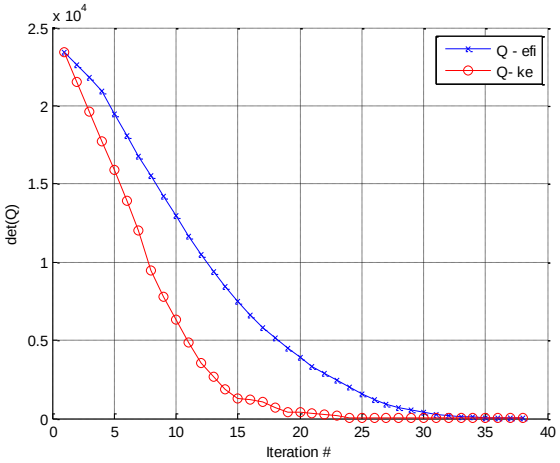

Figure 5-The determinant of the Fisher information matrix for each iteration of the d

Both these DOF sets seem comparable. The difference in determinant of the Fisher Information Matrix seems small when the set is so reduced, and the self-MAC between both modes is relatively comparable, although the MKE has fewer high valued off-diagonal entries. The Modal Kinetic Energy set seems to correlate to more modes, with lower frequency error. In terms of signal-strength, Effective Independence might provide a better DOF set, but it doesn't seem to be by much.

APPENDIX A

Static Reduction

To statically reduce a system, begin by partitioning your DOF vector into the DOF you wish to keep, and those you wish to reduce out of your system: ($u_a -$ $Kept\ DOF$, $u_d -$ $Condensed\ DOF$)

$$u = \begin{Bmatrix} u_a \\ u_d \end{Bmatrix}$$

We assume the DOF being reduced possess little mass and that no load is applied to them. The EOM for an undamped system with this new partitioned DOF vector, it will look as follows:

$$\underset{M-sorted}{\begin{bmatrix} M_{aa} & M_{ad} \\ M_{da} & M_{dd} \end{bmatrix}} \begin{Bmatrix} \ddot{u}_a \\ \ddot{u}_d \end{Bmatrix} + \underset{K-sorted}{\begin{bmatrix} K_{aa} & K_{ad} \\ K_{da} & K_{dd} \end{bmatrix}} \begin{Bmatrix} u_a \\ u_d \end{Bmatrix} = \begin{Bmatrix} P_a \\ 0 \end{Bmatrix}$$

Working with the second equation, we can derive the following Ritz transformation matrix:

$$T = [\ I_{N_a x N_a}\ ;\ -K_{dd}^{-1}\ K_{da}\ ]$$

Which, when applied to our sorted Mass and Stiffness matrices, will yield the matrices for our reduced system:

$$\widehat{M} = T^T M T$$
$$\widehat{K} = T^T K T$$

MAC, SELF AND CROSS-ORTHOGONALITY

Modal comparison between FEM and test modes can be made by computing the Modal Assurance Criterion (MAC). MAC stems from the inequality of vector products:

$$|\phi^T \phi| \leq |\phi||\phi|$$
$$\frac{|\phi^T \phi|}{|\phi||\phi|} \leq 1$$

Which, in our case, we are working with the squares of the modes, so our expression becomes:

$$0 \leq \frac{(\phi_{FEM}^T \phi_{TEST})^2}{(\phi_{FEM}^T \phi_{FEM})(\phi_{TEST}^T \phi_{TEST})} \leq 1$$

Where a value of 0 indicates that modes are orthogonal and a value of 1 indicates modes are parallel.

After modes have been matched by using MAC, a mass-weighted cross and self-orthogonality test will can done. The expressions used will be:

$$\phi_{TEST}^T M \phi_{TEST} \quad \& \quad \phi_{TEST}^T M \phi_{FEM}$$

In an ideal world, the self-orthogonality check for test modes will yield an identity; however, this will rarely be the case. In the cross-orthogonality computation, the criteria for ideal mode matching is to have diagonal values of ≥ 0.9 and off-diagonal values of ≤ 0.1.

Frequency errors will be computed between the FEM and TEST frequencies (in Hertz), by using the formula:

$$\frac{\omega_{FEM} - \omega_{TEST}}{\omega_{TEST}} * 100$$

APPENDIX B – Numerical Values for MAC, SO, CO

Cross-Orthogonality for Effective Independence. The Rows correspond to the EFI and KE set modes, while the Columns correspond to the target modes.

EFI Cross-Orthogonality

1.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.004	0.000	0.000	0.005	0.000	0.002
0.000	1.000	0.001	0.002	0.008	0.004	0.007	0.010	0.001	0.000	0.001	0.001	0.007	0.000
0.000	0.001	1.000	0.057	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.011	0.000	0.014
0.000	0.004	0.000	0.000	1.000	0.006	0.000	0.009	0.000	0.000	0.001	0.000	0.003	0.002
0.000	0.001	0.023	0.978	0.000	0.002	0.004	0.002	0.221	0.013	0.128	0.627	0.001	0.712
0.000	0.001	0.000	0.000	0.003	1.000	0.000	0.005	0.001	0.000	0.001	0.002	0.008	0.001
0.000	0.000	0.006	0.199	0.000	0.003	0.008	0.002	0.204	0.008	0.102	0.499	0.000	0.565
0.000	0.001	0.000	0.002	0.000	0.000	1.000	0.008	0.004	0.001	0.002	0.004	0.009	0.004
0.000	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.010	0.022	0.001	0.001	0.000	0.002
0.000	0.000	0.000	0.000	0.000	0.000	0.001	1.000	0.010	0.000	0.000	0.003	0.001	0.004
0.000	0.000	0.001	0.022	0.000	0.000	0.001	0.009	0.954	0.007	0.054	0.242	0.001	0.275
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	1.000	0.000	0.008	0.000	0.001
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.985	0.025	0.000	0.025
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.996	0.001
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.007	0.546	0.003	0.312
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.087	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.009	0.000	0.010
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.003	0.000	0.005

KE Cross-Orthogonality

1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
0.000	1.000	0.006	0.006	0.007	0.001	0.005	0.012	0.002	0.000	0.000	0.000	0.119	0.001
0.000	0.006	1.000	0.008	0.000	0.000	0.001	0.000	0.002	0.000	0.001	0.004	0.000	0.012
0.000	0.005	0.006	1.000	0.014	0.005	0.024	0.002	0.048	0.000	0.010	0.070	0.004	0.199
0.000	0.003	0.000	0.009	1.000	0.000	0.005	0.013	0.006	0.000	0.000	0.001	0.133	0.002
0.000	0.000	0.000	0.004	0.001	0.187	0.014	0.000	0.008	0.000	0.010	0.063	0.025	0.199
0.000	0.000	0.000	0.002	0.000	0.982	0.009	0.003	0.005	0.000	0.002	0.012	0.123	0.038
0.000	0.001	0.000	0.004	0.001	0.003	1.000	0.007	0.022	0.000	0.001	0.001	0.009	0.003
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.023	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.001	0.000	0.000	0.003	0.002	0.060	0.000	0.013	0.003	0.001	0.059
0.000	0.000	0.000	0.000	0.000	0.001	0.001	1.000	0.012	0.000	0.000	0.001	0.120	0.002
0.000	0.000	0.000	0.001	0.000	0.000	0.004	0.011	0.997	0.000	0.003	0.059	0.001	0.147
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.001
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	1.000	0.041	0.000	0.061
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.012	0.990	0.000	0.334
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.968	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.069	0.000	0.883
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.017	0.000

Cross-Modal Assurance Criterion, The rows represent the EFI and KE modes while the columns represent the target modes.

EFI - Cross MAC													
1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	1.000	0.000	0.000	0.001	0.187	0.374	0.079	0.000	0.000	0.000	0.000	0.033	0.000
0.000	0.000	1.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	1.000	0.526	0.125	0.315	0.000	0.000	0.000	0.001	0.162	0.001
0.000	0.000	0.001	0.934	0.000	0.000	0.000	0.000	0.494	0.000	0.028	0.171	0.000	0.216
0.000	0.188	0.000	0.001	0.515	1.000	0.545	0.735	0.000	0.000	0.000	0.001	0.418	0.002
0.000	0.000	0.001	0.629	0.000	0.000	0.000	0.000	0.762	0.000	0.063	0.396	0.000	0.503
0.000	0.366	0.000	0.001	0.122	0.547	0.999	0.481	0.001	0.000	0.000	0.001	0.289	0.001
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.017	0.000	0.003	0.000	0.001
0.000	0.080	0.000	0.000	0.309	0.733	0.475	1.000	0.000	0.000	0.000	0.001	0.335	0.001
0.000	0.000	0.000	0.283	0.000	0.000	0.000	0.000	0.991	0.000	0.035	0.207	0.000	0.261
0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.001	0.000	0.002
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.994	0.002	0.000	0.000	0.002
0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.009	0.002	0.957	0.002	0.000	0.015
0.000	0.028	0.000	0.001	0.135	0.355	0.244	0.287	0.000	0.000	0.000	0.002	0.989	0.003
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.320	0.000	0.087
0.000	0.038	0.000	0.001	0.189	0.510	0.350	0.408	0.000	0.000	0.000	0.003	0.557	0.004
0.000	0.000	0.000	0.014	0.023	0.015	0.112	0.011	0.025	0.004	0.003	0.027	0.015	0.034
0.000	0.006	0.000	0.003	0.015	0.000	0.017	0.000	0.023	0.000	0.003	0.046	0.000	0.059

KE - Cross MAC													
1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	1.000	0.000	0.000	0.009	0.171	0.351	0.065	0.000	0.000	0.000	0.000	0.113	0.000
0.000	0.000	1.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001
0.000	0.000	0.000	1.000	0.001	0.000	0.005	0.000	0.208	0.000	0.036	0.205	0.000	0.517
0.000	0.007	0.000	0.000	1.000	0.526	0.091	0.299	0.000	0.000	0.000	0.000	0.350	0.000
0.000	0.006	0.001	0.661	0.017	0.033	0.017	0.024	0.660	0.000	0.053	0.288	0.022	0.738
0.000	0.169	0.000	0.025	0.504	0.964	0.553	0.703	0.025	0.000	0.002	0.011	0.628	0.028
0.000	0.348	0.000	0.011	0.086	0.559	0.996	0.483	0.004	0.000	0.000	0.001	0.373	0.003
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.016	0.000	0.000	0.000	0.000
0.000	0.000	0.001	0.445	0.000	0.000	0.000	0.000	0.488	0.000	0.037	0.197	0.000	0.497
0.000	0.068	0.000	0.000	0.303	0.735	0.491	1.000	0.000	0.000	0.000	0.000	0.556	0.000
0.000	0.000	0.000	0.223	0.000	0.000	0.000	0.000	0.998	0.000	0.044	0.238	0.000	0.606
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.994	0.001	0.000	0.000	0.000
0.000	0.000	0.000	0.008	0.000	0.000	0.000	0.000	0.012	0.001	0.985	0.000	0.000	0.012
0.000	0.000	0.000	0.007	0.000	0.000	0.000	0.000	0.006	0.000	0.001	0.774	0.000	0.006
0.000	0.004	0.000	0.000	0.013	0.024	0.014	0.021	0.000	0.000	0.000	0.000	0.487	0.000
0.000	0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.003	0.000	0.004	0.009	0.000	0.251
0.000	0.006	0.000	0.000	0.031	0.075	0.063	0.055	0.000	0.000	0.000	0.000	0.053	0.000

Self-Modal Assurance Criterion. The columns and rows represent the EFI and KE modes.

Self MAC EFI

1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	1.000	0.000	0.000	0.000	0.187	0.000	0.366	0.000	0.079	0.000	0.000	0.000	0.000	0.028	0.000	0.038	0.000	0.006
0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	1.000	0.000	0.525	0.000	0.128	0.000	0.316	0.000	0.000	0.000	0.000	0.137	0.000	0.193	0.023	0.015
0.000	0.000	0.000	0.000	1.000	0.000	0.848	0.001	0.000	0.000	0.461	0.000	0.000	0.005	0.000	0.000	0.000	0.023	0.013
0.000	0.187	0.000	0.525	0.000	1.000	0.000	0.547	0.000	0.734	0.000	0.000	0.000	0.000	0.356	0.000	0.510	0.015	0.000
0.000	0.000	0.000	0.000	0.848	0.000	1.000	0.001	0.000	0.000	0.701	0.000	0.000	0.011	0.000	0.000	0.000	0.032	0.033
0.000	0.366	0.000	0.128	0.001	0.547	0.001	1.000	0.000	0.479	0.000	0.002	0.000	0.000	0.246	0.000	0.353	0.116	0.016
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.066	0.000
0.000	0.079	0.000	0.316	0.000	0.734	0.000	0.479	0.000	1.000	0.000	0.000	0.000	0.000	0.286	0.000	0.407	0.012	0.000
0.000	0.000	0.000	0.000	0.461	0.000	0.701	0.000	0.000	0.000	1.000	0.000	0.000	0.008	0.000	0.000	0.000	0.020	0.018
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.012	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.000	0.000	0.000	1.000	0.002	0.000	0.000	0.000	0.002	0.000
0.000	0.000	0.000	0.000	0.005	0.000	0.011	0.000	0.000	0.000	0.008	0.000	0.002	1.000	0.000	0.005	0.000	0.000	0.000
0.000	0.028	0.000	0.137	0.000	0.356	0.000	0.246	0.000	0.286	0.000	0.000	0.000	0.000	1.000	0.000	0.456	0.012	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.000	1.000	0.000	0.000	0.000
0.000	0.038	0.000	0.193	0.000	0.510	0.000	0.353	0.000	0.407	0.000	0.000	0.000	0.000	0.456	0.000	1.000	0.019	0.000
0.000	0.000	0.000	0.023	0.023	0.015	0.032	0.116	0.066	0.012	0.020	0.012	0.002	0.000	0.012	0.000	0.019	1.000	0.003
0.000	0.006	0.000	0.015	0.013	0.000	0.033	0.016	0.000	0.000	0.018	0.000	0.000	0.000	0.000	0.000	0.000	0.003	1.000

Self MAC KE

1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	1.000	0.000	0.000	0.008	0.005	0.166	0.345	0.000	0.000	0.066	0.000	0.000	0.000	0.000	0.000	0.004	0.000	0.006
0.000	0.000	1.000	0.001	0.000	0.001	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.001	1.000	0.000	0.672	0.023	0.013	0.000	0.449	0.000	0.230	0.000	0.000	0.008	0.007	0.000	0.003	0.000
0.000	0.008	0.000	0.000	1.000	0.014	0.510	0.087	0.000	0.000	0.304	0.000	0.000	0.000	0.000	0.000	0.013	0.000	0.031
0.000	0.005	0.001	0.672	0.014	1.000	0.000	0.035	0.000	0.530	0.022	0.680	0.000	0.000	0.013	0.008	0.001	0.004	0.002
0.000	0.166	0.000	0.023	0.510	0.000	1.000	0.525	0.000	0.019	0.712	0.022	0.000	0.000	0.000	0.000	0.023	0.000	0.072
0.000	0.345	0.000	0.013	0.087	0.035	0.525	1.000	0.000	0.001	0.483	0.005	0.000	0.000	0.000	0.000	0.014	0.000	0.062
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.001	0.449	0.000	0.530	0.019	0.001	0.000	1.000	0.000	0.527	0.000	0.000	0.007	0.008	0.000	0.003	0.000
0.000	0.066	0.000	0.000	0.304	0.022	0.712	0.483	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.021	0.000	0.055
0.000	0.000	0.000	0.230	0.000	0.680	0.022	0.005	0.000	0.527	0.000	1.000	0.000	0.000	0.012	0.006	0.000	0.004	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	1.000	0.001	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.008	0.000	0.013	0.000	0.000	0.000	0.007	0.000	0.012	0.000	0.001	1.000	0.003	0.000	0.003	0.000
0.000	0.000	0.000	0.007	0.000	0.008	0.000	0.000	0.000	0.008	0.000	0.006	0.000	0.000	0.003	1.000	0.000	0.000	0.000
0.000	0.004	0.000	0.000	0.013	0.001	0.023	0.014	0.000	0.000	0.021	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.003
0.000	0.000	0.000	0.003	0.000	0.004	0.000	0.000	0.000	0.003	0.000	0.004	0.000	0.000	0.003	0.000	0.000	1.000	0.000
0.000	0.006	0.000	0.000	0.031	0.002	0.072	0.062	0.000	0.000	0.055	0.000	0.000	0.000	0.000	0.000	0.003	0.000	1.000

APPENDIX C – MATLAB CODE

getEffectiveIndependence

```
function [Efi,DOF,xtraDOF,detFish,kedetFish] = getEffectiveIndependence(PHI,M,K, ntargetdof,DOF)
%Effective Independce can be explained here:
%  https://ay16-17.moodle.wisc.edu/prod/pluginfile.php/171634/
%  mod_resource/content/3/eCOWI_Resources/lecturepre/
%  Topic%2010.2%20-%20Sensor%20Placement%20using%20Effective
%  %20Independence%20Pres.pdf

%And here:
%  https://ay16-17.moodle.wisc.edu/prod/pluginfile.php/171635/
%  mod_resource/content/1/AIAA-JGCD-91.pdf


%It uses Modal Kinetic Energy to pick an initial candidate set of DOF which
%is 2x ntargetdof. It then iteratively removes dof by ranking via effective
%independence until a set of target dof is chosen (with ntargetdof dof).

format long;

%xtraDOF - DOF FOR HW3. SET OF nM+5 dof based solely on MKE.

[n,nM] = size(PHI);
iniset = 3*ntargetdof;
% iniset = n;
for i = 1:n
    for j = 1:nM
        KE(i,j) = PHI(i,j)*M(i,:)*PHI(:,j);
    end
end

KEnorm = (1/nM)*sum(KE,2);
%idx is the index of the positions in the PHI vector of the sorted kinetic
%energy by modes.
[KEsort, idx] = sort(KEnorm,'descend');
xtraDOF = DOF(idx(1:ntargetdof));
%DOF for only the initial candidate set of modes
DOF = DOF(idx(1:iniset));
PHIb = PHI;

%Reduce our target modes to the initial set of DOF selected by Modal
%Kinetic Energy
PHI = PHI(idx(1:iniset),:);
k = iniset;
detFish = zeros(iniset-ntargetdof,1);
kedetFish = zeros(iniset-ntargetdof,1); %HW 3 only
j = 0;
figure; hold on;
while k > ntargetdof


    PHIKE = PHIb(idx(1:iniset-j),:);
    B = PHIKE'*PHIKE;
    kedetFish(iniset-k+1) = det(B); %HW 3 only
    j = j+1;

    %     A = zeros(nM,nM);
%     for i = 1:size(PHI,1)
%         A = A+PHI(i,:)'*PHI(i,:);
%     end
    A = PHI'*PHI;
    [V,D] = eig(A);
    [D, ascindx] = sort(diag(D),'ascend'); %Sort eigenvalues.
    V = V(:,ascindx); %sort eigenvectors accordingly.
    G = (PHI*V).^2;
    % F = G.*repmat((1./D'),38,1);
    F = G*inv(diag(D));
    Efi = F*ones(nM,1);
```

```
    [Efi,index] = sort(Efi,'descend');
%Remove the lowest contributing dof from original DOF list and from PHI.
    DOF = DOF(index(1:end-1));
    PHI = PHI(index(1:end-1),:);
    detFish(iniset-k+1) = det(A);
    k = k-1;

end
    plot(detFish,'bx-');
    plot(kedetFish,'ro-');
    xlabel('Iteration #');
    ylabel('det(Q)');
    legend('Q - efi' ,'Q- ke');
end
```

getEffectiveMass

```
function [E,MR, MRo,M,K,dofA,dofO] = getEffectiveMass(M,K,PHI, DOFint, DOF, PHIR )

% Effective Mass can be described here:
%  https://ay16-17.moodle.wisc.edu/prod/pluginfile.php/171644/...
%  mod_resource/content/1/Topic%2012.1%20-%20Selection%20of%20Target...
%  %20Modes%20-%20Effective%20Mass%20Pres.pdf

% Returns E, the effective mass matrix. It's columns sum to the diagonal
% terms of the rigid body mass matrix Mro (PHI'*Moo*PHI)
% Returns MR, the rigid body mass matrix, MRo, the o-set partition of MR.
% M and K, sorted mass and stiffness.

% Takens in the mass matrix M, FIXED-INTERFACE MODES PHIro,
% modes PHI, the list of DOF at the interface DOFint and
% the total list of DOF, DOF.
% Mass matrix and rows of PHI should be sorted according to DOF.

nDOFint = length(DOFint); % # kept dof.
nDOF = length(DOF); % # dof total.
dofA = zeros(nDOFint,1); % INDEX OF ASET wrt M and K
for i = 1:nDOFint
    [tmpind,~] = find(DOF==DOFint(i));
    dofA(i) = tmpind;
end
dofO = setxor(1:nDOF,dofA); % INDEX OSET DOF wrt M and K
DOF = DOF([dofA;dofO],:); % ordered DOF list, [ASET;OSET]
Moo = M(dofO,dofO);
for i = 1:numel(PHI(1,:))
 mnorm(i) = 1/(sqrt(PHI(:,i)'*Moo*PHI(:,i)));
 PHI(:,i) = mnorm(i)*PHI(:,i);
end
% Moa = M(DOFind,DOFCOMP);
% Mao = M(DOFCOMP,DOFind);
% Maa = M(DOFCOMP,DOFCOMP); `

%M AND K PARTITIONED ACCORDING TO DOF
M = [ M(dofA,dofA), M(dofA,dofO); ...
    M(dofO,dofA),M(dofO,dofO)];
K = [ K(dofA,dofA), K(dofA,dofO); ...
    K(dofO,dofA),K(dofO,dofO)];

%Orthogonality Check- turn on to see if PHI is orthogonal in a general
%sense.
% surf(PHI'*Moo*PHI);

% OSET MASS WEIGHTED FIXED INTERFACE MODES ORTHOGONALITY CHECK.

%Compute PHIro *** NOTE, this restricts rigid body modes from the first 6
%DOF of the interface.

% PHIr = [ eye(6); -inv(K(7:end,7:end))*K(7:end,1:6)];
% PHIRo =  PHIR(nDOFint+1:end,:);
```

```
PHIRo = PHIR(dofO,:);
PHIR = PHIR([dofA; dofO],:);
MR = PHIR'*M*PHIR;
MRo = PHIRo'*Moo*PHIRo; %Modal mass matrix contributing to o-set partition.


%%
A = repmat((1./diag(MRo))',30,1);
E = ([PHI'*Moo*PHIRo].^2).*A;
% E = ([PHI'*Moo*PHIRo].^2)*(1./diag(MRo));
end
```

Wrapper Function

```
close all; clear all; clc;
load('gpsc.mat');

wtmass = 386.4; %parameter to convert weight to mass for M.
% M = wtmass.*M; %Convert to propper units.

%Sort DOF1 and DOF2 in ascending order. Reorder M and K according to DOF1
%sort, sort PHI according to DOF2 sort.
[DOF1, DOF1ind] = sort(DOF1,'ascend');
[DOF2, DOF2ind] = sort(DOF2,'ascend');
M = M(DOF1ind,DOF1ind);
K = K(DOF1ind,DOF1ind);
PHI = PHI(DOF2ind,:);
PHIFI = PHI; %fixed interface modes;

%interface dof set
aSET = [44.1; 44.2; 44.3; 45.1; 45.2; 45.3; 48.1; 48.2; 48.3; 49.1; ...
    49.2; 49.3];

%Compute RBM about node 50
PHIR = [  -inv(K(1:end-6,1:end-6))*K(1:end-6,end-5:end); eye(6)];
PHIR = PHIR(1:end-3,:); %Remove the rotational DOF from PHIR
DOF1 = DOF1(1:end-3); %Remove the rotation DOF from list.



%Static reduction of the 3 rotations at DOF 50 (50.4,50.5,50.6);
T = [eye(150); -inv(K(151:end,151:end))*K(151:end,1:150)];
M = T'*M*T;
K = T'*K*T;



%% Generate a function that computes effective mass and use it to rank the
%fixed interface modes of the craft (PHIsort). Select a set of target modes
%based on the fact that each of them needs at least 5% of the effective
%mass in each of the six rigid body directions. Determine the total
%effective mass for the target mode set in each rigid body direction and
%comment on the sets dynamic completeness.



% M = wtmass.*M; %Convert to propper units.
[E,MR,MRo,Msort,Ksort,dofA,dofO] = getEffectiveMass(M,K,PHIFI,aSET,DOF1,PHIR);
Esum = sum(E);
%%
% DOF1 = dofO;
% MR = wtmass.*MR;
% MRo = wtmass.*MRo;
%%
disp('Effective Mass (30 modes):');
disp(Esum);

%Selects all modes that contribute to more than 4.65% of mass in any rigid
%body direction. The assignment specifies 5%, however Dr. Kammer suggested
%"Things close to 5% RBM".
I = find(E>=0.048);
```

```
modindx = sort(mod(I,numel(E(:,1))),'ascend');
modindx = unique(modindx);
masscmplt = sum(E(modindx,:));
disp('Selected Modes Total Effective Mass:');
disp(masscmplt);
disp('The Modes that should be kept are:');
disp(modindx');

%Fixed interface M and K. These matrices were sorted in the
%getEffectiveMass call. (O-set partition)
PHIK = PHIFI(:,modindx); %Kept fixed-interface modes.
MF = Msort(13:end,13:end);
KF = Ksort(13:end,13:end);



%Generate functions to rank candidate sensor locations based on Modal
%Kinetic Energy and Effective Independence. Pick a single initial candidate
%set of sensor locations and use these functions to select a final set that
%will identify your target modes. The final sensor location should have
%n-target modes + 5 sensors.

[Efi,EFIdof,KEdof,efiDetQ,keDetQ] = getEffectiveIndependence(PHIK,MF,KF,numel(modindx)+5,dofO);


%Use MAC and any other measure to determine which of the sensor sets
%(initial and final) produces the most independent target mode partitions
%and the greatest target mode response signal strength.

EFIdof = sort(EFIdof,'ascend'); %DOF in original 150x150 mass matrix.
KEdof = sort(KEdof,'ascend');

for i = 1:numel(EFIdof)
    [efiDOFind(i),~] = find(EFIdof(i)==dofO);
end
for i = 1:numel(KEdof)
    [keDOFind(i),~] = find(KEdof(i)==dofO);
end
efiDOFind = efiDOFind';
keDOFind = keDOFind';
disp('The DOF set selected by EFI are:');
disp(DOF1(EFIdof)');
disp('The DOF set selected by KE are:');
disp(DOF1(KEdof)');
grid on;

%% DEBUG ONLY
% i = [ 130:135, 142:147]';
% j = setxor(1:150,i);
% [PHI,D,wn,wnhz,sortindx] = getEigSort(K(j,j),M(j,j));
%%


%Static reductions to EFIdof and KEdof.
[Kke,Mke,keREDind,keREDcomp,Kkesort,Mkesort] = ...
getStaticTAM(KF,MF,[1:138]',keDOFind);
[Kefi,Mefi,efiREDind,efiREDcomp,Kefisort,Mefisort] = ...
getStaticTAM(KF,MF,[1:138]',efiDOFind);
%Eigenproblem for EFIdof and KEdof.
[efiPHI,efiD,efiwn,efiwnhz,efisortindx] = getEigSort(Kefi,Mefi);
[kePHI,keD,kewn,kewnhz,kesortindx] = getEigSort(Kke,Mke);
PHImacefi = PHI(efiREDind,modindx);
PHImacke = PHI(keREDind,modindx);

%MAC
efimac = mac(efiPHI,PHImacefi);
set(gca, 'XTickLabel', modindx);
title('Modal Assurance Criterion (MAC) between PHI_e_f_i and PHI');
xlabel('Target Fixed-Interface Mode #');

efiSmac = mac(efiPHI,efiPHI);
```

```
title('Self - Modal Assurance Criterion (MAC) for PHI_e_f_i');
ylabel('PHI_e_f_i Mode #');
xlabel('PHI_e_f_i Mode #');

kemac = mac(kePHI,PHImacke);
set(gca, 'XTickLabel', modindx);
title('Modal Assurance Criterion (MAC) between PHI_k_e and PHI');
xlabel('Target Fixed-Interface Mode #');

keSmac = mac(kePHI,kePHI);
title('Self - Modal Assurance Criterion (MAC) for PHI_k_e');
ylabel('PHI_k_e Mode #');
xlabel('PHI_k_e Mode #');

%Cross and Self Orthogonality
% [keCO] = corl8(kePHI,PHImacke,Mke);
[Cke,~,~] = Kammercorl8(kePHI,Mke,PHImacke);
set(gca, 'XTickLabel', modindx);
title('Cross Orthogonality (CO) between PHI_k_e and PHI');
ylabel('PHI_k_e Mode #');
xlabel('Target Fixed-Interface Mode #');

% [keSO] = corl8(kePHI,kePHI,Mke);
[Sefi,~,~] = Kammercorl8(kePHI,Mke,kePHI);
% set(gca, 'XTickLabel', modindx);
title('Self Orthogonality (SO) for PHI_k_e');
ylabel('PHI_k_e Mode #');
xlabel('Target Fixed-Interface Mode #');

% [efiCO] = corl8(efiPHI,PHImacefi,Mefi);
[Cefi,p1,p2] = Kammercorl8(efiPHI,Mefi, PHImacefi);
set(gca, 'XTickLabel', modindx);

title('Cross Orthogonality (CO) between PHI_e_f_i and PHI');
ylabel('PHI_e_f_i Mode #');
xlabel('Target Fixed-Interface Mode #');

% [efiSO] = corl8(efiPHI,efiPHI,Mefi);
[Sefi,~,~] = Kammercorl8(efiPHI,Mefi, efiPHI);
% set(gca, 'XTickLabel', modindx);
title('Self Orthogonality (SO) for PHI_e_f_i');
ylabel('PHI_e_f_i Mode #');
xlabel('Target Fixed-Interface Mode #');

%Frequency error
% efimodematching = [1,1; 2,4; 3,3; 4,5; 5,6; 7,8; 10,9; 11,12; 12,13; 15,18; 16,19]; %FULL SET
kemodematching = [1,1; 2,3; 3,4; 4,5; 5,6; 7,8; 8,9; 11,12; 12,13; 14,18; 15,19; 16,20; 17,21;
18,22]; %FULL SET (REDUCED IS THE SAME)

efimodematching = [1,1; 2,3; 3,4; 4,6; 5,5; 6,8; 8,9; 10,12; 11,13; 13,18; 14,19; 15,21];
%REDUCED BEGINING SET

% kemodematching = [1,1; 2,3; 3,4; 4,5; 5,6; 7,9; 11,12; 12,13; 14,18; 15,19;16,20;17,21; 18,22];
eficompare = [w(efimodematching(:,2)) efiwnhz(efimodematching(:,1))];
kecompare = [w(kemodematching(:,2)) kewnhz(kemodematching(:,1))];
```