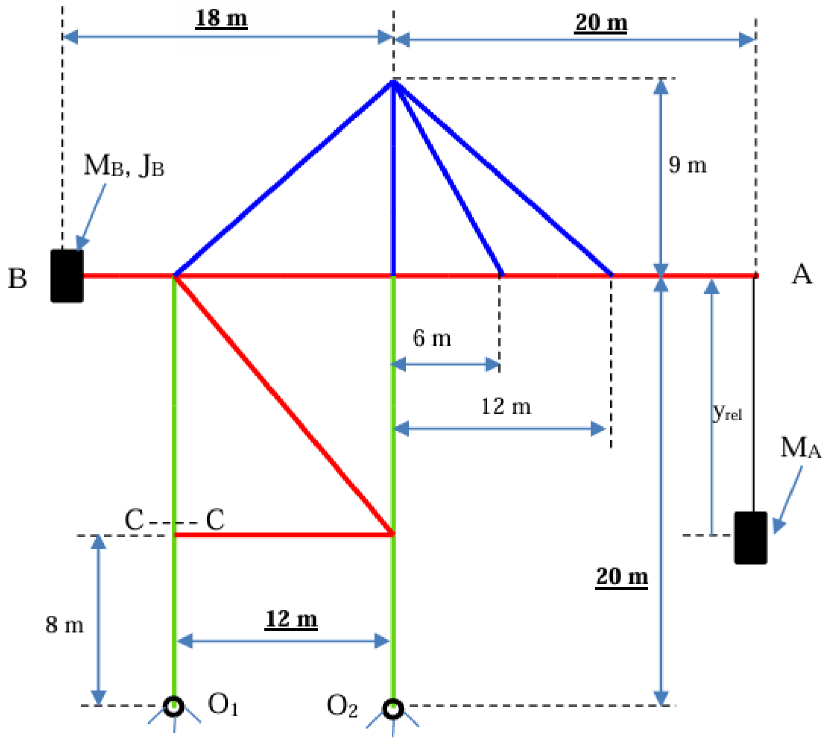
Year Work

Lazzaro Francesco 11095668



Computation of Input File

Initial Beams Parameters

```
clear all;

E = 2.06E11; % Young's modulus [N/m^2]
rho = 7800;  % Density of the material [kg/m^3]

% Blue beams (IPE240)
A_IPE_240 = 3.912E-3;  % Area [m^2]
I_IPE_240 = 3.892E-5;  % Moment of inertia (cross-section) [m^4]
mL_IPE_240 = rho*A_IPE_240; % Linear mass [kg/m]
EA_IPE_240 = E*A_IPE_240; % Axial stiffness [N]
EJ_IPE_240 = E*I_IPE_240; % Bending stiffness [Nm^2]
beam_IPE_240 = [A_IPE_240 I_IPE_240 mL_IPE_240 EA_IPE_240 EJ_IPE_240];

% Red beams (IPE300)
A_IPE_300 = 5.381E-3; % Area [m^2]
I_IPE_300 = 8.356E-5; % Moment of inertia (cross-section) [m^4]
mL_IPE_300 = rho*A_IPE_300; % Linear mass [kg/m]
EA_IPE_300 = E*A_IPE_300; % Axial stiffness [N]
EJ_IPE_300 = E*I_IPE_300; % Bending stiffness [Nm^2]
beam_IPE_300 = [A_IPE_300 I_IPE_300 mL_IPE_300 EA_IPE_300 EJ_IPE_300];

% Green beams (IPE550)
A_IPE_550 = 1.344E-2; % Area [m^2]
I_IPE_550 = 6.712E-4; % Moment of inertia (cross-section) [m^4]
mL_IPE_550 = rho*A_IPE_550; % Linear mass [kg/m]
```

```
EA_IPE_550 = E*A_IPE_550; % Axial stiffness [N]
EJ_IPE_550 = E*I_IPE_550; % Bending stiffness [Nm^2]
beam_IPE_550 = [A_IPE_550 I_IPE_550 mL_IPE_550 EA_IPE_550 EJ_IPE_550];
```

In order to compute the Maximum Lenght for each Finite Element $\overline{\Omega}_{1,\min}$ is needed

$$\Omega_{1_{\min}} = 2\pi c f_{\max}$$

```
f_max = 10; % Max frequency [Hz]
c = 2; % Safety coefficient
Omega_1_min = 2*pi*c*f_max;
```

For the computation of the position of each beam the Origin is considered in O1

```
% Beams
% Initial Node - Final Node - Beam Type
beams = [0 0 0 8 beam_IPE_550;
         0 8 0 20 beam_IPE_550;
         12 8 12 0 beam_IPE_550;
         12 20 12 8 beam_IPE_550;
         0 20 12 8 beam_IPE_300;
        -6 20 0 20 beam_IPE_300;
         0 20 12 20 beam_IPE_300;
         12 20 18 20 beam_IPE_300;
         18 20 24 20 beam_IPE_300;
         24 20 32 20 beam_IPE_300;
         0 8 12 8 beam_IPE_300;
         0 20 12 29 beam_IPE_240;
         18 20 12 29 beam_IPE_240;
         24 20 12 29 beam_IPE_240;
         12 29 12 20 beam_IPE_240];
```

Calculation of Max Lenght of FEs

$$L_{\max_j} = \sqrt{\frac{\pi^2}{\Omega_{1_{\min}}}\sqrt{\frac{\mathrm{EI}_j}{\rho A_j}}}$$

```

% Inside "beams" will be stored all the information needed for all upcoming
% calculations

% "Beams" Organization
for i=1:length(beams)
    % Calculation of the length
    beams(i,10) = sqrt((beams(i,1)-beams(i,3))^2+(beams(i,2)-beams(i,4))^2);
    % Calculation of the Mass
    beams(i,11) = rho*beams(i,5)*beams(i,10);
    % Calculation of Max Length
    beams(i,12) = sqrt(((pi^2)/Omega_1_min)*sqrt(E*beams(i,6)/beams(i,7)));
    % Number of finite Elements for each beam
    beams(i,13) = ceil(beams(i,10)/beams(i,12));
end
```

Nodal coordinates

```
% Total number of finite elements
FE_number = sum(beams(:,13));

% XY coordinates for each node
nodes = zeros(FE_number+length(beams),2);
j = 1;

for i=1:length(beams)
    number_nodes = beams(i,13)+1; % Number of the node
    ith_nodes_x = linspace(beams(i,1),beams(i,3),number_nodes); % x
    ith_nodes_y = linspace(beams(i,2),beams(i,4),number_nodes); % y
    nodes(j:j+number_nodes-1,:) = [ith_nodes_x' ith_nodes_y'];
    j = j+number_nodes;
end
```

Development of Specified Format Inp FIle

```
% FileGenerator
```

With Nodes and Beams files it is possible to generate the final .inp file, suitable for ***dmb_fem2*** software

Input File

! Yearwork structure

! nodes list :

! node nr. - boundary conditions codes: x,y,theta x y

*NODES

1 1 1 0 0.0 0.0

2 0 0 0 0.0 8.0

3 0 0 0 0.0 14.0

4 0 0 0 0.0 20.0

5 0 0 0 12.0 8.0

6 1 1 0 12.0 0.0

7 0 0 0 12.0 20.0

8 0 0 0 12.0 14.0

9 0 0 0 4.0 16.0

10 0 0 0 8.0 12.0

11 0 0 0 -6.0 20.0

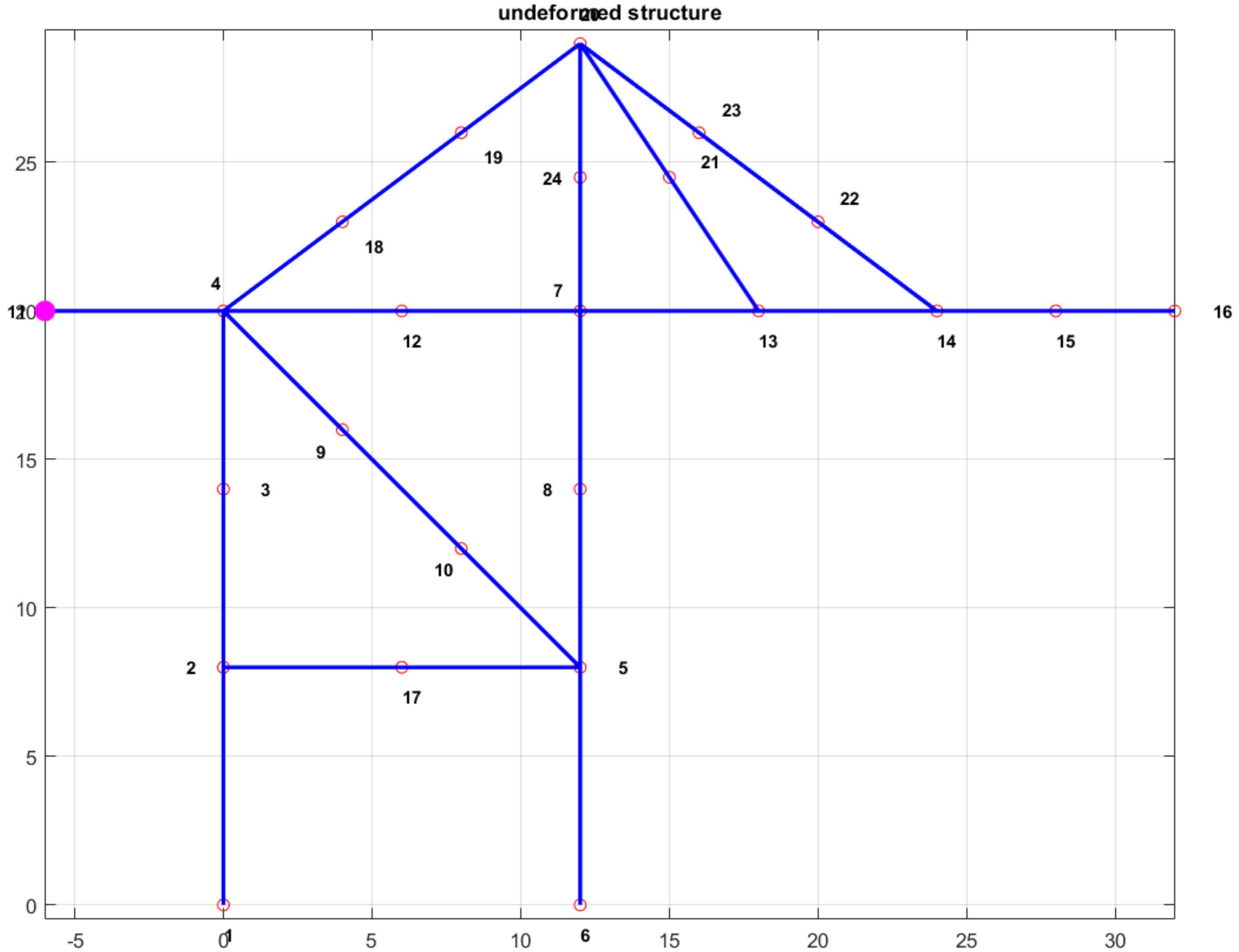
```
12              0 0 0   6.0    20.0
13              0 0 0   18.0    20.0
14              0 0 0   24.0    20.0
15              0 0 0   28.0    20.0
16              0 0 0   32.0    20.0
17              0 0 0   6.0    8.0
18              0 0 0   4.0    23.0
19              0 0 0   8.0    26.0
20              0 0 0   12.0    29.0
21              0 0 0   15.0    24.5
22              0 0 0   20.0   23.0
23              0 0 0   16.0    26.0
24              0 0 0   12.0    24.5
*ENDNODES

! beams list :
! beam nr.  i-th node nr.  j-th node nr.     mass [kg/m]   EA [N]  EJ [Nm^2]
*BEAMS
1    1    2    1.048320e+02   2.8e+09    1.4E+08
2    2    3    1.048320e+02   2.8e+09    1.4E+08
3    3    4    1.048320e+02   2.8e+09    1.4E+08
4    5    6    1.048320e+02   2.8e+09    1.4E+08
5    7    8    1.048320e+02   2.8e+09    1.4E+08
6    8    5    1.048320e+02   2.8e+09    1.4E+08
7    4    9    4.197180e+01   1.1e+09    1.7E+07
8    9    10    4.197180e+01   1.1e+09    1.7E+07
9    10    5    4.197180e+01   1.1e+09    1.7E+07
10    11    4    4.197180e+01   1.1e+09    1.7E+07
11    4    12    4.197180e+01   1.1e+09    1.7E+07
12    12    7    4.197180e+01   1.1e+09    1.7E+07
13    7    13    4.197180e+01   1.1e+09    1.7E+07
14    13    14    4.197180e+01   1.1e+09    1.7E+07
```

```
15    14    15    4.197180e+01   1.1e+09    1.7E+07
16    15    16    4.197180e+01   1.1e+09    1.7E+07
17    2     17    4.197180e+01   1.1e+09    1.7E+07
18    17    5     4.197180e+01   1.1e+09    1.7E+07
19    4     18    3.051360e+01   8.1e+08    8.0E+06
20    18    19    3.051360e+01   8.1e+08    8.0E+06
21    19    20    3.051360e+01   8.1e+08    8.0E+06
22    13    21    3.051360e+01   8.1e+08    8.0E+06
23    21    20    3.051360e+01   8.1e+08    8.0E+06
24    14    22    3.051360e+01   8.1e+08    8.0E+06
25    22    23    3.051360e+01   8.1e+08    8.0E+06
26    23    20    3.051360e+01   8.1e+08    8.0E+06
27    20    24    3.051360e+01   8.1e+08    8.0E+06
28    24    7     3.051360e+01   8.1e+08    8.0E+06
*ENDBEAMS

! alpha and beta values to define the damping matrix
*DAMPING
0.2 4.0e-4

! Masses
*MASSES
1    11    2000.0    80.0
*ENDMASSES
```

Important Results

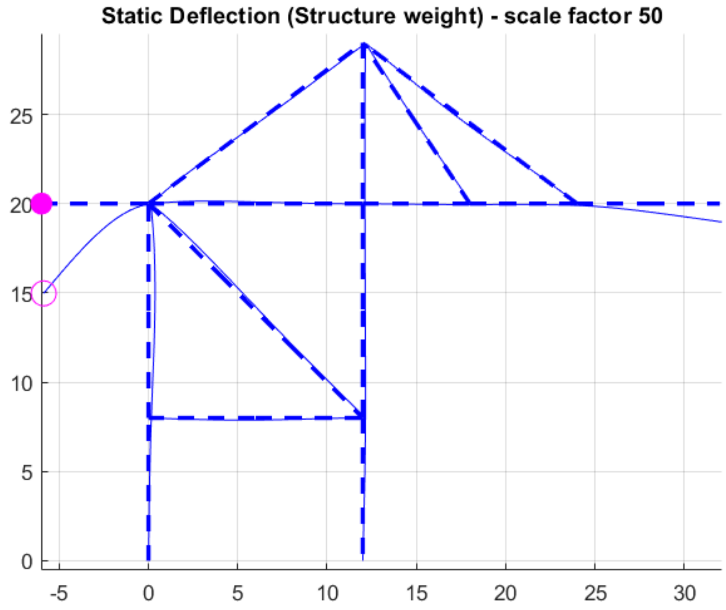
```
Mass of the Beams: 8524.24051376353 kg

Number of Finite Elements Employed: 28

Number of Finite Nodes: 24

Number of Degrees of Freedom (DOF): 68

Max_length for IPE550 Beams: 6.34499948927024 m

Max_length for IPE240 Beams: 7.09205271304674 m

Max_length for IPE300 Beams: 9.4973242344486 m
```

1) Undeformed Structure of the Unloaded Crane



Point B: Node n.11

Point A: Node n.16



2) Natural Frequencies & Modal Shapes

The software produces 6 acceptable natural frequencies

$\omega_1 = 1.3459$

$\omega_2 = 1.586$

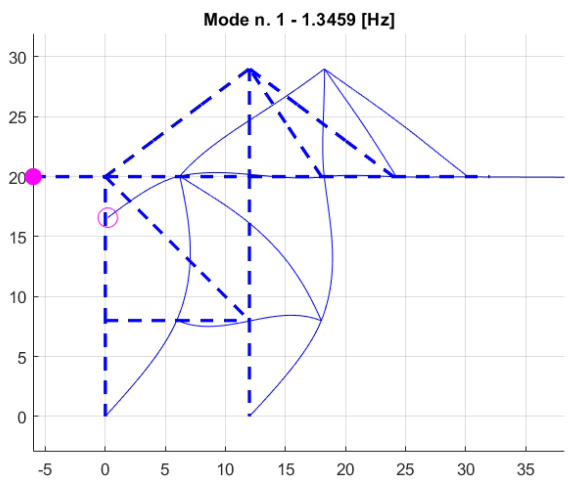
$\omega_3 = 4.0134$

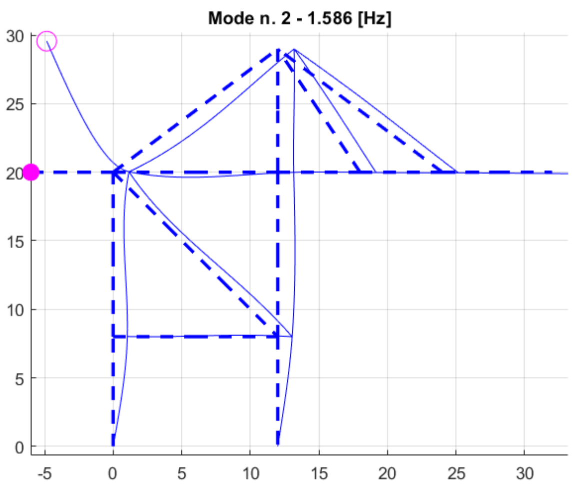
$\omega_4 = 6.5149$

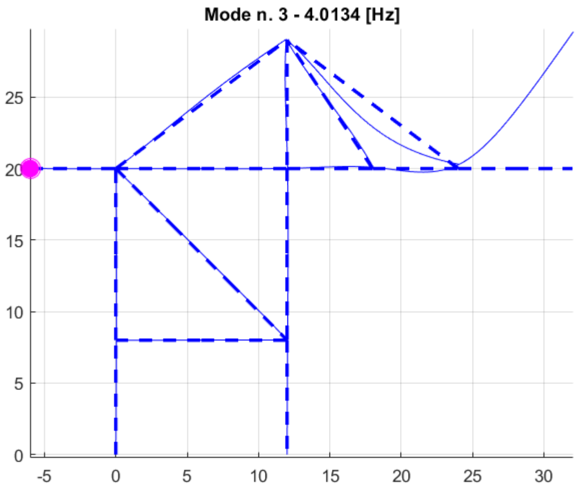
$\omega_5 = 7.5404$

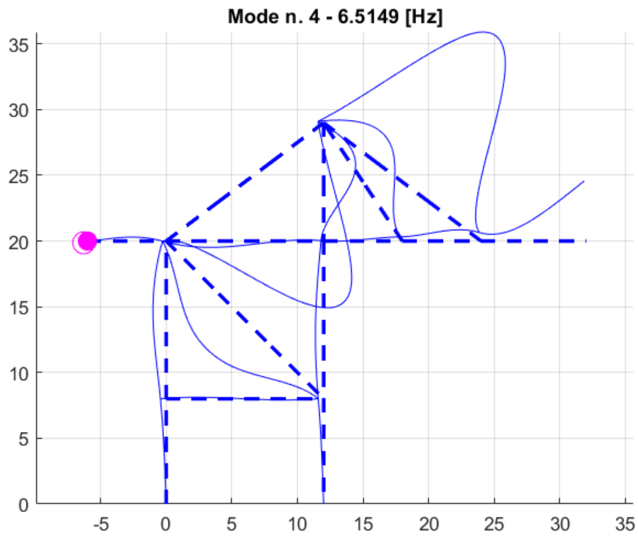
$\omega_6 = 7.8943$

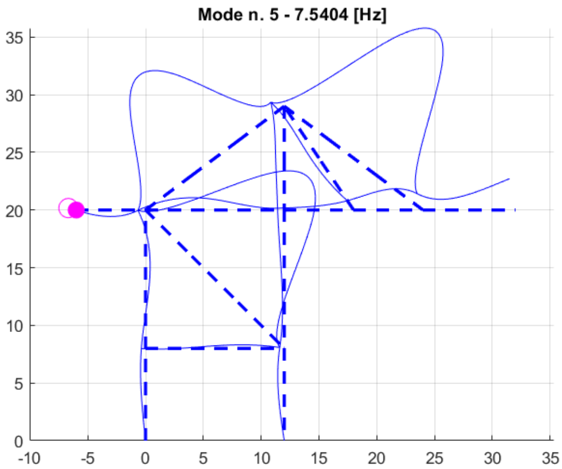
Scale Factor set to 30

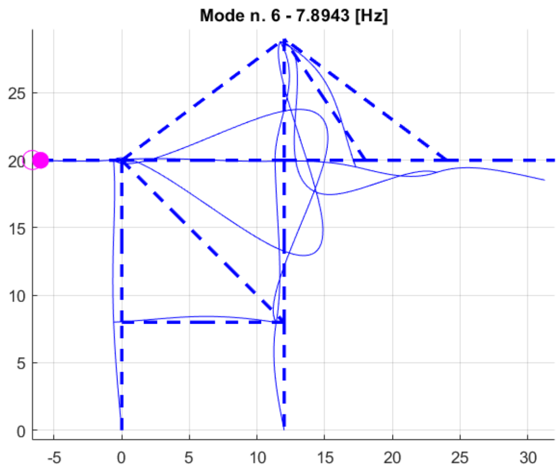












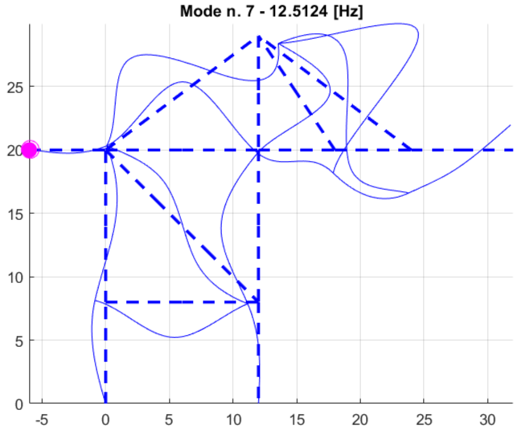
Mode Shape Analysis

Each mode corresponds to a specific deformation pattern, describing how the structure oscillates when excited at its natural frequency

Understanding the characteristics of each mode helps in interpreting the dynamics of the structure

Observations:

- **Mode 1 (1.3459 Hz)**:
- This is the fundamental mode, usually involving large-scale, global deformation (e.g., bending or overall translation)
- The structure likely flexes uniformly, and the static mass moves in-phase with nearby elements
- **Mode 2 (1.586 Hz)**:
- This mode often represents a secondary bending mode. Here, the static mass may act as a localized center of inertia, affecting how adjacent elements deform
- **Modes 3-6 (Higher frequencies)**:
- These modes involve more localized and complex patterns, such as higher-order bending, torsion, or twisting

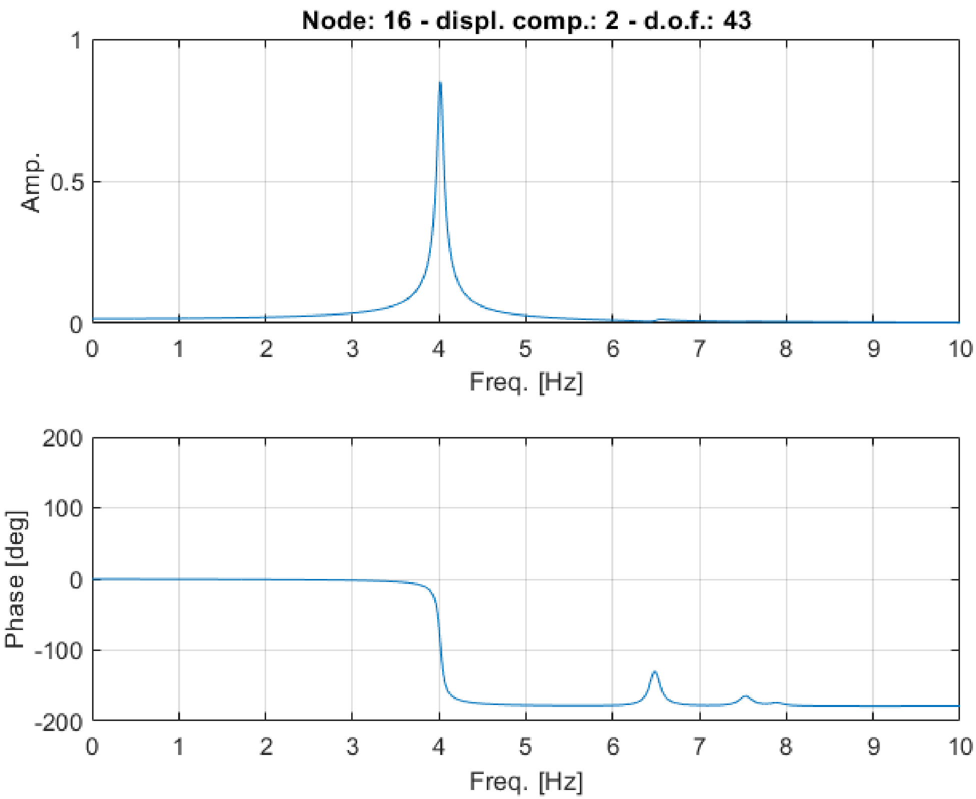


Since the requirements set the frequency range as 0 - 10 Hz starting from mode n. 7 results would be pointless

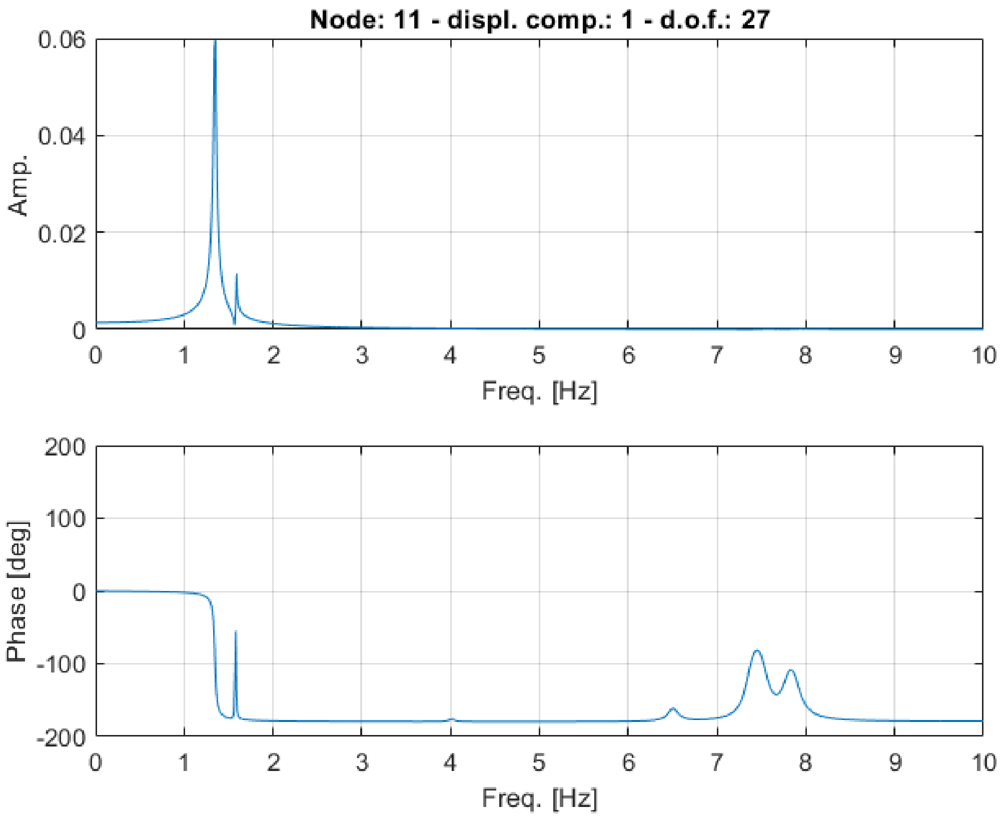
3) FRF 0 ÷ 10 Hz

Force Amplitude set to 1000 N, Scale Factor set to 30

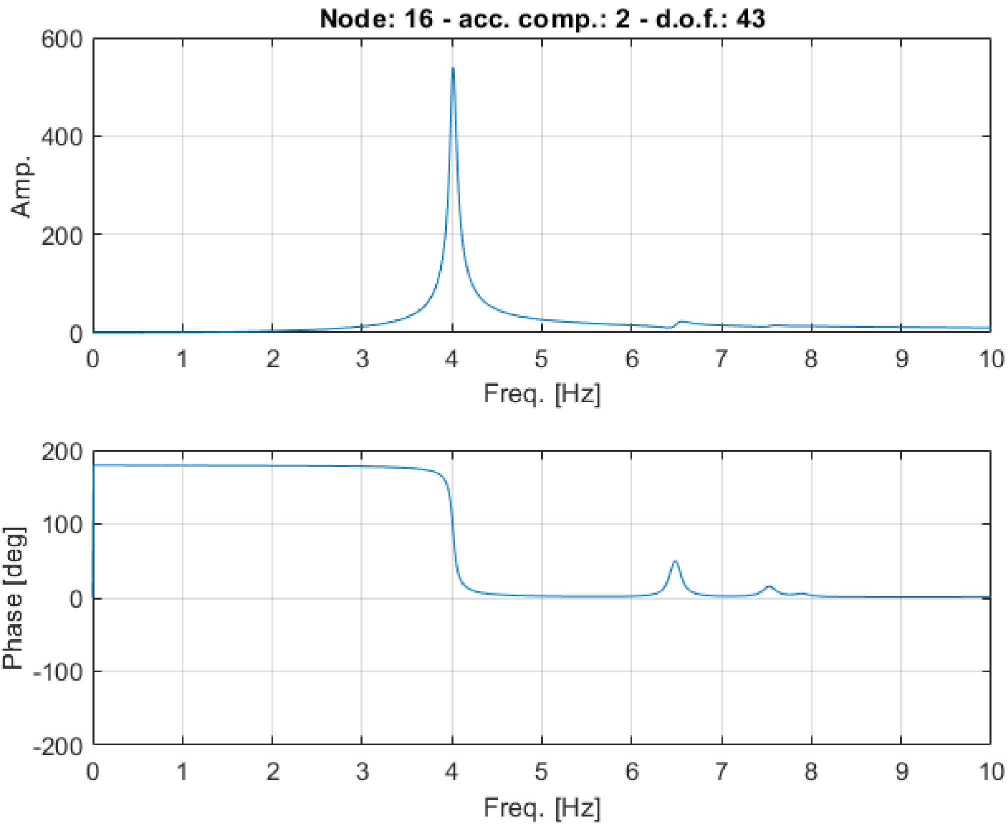
Input: vertical force at point A; Output: vertical displacement of point A



Input: horizontal force at point A; Output: horizontal displacement of point B



Input: vertical force at point A; Output: vertical acceleration of point A



4) Natural Frequencies and Damping Ratios of the Damped System

```
clear all;
```

```
load YWStructure_mkr.mat;
C=R;
```

M C and K are the $72x72$ structural matrixes

YearWork_Lazzaro.mlx | idb = 24×3

	1	2	3
1	69	70	1
2	2	3	4
3	5	6	7
4	8	9	10
5	11	12	13
6	71	72	14
7	15	16	17
8	18	19	20
9	21	22	23
10	24	25	26
11	27	28	29
12	30	31	32
13	33	34	35
14	36	37	38
15	39	40	41
16	42	43	44
17	45	46	47
18	48	49	50
19	51	52	53
20	54	55	56
21	57	58	59
22	60	61	62
23	63	64	65
24	66	67	68

Matrixes need to be partitioned to go on with the calculations

Total number of free nodal coordinates is equal to 3 times the number o nodes minus the number of constraints

$n_f = 24 * 3 - 4 = 68$

Partitioning

```
ndgf=68; % Free nodal coordinates
ntot=72; % Total nodal coordinates

% Mass Matrix
MFF=M(1:ndgf,1:ndgf);
MFC=M(1:ndgf,ndgf+1:ntot);
MCF=M(ndgf+1:ntot,1:ndgf);
MCC=M(ndgf+1:ntot,ndgf+1:ntot);

% Stiffness Matrix
KFF=K(1:ndgf,1:ndgf);
KFC=K(1:ndgf,ndgf+1:ntot);
```

```
KCF=K(ndgf+1:ntot,1:ndgf);
KCC=K(ndgf+1:ntot,ndgf+1:ntot);

% Damping Matrix
CFF=C(1:ndgf,1:ndgf);
CFC=C(1:ndgf,ndgf+1:ntot);
CCF=C(ndgf+1:ntot,1:ndgf);
CCC=C(ndgf+1:ntot,ndgf+1:ntot);
```

Computation of Frequencies, Modes and Damping Ratios

The natural frequencies of the damped system can be found taking into account the free system, as it can be shown in the following equation

$$[m_{\mathrm{FF}}]\underline{\ddot{x_F}} + [c_{\mathrm{FF}}]\underline{\dot{x_F}} + [k_{\mathrm{FF}}]\underline{x} = 0$$

Which be written in State-Space form taking the following states

$$\underline{z} = \begin{bmatrix} \underline{\psi} \\ \underline{\varsigma} \end{bmatrix} = \begin{bmatrix} \underline{\dot{x_F}} \\ \underline{x_F} \end{bmatrix}$$

Indeed the system equation can be rewritten

$$[m_{\mathrm{FF}}]\underline{\dot{\psi}} + [c_{\mathrm{FF}}]\underline{\psi} + [k_{\mathrm{FF}}]\underline{\varsigma} = 0$$

$$\underline{\dot{\psi}} = -[m_{\mathrm{FF}}]^{-1}[c_{\mathrm{FF}}]\underline{\psi} - [m_{\mathrm{FF}}]^{-1}[k_{\mathrm{FF}}]\underline{\varsigma}$$

In this way, the final representation of the system in State-Space matrixes is

$$\underline{\dot{z}} = \underbrace{\begin{bmatrix} -[m_{\mathrm{FF}}]^{-1}[c_{\mathrm{FF}}] & -[m_{\mathrm{FF}}]^{-1}[k_{\mathrm{FF}}] \\ I & 0 \end{bmatrix}}_{\text{Damped System Matrix}} \begin{bmatrix} \underline{\dot{x_F}} \\ \underline{x_F} \end{bmatrix}$$

Finally the natural frequencies and the damping factors can be found as the eigen values of the Damped System Matrix, employing the imaginary an real parts of them respectively

```
% State Matrix for free dumped system

A=[-MFF\CFF -MFF\KFF;
    eye(ndgf) zeros(ndgf)];

[eigenvectors, eigenvalues]=eig(A);

% Natural Frequencies and Modes of Vibration
damp_lambda=diag(eigenvalues)/(2*pi);
damp_modes = eigenvectors;

% Values up to 10 Hz
fMax = 10;
freq_Analysis = abs(imag(damp_lambda))<fMax;
sum(freq_Analysis);  % Number of frequencies below 10 Hz
damp_lambda = damp_lambda(freq_Analysis,1)
```

```
damp_lambda = 12×1 complex
```

```
-0.0362 + 4.0132i
-0.0362 - 4.0132i
-0.0692 + 6.5145i
-0.0692 - 6.5145i
-0.0941 + 7.8937i
-0.0941 - 7.8937i
-0.0872 + 7.5399i
-0.0872 - 7.5399i
-0.0143 + 1.3458i
-0.0143 - 1.3458i
    ⋮
```

```
% Natural Frequencies
[natural_frequencies, unique_Indexes] = unique(abs(imag(damp_lambda)))
```

```
natural_frequencies = 6×1
    1.3458
    1.5859
    4.0132
    6.5145
    7.5399
    7.8937
unique_Indexes = 6×1
     9
    11
     1
     3
     7
     5
```

```
% Modes of Vibration
vibration_Modes = zeros(length(damp_modes),length(unique_Indexes));
for j = 1:length(unique_Indexes)
    vibration_Modes(:,j) = damp_modes(:,unique_Indexes(j));
end
vibration_Modes;
% Dumping Ratios
damping_ratio = unique(abs(real(damp_lambda))./abs(damp_lambda))
```

```
damping_ratio = 6×1
    0.0028
    0.0090
    0.0106
    0.0106
    0.0116
    0.0119
```

The verification of this result can be carried out employing the linear model used in the ***dmb_fem2*** software, and taking into account that these are diagonal matrices, the followign equation holds.

$$\frac{c_i}{c_{i_{\mathrm{cr}}}} = \frac{\alpha}{2 \cdot w_{n_i}} + \frac{\beta \cdot w_{n_i}}{2}$$

```
alpha = 0.2;
beta = 4E-4;

% Natural Freq of Undamped System
[~, eigenvalues]=eig(MFF\KFF);
```

```
nf=sqrt(diag(eigenvalues));
% Frequencies to be taken into account
omegaMax = 2*pi*10; % [rad/s]
freq_Analysis = nf<omegaMax;
nf = nf(freq_Analysis,1);

% Validation
damping_verification = alpha./(2*nf)+beta.*nf/2;
damping_validation = [damping_ratio damping_verification]
```

```
damping_validation = 6×2
    0.0028    0.0135
    0.0090    0.0120
    0.0106    0.0090
    0.0106    0.0106
    0.0116    0.0116
    0.0119    0.0119
```

5) 6) & 7) FRF considering the Damped System

FRF Computation General Parameters

```
% Harmonic Force applied
Force = 1000; % [N]

% Selection Made through Graphic Analysis
% Analized Nodes Index
nodeA = 16;
nodeB = 11;
hinge1 = 1;
hinge2 = 6;

% Bending Beam at section C-C
nodeBeam_ith = 2;
nodeBeam_jth = nodeBeam_ith+1;

% Beam length
L_k = 6;
```

Modes of vibration selection

The analysis is limited only to the first 4 modes of vibration obtained for the first four natural frequencies of the system

Employing the ones from the undamped system

```
% Natural frequencies and modes of vibration

[eigenvectors, eigenvalues]=eig(MFF\KFF);

freq=sqrt(diag(eigenvalues))/2/pi;
modes = eigenvectors;

% Usable Frequencies
fMax = 10;
freq_Analysis = freq<fMax;
sum(freq_Analysis);
```

```
freq = freq(freq_Analysis,1)
```

```
freq = 6×1
    1.3459
    1.5860
    4.0134
    6.5149
    7.5404
    7.8943
```

```
modes = modes(:,freq_Analysis)
```

```
modes = 68×6
   -0.0301   -0.0161   -0.0006    0.0016    0.0046    0.0042
    0.1984    0.0916    0.0037   -0.0196   -0.0153   -0.0255
    0.0005    0.0011   -0.0004   -0.0001   -0.0005    0.0003
   -0.0143   -0.0022   -0.0002    0.0043   -0.0031    0.0017
    0.2356    0.0737    0.0039   -0.0446    0.0150   -0.0261
    0.0009    0.0020   -0.0007   -0.0002   -0.0009    0.0006
    0.0007    0.0037    0.0002    0.0013   -0.0029   -0.0012
    0.2067    0.1029    0.0019   -0.0126   -0.0275   -0.0256
    0.0012    0.0028   -0.0009   -0.0002   -0.0013    0.0008
    0.0073   -0.0181    0.0005   -0.0129    0.0205    0.0031
      ⋮
```

The system will be restricted to 4dof by doing a linear transformation employing the first four Modes of vibration

$$\widehat{\Phi} = \left[\underline{x}^{(1)} \quad \underline{x}^{(2)} \quad \underline{x}^{(3)} \quad \underline{x}^{(4)}\right]$$

```
% Modes that need to be considered
Phi_4 = modes(:,1:4)
```

```
Phi_4 = 68×4
   -0.0301   -0.0161   -0.0006    0.0016
    0.1984    0.0916    0.0037   -0.0196
    0.0005    0.0011   -0.0004   -0.0001
   -0.0143   -0.0022   -0.0002    0.0043
    0.2356    0.0737    0.0039   -0.0446
    0.0009    0.0020   -0.0007   -0.0002
    0.0007    0.0037    0.0002    0.0013
    0.2067    0.1029    0.0019   -0.0126
    0.0012    0.0028   -0.0009   -0.0002
    0.0073   -0.0181    0.0005   -0.0129
      ⋮
```

$$\widehat{\Phi}^T[m_{\mathrm{FF}}]\widehat{\Phi}\underline{\ddot{q}} + \widehat{\Phi}^T[c_{\mathrm{FF}}]\widehat{\Phi}\underline{\dot{q}} + \widehat{\Phi}^T[k_{\mathrm{FF}}]\widehat{\Phi}\underline{q} = \widehat{\Phi}^T\underline{F_F}(t)$$

```
M_Phi_4 = Phi_4'*MFF*Phi_4
```

```
M_Phi_4 = 4×4
10^3 ×
    0.4549   -0.0000    0.0000    0.0000
   -0.0000    1.5867   -0.0000   -0.0000
    0.0000   -0.0000    0.0817    0.0000
    0.0000   -0.0000    0.0000    0.1460
```

```
K_Phi_4 = Phi_4'*KFF*Phi_4
```

```
K_Phi_4 = 4×4
10^5 ×
    0.3253   -0.0000    0.0000    0.0000
```

```
   -0.0000    1.5756   -0.0000    0.0000
    0.0000   -0.0000    0.5195    0.0000
    0.0000    0.0000    0.0000    2.4464
```

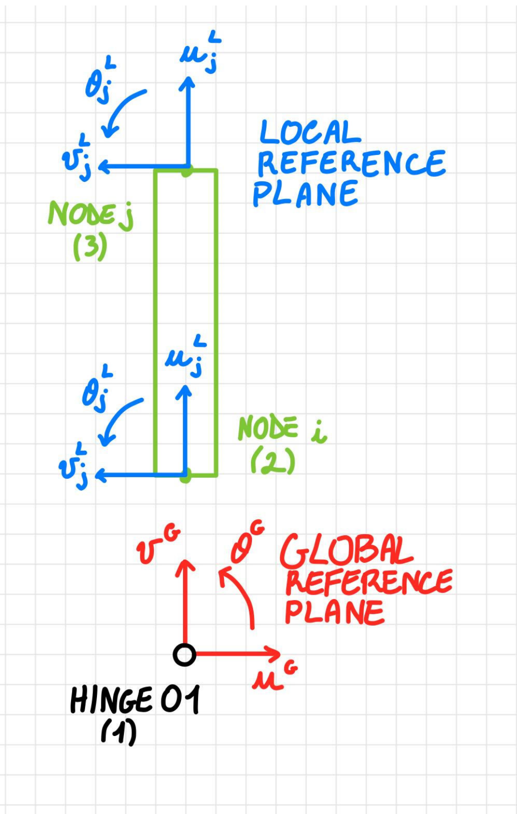
```
C_Phi_4 = Phi_4'*CFF*Phi_4
```

```
C_Phi_4 = 4×4
   81.6549   30.2853   -0.1159    0.7870
   30.2853   87.1036   -0.3661    2.5911
   -0.1159   -0.3661   37.1196    0.0117
    0.7870    2.5911    0.0117  126.9741
```

Bending Moment Analysis

In order to compute the Bending Moment for any of the beams of the system, it is necessary to translate the global reference plane to the local reference plane of each beam

The rotation required is shown in the following image



This is a rotation of 90 degrees around the Z axis, and the relationship between local and global coordinates is

$$\underline{x}^L = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underline{x}^G$$

```
rotMatrix = [0 1 0;
            -1 0 0;
             0 0 1]
```

```
rotMatrix = 3×3
     0     1     0
    -1     0     0
     0     0     1
```

Once the Local Displacements of the nodes are obtained, the shape functions for the bending moment can be used to obtain the bending displacement of the beam element

$$v(\xi, t) = f_v(\xi)^T \underline{x_k}^L$$

With x_k^L having the six nodal coordinates associated to the i-th and j-th node of the beam (2 and 3)

$$\underline{x_k}^L = \begin{bmatrix} \underline{x_i}^L \\ \underline{x_j}^L \end{bmatrix}$$

The shape function for bending motion is

$$f_v(\xi) = \begin{bmatrix} 0 \\ 2\left(\frac{\xi}{L_k}\right)^3 - 3\left(\frac{\xi}{L_k}\right)^2 + 1 \\ L_k\left[\left(\frac{\xi}{L_k}\right)^3 - 2\left(\frac{\xi}{L_k}\right)^2 + \frac{\xi}{L_k}\right] \\ 0 \\ -2\left(\frac{\xi}{L_k}\right)^3 + 3\left(\frac{\xi}{L_k}\right)^2 \\ L_k\left[\left(\frac{\xi}{L_k}\right)^3 - \left(\frac{\xi}{L_k}\right)^2\right] \end{bmatrix}$$

Finally, in order to obtain the final bending moment on the required part of the beam, the previous equation can be derived two times w.r.t ξ

$$M(\xi, t) = \mathrm{EI}\frac{\partial^2}{\partial \xi^2} v(\xi, t) = \mathrm{EI}\frac{\partial^2}{\partial \xi^2} f_v(\xi) \underline{x_k}^L$$

```
syms xi
f_v = [0;
       2*((xi/L_k)^3)-3*((xi/L_k)^2)+1;
       L_k*((xi/L_k)^3-2*((xi/L_k)^2)+(xi/L_k));
       0;
       -2*((xi/L_k)^3)+3*((xi/L_k)^2);
       L_k*((xi/L_k)^3-((xi/L_k)^2))]
```

f_v =

$$\begin{pmatrix} 0 \\ \frac{\xi^3}{108} - \frac{\xi^2}{12} + 1 \\ \frac{\xi^3}{36} - \frac{\xi^2}{3} + \xi \\ 0 \\ \frac{\xi^2}{12} - \frac{\xi^3}{108} \\ \frac{\xi^3}{36} - \frac{\xi^2}{6} \end{pmatrix}$$

```
df_vdxi = diff(f_v,xi);
d2f_vddxi = diff(df_vdxi,xi)
```

d2f_vddxi =

$$\begin{pmatrix} 0 \\ \frac{\xi}{18}-\frac{1}{6} \\ \frac{\xi}{6}-\frac{2}{3} \\ 0 \\ \frac{1}{6}-\frac{\xi}{18} \\ \frac{\xi}{6}-\frac{1}{3} \end{pmatrix}$$

As the analized part is really near to the i-th node of the beam element, we have to evaluate the derivatives at $\xi = 0$

```
f_vdd = double(subs(d2f_vddxi, xi, 0))
```

```
f_vdd = 6×1
         0
   -0.1667
   -0.6667
         0
    0.1667
   -0.3333
```

FRF for the Vertical Force Input in point A

```

% Initial values
F = zeros(length(MFF),1);

% Vertical Force in Node A
forcedNode = 16;
F(idb(forcedNode,2)) = Force;
F_Phi_4 = Phi_4'*F
```

```
F_Phi_4 = 4×1
   -1.8823
   -9.2983
  893.6881
  204.9100
```

```

% FRF Calculation
i=sqrt(-1);
vect_f=0:0.01:10;

% Memory preallocation
y_A = zeros(size(vect_f));
ydd_A = zeros(size(vect_f));
H_0_1 = zeros(size(vect_f));

for j=1:length(vect_f)
    ome=vect_f(j)*2*pi;
```

```
    A=-ome^2*M_Phi_4+i*ome*C_Phi_4+K_Phi_4;
    q=A\F_Phi_4;

    % Inverse transform
    x_f = Phi_4*q;
    x_fd = i*ome.*x_f;
    x_fdd = -ome^2.*x_f;

    % Constraint Forces Calculation
    F_c = MCF*x_fdd + CCF*x_fd + KCF*x_f;

    % Solutions
    y_A(j) = x_f(idb(nodeA,2));
    ydd_A(j) = x_fdd(idb(nodeA,2));
    H_O_1(j) = F_c(idb(hinge1,2)-ndgf);
end
```

FRF for the Horizontal Force Input in point A

```
%...............................................
% Frequency Response Function (FRF) For Horizontal External Force at point A

% Initial values
F = zeros(length(MFF),1);
% Vertical Force in Node A
forcedNode = 16;
F(idb(forcedNode,1)) = Force;
F_Phi_4 = Phi_4'*F
```

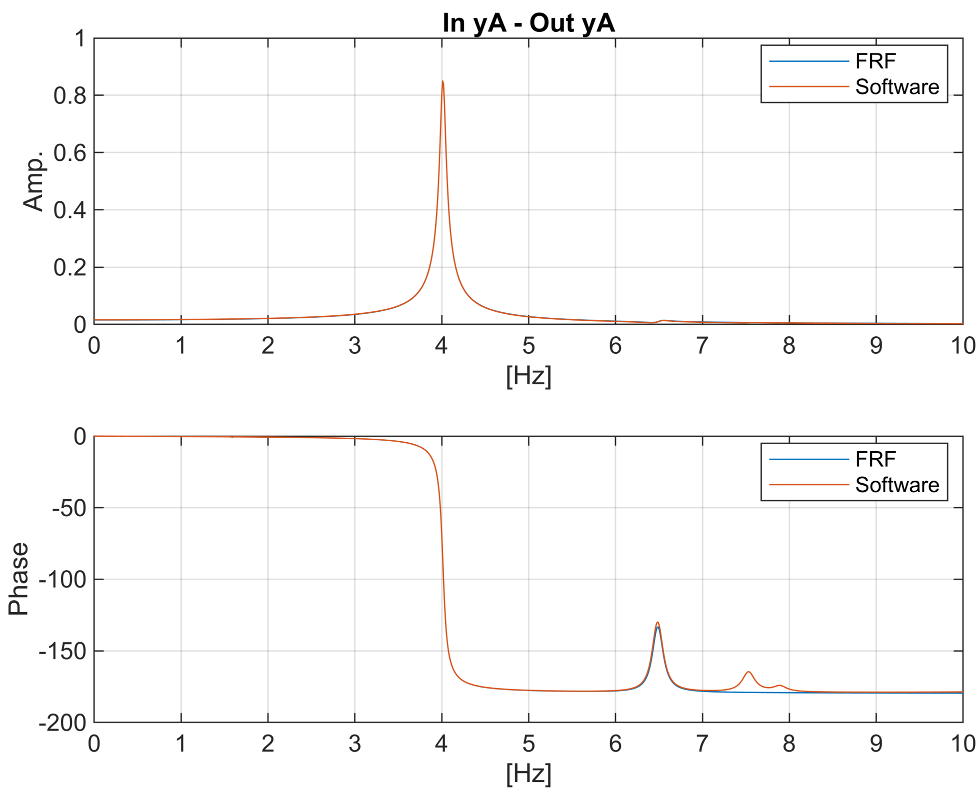
```
F_Phi_4 = 4×1
  207.8358
  103.4601
    5.5569
   -5.9247
```

```
% FRF Calculation
i=sqrt(-1);
vect_f=0:0.01:10;

% Memory preallocation
x_B = zeros(size(vect_f));
H_O_2 = zeros(size(vect_f));

for j=1:length(vect_f)
    ome=vect_f(j)*2*pi;
    A=-ome^2*M_Phi_4+i*ome*C_Phi_4+K_Phi_4;
    q=A\F_Phi_4;

    % Inverse transform
    x_f = Phi_4*q;
    x_fd = i*ome.*x_f;
    x_fdd = -ome^2.*x_f;

    % Constraint Forces Calculation
    F_c = MCF*x_fdd + CCF*x_fd + KCF*x_f;
```

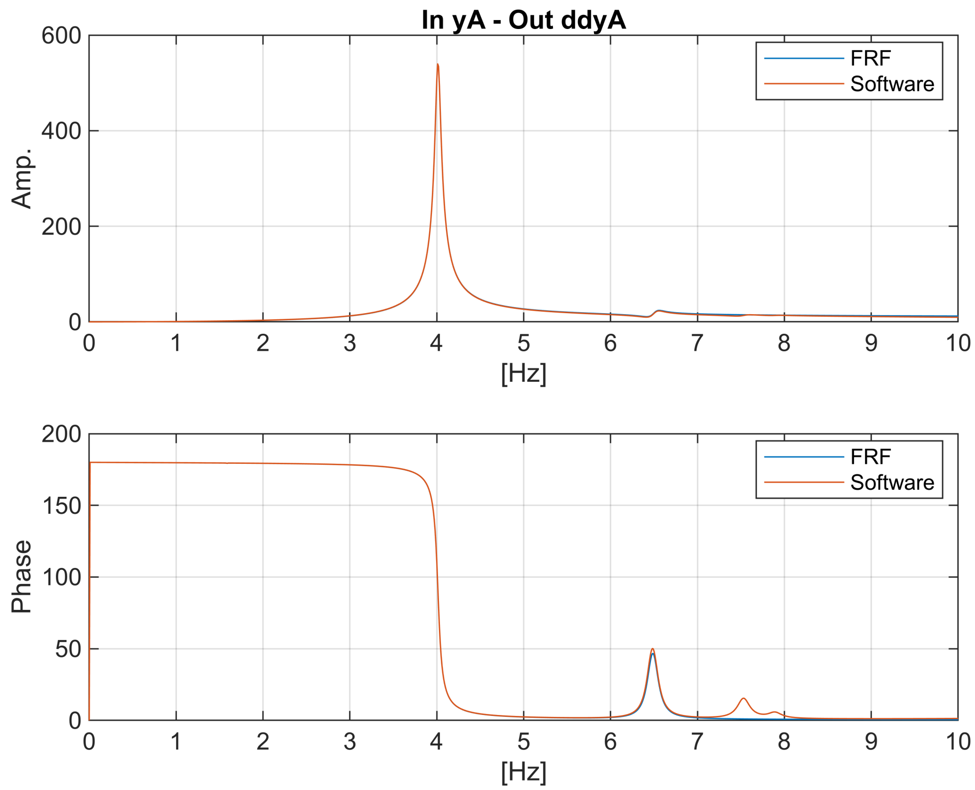
```
    % Bending Moment
    x_i_L = rotMatrix*x_f(idb(nodeBeam_ith,1):idb(nodeBeam_ith,3));
    x_j_L = rotMatrix*x_f(idb(nodeBeam_jth,1):idb(nodeBeam_jth,3));
    x_k_L = [x_i_L; x_j_L];

    % Solutions
    x_B(j) = x_f(idb(nodeB,1));
    H_O_2(j) = F_c(idb(hinge2,2)-ndgf);

    EJ_IPE_550 =(6.712E-4)*2.06E11;
    M_C_C(j) = EJ_IPE_550*f_vdd'*x_k_L;
end
```

Bode Diagrams

```
% In yA - Out yA
[xData, yData] = extract_all_graph_data('vA_vA.fig');
figure
subplot 211;plot(vect_f,abs(y_A));grid;xlabel('[Hz]');ylabel('Amp.');title('In
yA - Out yA')
hold on
plot(xData{1}, yData{2})
legend({'FRF', 'Software'}, 'Location', 'best');
subplot 212;plot(vect_f,(180/pi)*angle(y_A));grid;xlabel('[Hz]');ylabel('Phase')
hold on
plot(xData{2}, yData{1})
legend({'FRF', 'Software'}, 'Location', 'best');
```

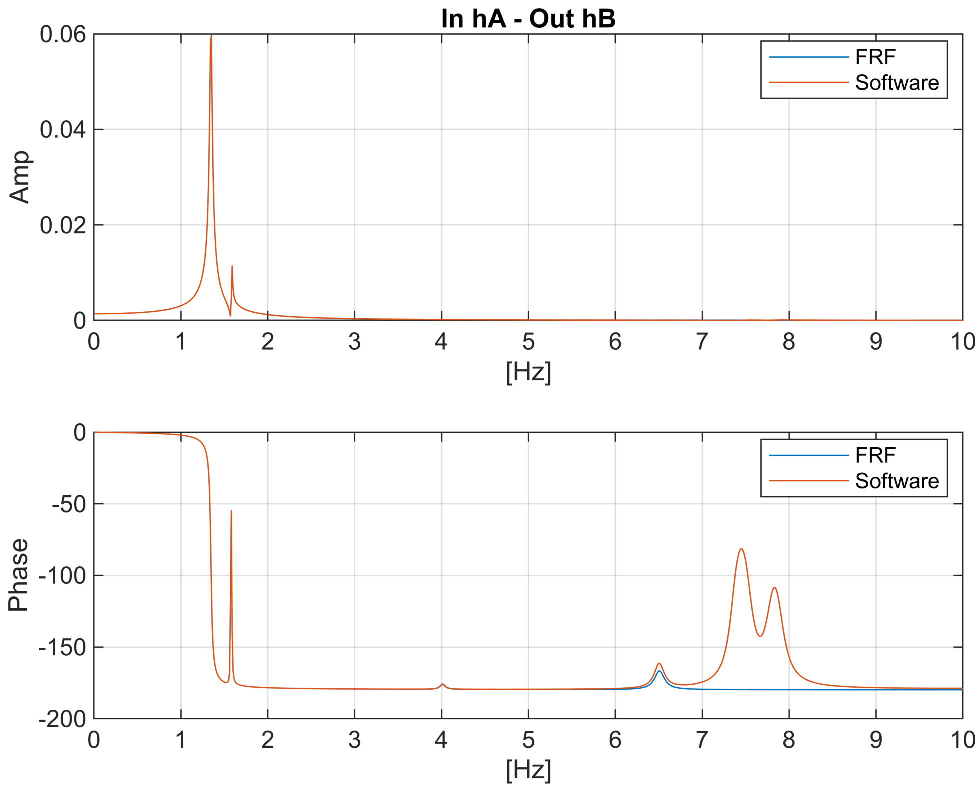


```
% In yA - Out ddyA
[xData, yData] = extract_all_graph_data('vA_accvA.fig');
figure
grid on
subplot 211;plot(vect_f,abs(ydd_A));grid;xlabel('[Hz]');ylabel('Amp.');title('In
yA - Out ddyA')
hold on
plot(xData{1}, yData{2})
legend({'FRF', 'Software'}, 'Location', 'best');
subplot 212;plot(vect_f,(180/
pi)*angle(ydd_A));grid;xlabel('[Hz]');ylabel('Phase')
hold on
plot(xData{2}, yData{1})
legend({'FRF', 'Software'}, 'Location', 'best');
```



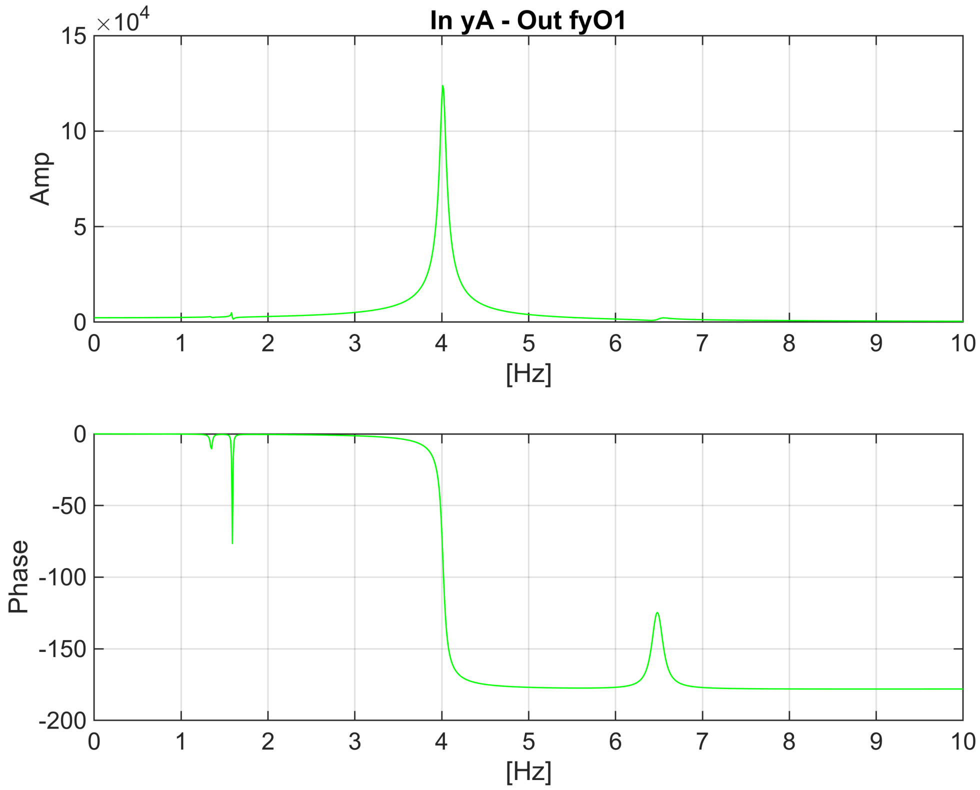

```
% In hA - Out hB
[xData, yData] = extract_all_graph_data('hA_hB.fig');
figure
grid on
subplot 211;plot(vect_f,abs(x_B));grid;xlabel('[Hz]');ylabel('Amp');title('In hA
- Out hB')
hold on
plot(xData{1}, yData{2})
legend({'FRF', 'Software'}, 'Location', 'best');
subplot 212;plot(vect_f,(180/pi)*angle(x_B));grid;xlabel('[Hz]');ylabel('Phase')
hold on
plot(xData{2}, yData{1})
```

```
legend({'FRF', 'Software'}, 'Location', 'best');
```

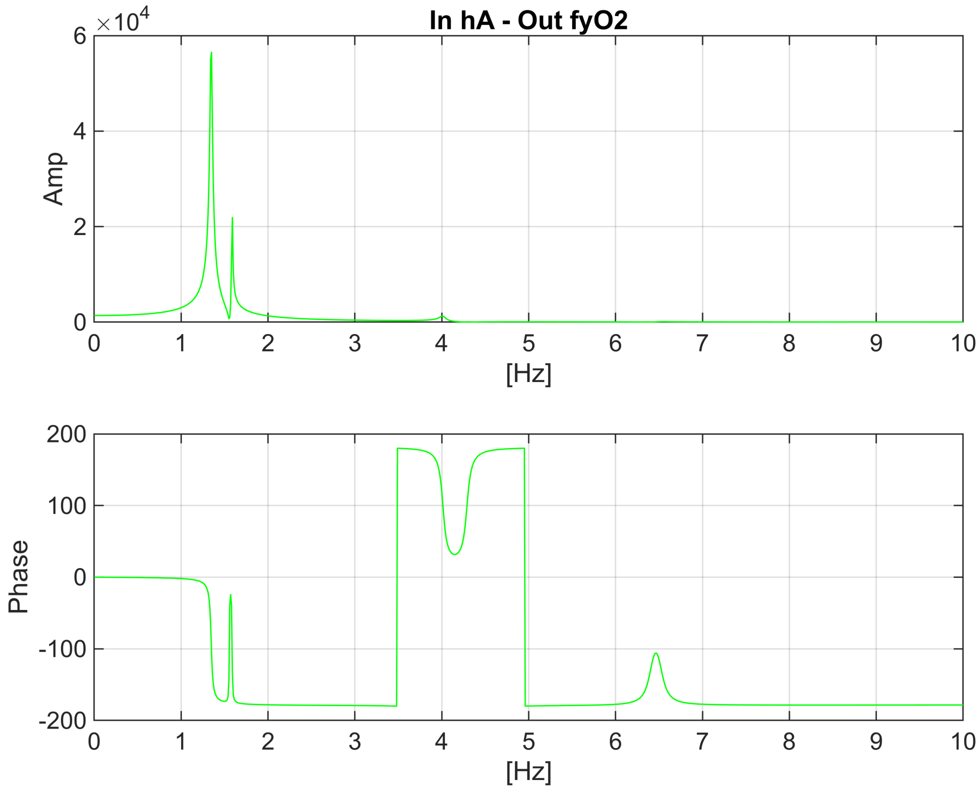


Point 6)

```
% In yA - Out fyO1
figure
grid on
subplot
211;plot(vect_f,abs(H_O_1),'g');grid;xlabel('[Hz]');ylabel('Amp');title('In yA -
Out fyO1')
subplot 212;plot(vect_f,(180/
pi)*angle(H_O_1),'g');grid;xlabel('[Hz]');ylabel('Phase')
```

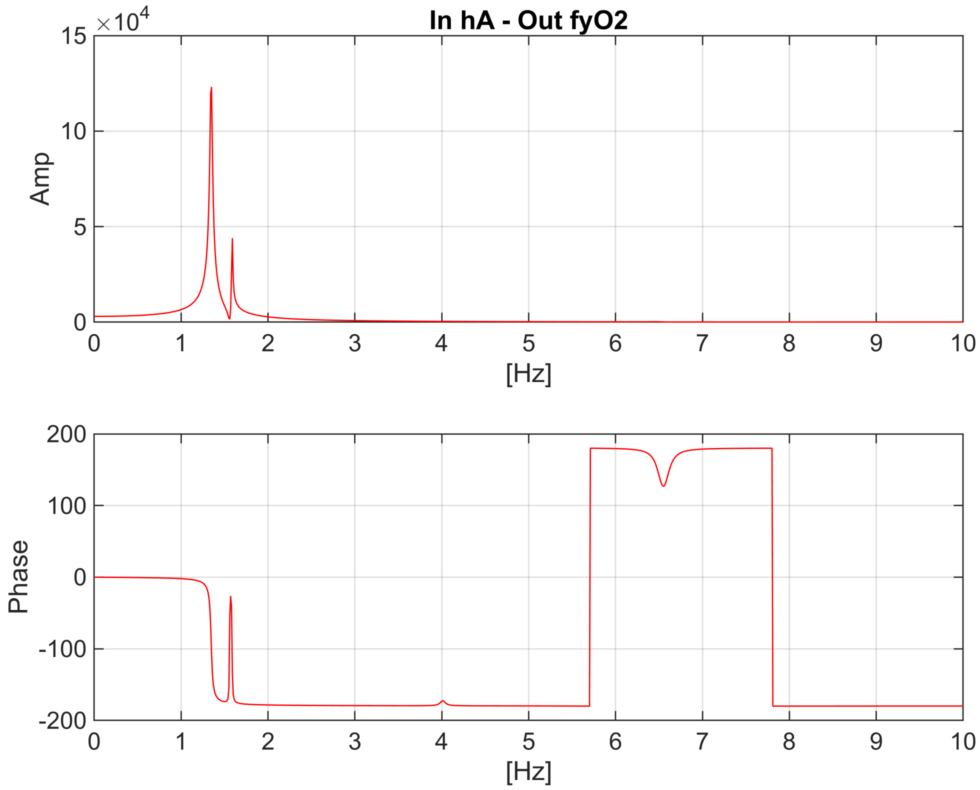



```
% In hA - Out fyO2
figure
grid on
subplot
211;plot(vect_f,abs(H_O_2),'g');grid;xlabel('[Hz]');ylabel('Amp');title('In hA -
Out fyO2')
subplot 212;plot(vect_f,(180/
pi)*angle(H_O_2),'g');grid;xlabel('[Hz]');ylabel('Phase')
```



Point 7)

```
% In hA - Out fyO2
figure
grid on
subplot
211;plot(vect_f,abs(M_C_C),'r');grid;xlabel('[Hz]');ylabel('Amp');title('In hA -
Out fyO2')
subplot 212;plot(vect_f,(180/
pi)*angle(M_C_C),'r');grid;xlabel('[Hz]');ylabel('Phase')
```


In hA - Out fyO2
×10^4
15
10
5
0
Amp
0
1
2
3
4
5
6
7
8
9
10
[Hz]
200
100
0
-100
-200
Phase
0
1
2
3
4
5
6
7
8
9
10
[Hz]