Subject: ACS 547, Noise Control Applications - Module 2 Assignment

Date: February 24, 2025 (Submitted)

Problem 1 - Modal Behaviour of a Cylindrical Room

The Matlab code for this problem is listed in Appendix 1.

Problem 1a

Table 1 lists the ten lowest resonance mode orders for the room and the respective frequency.

Index	**Mode (n_x, n_ϑ, n_r)**	**Frequency [Hz]**
0	0, 0, 0	0
1	1, 0, 0	17.2
2	0, 1, 0	33.5
3	2, 0, 0	34.3
4	1, 1, 0	37.6
5	2, 1, 0	48.0
6	3, 0, 0	51.5
7	0, 2, 0	55.6
8	1, 2, 0	58.2
9	3, 1, 0	61.4
10	2, 2, 0	65.3

Table 1: Resonant modes of the cylindrical room.

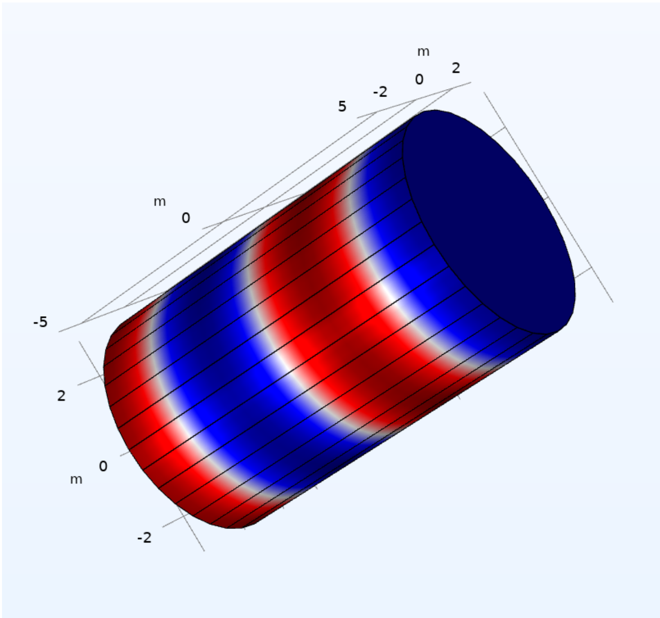
Problem 1b

The two closest modes are (3, 0, 0) and (0, 2, 0) with frequencies of 51.5 Hz and 55.6 Hz, respectively.

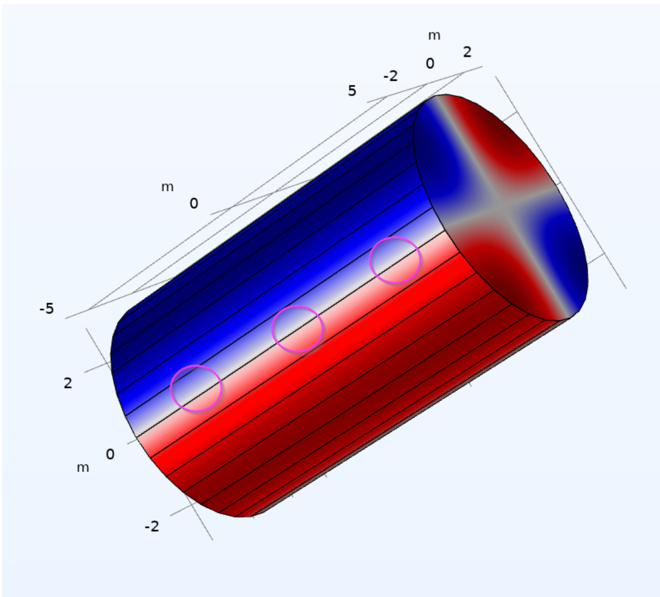
Problem 1c

Figure 1 illustrates the (3, 0, 0) and (0, 2, 0) modes. The white lines in each figure show the modal lines for that mode. **The machine can be placed where the modal lines for each mode overlap**. The figures were produced using the Room Eigenmode Simulator Version 1.1 software package.

The pink rings in Figure 1b indicate 3 possible places where the machine could be placed. These points coincide with the three modal planes shown in Figure 1a. Theoretically, there are an infinite number of places where the machine could be placed. However, placement would take into account practical considerations such as accessibility, etc.



(a) Mode (3, 0, 0)



(b) Mode (0, 2, 0)

Figure 1: Visualization of modes. (a.) Mode (3, 0, 0). (b.) Model (0, 2, 0). The pink circles in 1b illustrate a few modal line intersections where the machine could be placed.

Problem 2 - Sabine Room

The Matlab code for this problem is listed in Appendix 2.

Problem 2a

The reverberant field sound pressure level is approximately 98.3 dB SPL.

Problem 2b

Figure 2 shows the direct, reverberant, and total sound pressure levels for a 25 mW, 125 Hz, broadband, omnidirectional source placed centrally in the room (i.e. the directivity factor is 1).

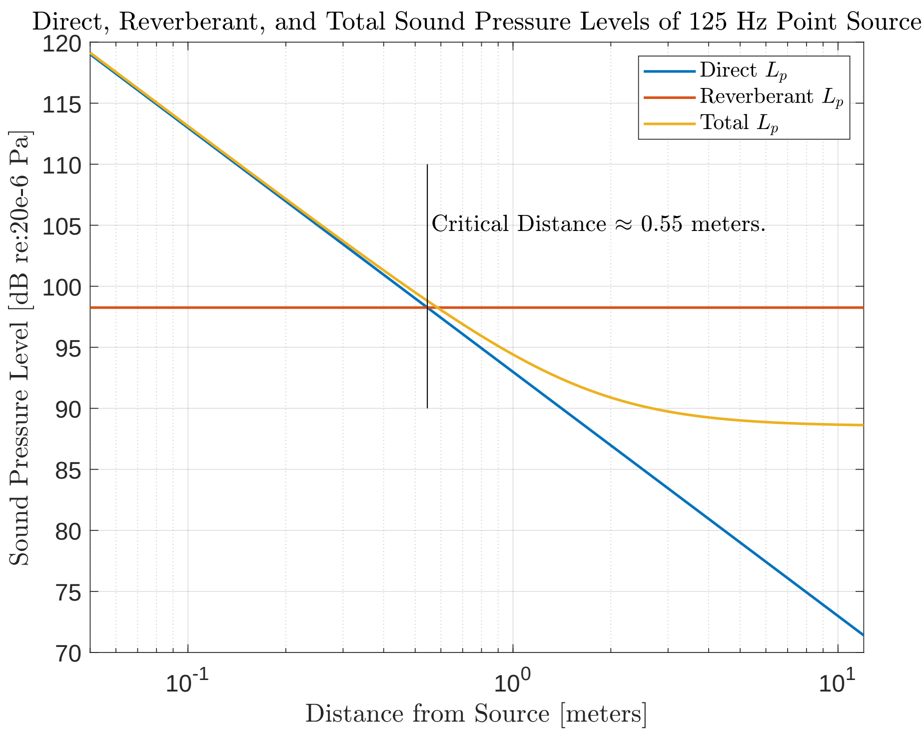


Figure 2: Sound levels for the room produced by a 25 mW, 125 Hz omnidirectional, broadband source.

Problem 3 - Transmission Loss Measurement

The Matlab code for this problem is listed in Appendix 3.

Figure 3 shows the average absorption per octave band based on the T60 data. The calibration plate isolates the receiver room and the absorption calculation does not consider the calibration plate.

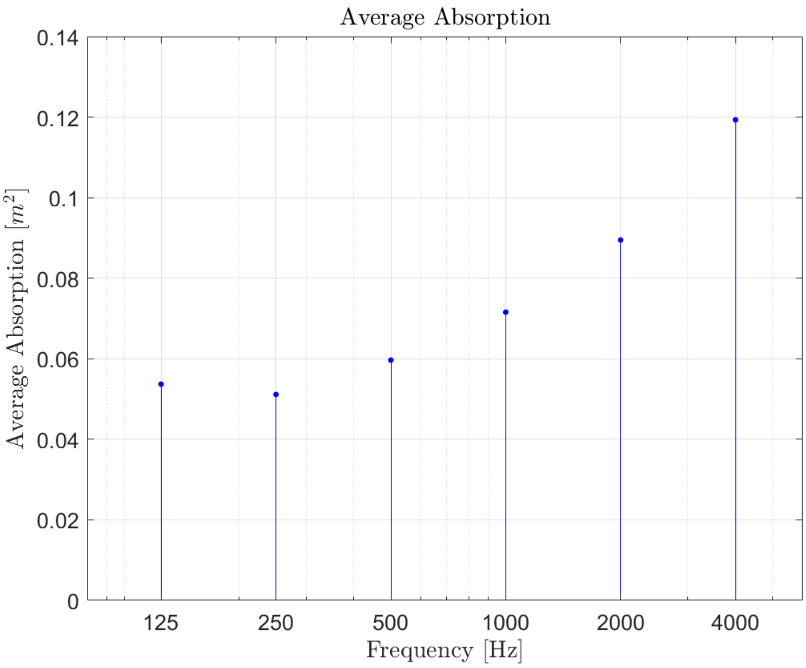


Figure 3: Average absorption per octave band.

Figure 4 shows the transmission loss per octave band.

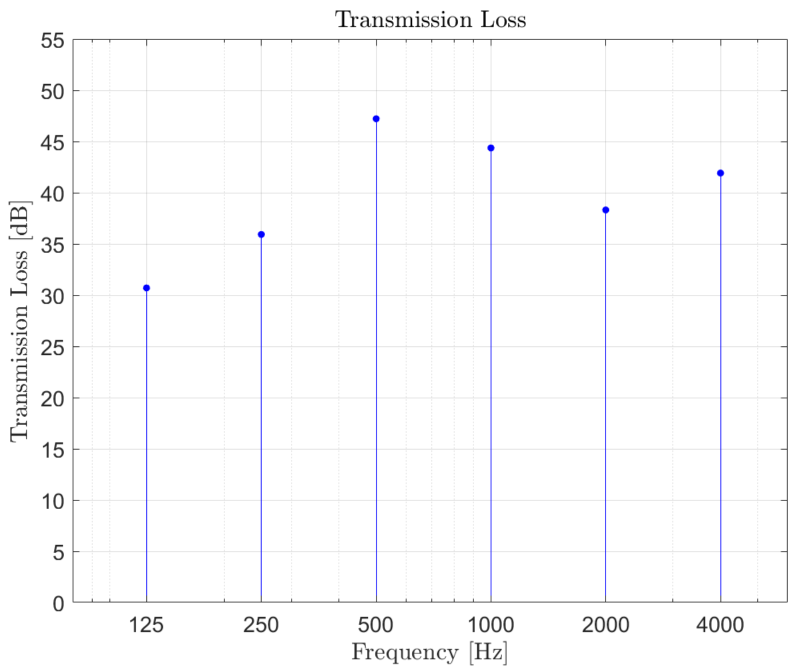


Figure 4: Transmission loss per octave band.

Problem 4 - Panel Transmission Loss

The Matlab code for this problem is listed in Appendix 4.

Problem 4a

The resonance frequency, f_o, of the galvanized steel panel is 4 Hz or 22.6 $\frac{\text{radians}}{\text{s}}$.

Problem 4b

i - Critical Frequency and Coincidence Frequency at 75°

The critical frequency is 10,216 Hz and the coincidence frequency at 75° is 10,494 Hz.

ii - Transmission Loss at Angle of Incidence of 75°

Figure 5 shows the transmission loss for an angle of incidence of 75°.

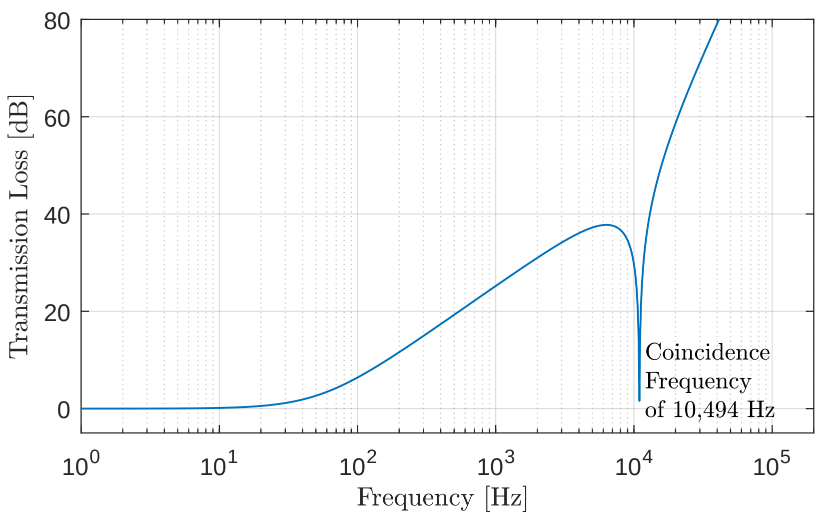


Figure 5: Flexible Panel Transmission Loss for a 75° Incidence Angle

iii - Transmission Loss for Angles of Incidence between 0-90°

Figure 6 shows the transmission loss angles of incidence from 0 to 90° in steps of 10°.

iv - Diffuse Transmission Loss

See the Matlab code in the Appendix 4

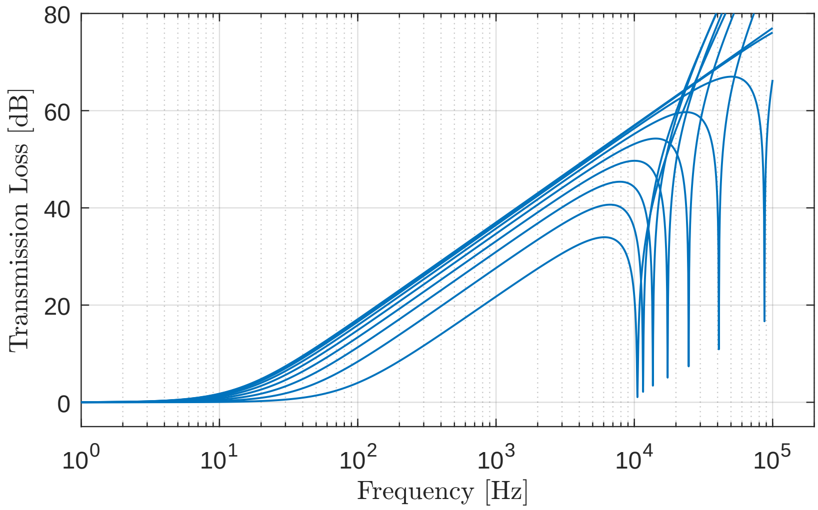


Figure 6: Flexible Panel Transmission Loss for angles of incidence between 0 and 90°.

Problem 4c

See the Matlab code in the Appendix 4

Problem 4d

Figure 7 shows the transmission loss angles of incidence from 0 to 90° in steps of 10°.

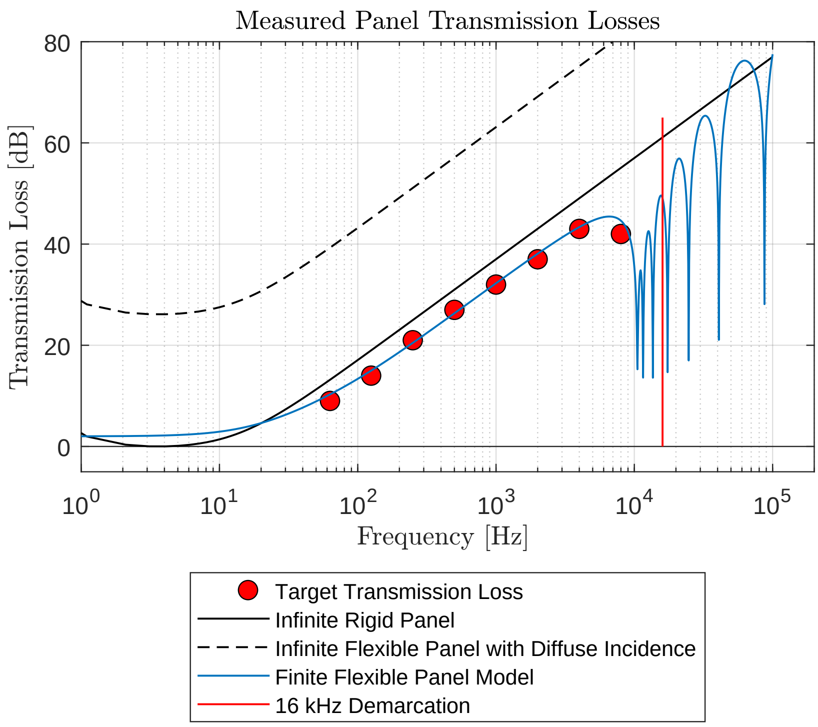


Figure 7: Measured data and the modeled responses.

The finite flexible panel appears to be the most appropriate model. As noted in class, the measured transmission loss at 8 kHz is smaller than the loss a 4 kHz. This indicates that the loss at 16 kHz should be less than the response of the infinite rigid panel at 16 kHz.

Problem 5 - Large Enclosure Design

The Matlab code for this problem is listed in Appendix 5.

My enclosure is a 2 m cube that sits on the ground (5 sides of enclosure material) over the machine. The machine is suspended above the pad (directivity factor of unity). It is assumed that there is no noise transmission through the ground.

Table 2 lists the target insertion losses, the calculated insertion losses for four materials, and the loss difference between the target and each type of material. *A positive value indicates that the target insertion loss for the given octave band was not met.*

Figure 8 shows calculated insertion loss differences for the data in Table 2.

Octave Band [Hz]	Target IL [dB]	QBV 2 [dB]	QBV 3 [dB]	HTL (100MM) [dB]	HTL 4 [dB]
250	44	28.0	19.7	10.6	5.6
500	54	26.7	22.2	6.6	-4.4
1,000	45	5.6	1.4	-15.4	-22.4
2,000	47	-8.4	-2.2	-18.4	-19.4
4,000	58	-2.4	5.7	-11.0	-13.0

Table 2: Calculated insertion losses for the 4 enclosure materials.

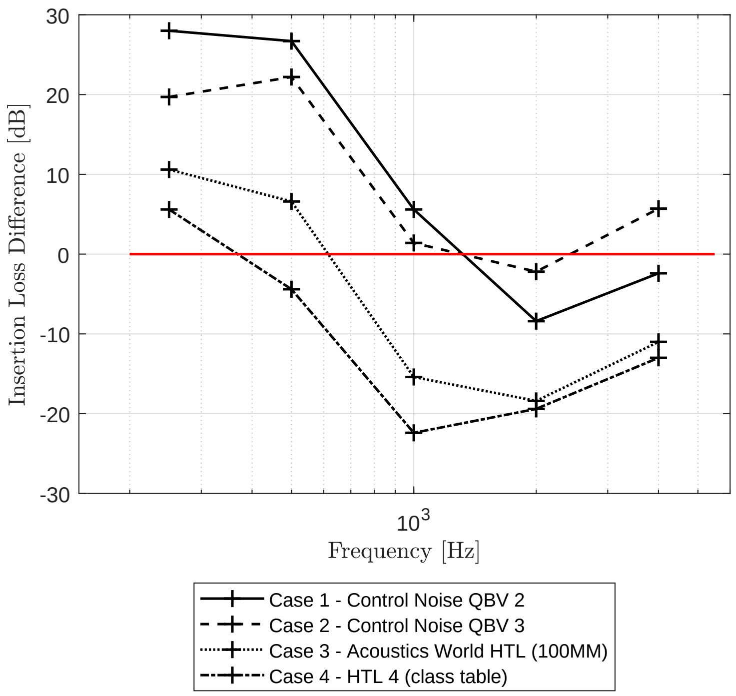


Figure 8: Insertion loss differences.

The most restrictive case given the 2 cubic-meter, 5-sided enclosure, is the HTL 4 material, Case 4, which was presented in class. The target insertion loss was reached by all of the octave bands except the 250 Hz band.

With the same enclosure dimensions and machine orientation, the HTL (100MM) material, Case 3, was found to be the second best material, not meeting the 250 Hz and 500 Hz targets. Figure 9 shows the transmission loss and absorption coefficient data for the material selected from Acoustics World.

The QBV-2 and QBV-3 , Case 1 and Case 2, respectively, had the poorest performance. Figure 10 shows the transmission loss and absorption coefficient data for these material from Control Noise.

Using the HTL 4 material from class, a cube with 2 m sides appeared to produce the optimal insertion loss for the 250 Hz octave band. Making the size of the enclosure larger does not reduce this insertion loss and does not seem to be practical. A supplementary approach for this octave band should be considered.

NRC Rating

Sound Absorption Coefficients (ASTM C423)

Acoustic Panel Type	Panel Construction	125 Hz	250Hz	500Hz	1000 Hz	2000 Hz	4000 Hz	NRC
STL (100MM)	18 ga. solid / 22 ga. perforated	0.60	1.13	1.12	1.09	1.03	0.91	1.00
STL (100MM)	16 ga. solid / 22 ga. perforated	0.60	1.13	1.12	1.09	1.03	0.91	1.00
HTL (100MM)	16 ga. solid / 22 ga. perforated with HD Soundbloc Layer	0.60	1.13	1.12	1.09	1.03	0.91	1.00

STC Rating

Sound Transmission Class

Acoustic Panel Type	Panel Construction	125 Hz	250Hz	500Hz	1000 Hz	2000 Hz	4000 Hz	STC
STL (100MM)	18 ga. solid / 22 ga. perforated	21	28	39	48	56	58	40
STL (100MM)	16 ga. solid / 22 ga. perforated	24	32	41	51	60	66	43
HTL (100MM)	16 ga. solid / 22 ga. perforated with HD Soundbloc Layer	27	34	48	61	66	70	48

Figure 9: HTL (100MM) data from Acoustics World.

ACOUSTICAL DATA:

The **most effective** noise reduction products combine **both sound absorption and noise barrier properties.** Tested under strict compliance to appropriate ASTM standards, we offer the following results:

Sound Transmission Loss (dB) per Octave Band Frequency

NetWell Model #	THK.	WT.	125	250	500	1000	2000	4000	STC
QBV-1	1"	1.3	11	16	24	30	35	35	27
QBV-2	2"	1.5	13	20	29	40	50	55	32
QBV-3	2.5"	2	19	25	33	46	53	58	37
QBS-1	2"	1.5	12	16	27	40	44	43	29
QBS-2	2"	2.5	19	22	28	40	56	61	33
Roof Panel	2"	2	18	24	28	37	45	46	31

Per ASTM E 90

Sound Absorption Data – Absorber Component Random Incident Sound Absorption Octave Band Center Frequencies (Hz)

Product	THK.	125	250	500	1000	2000	4000	NRC
QBV-1	1"	.12	.47	.85	.84	.64	.62	.70
QBV-2	2"	.07	.27	.96	1.13	1.08	.99	.85
QBS-1 & 2	2"	.45	.96	.87	.66	.47	.30	.75
QB-4	4"	.21	.89	1.09	1.17	1.13	1.07	1.05

Per ASTM C 423

Flammability Ratings

Product	Descriptor	Flame Spread	Smoke Developed
QBS-1	Vinyl faced 1" quilted fiberglass on both sides of a 1 lb. PSF non-reinforced loaded vinyl barrier septum	23	30
QBS-HT	Silicone faced 1" quilted fiberglass on both sides of a 1 lb. PSF non-reinforced noise barrier septum	4	19
QBV-2	Vinyl faced 2" quilt on one side of a 1 lb. reinforced loaded vinyl noise barrier	23	12
QBV-1	Vinyl faced 1" quilt on one side of a 1lb. reinforced loaded vinyl noise barrier	23	30

Above table shows flame spread and smoke developed ratings per ASTM Designation E84; Surface Burning Characteristics of Building Materials. Note: Class A rating applies to products with a flame spread index of 25 or less, and a smoke developed index of 450 or less

Additional products tested to ASTM E 162 and ASTM E 662, test reports available on request

Figure 10: QBV-2 and QBV-3 data from Control Noise.

Problem 6 - Close-fitting Enclosure Design

The Matlab code for this problem is listed in Appendix 6.

1 Appendix - Matlab Code for Problem 1

```matlab



%% Synopsis

% Problem 1 - Modal Behaviour of a Cylindrical Room



%% Environment

close all; clear; clc;
% restoredefaultpath;

% addpath( genpath( '' ), '-begin' );
addpath( genpath( './00 Support' ), '-begin' );

% set( 0, 'DefaultFigurePosition', [  400  400  900  400  ] );  % [ left bottom width height ]
set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
set( 0, 'DefaultFigureWindowStyle', 'normal' );
set( 0, 'DefaultLineLineWidth', 1.5 );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );



%% Define Room

room.radius = 3;  % m
room.length = 10;  % m



%% Test Circular Mode Function

% psi = circular_mode_shape( 3, 1, 2, false );  % 3.7261 - CHECKED FROM CLASS (PLOT NOT CREATED)
% psi = circular_mode_shape( 3, 1, 2, true );  % 3.7261 - CHECKED FROM CLASS (PLOT CREATED)



%% Define Anonymous Function for the Natural Frequencies

h_natural_frequencies = @( c, nx, ntheta, nr, Lx, cylinder_radius, plot_flag )  (c/2) .* sqrt( (nx/Lx).^2  + (circular_mode_shape(nr, ntheta, cylinder_radius, plot_flag )/cylinder_radius).^2 );



%% Calculate the Natural Frequencies

% The maximum number of radial modes is 5 (indexed from 0 to 4).
% The maximum number of angular modes is 8 (indexed from 0 to 7).

NX_SIZE = 20;
NTHETA_SIZE = 7;
NR_SIZE = 4;
    natural_frequencies = nan( NX_SIZE, NTHETA_SIZE, NR_SIZE );

for nx = 0:1:NX_SIZE
    for ntheta = 0:1:NTHETA_SIZE
        for nr = 0:1:NR_SIZE
            natural_frequencies( nx+1, ntheta+1, nr+1 ) = h_natural_frequencies( 343, nx, ntheta, nr, 10, 3, false );
        end
    end
end



%% Part a - Find 10 Lowest Resonance Frequencies

NUMBER_OF_LOWEST_FREQUENCIES = 11;
```

```matlab
    mode_indices =  ( 1:1:NUMBER_OF_LOWEST_FREQUENCIES ).';

[ sortedValues, sortedIndices ] = sort( natural_frequencies(:) );  % 21-by-8-by-5 -> 840 elements

smallestValues = sortedValues( 1:NUMBER_OF_LOWEST_FREQUENCIES );
    % [ mode_indices   round( smallestValues, 1 ) ]
    %
    % 1            0
    % 2         17.2
    % 3         33.5
    % 4         34.3
    % 5         37.6
    % 6         47.9
    % 7         51.5
    % 8         55.6
    % 9         58.2
    % 10         61.4
    % 11         65.3

smallestIndices = sortedIndices( 1:NUMBER_OF_LOWEST_FREQUENCIES );

[ x, y, z ] = ind2sub( size(natural_frequencies), smallestIndices );
    % ( [ x y z ] - 1 )

% Verify the calculated mode indices.
h_natural_frequencies( 343, 0, 0, 0, 10, 3, false );  % 0 Hz
h_natural_frequencies( 343, 1, 0, 0, 10, 3, false );  % 17.2 Hz
h_natural_frequencies( 343, 0, 1, 0, 10, 3, false );  % 33.5 Hz
h_natural_frequencies( 343, 2, 0, 0 , 10, 3, false );  % 34.3 Hz
h_natural_frequencies( 343, 1, 1, 0 , 10, 3, false );  % 37.6 Hz
h_natural_frequencies( 343, 2, 1, 0 , 10, 3, false );  % 48.0 Hz
h_natural_frequencies( 343, 3, 0, 0 , 10, 3, false );  % 51.5 Hz
h_natural_frequencies( 343, 0, 2, 0 , 10, 3, false );  % 55.6 Hz
h_natural_frequencies( 343, 1, 2, 0 , 10, 3, false );  % 58.2 Hz
h_natural_frequencies( 343, 3, 1, 0 , 10, 3, false );  % 61.4 Hz
h_natural_frequencies( 343, 2, 2, 0 , 10, 3, false );  % 65.3 Hz



%% Part b - Two

% [ (1:11).'  abs( smallestValues - 53 ) ]

temp = [ x y z ] - 1;  temp( 7:8, :, : )

h_natural_frequencies( 343, 3, 0, 0, 10, 3, false );  % 51.5 Hz, (3, 0, 0)
h_natural_frequencies( 343, 0, 2, 0, 10, 3, false );  % 55.6 Hz, (0, 2, 0)



%% Part c

% See the report.



%% Clean-up

if ( ~isempty( findobj( 'Type', 'figure' ) ) )
    monitors = get( 0, 'MonitorPositions' );
        if ( size( monitors, 1 ) == 1 )
            autoArrangeFigures( 2, 2, 1 );
        elseif ( 1 < size( monitors, 1 ) )
            autoArrangeFigures( 2, 2, 1 );
        end
end


fprintf( 1, '\n\n\n*** Processing Complete ***\n\n\n' );



%% Reference(s)

% https://www.mathworks.com/matlabcentral/answers/1883747-how-to-find-the-5-minimum-values-in-a-
    multidimensional-matrix-and-the-indices-to-which-these-entries
```

2 Appendix - Matlab Code for Problem 2

```
%% Synopsis

% Problem 2 - Sabine Room



%% Environment

close all; clear; clc;
% restoredefaultpath;

% addpath( genpath( '' ), '-begin' );
addpath( genpath( '../00 Support' ), '-begin' );

% set( 0, 'DefaultFigurePosition', [  400  400  900  400  ] );  % [ left bottom width height ]
set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
set( 0, 'DefaultFigureWindowStyle', 'normal' );
set( 0, 'DefaultLineLineWidth', 1.5 );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );



%% Define Room

room.width = 8;  % m
room.length = 6;  % m
room.height = 3;  % m
    room.volume = room.width * room.length * room.height;  % 144 m^3
    room.area = 2*(room.width*room.height)  +  2*(room.length*room.height)  + 2*(room.width*room.length);  % 180 m^2

alpha_average_walls_and_floor = 0.05;  % For the walls and the floor.
alpha_average_ceiling = 0.15;  % For the ceiling.

% For the 125 Hz octave band.



%% Part a - Estimate the Reverberant Sound Pressure Level

Lw = 10*log10( 25e-3 / 1e-12 );  % 103.98 dB

average_absorption_coefficient = (  (room.width*room.length)*alpha_average_ceiling  +  (room.width*room.length  +  2*(room.width*room.height)  + 2*(room.length*room.height) )*alpha_average_walls_and_floor  ) / room.area;  % 0.076667 unitless

room_constant = room.area * average_absorption_coefficient / ( 1 - average_absorption_coefficient );  % 14.9 m^2 or Sabin


sound_pressure_level = Lw + 10*log10( 4 / room_constant );  % 98.3 dB



%% Part b - Calculate the Critical Distance

D0 = 1;

r = 0:0.05:12;  % m

h_Lp_direct = @( Lw, D0, r )  Lw + 10*log10( D0./(4.*pi.*r.^2) );
h_Lp_reverberant = @( Lw, room_constant )  Lw + 10*log10( 4 ./ room_constant );
%
h_Lp_net = @( Lw, D0, r, room_constant )  Lw + 10*log10( D0./(4.*pi.*r.^2) + 4/room_constant ) + 10*log10( 343*1.2/400 );


figure( ); ...
```

```
    h1 = plot( r, h_Lp_direct( Lw, D0, r ) );  hold on;
    h2 = plot( r, ones( size(r) ).*h_Lp_reverberant( Lw, room_constant) );
    h3 = plot( r, h_Lp_net( Lw, D0, r, room.volume ));  grid on;
        legend( [ h1, h2 h3 ], { 'Direct $L_p$', 'Reverberant $L_p$', 'Total $L_p$' }, 'Interpreter', 'Latex' );
    %
    text( 0.56, 105, 'Critical Distance $\approx$ 0.55 meters.', 'Interpreter', 'Latex' );
        line( [ 0.545 0.545 ], [ 90 110 ], 'Color', 'k', 'LineWidth', 0.6 );
    %
    xlabel( 'Distance from Source [meters]' );  ylabel( 'Sound Pressure Level [dB re:20e-6 Pa]' );
    title( 'Direct, Reverberant, and Total Sound Pressure Levels of 125 Hz Point Source' );
    %
    set( gca, 'XScale', 'log' );


% Estimate the critical distance (see page 84 of "06-Indoors.pdf" notes for ACS 537).
rc = 0.141 * sqrt( D0 * room_constant );  % 0.5451 meters

return

%% Clean-up

if ( ~isempty( findobj( 'Type', 'figure' ) ) )
    monitors = get( 0, 'MonitorPositions' );
        if ( size( monitors, 1 ) == 1 )
            autoArrangeFigures( 2, 2, 1 );
        elseif ( 1 < size( monitors, 1 ) )
            autoArrangeFigures( 2, 2, 1 );
        end
end


fprintf( 1, '\n\n\n*** Processing Complete ***\n\n\n' );



%% Reference(s)
```

3 Appendix - Matlab Code for Problem 3

```matlab



%% Synopsis

% Question 3 - Transmission Loss Measurement


% Note(s):
%
%   1.)  Lp1 depends on the transmission loss.
%
%        If the transmission loss is low, then more energy goes to room 2 (i.e., the receiver room ).
%
%        The noise reduction from the source room to the receiver room.
%
%        Adding the barrier will change the level in the source room.  Typically making the sound level higher in the source room.



%% Environment

close all; clear; clc;
% restoredefaultpath;

% addpath( genpath( '' ), '-begin' );
addpath( genpath( '../00 Support' ), '-begin' );

% set( 0, 'DefaultFigurePosition', [  400  400  900  400  ] );  % [ left bottom width height ]
set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
set( 0, 'DefaultFigureWindowStyle', 'normal' );
set( 0, 'DefaultLineLineWidth', 1.2 );
set( 0, 'DefaultLineMarker', 'x' );
set( 0, 'DefaultLineMarkerSize', 15 );
% set( 0, 'DefaultAxesLineStyleOrder', { '-' '--o' '+' } );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );



%% Dimensions of Rooms and Panel

room.length = 4;  % m
room.width = 4;  % m
room.height = 4;  % m
    room.volume = room.length * room.width * room.height;  % 64 m^3
    room.area = 2*(room.length * room.width) + 2*(room.length * room.height) + 2*(room.width * room.height);  % 96 m^2

panel.width = 0.8;  % m
panel.height = 0.8;  % m
    panel.area = panel.width * panel.height;  % 0.64 m^2

c = 343;  % m/s



%% Measurement Data

octave_band_frequencies = [ 125  250  500  1000  2000  4000 ].';  % Hz
T60 = [ 2.0  2.1  1.8  1.5  1.2  0.9 ].';  % seconds
spl.source_room = [ 90  95  103  105  100  93 ].';  % dB re: 20e-6 Pa
spl.receiver_room = [ 50  50  46  50  50  38 ].';  % dB re: 20e-6 Pa
    % [ octave_band_frequencies  T60  spl.source_room  spl.receiver_room  (spl.source_room - spl.receiver_room) ]



%% Calculate Average Absorption in the Receiver Room using Reverberation Time Measurements

```

```matlab
average_absorption = @( volume, area, c, T60 ) ( 55.25 .* volume ) ./ ( area .* c .* T60 );

receiver_room.average_absorption = average_absorption( room.volume, room.area, c, T60 );

% Assumption: Calibration panel has very high transmission loss.

figure( ); ...
    stem( octave_band_frequencies, receiver_room.average_absorption, 'Marker', '.', 'MarkerSize', 12, 'Color', 'b' ); grid on;
    xlabel( 'Frequency [Hz]' ); ylabel( 'Average Absorption [$m^2$]' );
    title( 'Average Absorption' );
    %
    xticks( octave_band_frequencies ); xticklabels( num2cell( octave_band_frequencies ) );
    set( gca, 'XScale', 'log' );
    xlim( [ 80 6e3 ] ); ylim( [ 0 0.14 ] );



%% Calculate the Receiver Room Constant

% The calibration plate isolates the receiver room.

% The receiver room does not consider the calibration plate.

room_constant = @( average_absorption, area ) ( average_absorption * area ) ./ ( 1 - average_absorption ); % Unitless

receiver_room.room_constant = room_constant( receiver_room.average_absorption, room.area );



%% Calculate the Transmission Loss in Each Octave Band

transmission_coefficient = @( receiver_room_pressure, source_room_pressure, panel_area, receiver_room_constant ) ( ( receiver_room_pressure ./ source_room_pressure ) .* receiver_room_constant ) ./ panel_area;

tau = transmission_coefficient( 10.^(spl.receiver_room./10)*20e-6, 10.^(spl.source_room/10)*20e-6, panel.area, receiver_room.room_constant );

TL = -10*log10( tau );

figure( ); ...
    stem( octave_band_frequencies, TL, '.', 'MarkerSize', 15, 'Color', 'b' ); grid on;
    xlabel( 'Frequency [Hz]' ); ylabel( 'Transmission Loss [dB]' );
    title( 'Transmission Loss' );
    %
    xticks( octave_band_frequencies ); xticklabels( num2cell( octave_band_frequencies ) );
    set( gca, 'XScale', 'log' );
    xlim( [ 80 6e3 ] ); ylim( [ 0 55 ] );



%% Clean-up

if ( ~isempty( findobj( 'Type', 'figure' ) ) )
    monitors = get( 0, 'MonitorPositions' );
        if ( size( monitors, 1 ) == 1 )
            autoArrangeFigures( 2, 2, 1 );
        elseif ( 1 < size( monitors, 1 ) )
            autoArrangeFigures( 2, 2, 1 );
        end
end


fprintf( 1, '\n\n\n*** Processing Complete ***\n\n\n' );



%% Reference(s)
```

4 Appendix - Matlab Code for Problem 4

```matlab



%% Synopsis

% Problem 4 - Panel Tramission Loss



%% Environment

close all; clear; clc;
% restoredefaultpath;

% addpath( genpath( '' ), '-begin' );
addpath( genpath( '../40 Assignments/00 Support' ), '-begin' );

% set( 0, 'DefaultFigurePosition', [  400  400  900  400  ] ); % [ left bottom width height ]
set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
set( 0, 'DefaultFigureWindowStyle', 'normal' );
set( 0, 'DefaultLineLineWidth', 0.8 );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );



%% Define Panel

panel.length = 80e-2;  % m
panel.E = 200e9;  % Pa
panel.density = 7800;  % kg/m^3
panel.v = 0.29;  % Poisson's Ratio (unitless)
panel.thickness = 1.2e-3;  % m
panel.eta = 0.001;  % Loss factor (unitless)

c = 343;  % m/s
rho0 = 1.21;  % kg/m^3



%% Measured Panel Data

octave_band_frequencies = [ 63  125  250  500  1000  2000  4000  8000 ].';  % Hz
TL = [ 9  14  21  27  32  37  43  42 ].';  % dB



%% Problem 4a - Infinite, Rigid Panel Model with Normal Incidence

D = ( panel.E * panel.thickness.^3 ) / ( 12 * ( 1 - panel.v^2 ) );  % 31.4

ms = panel.density * panel.thickness;  % 9.4 kg/m^2

wo = pi^2 / panel.length * sqrt( D / ms );  % 22.6 radians/s
    s = wo^2 * ms;  % 4,785.9 kg radians / m^2s^2

fo = wo / (2*pi);  % 4 Hz


% Define an Anonymous function for the rigid panel with normal incidence.
h_tau_infinite_rigid_panel = @( f, wo, ms, s, rho0, c, eta)  4 ./ ( ( (2*pi.*f*ms - s./(2*pi.*f))
    ./ (rho0 * c) ).^2 + ( (wo*ms*eta) ./ (rho0*c) + 2 ).^2 );



%% Problem 4b - Infinite, Flexible Panel Model with Random Incidence

% The panel has bending waves.


% Part (i.)

```

```matlab
% The critical frequency.
 critical_frequency = c^2 / (2*pi) * sqrt( ms / D );  % 10.22 kHz

% Verify the critical frequency using a 90 degree angle of incidence.
critical_frequency_verify_1 =  1./(2*pi) .* sqrt( ms / D ) .* ( c / sind( 90 )).^2;  % 10.22 kHz

% Verify the critical frequency using the properties of the panel.
critical_frequency_verify_2 = c^2 / ( 1.8 * panel.thickness * sqrt( panel.E / ( panel.density * ( 1 - panel.v^2) ) ) );  % 10.3 kHz


% Coincidence frequency for a 75 degree angle of incidence.
phi = 75;
    h_coincidence_frequency =  @( ms, D, c, phi )  1./(2*pi) * sqrt( ms / D ) .* ( c ./ sind( phi )).^2;
        h_coincidence_frequency( ms, D, c, phi );  % 10,949 Hz


% Define an Anonymous function for the flexible panel with random incidence.
h_tau_infinite_flexible_panel = @( f, c, phi, D, eta )  ( 2*rho0.*c*secd(phi)).^2 ./ ( (2*rho0.*c*secd(phi) + D*eta*(2*pi.*f./c).^4./(2*pi.*f)*sind(phi)^4).^2  +  ...
    (2*pi.*f*ms - D*(2*pi.*f./c).^4./(2*pi.*f)*sind(phi)^4).^2 );



% Part (ii.) - Transmission loss for a 75 degree angle of incidence.
f = 0.1:1:100e3;

figure( ); ...
    plot( f, -10*log10( h_tau_infinite_flexible_panel( f, rho0, c, phi, D, panel.eta ) ), 'LineStyle', '-', 'Marker', 'none', 'Color', [0.00, 0.45, 0.74] );  grid on;
    text( 12e3, 5, sprintf( 'Coincidence\nFrequency\nof 10,494 Hz' ) );
    xlabel( 'Frequency [Hz] ' );  ylabel( 'Transmission Loss [dB]' );
    set( gca, 'XScale', 'log' );
    axis( [ 1 200e3 -5 80 ] );
    %
    % Textheight:  744 pt. and Textwidth:  493 pt. from LaTex document
    %
    % set( gcf, 'units', 'point', 'pos', [ 200 200    493*0.8  744*0.3 ] );
    %     pos = get( gcf, 'Position' );
    %         set( gcf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'points', 'PaperSize', [pos(3), pos(4)] );
    %             print(gcf, 'Q4 TL for 75 AOI', '-dpdf', '-r0' );
%
% https://tex.stackexchange.com/questions/179382/best-practices-for-using-matlab-images-in-latex



% Part (iii.)
f = 0.1:1:100e3;

eta = panel.eta;  phi_set = 0:10:90;  t_set = [ ];

figure( ); ...
    hold on;
    for phi = phi_set
        plot( f, -10*log10( h_tau_infinite_flexible_panel( f, rho0, c, phi, D, panel.eta ) ), 'LineStyle', '-', 'Marker', 'none', 'Color', [0.00, 0.45, 0.74] );
            t_set = [ t_set;  h_tau_infinite_flexible_panel( f, rho0, c, phi, D, panel.eta ) ];
    end
    %
    grid on;  box on;
    xlabel( 'Frequency [Hz] ' );  ylabel( 'Transmission Loss [dB]' );
    set( gca, 'XScale', 'log' );
    axis( [ 1 200e3 -5 80 ] );
    %
    % Textheight:  744 pt. and Textwidth:  493 pt. from LaTex document
    %
    % set( gcf, 'units', 'point', 'pos', [ 200 200    493*0.8  744*0.3 ] );
    %     pos = get( gcf, 'Position' );
    %         set( gcf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'points', 'PaperSize', [pos(3), pos(4)] );
    %             print(gcf, 'Q4iii TL for 75 AOI', '-dpdf', '-r0' );
%
% https://tex.stackexchange.com/questions/179382/best-practices-for-using-matlab-images-in-latex


tau_d = nanmean( t_set .* sind( 2*phi_set ).', 1 );
```

```matlab
%% Combined Transmission Loss Plot

figure( ); ...
    h1 = stem( octave_band_frequencies, TL, 'LineStyle', 'none', 'Marker', 'o', 'MarkerSize', 8, 'MarkerFaceColor', 'r', 'MarkerEdgeColor', 'k' );  hold on;
    h2 = plot( f, -10*log10( h_tau_infinite_rigid_panel( f, wo, ms, s, rho0, c, panel.eta ) ), 'LineStyle', '-', 'Color', 'k' );
    h3 = plot( f, -10*log10( h_tau_infinite_rigid_panel( f, wo, ms, s, rho0, c, panel.eta ) ./(200*panel.eta) * ( 4*panel.length / ( panel.length^2 * critical_frequency ) ) ), 'LineStyle', '--', 'Marker', 'none', 'Color', 'k' );
    h4 = plot( f, -10*log10( tau_d ), 'LineStyle', '-', 'Marker', 'none', 'Color', [0.00, 0.45, 0.74] );
    h5 = line( [ 16e3 16e3 ], [ 0 65 ], 'Color', 'r' );  grid on;  % 16 kHz Demarcation
    legend( ...
        [ h1, h2, h3, h4, h5 ], ...
        'Target Transmission Loss', ...
        'Infinite Rigid Panel', ...
        'Infinite Flexible Panel with Diffuse Incidence', ...
        'Finite Flexible Panel Model', ...
        '16 kHz Demarcation', ...
        'Location', 'SouthOutside' );

    xlabel( 'Frequency [Hz] ' );  ylabel( 'Transmission Loss [dB]' );
    title( 'Measured Panel Transmission Losses' );
    set( gca, 'XScale', 'log' );
    axis( [ 1 200e3 -5 80 ] );
    %
    % Textheight:  744 pt. and Textwidth:  493 pt. from LaTex document
    %
    % set( gcf, 'units', 'point', 'pos', [ 200 200    493*0.8 744*0.45 ] );
    %     pos = get( gcf, 'Position' );
    %         set( gcf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'points', 'PaperSize', [pos(3), pos(4)] );
    %             print(gcf, 'Q4d TL for 75 AOI', '-dpdf', '-r0' );
%
% https://tex.stackexchange.com/questions/179382/best-practices-for-using-matlab-images-in-latex



%% Plot Data and Model - Different Side Materials

% h_tau_infinite_rigid_panel_side_materials = @( f, wo, ms, s, rho0, c, eta, n )  (4*n) ./ ( ( (2*pi.*f*ms - s./(2*pi.*f)) ./ (rho0 * c) ).^2  +  ( (wo*ms*eta) ./ (rho0*c) + n + 1 ).^2 );

% f = 1e-2:1e-2:20e3;
%
% phi = 15;
% eta = panel.eta;
%
%
% figure( ); ...
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel_side_materials( f, wo, ms, s, rho0, c, panel.eta, 1 ) ), 'LineStyle', '-' );  hold on;
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel_side_materials( f, wo, ms, s, rho0, c, panel.eta, 1/3600 ) ), 'LineStyle', '-' );
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel_side_materials( f, wo, ms, s, rho0, c, panel.eta, 3600 ) ), 'LineStyle', '-' );  grid on;
%         %
%         legend( ...
%             'Same Fluid', ...
%             'Water to Air', ...
%             'Air to Water', ...
%             'Location', 'North' );
%     %
%     xlabel( 'Frequency [$\frac{\omega}{\omega_o}$] ' );  ylabel( 'Transmission Loss [dB]' );
%     title( 'Measured Panel Transmission Losses' );
%     set( gca, 'XScale', 'log' );
%     % axis( [ 40 12e3 -5 45] );



%% Change is Stiffness

% figure( ); ...
%     stem( octave_band_frequencies ./ (wo / (2*pi) ), TL, 'LineWidth', 0.5, 'Marker', 'o', '
```

```
MarkerSize', 8, 'MarkerEdgeColor', 'b', 'MarkerFaceColor', 'r' );  hold on;
%     %
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel( f, wo, ms, s, rho0, c, panel.eta ) ), 'LineStyle', '-' );
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel( f, wo, ms, s*100, rho0, c, panel.eta ) ), 'LineStyle', '--' );
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel( f, wo, ms, s*1e-2, rho0, c, panel.eta ) ), 'LineStyle', '--' );
%         %
%         legend( ...
%             'Target TL Values', ...
%             'Infinite Rigid Panel with Normal Incidence Sound', ...
%             'Infinite Rigid Panel with Normal Incidence Sound (s * 100)', ...
%             'Infinite Rigid Panel with Normal Incidence Sound (s / 100)', ...
%             'Location', 'North' );
%     %
%     xlabel( 'Frequency [$\frac{\omega}{\omega_o}$] ' );  ylabel( 'Transmission Loss [dB]' );
%     title( 'Measured Panel Transmission Losses - Change in Stiffness' );
%     set( gca, 'XScale', 'log' );
%     % axis( [ 40 12e3 -5 45] );



%% Change in Mass

% figure( ); ...
%     stem( octave_band_frequencies ./ (wo / (2*pi) ), TL, 'LineWidth', 0.5, 'Marker', 'o', 'MarkerSize', 8, 'MarkerEdgeColor', 'b', 'MarkerFaceColor', 'r' );  hold on;
%     %
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel( f, wo, ms*100, s, rho0, c, panel.eta ) ), 'LineStyle', '-' );
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel( f, wo, ms, s, rho0, c, panel.eta ) ), 'LineStyle', '-' );
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel( f, wo, ms*1e-2, s, rho0, c, panel.eta ) ), 'LineStyle', '-' );
%         %
%         legend( ...
%             'Target TL Values', ...
%             'Infinite Rigid Panel with Normal Incidence Sound', ...
%             'Infinite Rigid Panel with Normal Incidence Sound (s * 100)', ...
%             'Infinite Rigid Panel with Normal Incidence Sound (s / 100)', ...
%             'Location', 'North' );
%     %
%     xlabel( 'Frequency [$\frac{\omega}{\omega_o}$] ' );  ylabel( 'Transmission Loss [dB]' );
%     title( 'Measured Panel Transmission Losses - Change in Mass' );
%     set( gca, 'XScale', 'log' );
%     % axis( [ 40 12e3 -5 45] );



%% Change in Loss Factor

% figure( ); ...
%     stem( octave_band_frequencies ./ (wo / (2*pi) ), TL, 'LineWidth', 0.5, 'Marker', 'o', 'MarkerSize', 8, 'MarkerEdgeColor', 'b', 'MarkerFaceColor', 'r' );  hold on;
%     %
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel( f, wo, ms, s, rho0, c, panel.eta ) ), 'LineStyle', '-' );
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel( f, wo, ms, s, rho0, c, panel.eta*1e2 ) ), 'LineStyle', ':' );
%     plot( f ./ (wo / (2*pi) ), -10*log10( h_tau_infinite_rigid_panel( f, wo, ms, s, rho0, c, panel.eta*1e-2 ) ), 'LineStyle', ':' );
%         %
%         legend( ...
%             'Target TL Values', ...
%             'Infinite Rigid Panel with Normal Incidence Sound', ...
%             'Infinite Rigid Panel with Normal Incidence Sound (eta * 100)', ...
%             'Infinite Rigid Panel with Normal Incidence Sound (eta / 100)', ...
%             'Location', 'North' );
%     %
%     xlabel( 'Frequency [$\frac{\omega}{\omega_o}$] ' );  ylabel( 'Transmission Loss [dB]' );
%     title( 'Measured Panel Transmission Losses - Change in Loss Factor' );
%     set( gca, 'XScale', 'log', 'YScale', 'log' );
%     % axis( [ 40 12e3 -5 45] );



%% Clean-up
```

```
% if ( ~isempty( findobj( 'Type', 'figure' ) ) )
%     monitors = get( 0, 'MonitorPositions' );
%         if ( size( monitors, 1 ) == 1 )
%             autoArrangeFigures( 2, 2, 1 );
%         elseif ( 1 < size( monitors, 1 ) )
%             autoArrangeFigures( 2, 2, 1 );
%         end
% end


fprintf( 1, '\n\n\n*** Processing Complete ***\n\n\n' );



%% Reference(s)
```

5 Appendix - Matlab Code for Problem 5

```matlab



%% Synopsis

% Problem 5 - Large Enclosure Design



%% Environment

close all; clear; clc;
% restoredefaultpath;

% addpath( genpath( '' ), '-begin' );
addpath( genpath( '../40 Assignments/00 Support' ), '-begin' );

% set( 0, 'DefaultFigurePosition', [ 400 400 900 400 ] ); % [ left bottom width height ]
set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
set( 0, 'DefaultFigureWindowStyle', 'normal' );
set( 0, 'DefaultLineLineWidth', 1.2 );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );



%% Define Machine

machine.area = 3; % m^2
machine.absorption = 0.07; % m^2 or Sabin
machine.D = 1; % Unitless - In air.

machine.distance = 10; % m



%% Measurement Data

octave_band_frequencies = [ 250 500 1000 2000 4000 ].'; % Hz
Lw = [ 105 115 106 108 119 ].'; % dB re: 1 pW



%% Per Octave Band Insertion Loss

Lp_10_meters = Lw + 10*log10( machine.D /( 4 * pi * machine.distance^2 ) ); % dB re: 20e-6 Pa
%
% The value of R is infinite. The machine is outside in open air.

octave_band_IL = Lp_10_meters - 30;



%% Define Anonymous Function for Insertion Loss

h_IL_large = @( Sw, alpha_w, Si, alpha_i, TL ) 10*log10( 1 + (Sw*alpha_w + Si*alpha_i)./(Sw
    + Si)*10^(TL/10) );



%% Find Values of TL and Aborption that will Meet the Target Insertion Loss - Ground Reflecting

% Assumption(s):
%
% 1.) The enclosure is a cube.
% 2.) The machine sits on the ground.
% 2.) There is no noise transmission through the ground.

enclosure.dimension = 2.0; % m
    enclosure.area = 5 * enclosure.dimension^2; % 20 m^2

% Volume of the enclosure is much bigger than the machine Diffuse sound field in the enclosure.
```

```

switch ( 4 )

    case 1

        % https://www.controlnoise.com/wp-content/uploads/2022/02/Acoustic-Enclosures-Datasheet.pdf

        % 250 Hz - QBV-2
        alpha_w = 0.27;  % From specification sheet.
        TL = 22;  % From specification sheet.
            IL_estimates(1) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );

        % 500 Hz - QBV-2
        alpha_w = 0.96;  % From specification sheet.
        TL = 28;  % From specification sheet.
            IL_estimates(2) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );

        % 1 kHz - QBV-2
        % alpha_w = 1.13;  % From specification sheet.
        alpha_w = 0.99;  % From specification sheet.  See comment on slide 28 of Lecture 10.
        TL = 40;  % From specification sheet.
            IL_estimates(3) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );

        % 2 kHz - QBV-2
        % alpha_w = 1.08;  % From specification sheet.
        alpha_w = 0.99;  % From specification sheet.  See comment on slide 28 of Lecture 10.
        TL = 56;  % From specification sheet.
            IL_estimates(4) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );

        % 4 kHz - QBV-2
        alpha_w = 0.99;  % From specification sheet.
        TL = 61;  % From specification sheet.
            IL_estimates(5) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );


    case 2

        % https://www.controlnoise.com/wp-content/uploads/2022/02/Acoustic-Enclosures-Datasheet.pdf

        % Note:  Abosrption values are carried over from QBV-2.

        % 250 Hz - QBV-3
        alpha_w = 0.96;  % From specification sheet.
        TL = 25;  % From specification sheet.
            IL_estimates(1) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );

        % 500 Hz - QBV-3
        alpha_w = 0.87;  % From specification sheet.
        TL = 33;  % From specification sheet.
            IL_estimates(2) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );

        % 1 kHz - QBV-3
        % alpha_w = 1.13;  % From specification sheet.
        alpha_w = 0.66;  % From specification sheet.  See comment on slide 28 of Lecture 10.
        TL = 46;  % From specification sheet.
            IL_estimates(3) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );

        % 2 kHz - QBV-3
        % alpha_w = 1.08;  % From specification sheet.
        alpha_w = 0.47;  % From specification sheet.  See comment on slide 28 of Lecture 10.
        TL = 53;  % From specification sheet.
            IL_estimates(4) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );

        % 4 kHz - QBV-3
        alpha_w = 0.30;  % From specification sheet.
        TL = 58;  % From specification sheet.
```

```
            IL_estimates(5) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                absorption, TL );


        case 3

            % https://www.acousticsworld.com/machine-acoustic-enclosures/

            % 250 Hz
            % alpha_w = 1.13;  % From specification sheet.
            alpha_w = 0.99;
            TL = 34;  % From specification sheet.
                IL_estimates(1) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                    absorption, TL );

            % 500 Hz
            % alpha_w = 1.12;  % From specification sheet.
            alpha_w = 0.99;
            TL = 48;  % From specification sheet.
                IL_estimates(2) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                    absorption, TL );

            % 1 kHz
            % alpha_w = 1.09;  % From specification sheet.
            alpha_w = 0.99;
            TL = 61;  % From specification sheet.
                IL_estimates(3) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                    absorption, TL );

            % 2 kHz
            % alpha_w = 1.03;  % From specification sheet.
            alpha_w = 0.99;
            TL = 66;  % From specification sheet.
                IL_estimates(4) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                    absorption, TL );

            % 4 kHz
            alpha_w = 0.91;  % From specification sheet.
            TL = 70;  % From specification sheet.
                IL_estimates(5) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                    absorption, TL );


        case 4

            % HTL-4 from specification sheet shown in class.

            % 250 Hz
            % alpha_w = 1.13;  % From specification sheet.
            alpha_w = 0.99;
            TL = 39;  % From specification sheet.
                IL_estimates(1) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                    absorption, TL );

            % 500 Hz
            % alpha_w = 1.12;  % From specification sheet.
            alpha_w = 0.99;
            TL = 59;  % From specification sheet.
                IL_estimates(2) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                    absorption, TL );

            % 1 kHz
            % alpha_w = 1.09;  % From specification sheet.
            alpha_w = 0.99;
            TL = 68;  % From specification sheet.
                IL_estimates(3) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                    absorption, TL );

            % 2 kHz
            % alpha_w = 1.03;  % From specification sheet.
            alpha_w = 0.99;
            TL = 67;  % From specification sheet.
                IL_estimates(4) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.
                    absorption, TL );

            % 4 kHz
            alpha_w = 0.91;  % From specification sheet.
            TL = 72;  % From specification sheet.
```

```
            IL_estimates(5) = h_IL_large( enclosure.area, alpha_w, machine.area, machine.absorption, TL );

end



%% Calculate the Differences

[ octave_band_IL    IL_estimates.'    (octave_band_IL - IL_estimates.') ]

data = [ ...
    250      44     28.0     19.7    10.6     5.6; ...
    500      54     26.7     22.2     6.6     -4.4; ...
    1000    45     5.6     1.4     -15.4     -22.4; ...
    2000    47     -8.4     -2.2     -18.4     -19.4; ...
    4000    58     -2.4     5.7      -11.0     -13.0 ];

figure( ); ...
    h1 = plot( data(:, 1 ), data( :, 3 ), 'LineStyle', '-', 'Color', 'k', 'Marker', '+', 'MarkerSize', 8 );  hold on;
    h2 = plot( data(:, 1 ), data( :, 4 ), 'LineStyle', '--', 'Color', 'k', 'Marker', '+', 'MarkerSize', 8 );
    h3 = plot( data(:, 1 ), data( :, 5 ), 'LineStyle', ':', 'Color', 'k', 'Marker', '+', 'MarkerSize', 8 );
    h4 = plot( data(:, 1 ), data( :, 6 ), 'LineStyle', '-.', 'Color', 'k', 'Marker', '+', 'MarkerSize', 8 );  grid on;
    line( [200 5500], [0 0 ], 'Color', 'r' );
    legend( [ h1 h2 h3 h4 ], { 'Case 1 - Control Noise QBV 2', 'Case 2 - Control Noise QBV 3', 'Case 3 - Acoustics World HTL (100MM)', 'Case 4 - HTL 4 (class table)' }, 'Location', 'SouthOutside' );
    xlabel( 'Frequency [Hz]' );  ylabel( 'Insertion Loss Difference [dB]' );
    %
    axis( [ 150 6e3  -30 30 ] );
    set( gca, 'XScale', 'log' );
    %
    % Textheight:  744 pt. and Textwidth:  493 pt. from LaTex document
    %
    set( gcf, 'units', 'point', 'pos', [ 200 200    493*0.8 744*0.5 ] );
        pos = get( gcf, 'Position' );
            set( gcf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'points', 'PaperSize', [pos(3), pos(4)] );
    %           print(gcf, 'Q5 IL Plot', '-dpdf', '-r0' );
%
% https://tex.stackexchange.com/questions/179382/best-practices-for-using-matlab-images-in-latex



%% Clean-up

% if ( ~isempty( findobj( 'Type', 'figure' ) ) )
%     monitors = get( 0, 'MonitorPositions' );
%         if ( size( monitors, 1 ) == 1 )
%             autoArrangeFigures( 2, 2, 1 );
%         elseif ( 1 < size( monitors, 1 ) )
%             autoArrangeFigures( 2, 2, 1 );
%         end
% end


fprintf( 1, '\n\n\n*** Processing Complete ***\n\n\n' );



%% Reference(s)

    %
    % Textheight:  744 pt. and Textwidth:  493 pt. from LaTex document
    %
    % set( gcf, 'units', 'point', 'pos', [ 200 200    493*0.8 744*0.45 ] );
    %     pos = get( gcf, 'Position' );
    %         set( gcf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'points', 'PaperSize', [pos(3), pos(4)] );
    %           print(gcf, 'Q4d TL for 75 AOI', '-dpdf', '-r0' );
%
% https://tex.stackexchange.com/questions/179382/best-practices-for-using-matlab-images-in-latex
```

6 Appendix - Matlab Code for Problem 6

```
%% Synopsis

% Lecture 11, Wednesday, February 19, 2025

% The compressor elevated above the ground.



%% Environment

close all; clear; clc;
% restoredefaultpath;

% addpath( genpath( '' ), '-begin' );
addpath( genpath( '../40 Assignments/00 Support' ), '-begin' );

% set( 0, 'DefaultFigurePosition', [  400  400  900  400  ] ); % [ left bottom width height ]
set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
set( 0, 'DefaultFigureWindowStyle', 'normal' );
set( 0, 'DefaultLineLineWidth', 0.8 );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );

PRINT_FIGURES = 0;



%% Define Anonymous Functions

h_RA_term_1 = @( rho0, c , S, k, delta_mu, D, w ) ( rho0*c/S ) * ( (k * sqrt( (2*3.178e-5) / ( rho0*w) ) * D * 0.004 ) / (2*S) *1.4364 );
h_RA_term_2 = @( rho0, c , S, k, delta_mu, D, w, h ) ( rho0*c/S ) * 0.288*k*3.178e-5*log10((4*S)/(pi*h^2));
h_RA_term_3 = @( rho0, c , S, k, delta_mu, D, w, h ) ( rho0*c/S ) * (0.5*S*k^2)/(2*pi);



%% Define Compressor

compressor.width = 1;  % m
compressor.depth = 1;  % m
compressor.height = 2;  % m
    compressor.area = 2*(compressor.width * compressor.depth) + 2*(compressor.width * compressor.height) + 2*(compressor.depth * compressor.height);  % 3 m^2
    compressor.volume = compressor.width * compressor.depth * compressor.height;  % m^3

compressor.power_level = 105;  % dB re: 1e-12 Watts
compressor.frequency = 50;  % Hz

c = 343;  % m/s

rho0 = 1.2;  % kg/m^3  CHECK



%% Sound Level Target

sound_level_target = 82;  % dB re: 20e-6 Pascals



%% Define Workshop

R = 40;  % m^2 or Sabins



%% Define Close-fitting Enclosure

```

```
helmholtz_factor = (2 * pi * compressor.frequency) / c;  % 0.92 m
%
%  For a small enclosure, k*d << 1.  Therefore d << 1.1.
d = 0.75;
d = 0.25;
% d = 1;
    % helmholtz_factor * d;  % 0.69

enclosure.width = compressor.width + d;  % m
enclosure.depth = compressor.depth + d;  % m
enclosure.height = 3;  % m;  compression height is 2 m
% enclosure.height = compressor.height + d;
    enclosure.area = 2*(enclosure.width * enclosure.depth) + 2*(enclosure.width * enclosure.height) + 2*(enclosure.depth * enclosure.height);
    enclosure.volume = enclosure.width * enclosure.depth * enclosure.height;

enclosure.E = 3.6e9;  % Pascals
enclosure.thickness = 3.81e-2;  % m
enclosure.density = 800;  % kg/m^3
enclosure.poisson_ratio = 0.25;  % Unitless

% Clamped boundary conditions.



%% Calculate Diffuse Sound Pressure Level

% Assume distance is beyond the critical distance, so the distance value is
% large and its associated term is not relevant.

sound_pressure_level = 105  +  10*log10( 4/R );  % 95 dB SPL



%% Calculate the Required Insertion Loss

target_insertion_loss = sound_pressure_level - 82  % 13 dB



%% Insertion Loss

% For the insertion loss to be high, we need:
%
%   1.)  Compliance of the air to be high;  volume of enclosure must be large.
%   2.)  Compliance of each enclosure wall to be low;  low area, high stiffness, edges clamped).
%             AREA IS THE DOMINATE FACTOR OVER VOLUME.


% The correction factor for clamped walls.  See Figure 12.4 on slide 9 of the Lecture 11 notes.
aspect_ratio = enclosure.height / enclosure.width;  % 1.7
    correction_factor = 2;  % Approximate value read from the Figure 12.4.

bending_stiffness = ( enclosure.E * enclosure.thickness^3 ) / ( 12*( 1 - enclosure.poisson_ratio^2 ) );  % 1.78e7
    h_wall_compliance = @( wall_area, correction_factor )  ( 0.001 * wall_area^3 * correction_factor ) / bending_stiffness;

Ca = enclosure.volume / ( rho0 * c^2 );


% Top
top.area = enclosure.width * enclosure.depth;
top.aspect_ratio = max( enclosure.width, enclosure.depth ) / min( enclosure.width, enclosure.depth );
top.correction_factor = 3.8;
    top.compliance = h_wall_compliance( top.area, top.correction_factor );


% Side 1
side_1.area = enclosure.depth * enclosure.height;
side_1.aspect_ratio = max( enclosure.width, enclosure.height ) / min( enclosure.width, enclosure.height );
side_1.correction_factor = 2;
    side_1.compliance = h_wall_compliance( side_1.area, side_1.correction_factor );

% Side 2
side_2.compliance = side_1.compliance;
```

```matlab


% Side 3
side_3.area = enclosure.width * enclosure.height;
side_3.aspect_ratio = max( enclosure.width, enclosure.height ) / min( enclosure.width, enclosure.height );
side_3.correction_factor = 2;
    side_3.compliance = h_wall_compliance( side_3.area, side_3.correction_factor );

% Side 4
side_4.compliance = side_3.compliance;


estimated_insertion_loss = 20*log10( 1 + Ca / ( top.compliance + 2*side_1.compliance + 2* side_3.compliance ) );  % 59.2 dB



%% Compliance of the Air Intake

air_intake_radius = 10e-2;  % m
air_intake_thickness = enclosure.thickness;  % m
air_intake_frequency = 50;  % Hz
    air_intake_angular_frequency = 2*pi*air_intake_frequency;  % radians/s

viscosity = 1.5e-5;  % m^2/s

h = 0.3;  % CHECK

f = 50;
    term_1 = h_RA_term_1( rho0, c, pi*(air_intake_radius*2)^2/4, 2*pi*f/c, sqrt( (2 * 3.178e-5 ) / ( 2*pi*f * rho0 ) ), pi * 0.1, 2*pi*f );
    term_2 = h_RA_term_2( rho0, c, pi*(air_intake_radius*2)^2/4, 2*pi*f/c, sqrt( (2 * 3.178e-5 ) / ( 2*pi*f * rho0 ) ), pi * 0.1, 2*pi*f, 0.3 );
    term_3 = h_RA_term_3( rho0, c, pi*(0.1)^2/4, 2*pi*f/c, sqrt( (2 * 3.178e-5 ) / ( 2*pi*f * rho0 ) ), pi * 0.1, 2*pi*f, 0.3 );
        impedance.real = term_1 + term_2 + term_3;


% Deng (1998)
epsilon = 1;
    L_o = air_intake_radius * ( 1.27 / (1 + 1.92*epsilon) - 0.086 );

L_e =  enclosure.thickness + 2*L_o;
    impedance.imaginary = 1j * rho0 * (2 * pi * f) * L_e / ( pi*0.1^2/4 );


impedance.net = impedance.real + impedance.imaginary;
    compliance_of_hole = 1 / impedance.net;
        Cl = abs( compliance_of_hole );

estimated_insertion_loss
estimated_insertion_loss_with_hole = 20*log10( (Cl + Ca)  / ( Cl + ( top.compliance + 2*side_1.compliance + 2* side_3.compliance ) ) )


13 - estimated_insertion_loss_with_hole;



critical_frequency = c^2/(2*pi)*sqrt( enclosure.density * enclosure.thickness / bending_stiffness );
%
% The critical frequency is 25 Hz.
%
% The frequency of the compressor is 50 Hz.





%% Clean-up

if ( ~isempty( findobj( 'Type', 'figure' ) ) )
    monitors = get( 0, 'MonitorPositions' );
        if ( size( monitors, 1 ) == 1 )
            autoArrangeFigures( 2, 2, 1 );
        elseif ( 1 < size( monitors, 1 ) )
```

```
            autoArrangeFigures( 2, 2, 1 );
        end
end


fprintf( 1, '\n\n\n*** Processing Complete ***\n\n\n' );



%% Reference(s)
```