DMS YEARWORK 2023/2024

Author: Andrea Ricciardi (10931392)

Question 1

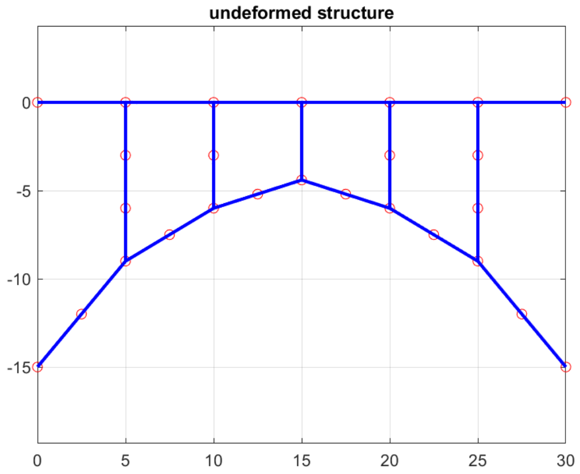

Figure 1: Undeformed Structure

Question 2

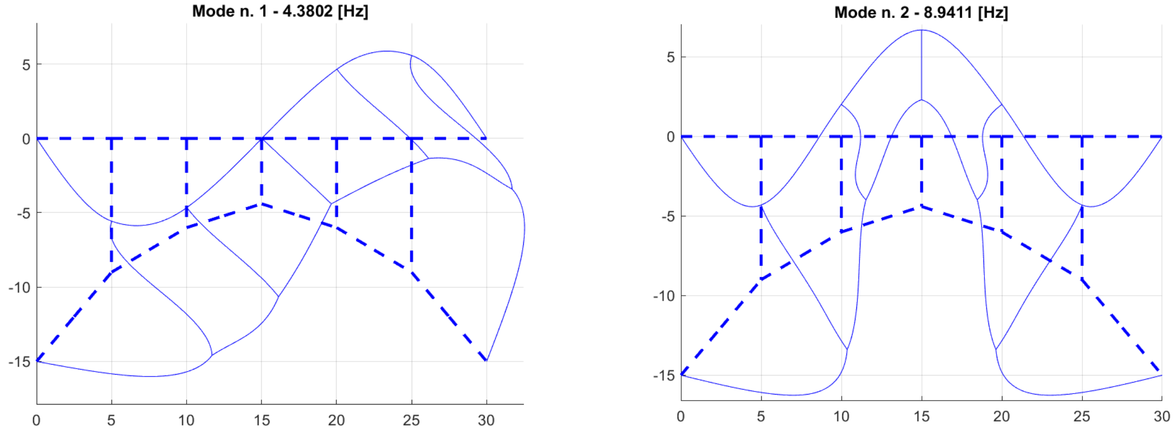

Figure 2: Modes 1 and 2

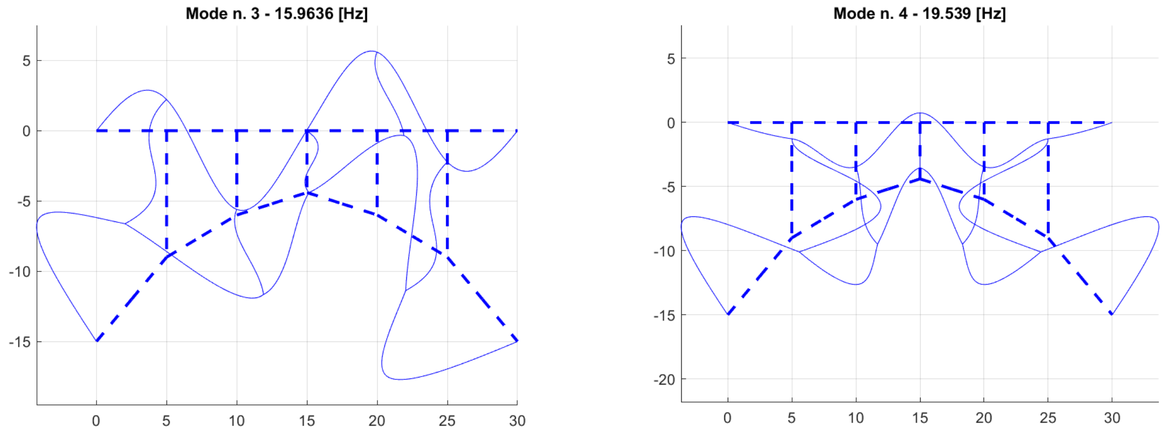

Figure 3: Modes 3 and 4

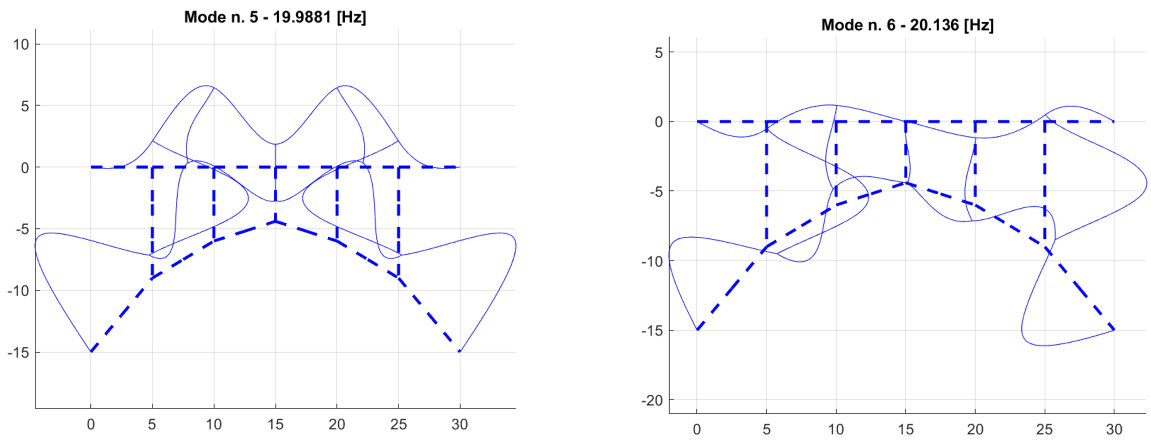

Figure 4: Modes 5 and 6

Question 3

```matlab
%% LOAD DATA

load("project_mkr.mat");

modal_matrix = load("modes.mat").phi;
natural_freq = load("modes.mat").freq;

%% MATRIX PARTITIONING

n = length(idb)*3;
n_doc = 4*2; % 4 hinges -> 8 doc
n_dof = n - n_doc;

MFF = M(1:n_dof, 1:n_dof);
CFF = R(1:n_dof, 1:n_dof);
KFF = K(1:n_dof, 1:n_dof);

MFC = M(1:n_dof, n_dof+1:n);
CFC = R(1:n_dof, n_dof+1:n);
KFC = K(1:n_dof, n_dof+1:n);

MCF = M(n_dof+1:n, 1:n_dof);
CCF = R(n_dof+1:n, 1:n_dof);
KCF = K(n_dof+1:n, 1:n_dof);

MCC = M(n_dof+1:n, n_dof+1:n);
CCC = R(n_dof+1:n, n_dof+1:n);
KCC = K(n_dof+1:n, n_dof+1:n);

%% ADIMENTIONAL DAMPING RATIOS and DAMPED FREQUENCIES

Mmod = modal_matrix'*MFF*modal_matrix;
Kmod = modal_matrix'*KFF*modal_matrix;
Cmod = modal_matrix'*CFF*modal_matrix;

alpha = 0.8;
beta = 3.0e-5;

for i=1:length(natural_freq)
    omega = 2*pi*natural_freq(i);
    % adimensional damping ratios (from alpha and beta):
    h(i) = alpha/(2*omega) + (beta*omega)/2;
    % adimensional damping ratios (from modal damping matrix):
    %h(i) = Cmod(i,i) / (2*Mmod(i,i)*omega);
    % damped natural frequencies (wd=wn*sqrt(1-h^2)):
    damped_freq(i) = natural_freq(i)*sqrt(1-h(i)^2);
end

% undamped natural frequencies:
undamped_freq = natural_freq(natural_freq <= 24)'
% get the frequencies up to 24Hz:
damped_freq = damped_freq(damped_freq <= 24)
% relative adimensional damping ratios:
h = h(1:length(damped_freq))
```

Figure 5: Computation of the damped frequencies and adimensional damping ratios

$$natural\_frequencies = [4.3802 \quad 8.9411 \quad 15.9636 \quad 19.5390 \quad 19.9881 \quad 20.1360]$$

$$damped\_frequencies = [4.3797 \quad 8.9409 \quad 15.9634 \quad 19.5387 \quad 19.9878 \quad 20.1358]$$

$$h = [0.0149 \quad 0.0080 \quad 0.0055 \quad 0.0051 \quad 0.0051 \quad 0.0051]$$

Question 4

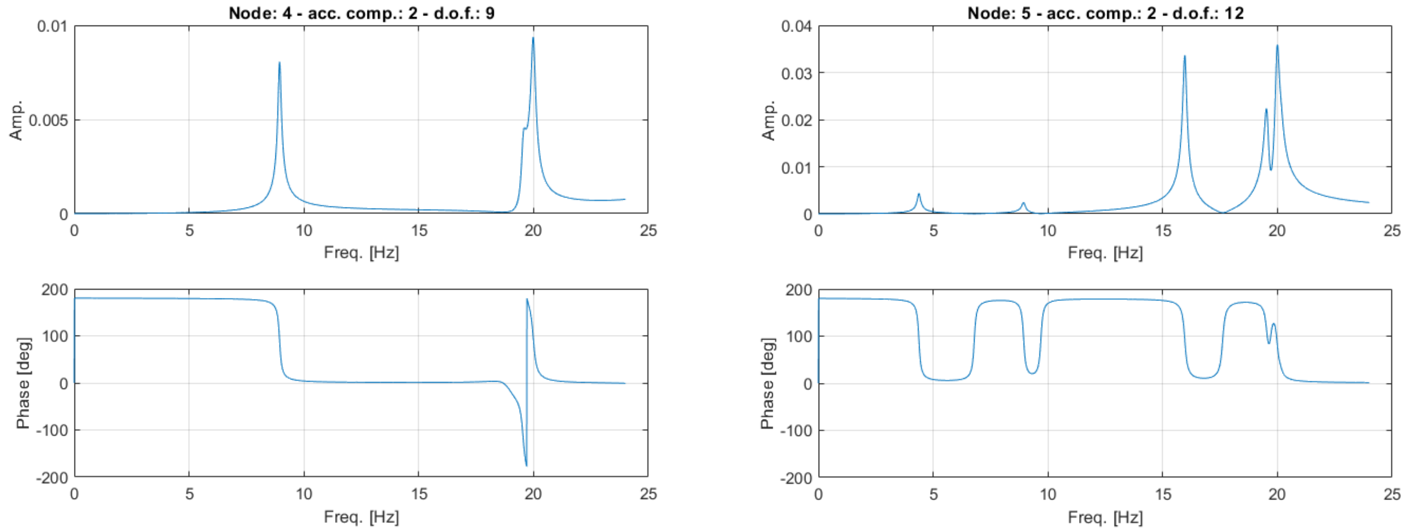

Figure 6: Vertical acceleration of point A and B to a vertical force applied in point B

Question 5

```
%% LOAD DATA

% load system matrices:
load("matrices.mat");

% load FRF of point A and B calculated on the previous point:
xA = load("FRFA.mat").x;
xB = load("FRFB.mat").x;
vect_freq = load("FRFA.mat").fre;

modal_matrix = load("modes.mat").phi; % modal matrix
natural_freq = load("modes.mat").freq; % natural frequencies

%% MATRIX PARTITIONING

n = length(idb)*3;
n_doc = 4*2; % 4 hinges -> 8 doc
n_dof = n - n_doc;

MFF = M(1:n_dof, 1:n_dof);
CFF = R(1:n_dof, 1:n_dof);
KFF = K(1:n_dof, 1:n_dof);

MFC = M(1:n_dof, n_dof+1:n);
CFC = R(1:n_dof, n_dof+1:n);
KFC = K(1:n_dof, n_dof+1:n);

MCF = M(n_dof+1:n, 1:n_dof);
CCF = R(n_dof+1:n, 1:n_dof);
KCF = K(n_dof+1:n, 1:n_dof);

MCC = M(n_dof+1:n, n_dof+1:n);
CCC = R(n_dof+1:n, n_dof+1:n);
KCC = K(n_dof+1:n, n_dof+1:n);

```

```
%% MODAL APPROACH

% extract the first three modes:
modal_matrix = modal_matrix(:, 1:3);

% modal matrices:
Mmod = modal_matrix'*MFF*modal_matrix;
Kmod = modal_matrix'*KFF*modal_matrix;
Cmod = modal_matrix'*CFF*modal_matrix;

% damping modal matrix (alternative approach):
% alpha = 0.8;
% beta = 3.0e-5;
%
% for k=1:3
%     Cmod(k,k) = alpha*Mmod(k,k) + beta*Kmod(k,k);
% end

F = zeros(length(natural_freq), 1);
F(idb(5, 2)) = 1; % force applied on node 5 in vertical direction: idb(5,2)
Fmod = modal_matrix'*F;

% modes superimposition:
i = sqrt(-1);
for k=1:length(vect_freq)
    % compute the modal coordinates:
    omega = 2*pi*vect_freq(k);
    A = (-omega^2*Mmod + i*omega*Cmod + Kmod);
    q = A \ Fmod;

    % superimposition response:
    x = modal_matrix*q;
    xdd = -omega^2*x;

    out1 = xdd(idb(4,2)); % vertical acceleration of point A (node4)
    out2 = xdd(idb(5,2)); % vertical acceleration of point B (node5)

    mod1(k) = abs(out1);
    fas1(k) = angle(out1);

    mod2(k) = abs(out2);
    fas2(k) = angle(out2);
end
```

Figure 7: Vertical acceleration of point A and B given a vertical force applied in point B (Modal Approach with only the first three modes)

```matlab
%% PLOT THE RESULT

% import old results
for k=1:length(vect_freq)
    omega = 2*pi*vect_freq(k);

    xAdd = -omega^2*xA(idb(4,2),k);
    xBdd = -omega^2*xB(idb(5,2),k);

    modA(k) = abs(xAdd);
    fasA(k) = angle(xAdd);

    modB(k) = abs(xBdd);
    fasB(k) = angle(xBdd);
end

% acceleration A:
figure

subplot 211;
plot(vect_freq, modA, 'b');
hold on;
plot(vect_freq, mod1, 'r');
grid; xlabel('[Hz]'); ylabel('Amp.');
title('F/accA');
legend('system response', 'modal approach');

subplot 212;
plot(vect_freq, fasA*180/pi, 'b');
hold on;
plot(vect_freq, fas1*180/pi, 'r');
grid; xlabel('[Hz]'); ylabel('Phase [deg]');
legend('system response', 'modal approach');

% acceleration B:
figure

subplot 211;
plot(vect_freq, modB, 'b');
hold on;
plot(vect_freq, mod2, 'r');
grid; xlabel('[Hz]'); ylabel('Amp.');
title('F/accB');
legend('system response', 'modal approach');

subplot 212;
plot(vect_freq, fasB*180/pi, 'b');
hold on;
plot(vect_freq, fas2*180/pi, 'r');
grid; xlabel('[Hz]'); ylabel('Phase [deg]');
legend('system response', 'modal approach');
```

Figure 8: Plotting (Question 5)

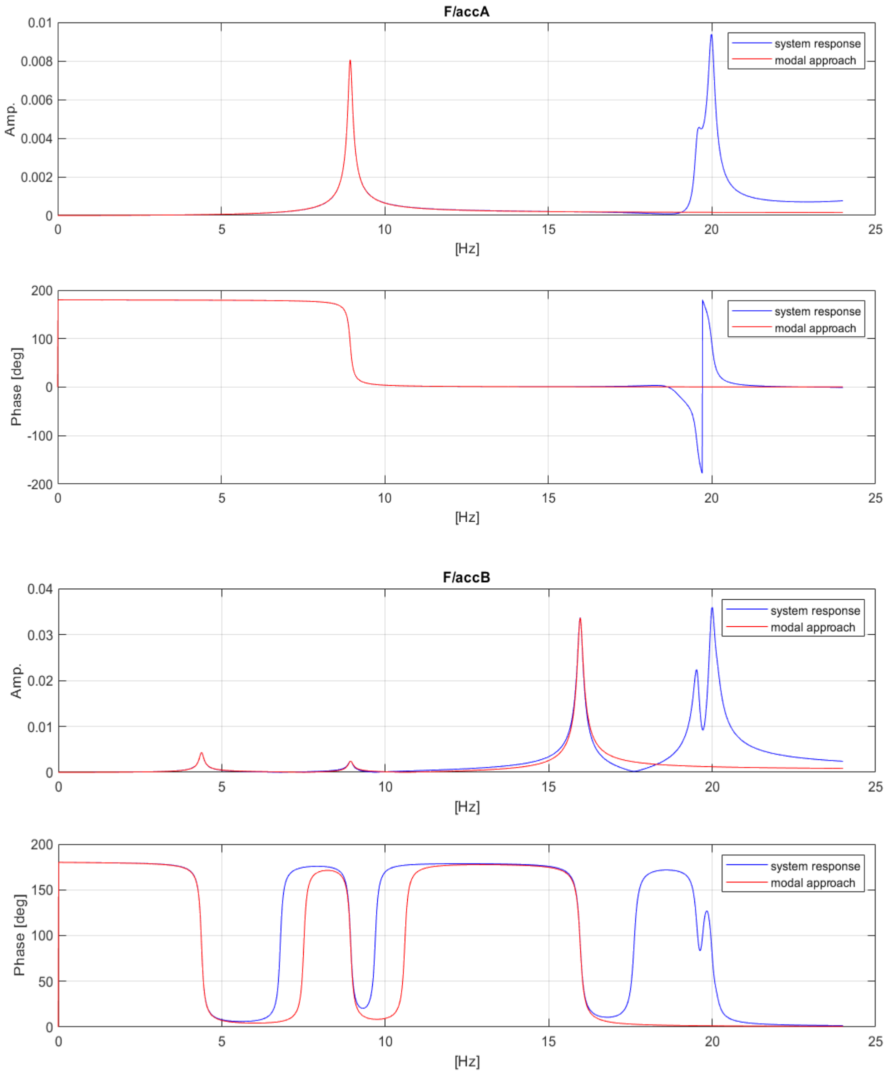

Figure 9: Obtained Results with comparisons (Question 5)

Question 6

```matlab
%% LOAD DATA

% numerical data:
E = 2.06e11;
I = 2.313e-4;
EI = E*I;
rho = 7800;
A = 8.446e-3;
m = rho*A;
L = 5;

load("matrices.mat"); % import idb matrix

% import frequency responses of nodes A and B:
xA = load("FRFA.mat").x;
xB = load("FRFB.mat").x;
vect_freq = load("FRFA.mat").fre;

figure
subplot 311; plot(vect_freq, abs(xA(idb(4,2),:))); grid; title("wA");
subplot 312; plot(vect_freq, abs(xB(idb(5,2),:))); grid; title("wB");

% verify if wC is equal to the interpolated value:
wA = xA(idb(4,2),:);
wB = xB(idb(5,2),:);
thetaA = xA(idb(4,3),:);
thetaB = xB(idb(5,3),:);
wC = (wA + wB)/2 + (L/8)*(thetaA - thetaB); % response with interpolation

subplot 313; plot(vect_freq, abs(wC)); grid; title("wC");
```

```matlab
%% FRF (BENDING MOMENT IN C)

% FRF responses of nodes A and B:
wA = xA(idb(4,2),:);
wB = xB(idb(5,2),:);
thetaA = xA(idb(4,3),:);
thetaB = xB(idb(5,3),:);

% second derivative of shape functions:
xsi = L/2;
f1 = 12*xsi/(L^3)-6/(L^2);
f2 = 6*xsi/(L^2)-4/L;
f3 = -12*xsi/(L^3)+6/(L^2);
f4 = 6*xsi/(L^2)-2/L;

% calculate displacement of C by interpolating with shape functions:
wCpp = f1*wA + f2*thetaA + f3*wB + f4*thetaB; % wCpp = -(1/L)*thetaA + (1/L)*thetaB;
Mc = EI*wCpp; % resultant bending moment

% plot the result:
figure
subplot 211; plot(vect_freq, abs(Mc)); grid;
title("Bending moment at node C given force F at B"); xlabel("Freq. [Hz]"); ylabel("Amplitude [Nm/N]")
subplot 212; plot(vect_freq, angle(Mc)/pi*180); grid;
xlabel("Freq. [Hz]"); ylabel("Phase [°]")
```

Figure 10: Bending Moment in point C given a vertical force applied in point B

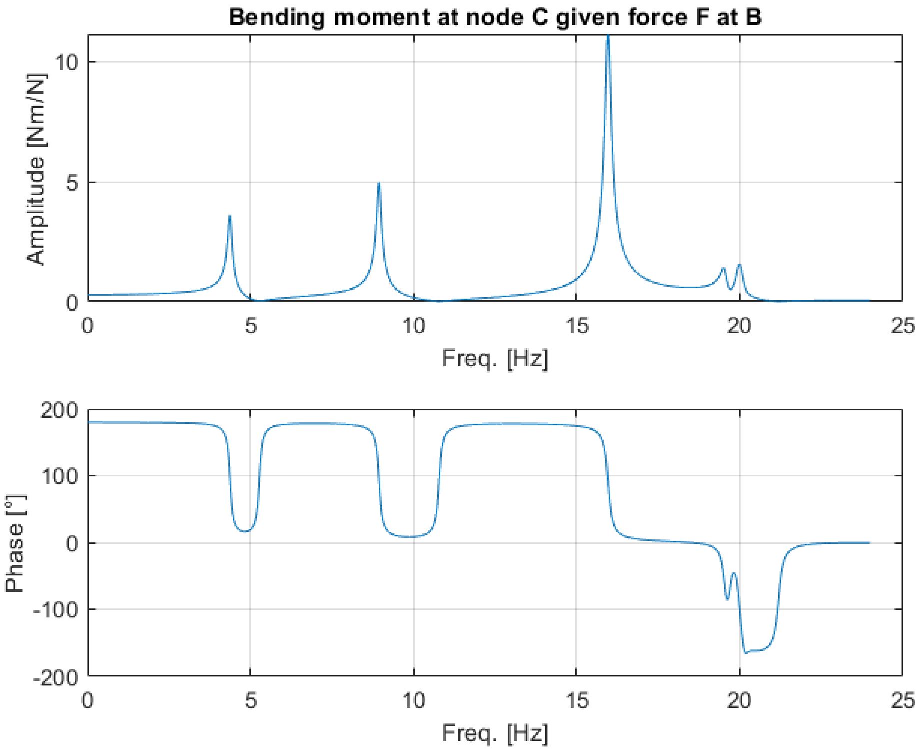

Figure 11: Bending moment in C given a vertical force in B

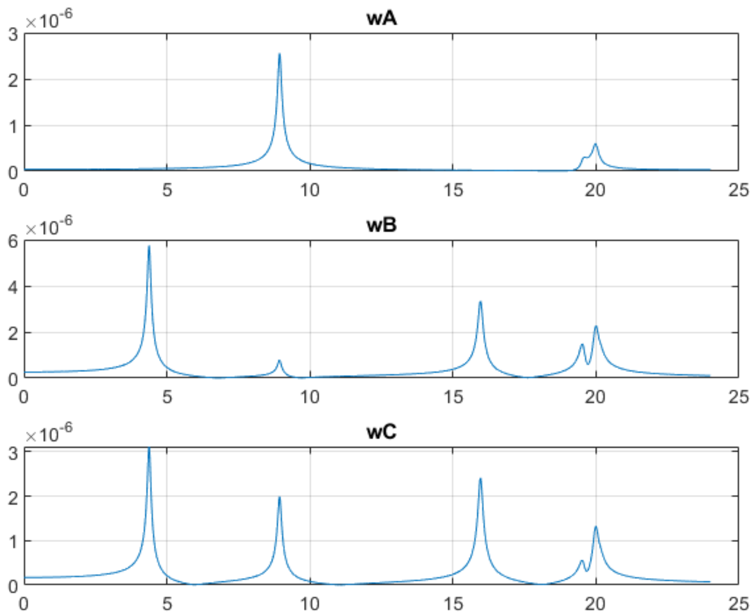

Figure 12: Vertical displacements in several points (Interplatiion for point C)

Question 7

```
%% LOAD DATA

% get constraint displacements:
data = load("seismic_displ.txt");
time = data(:, 1);
xc1 = data(:, 2); % displacement of points O11/O12
xc2 = data(:, 3); % displacement of points O21/O22

figure
subplot 211; plot(time, xc1); grid; xlabel("[s]"); ylabel("Y_{O_{11}/O_{12}} [m]"); title("Input seismic motion");
subplot 212; plot(time, xc2); grid; xlabel("[s]"); ylabel("Y_{O_{21}/O_{22}} [m]")

% get system matrices:
load("matrices.mat");
n = length(idb)*3;
n_doc = 4*2; % 4 hinges -> 8 doc
n_dof = n - n_doc;

MFF = M(1:n_dof, 1:n_dof);
CFF = R(1:n_dof, 1:n_dof);
KFF = K(1:n_dof, 1:n_dof);

MFC = M(1:n_dof, n_dof+1:n);
CFC = R(1:n_dof, n_dof+1:n);
KFC = K(1:n_dof, n_dof+1:n);

MCF = M(n_dof+1:n, 1:n_dof);
CCF = R(n_dof+1:n, 1:n_dof);
KCF = K(n_dof+1:n, 1:n_dof);

MCC = M(n_dof+1:n, n_dof+1:n);
CCC = R(n_dof+1:n, n_dof+1:n);
KCC = K(n_dof+1:n, n_dof+1:n);

```

Figure 13: Data Load (Question 7)

```
%% CONVERT TO FREQUENCY DOMAIN INPUT SIGNALS

% Sampling parameters:
dt = time(2)-time(1);    % Sampling distance
fs = 1/dt;               % Sampling frequency
N = length(time);        % Length of the signal (number of samples)
T = N/fs;                % Period of sampling
freq = (0:N-1)*(fs/N);   % Frequency vector (== (0:N-1)*(1/T))

% Perform FFT:
Xc1 = fft(xc1);
Xc2 = fft(xc2);

% Analysis of the spectrum of the signals (actually one-sided spectrums: values up to k=N/2-1):
Xc1_half = Xc1(1:N/2);
Xc2_half = Xc2(1:N/2);
freq_half = freq(1:N/2);
```

```
figure;
subplot 321; plot(time, xc1); grid;
title('Time-Domain Signal Y_{O11/O12}'); xlabel('Time [s]'); ylabel('Amplitude [m]');
subplot 323;
plot(freq_half, abs(Xc1_half)); grid;
title('Frequency-Domain Signal (Amplitude Spectrum)'); xlabel('Frequency [Hz]'); ylabel('Amplitude');
subplot 325;
plot(freq_half, angle(Xc1_half)); grid;
title('Frequency-Domain Signal (Phase Spectrum)'); xlabel('Frequency [Hz]'); ylabel('Phase');
subplot 322; plot(time, xc2); grid;
title('Time-Domain Signal Y_{O21/O22}'); xlabel('Time [s]'); ylabel('Amplitude  [m]');
subplot 324;
plot(freq_half, abs(Xc2_half)); grid;
title('Frequency-Domain Signal (Amplitude Spectrum)'); xlabel('Frequency [Hz]'); ylabel('Amplitude');
subplot 326;
plot(freq_half, angle(Xc2_half)); grid;
title('Frequency-Domain Signal (Phase Spectrum)'); xlabel('Frequency [Hz]'); ylabel('Phase');
```

Figure 14: Frequency Domain Conversion (FFT)

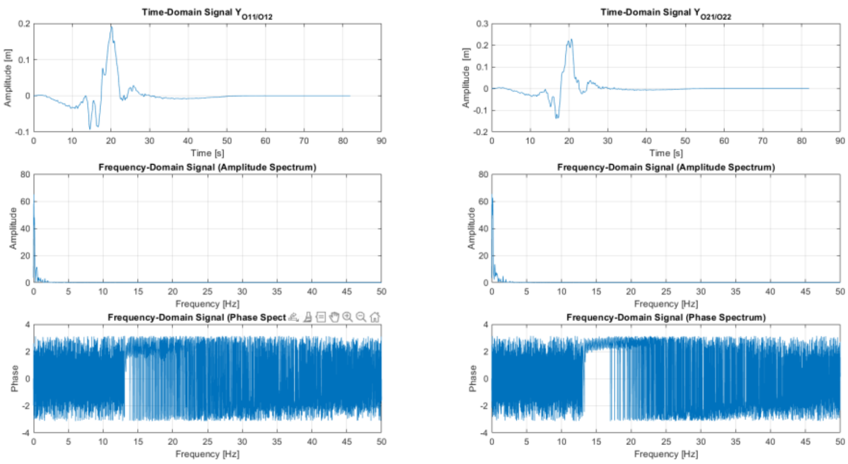

Figure 15: Time and Frequency domain plots of the Seismic Motion (Constraint displacement)

```matlab
%% FRF

i = sqrt(-1);

for k=1:(N/2)
    omega = 2*pi*freq(k); % omega = 2*pi*k/T

    A = -omega^2*MFF + i*omega*CFF + KFF;

    % constraint displacement vector (xc):
    x = zeros(n, 1);
    x(idb(1,2),1) = Xc1(k); % coordinate of vertical displacement of O11 (node 1)
    x(idb(8,2),1) = Xc1(k); % coordinate of vertical displacement of O12 (node 8)
    x(idb(7,2),1) = Xc2(k); % coordinate of vertical displacement of O21 (node 7)
    x(idb(20,2),1) = Xc2(k); % coordinate of vertical displacement of O22 (node 20)
    xc = x(n_dof+1:end, 1);

    FFC = -(-omega^2*MFC + i*omega*CFC + KFC)*xc;

    % system response:
    X = A \ FFC;
    XA(k) = X(idb(4,2));
    XAdd(k) = -omega^2*XA(k);
end

% spectrum of the signal:
XA = [XA(1) XA(2:end) fliplr(conj(XA(2:end)))];
XAdd = [XAdd(1) XAdd(2:end) fliplr(conj(XAdd(2:end)))];
% time series:
xA = ifft(XA);
xAdd = ifft(XAdd);
```

Figure 16: Computation of vertical displacement and acceleration of point A given the Seismic Motion (Constraint Displacement) in time domain

```matlab
% Confront response with inputs:
figure;
plot(time(1:end-1), xA); grid;
hold on;
plot(time, xc1); grid;
hold on;
plot(time, xc2); grid;
legend('y_A', 'O{11}/O{12}', 'O{21}/O{22}');
xlabel('Time [s]');ylabel('Displacement [m]');
title('Vertical displ. of A given the seism'); xlabel('Time [s]'); ylabel('Amplitude [m]');

% Plot the spectrum:
figure;
subplot 211; plot(freq(1:N/2+1), abs(XA(1:N/2+1))); grid;
title('Spectrum of the vertical displ. of A'); xlabel('Freq. [Hz]'); ylabel('Amplitude');
subplot 212; plot(freq(1:N/2+1), angle(XA(1:N/2+1))*180/pi); grid;
xlabel('Freq. [Hz]'); ylabel('Phase');

% Plot the time series:
figure;
subplot 211; plot(time(1:end-1), xA); grid;
title('Vertical displacement of A'); xlabel('Time. [s]'); ylabel('Amplitude [m]');
subplot 212; plot(time(1:end-1), xAdd); grid;
title('Vertical acceleration of A'); xlabel('Time. [s]'); ylabel('Amplitude [m/s^2]');
```

Figure 17: Plotting (Question 7)

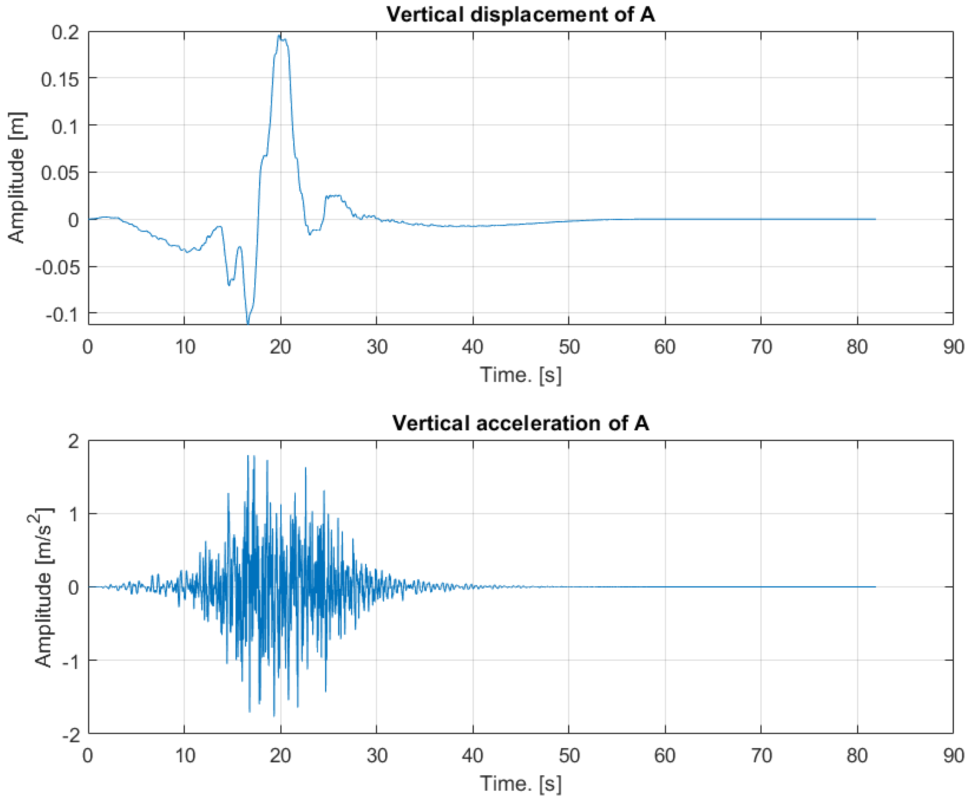

Figure 18: Vertical displacement and acceleration of point A in time domain

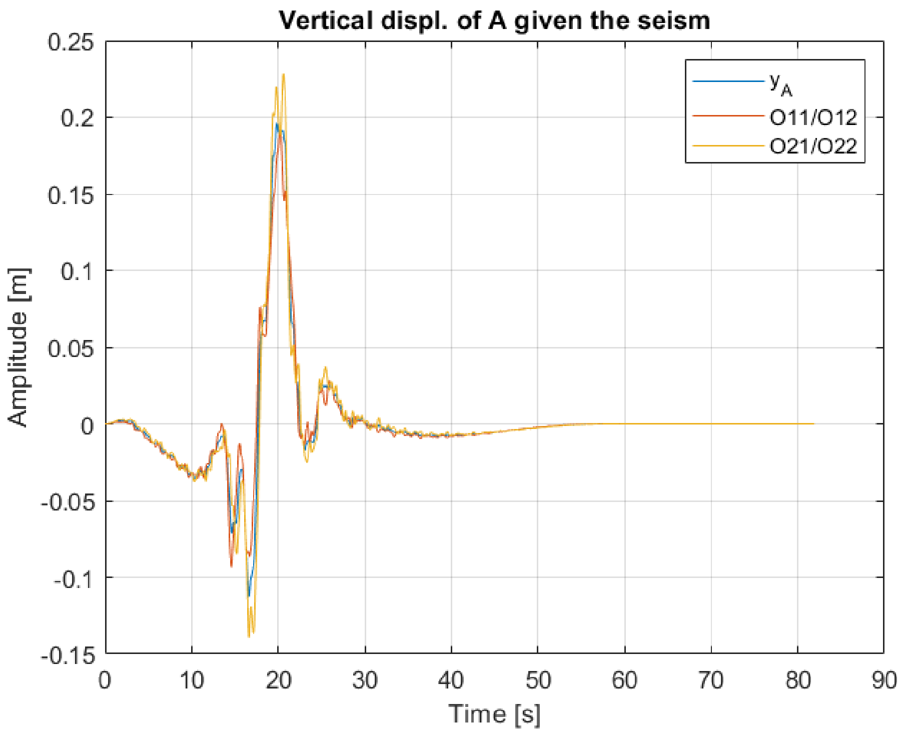

Figure 19: Vertical displacement of A vs Seismic Motion

Question 8

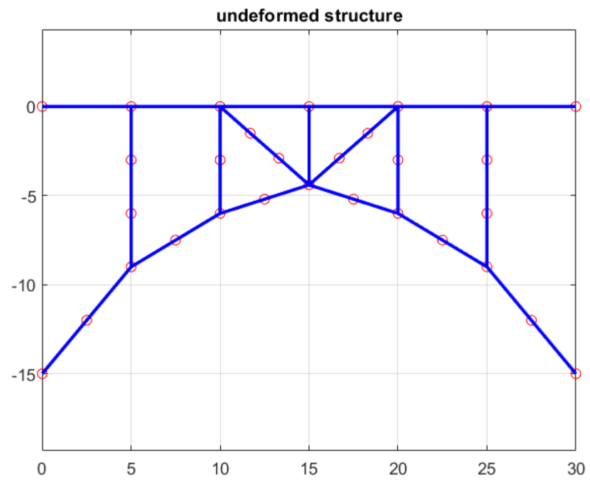

Figure 20: Changed Structure

```
%% COMPUTE TARGET AMPLITUDE

% import matrices:
load("matrices.mat");

% load the FRF data:
xB = load("FRFB.mat").x;
xB_reduced = load("FRFB_reduced.mat").x;
vect_freq = load("FRFB.mat").fre;


% compute the acceleration response:
for k=1:length(vect_freq)
    omega = 2*pi*vect_freq(k);

    xBdd = -omega^2*xB(idb(5,2),k); % compute acceleration
    modB(k) = abs(xBdd);

    xBdd_reduced = -omega^2*xB_reduced(idb(5,2),k);
    modB_reduced(k) = abs(xBdd_reduced);
end

figure
plot(vect_freq, modB); grid; title("FRF acc. B");

figure
plot(vect_freq, modB_reduced); grid; title("FRF acc. B reduced")

% get the maximum amplitude peak:
max_amp = max(modB)

% comute the target amplitude (-30%):
reduction = 0.3;
target_amp = max_amp*(1-reduction)

% maximum peak of the changed structure:
reduced_amp = max(modB_reduced)
```

Figure 21: Results Show-off Code (Question 8)

$$\max\ \_amp = 0.0359$$

$$target\_amp = 0.0251$$

$$reduced\_amp\ = 0.0194$$

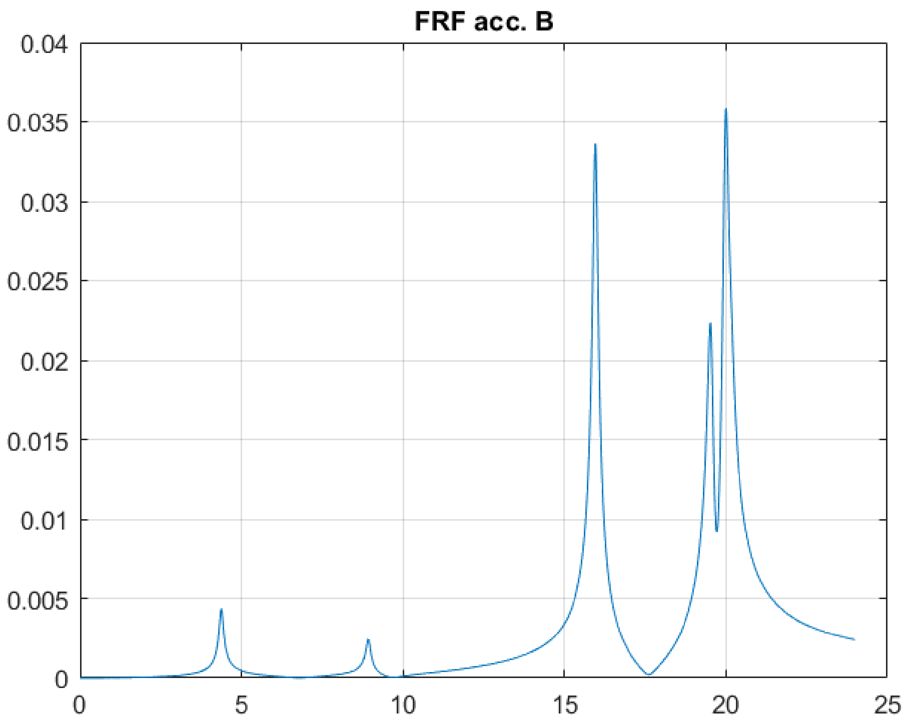

Figure 22: Vertical acceleration of point B given a vertical force in B (old structure)

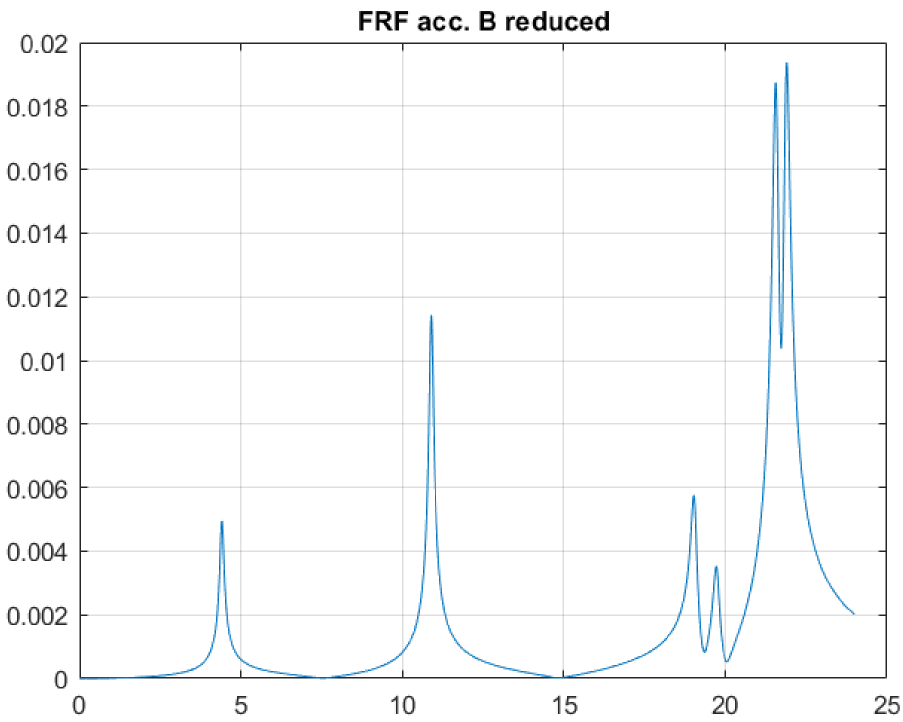

Figure 23: Vertical acceleration of point B given a vertical force in B (new structure)

Old Structure .inp file

```
*NODES
1                   1 1 0  0.0     0.0
2                   0 0 0  5.0     0.0
3                   0 0 0 10.0     0.0
4                   0 0 0 15.0     0.0
5                   0 0 0 20.0     0.0
6                   0 0 0 25.0     0.0
7                   1 1 0 30.0     0.0
8                   1 1 0  0.0   -15.0
9                   0 0 0  2.5   -12.0
10                  0 0 0  5.0    -9.0
11                  0 0 0  7.5    -7.5
12                  0 0 0 10.0    -6.0
13                  0 0 0 12.5    -5.2
14                  0 0 0 15.0    -4.4
15      0 0 0  17.5   -5.2
16      0 0 0  20.0   -6.0
17      0 0 0  22.5   -7.5
18      0 0 0  25.0   -9.0
19      0 0 0  27.5  -12.0
20      1 1 0  30.0  -15.0
21      0 0 0   5.0   -3.0
22      0 0 0   5.0   -6.0
23      0 0 0  10.0   -3.0
24      0 0 0  20.0   -3.0
25      0 0 0  25.0   -3.0
26      0 0 0  25.0   -6.0
*ENDNODES
```

*BEAMS

1	1	2	65.8788	1.7399e+09	47647800
2	2	3	65.8788	1.7399e+09	47647800
3	3	4	65.8788	1.7399e+09	47647800
4	4	5	65.8788	1.7399e+09	47647800
5	5	6	65.8788	1.7399e+09	47647800
6	6	7	65.8788	1.7399e+09	47647800
7	8	9	41.9718	1.1085e+09	17213360
8	9	10	41.9718	1.1085e+09	17213360
9	10	11	41.9718	1.1085e+09	17213360
10	11	12	41.9718	1.1085e+09	17213360
11	12	13	41.9718	1.1085e+09	17213360
12	13	14	41.9718	1.1085e+09	17213360
13	14	15	41.9718	1.1085e+09	17213360
14	15	16	41.9718	1.1085e+09	17213360
15	16	17	41.9718	1.1085e+09	17213360
16	17	18	41.9718	1.1085e+09	17213360
17	18	19	41.9718	1.1085e+09	17213360
18	19	20	41.9718	1.1085e+09	17213360
19	2	21	41.9718	1.1085e+09	17213360
20	21	22	41.9718	1.1085e+09	17213360
21	22	10	41.9718	1.1085e+09	17213360
22	3	23	41.9718	1.1085e+09	17213360
23	23	12	41.9718	1.1085e+09	17213360
24	4	14	41.9718	1.1085e+09	17213360
25	5	24	41.9718	1.1085e+09	17213360
26	24	16	41.9718	1.1085e+09	17213360
27	6	25	41.9718	1.1085e+09	17213360
28	25	26	41.9718	1.1085e+09	17213360
29	26	18	41.9718	1.1085e+09	17213360

*ENDBEAMS

*DAMPING

0.8 3.0e-5

New Structure .inp file

*NODES

1	1 1 0	0.0	0.0
2	0 0 0	5.0	0.0
3	0 0 0	10.0	0.0
4	0 0 0	15.0	0.0
5	0 0 0	20.0	0.0
6	0 0 0	25.0	0.0
7	1 1 0	30.0	0.0
8	1 1 0	0.0	-15.0
9	0 0 0	2.5	-12.0
10	0 0 0	5.0	-9.0
11	0 0 0	7.5	-7.5
12	0 0 0	10.0	-6.0
13	0 0 0	12.5	-5.2
14	0 0 0	15.0	-4.4
15	0 0 0	17.5	-5.2
16	0 0 0	20.0	-6.0
17	0 0 0	22.5	-7.5
18	0 0 0	25.0	-9.0
19	0 0 0	27.5	-12.0
20	1 1 0	30.0	-15.0
21	0 0 0	5.0	-3.0
22	0 0 0	5.0	-6.0

23	0 0 0	10.0	-3.0
24	0 0 0	20.0	-3.0
25	0 0 0	25.0	-3.0
26	0 0 0	25.0	-6.0
27	0 0 0	11.7	-1.5
28	0 0 0	13.3	-2.9
29	0 0 0	16.7	-2.9
30	0 0 0	18.3	-1.5

*ENDNODES

*BEAMS

1	1	2	65.8788	1.7399e+09	47647800
2	2	3	65.8788	1.7399e+09	47647800
3	3	4	65.8788	1.7399e+09	47647800
4	4	5	65.8788	1.7399e+09	47647800
5	5	6	65.8788	1.7399e+09	47647800
6	6	7	65.8788	1.7399e+09	47647800
7	8	9	41.9718	1.1085e+09	17213360
8	9	10	41.9718	1.1085e+09	17213360
9	10	11	41.9718	1.1085e+09	17213360
10	11	12	41.9718	1.1085e+09	17213360
11	12	13	41.9718	1.1085e+09	17213360
12	13	14	41.9718	1.1085e+09	17213360
13	14	15	41.9718	1.1085e+09	17213360
14	15	16	41.9718	1.1085e+09	17213360
15	16	17	41.9718	1.1085e+09	17213360
16	17	18	41.9718	1.1085e+09	17213360
17	18	19	41.9718	1.1085e+09	17213360
18	19	20	41.9718	1.1085e+09	17213360

19	2	21	41.9718	1.1085e+09	17213360
20	21	22	41.9718	1.1085e+09	17213360
21	22	10	41.9718	1.1085e+09	17213360
22	3	23	41.9718	1.1085e+09	17213360
23	23	12	41.9718	1.1085e+09	17213360
24	4	14	41.9718	1.1085e+09	17213360
25	5	24	41.9718	1.1085e+09	17213360
26	24	16	41.9718	1.1085e+09	17213360
27	6	25	41.9718	1.1085e+09	17213360
28	25	26	41.9718	1.1085e+09	17213360
29	26	18	41.9718	1.1085e+09	17213360
30	3	27	15.67	4.1385e+08	1790758
31	27	28	15.67	4.1385e+08	1790758
32	28	14	15.67	4.1385e+08	1790758
33	14	29	15.67	4.1385e+08	1790758
34	29	30	15.67	4.1385e+08	1790758
35	30	5	15.67	4.1385e+08	1790758

*ENDBEAMS

*DAMPING

0.8 3.0e-5