

Problem 1 - Cut-on Frequencies in Ducts and Pipes

The Matlab code for this problem is listed in Appendix [1](#).

Problem 1a

The lowest cut-on frequency for a rectangular duct with air flow is given by equation,

$$f_{\text{cut-on}} = 0.5 \cdot \frac{c}{L} \quad (1)$$

where c is the speed of sound in air, $343 \frac{\text{m}}{\text{s}}$, and L is the largest side of the rectangular cross-section.

With cross-sectional dimensions of $L_x = 12 \text{ cm}$ and $L_y = 20 \text{ cm}$, the lowest cut-on frequency for this rectangular duct is,

$$f_{\text{cut-on}} = 0.5 \cdot \frac{343 \frac{\text{m}}{\text{s}}}{0.20 \text{ m}} = \mathbf{857.5 \text{ Hz}}$$

Problem 1b

The lowest cut-on frequency for a circular duct with air flow with the same cross-sectional area as the rectangular duct in part (a.) can be calculated using equation,

$$f_{\text{cut-on}} = 0.568 \cdot \frac{c}{d} \quad (2)$$

where c is the speed of sound in air, $343 \frac{\text{m}}{\text{s}}$, and d is diameter of the circular duct.

The cross-sectional area of the rectangular duct is,

$$\text{Area}_{\text{rectangular duct}} = 0.12 \text{ m} \cdot 0.20 \text{ m} = 0.024 \text{ m}^2$$

The corresponding diameter for this area is,

$$\text{diameter} = \sqrt{\frac{0.024 \text{ m}^2}{\pi}} \cdot 2 = 0.17 \text{ m}$$

Using Eq. [2](#), the lowest cut-on frequency for this circular duct with air flow is,

$$f_{\text{cut-on}} = 0.568 \cdot \frac{343 \frac{\text{m}}{\text{s}}}{0.17 \text{ m}} = \mathbf{1,114.5 \text{ Hz}}$$

Problem 1c

The lowest cut-on frequency for this circular duct with water flow can be calculated using Eq. 2,

$$f_{\text{cut-on}} = 0.568 \cdot \frac{1,500 \frac{\text{m}}{\text{s}}}{0.17 \text{ m}} = 4,873.9 \text{ Hz}$$

The lowest cut-on frequency for water is considerable larger than it is for air flow.

Problem 1d

The speed of sound in air is calculated by,

$$c = \sqrt{\gamma \cdot R \cdot T_K} \quad (3)$$

where $\gamma = 1.4$ is the ratio of specific heats, $R = 287 \frac{\text{J}}{\text{kg} \cdot \text{K}}$ is the gas constant, and T_K is the absolute temperature in Kelvin.

Figure 1 illustrates how the lowest cut-on frequency changes as the air heats from 0° to 500° Celsius.

The square-root relationship between temperature and the speed of sound in air is apparent and governs the behaviour of the cut-on frequency.

Lowest Cut-on Frequency for a Circular Pipe with Air Flow Versus Air Temperature

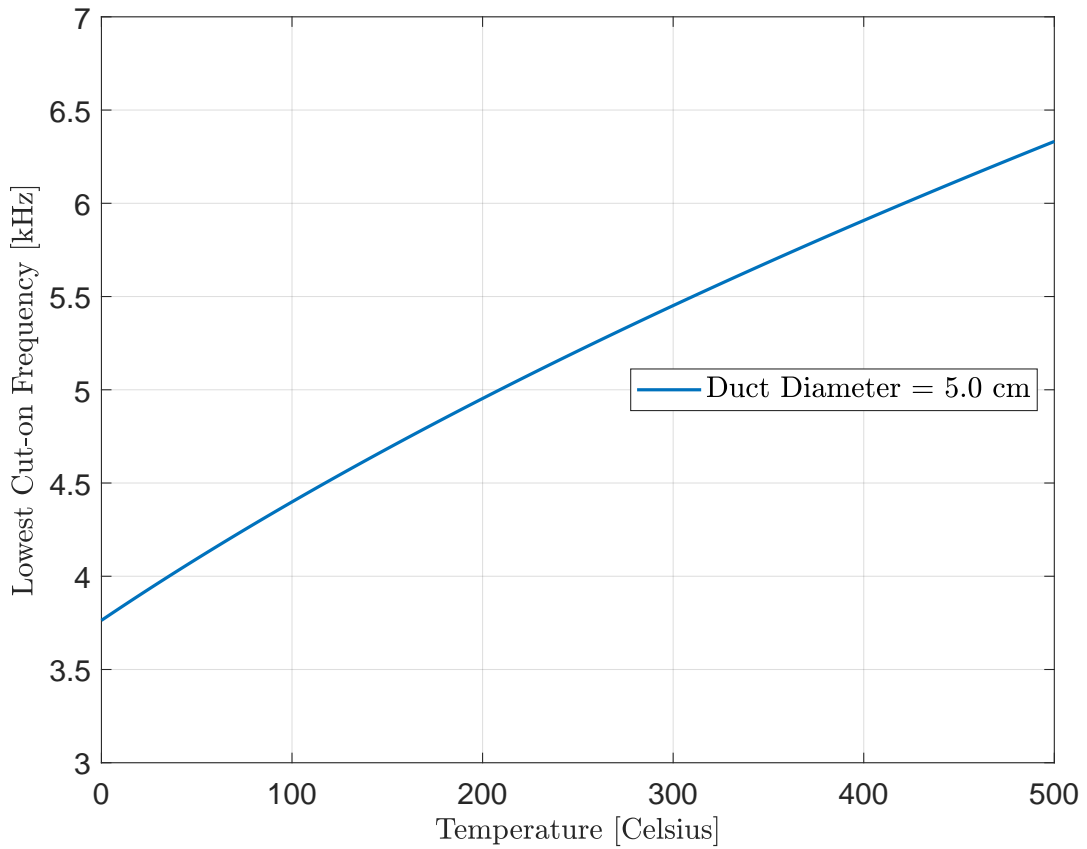


Figure 1: Lowest cut-on frequency for a circular 5 cm diameter duct versus air temperature.

Problem 1e

Question: Are cut-on frequencies higher for a circular or rectangular duct for a given cross-sectional area?

The lowest cut-on frequency is higher for a circular duct than for a rectangular duct for a given cross-sectional area.

For the dimensions given in class, the rectangular duct is not square. This produces a larger dimension and thus a smaller, lowest cut-on frequency. If the rectangular duct is square dimensions on the order of the circular duct diameter with the same cross-sectional area, the cut-on frequencies are approximately equal.

Question: What about in air versus water?

The lowest cut-on frequency is larger for water than for air. The cut-on frequency is proportional to the speed of sound and the speed of sound in water is greater than the speed of sound in air.

Question: What about cold versus hot air?

The lowest cut-on frequency is higher for warm air than it is for cold air.

Problem 2 - Muffler Design Comparison

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

For these muffler comparisons, the following assumptions were made:

- There is no flow.
- There are no resistive terms.
- The load impedance was not included because the transmission loss does not require them.
- For Parts b, c, and d, the side branch length offset, L_o , was set to zero.

Problems 2a, 2b, and 2c

Figure 2 shows the transmission loss profiles for a simple expansion chamber, a double-tuned expansion chamber, and a cascaded double-tuned expansion chamber muffler.

The peaks for the simple expansion chamber (red, dashed line) are approximately 22 dB and occur at frequencies with a wavelength that is a quarter of the length of the expansion chamber. Minimal loss occurs at half wavelength multiples.

The addition of the extension tube inside the muffler produces a quarter wavelength resonator. The side branch of Ji (2005; Slide 11, Lecture 3 notes) was used to calculate L_o . For the cascaded double-tuned expansion chamber, the extension tubes produce a secondary quarter wavelength resonator.

As noted in the office hours session, there is no damping which produces artificially high resonances.

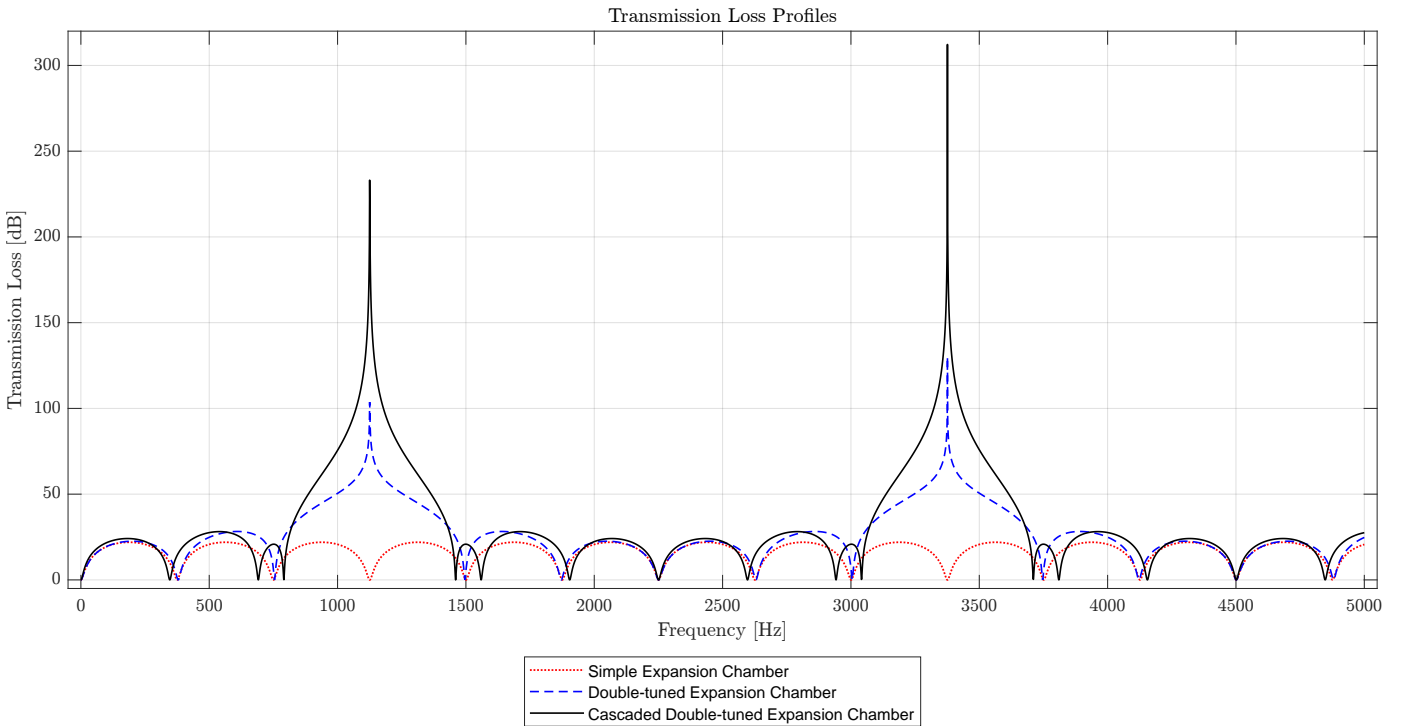


Figure 2: Transmission loss profiles for a simple expansion chamber, a double-tuned expansion chamber, and a cascaded double-tuned expansion chamber mufflers.

Problem 2d

Figure 3 shows the transmission loss profiles for a cascaded double-tuned expansion chamber, and a modified cascaded double-tuned expansion chamber muffler.

Two modifications were made to the original system:

1. The left 3" extension tube in the left chamber was shortened to 2" inches, making the respective muffler section 1" longer.
2. The left 3" extension tube in the right chamber was lengthened to 4", making the respective muffler section 1" shorter.

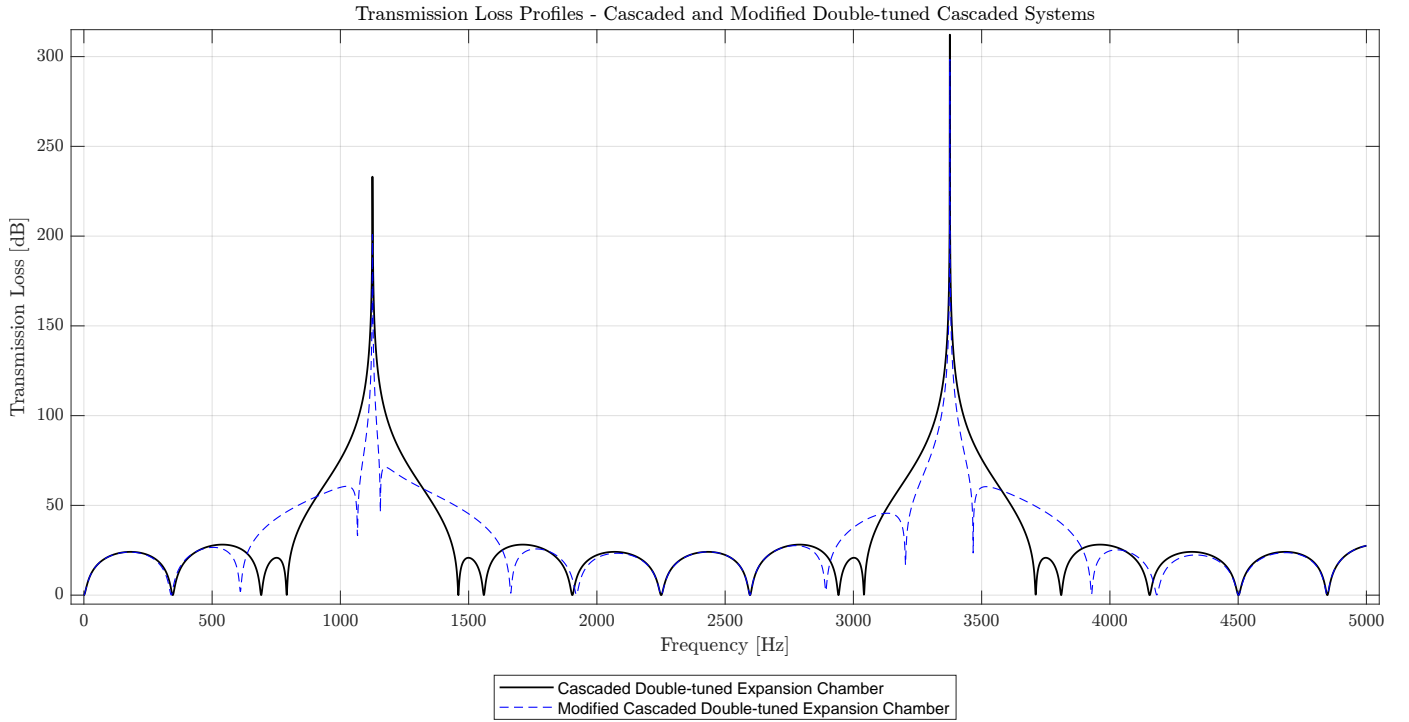


Figure 3: Transmission loss profiles for a cascaded double-tuned expansion chamber muffler and a modified version of this muffler.

These modifications change the symmetry of the cascaded system, and allow the resonate frequencies to be independently changed.

Problem 3 - Bugle Recorder

Problem 3a

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

Table 1 lists the length of the pipe section and the mouthpiece.

Item	Length [mm]
Pipe	145
Mouthpiece	90

Table 1: Calculated length of the pipe and length of the mouthpiece.

Problem 3b

Table 2 summarizes the placement of the holes for each note relative to the end of the pipe.

Note	Frequency [Hz]	Length from End of Pipe [mm]
C5	523	n/a
F5	698	87.75
A5	880	138.25
C6	1,046	0.168

Table 2: Hole placement distances.

Figure 4 shows the respective spectrum for each of the bugle recorder notes.

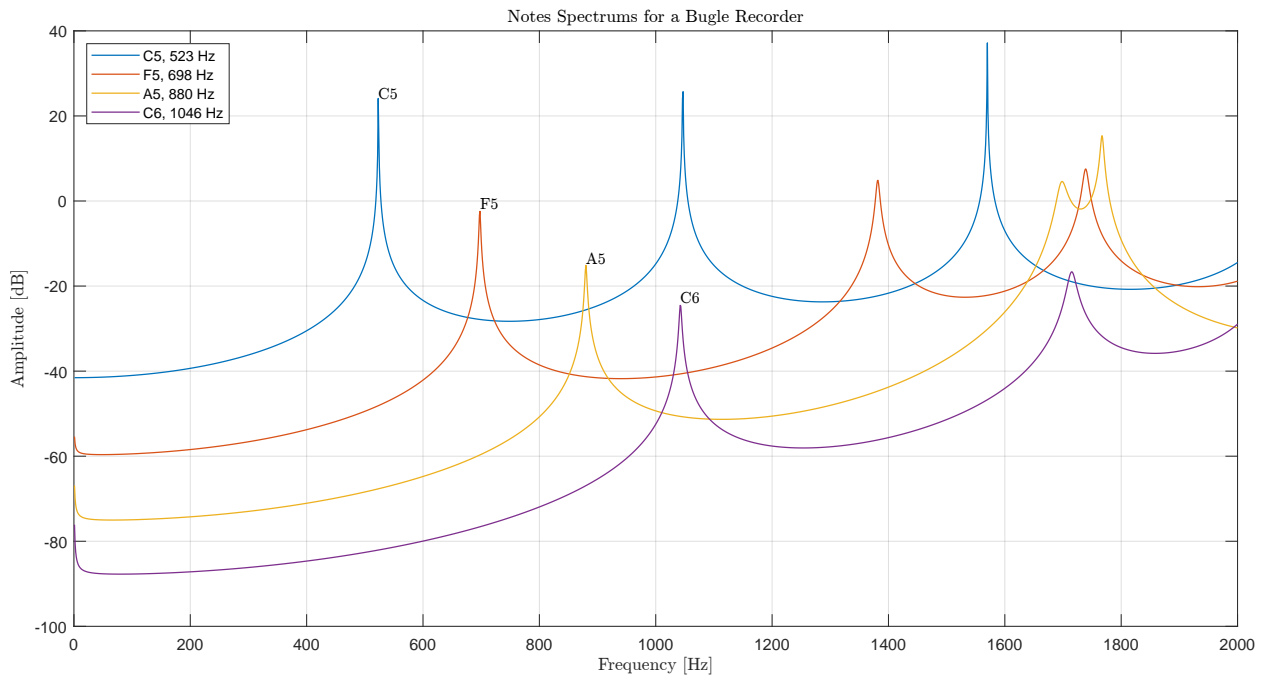


Figure 4: Spectrum for the C5, F5, A5, and C6 bugle recorder notes.

Problem 4 - Intake Duct

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

Problem 4a

Table 3 lists the Mach numbers for each pipe section.

Pipe	Mach Number [unitless]
Inlet	-0.10047
Outlet	-0.0062795

Table 3: Calculated Mach numbers.

Problem 4b

Problem 4c

Problem 4d

Problem 5 - Intake Duct Silencer

Problem 5a

Problem 5b

Problem 5c

Problem 5d

Problem 5e

1 Appendix - Matlab Code for Problem 1

```
1
2
3
4 %% Synopsis
5
6 % Question 1 – Cut-on Frequencies in Ducts and Pipes
7
8
9
10 %% Environment
11
12 close all; clear; clc;
13 % restoredefaultpath;
14
15 % addpath( genpath( '' ), '-begin' );
16 addpath( genpath( '../40 Assignments/00 Support' ), '-begin' );
17
18 % set( 0, 'DefaultFigurePosition', [ 400 400 900 400 ] ); % [ left bottom width height ]
19 set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
20 set( 0, 'DefaultFigureWindowStyle', 'normal' );
21 set( 0, 'DefaultLineLineWidth', 1.5 );
22 set( 0, 'DefaultTextInterpreter', 'Latex' );
23
24 format ShortG;
25
26 pause( 1 );
27
28 PRINT_FIGURES = 0;
29
30
31
32 %% Define Constants and Anonymous Functions
33
34 c_air = 343; % The speed of sound in air (meters per second).
35 c_water = 1500; % The speed of sound in water (meters per second).
36
37 gamma = 1.4; % The ratio of specific heats [unitless].
38 R = 287; % The gas constant [Joules per (kilogram * Kelvin)].
39
40
41 h_f_cut_on_rectangular_duct = @( c, L ) 0.5 .* c ./ L;
42 %
43 % c – The speed of sound.
44 % L – The largest cross-section dimension of the rectangular duct.
45
46
47 h_f_cut_on_circular_duct = @( c, d ) 0.568 .* c ./ d;
48 %
49 % c – The speed of sound.
50 % L – The diameter of the circular duct.
51
52
53 h_speed_of_sound_in_air = @( gamma, R, temperature_Kelvin ) sqrt( gamma .* R .*
    temperature_Kelvin );
54
55
56
57 %% Problem 1a
58
59 % The cross-sectional dimensions for the rectangular duct are:  $L_x = 12$  cm and  $L_y = 20$  cm.
60
61 % The largest dimension is  $L_y = 20$  cm or 0.2 m.
62
63 % The cut-on frequency is,
64 h_f_cut_on_rectangular_duct( c_air, 0.2 ); % 857.5 Hz (shown in class 858 Hz)
65 fprintf( 1, '\n Problem 1a: The lowest cut-on frequency for the rectangular pipe with air is
    %3.1f Hz.\n', h_f_cut_on_rectangular_duct( c_air, 0.2 ) );
66
67
68
69 %% Problem 1b
70
71 % The cross-sectional dimensions for the rectangular duct are:  $L_x = 12$  cm and  $L_y = 20$  cm.
72
73 % The cross-sectional area of the rectangular duct is  $12$  cm *  $20$  cm =  $240$  cm2 or  $0.024$  m2.
```

```

74 rectangular_duct_cross_sectional_area = 0.12 * 0.20; % 0.024 m^2
75
76 % The diameter of the circular pipe is,
77 circular_duct_diameter = sqrt( 0.024 / pi ) * 2; % 0.17481 meters
78 %
79 % Check:
80 % pi * ( circular_duct_diameter / 2 )^2 CHECKED
81
82
83 % The cut-on frequency for the circular duct is,
84 h_f_cut_on_circular_duct( c_air, circular_duct_diameter ); % 1,114.5 Hz
85 fprintf( 1, '\n Problem 1b: The lowest cut-on frequency for the circular pipe (of equal area
    ) with air is %3.1f Hz.\n', h_f_cut_on_circular_duct( c_air, circular_duct_diameter ) );
86
87
88
89 %% Problem 1c
90
91 % The cut-on frequency for the circular duct with water is,
92 h_f_cut_on_circular_duct( c_water, circular_duct_diameter ); % 4,873.9 Hz
93 fprintf( 1, '\n Problem 1c: The lowest cut-on frequency for the circular pipe (of equal area
    ) with water is %3.1f Hz.\n', h_f_cut_on_circular_duct( c_water, circular_duct_diameter ) );
94
95 % The cut-on frequency should be higher because it is proportional to the
96 % speed of sound in a given medium.
97
98
99
100 %% Problem 1d
101
102 fprintf( 1, '\n Problem 1d: See the figure.\n' );
103
104 temperature_range_celsius = 0:0.1:500; % Celsius
105 temperature_range_kelvin = temperature_range_celsius + 273.15; % Kelvin
106
107
108 FONT_SIZE = 14;
109
110 figure( ); ...
111 plot( temperature_range_celsius, h_f_cut_on_circular_duct( h_speed_of_sound_in_air( gamma, R,
    temperature_range_kelvin ), 0.05 ) ./ 1e3 ); grid on;
112 legend( 'Duct Diameter = 5.0 cm', 'Location', 'East', 'FontSize', FONT_SIZE, 'Interpreter
    ', 'Latex' );
113 set( gca, 'FontSize', FONT_SIZE );
114 %
115 xlabel( 'Temperature [Celsius]', 'FontSize', FONT_SIZE );
116 % xl = get( gca, 'xlabel' ); pxl = get( xl, 'position' ); pxl( 2 ) = 1.1 * pxl( 2 );
117 % set( xl, 'position', pxl );
118 %
119 ylabel( 'Lowest Cut-on Frequency [kHz]', 'FontSize', FONT_SIZE );
120 % yl = get( gca, 'ylabel' ); pyl = get( yl, 'position' ); pyl( 1 ) = 1.2 * pyl( 1 );
121 % set( yl, 'position', pyl );
122 %
123 caption = sprintf( 'Lowest Cut-on Frequency for a Circular Pipe with Air Flow Versus Air
    Temperature\n' );
124 title( caption, 'FontSize', FONT_SIZE );
125 %
126 ylim( [ 3 7 ] );
127
128
129
130 %% Problem 1e
131
132 fprintf( 1, '\n Problem 1e: See Section Problem 1e of the Matlab script for the answers.\n\n' );
133
134
135 % Question: Are cut-on frequencies higher for a circular or rectangular duct for a given cross-
    sectional area?
136
137 % The lowest cut-on frequency is higher for a circular duct than for a
138 % rectangular duct for a given cross-sectional area.
139
140 % For the dimensions given in class, the rectangular duct is not square.
141 % This produces a larger dimension and thus a smaller, lowest cut-on
142 % frequency.
143
144 % If the rectangular duct is square dimensions on the order of the circular
145 % duct diameter with the same cross-sectional area, the the cut-on

```

```

146 % frequencies are approximately equal.
147
148
149 % Question: What about in air versus water?
150
151 % The lowest cut-on frequency is larger with water than air. This due to
152 % the fact that the cut-on frequency is proportional to the speed of sound
153 % and the speed of sound in water is greater than it is in air.
154
155
156 % Question: What about cold versus hot air?
157
158 % For a circular pipe, the cut-on frequency is higher in warm air than cold
159 % air.
160
161
162
163 %% Clean-up
164
165 if ( ~isempty( findobj( 'Type', 'figure' ) ) )
166     monitors = get( 0, 'MonitorPositions' );
167     if ( size( monitors, 1 ) == 1 )
168         autoArrangeFigures( 2, 2, 1 );
169     elseif ( 1 < size( monitors, 1 ) )
170         autoArrangeFigures( 2, 2, 1 );
171     end
172 end
173
174 if ( PRINT_FIGURES == 1 )
175     saveas((gcf, 'Cut-on Frequency Versus Temperature - Sunday, January 19, 2025.pdf' );
176 end
177
178 fprintf( 1, '\n\n*** Processing Complete ***\n\n' );

```

2 Appendix - Matlab Code for Problem 2

```
1
2
3
4 %% Synopsis
5
6 % Question 2 – Muffler Design Comparison
7
8
9
10 %% Environment
11
12 close all; clear; clc;
13 % restoredefaultpath;
14
15 % addpath( genpath( '' ), '-begin' );
16 addpath( genpath( '../00 Support' ), '-begin' );
17
18 % set( 0, 'DefaultFigurePosition', [ 400 400 900 400 ] ); % [ left bottom width height ]
19 set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
20 set( 0, 'DefaultFigureWindowStyle', 'normal' );
21 set( 0, 'DefaultLineLineWidth', 1.5 );
22 set( 0, 'DefaultTextInterpreter', 'Latex' );
23
24 format ShortG;
25
26 pause( 1 );
27
28 PRINT_FIGURES = 0;
29
30
31
32 %% Constants
33
34 rho0 = 1.21; % Air density (kg per m^3).
35 c = 343; % Speed of sound in air (meters per second).
36
37 frequency_set = 0:1:5e3; % Hertz
38
39
40
41 %% Dimensions
42
43 convert.inches_to_meters = 0.0254;
44 convert.foot_to_meters = 0.3048;
45
46 dimensions.inlet_diameter_meters = 2 * convert.inches_to_meters; % 0.0508 meters
47 dimensions.inlet_length_meters = 6 * convert.foot_to_meters; % 1.82 meters
48
49 dimensions.muffler_diameter_meters = 10 * convert.inches_to_meters; % 0.254 meters
50 dimensions.muffler_length_meters = 18 * convert.inches_to_meters; % 0.4572 meters
51
52 dimensions.outlet_diameter_meters = 2 * convert.inches_to_meters; % 0.0508 meters
53 dimensions.outlet_length_meters = 1 * convert.foot_to_meters; % 0.3048 meters
54
55 outlet_flanged = false;
56
57 dimensions.overhang = 3 * convert.inches_to_meters; % 0.0762 meters
58
59 segment_diameters = [ ...
60     dimensions.outlet_diameter_meters, ...
61     dimensions.muffler_diameter_meters, ...
62     dimensions.inlet_diameter_meters, ...
63 ] .';
64 %
65 h_area_from_diameter = @( d ) pi .* d.^2 ./ 4;
66 %
67 segment_areas = h_area_from_diameter( segment_diameters );
68
69 segment_lengths = [ ...
70     dimensions.outlet_length_meters, ...
71     dimensions.muffler_length_meters, ...
72     dimensions.inlet_length_meters, ...
73     dimensions.overhang, ...
74 ] .';
75
```

```

76
77
78 %% Part a - Simple Expansion Chamber
79
80 nFreq = length( frequency_set );
81 TL = zeros( nFreq, 1 );
82
83 for frequency_index = 1:1:nFreq
84
85     f = frequency_set( frequency_index );
86
87     T_total = [ 1 0; 0 1 ];
88
89     T1 = duct_segment_transfer_matrix( f, rho0, c, 0.3048, 0.0020268 ); % Duct - Outlet
90     T2 = duct_segment_transfer_matrix( f, rho0, c, 0.4572, 0.050671 ); % Duct
91     T3 = duct_segment_transfer_matrix( f, rho0, c, 1.8288, 0.0020268 ); % Duct - Inlet
92
93     T_net = T3 * T2 * T1 * T_total;
94     % T_net = T_inlet * T_total; % Zero transmission loss for a straight duct.
95
96     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
97     TL( frequency_index ) = 10 * log10( abs( ( T11 + 0.0020268*T12/(rho0*c) + (rho0*c)*
98         T21/0.0020268 + T22 ) / 2 )^2 );
99     %
100     % The transmission loss calculation does not require a load impedance.
101 end
102
103 TL_parta = TL; % The maximum peak value should be about 22 (21.952) dB.
104 %
105 % Expected behaviour:
106 %
107 % 1.) 0 dB at 0 Hz.
108 % 2.) The transmission loss of a straight duct section is zero; energy out equals energy in.
109
110 max( TL_parta ); % 22 dB
111
112
113
114 %% Part b - Double-tuned Expansion Chamber
115
116 annulus_area_squared_meters = pi/4 * ( 0.254^2 - 0.0508^2 );
117
118 L_o = 0; % Assume that the L_o extension is negligible.
119
120
121 nFreq = length( frequency_set );
122 TL = zeros( nFreq, 1 );
123
124 for frequency_index = 1:1:nFreq
125
126     f = frequency_set( frequency_index );
127
128     T_total = [ 1 0; 0 1 ];
129
130     T1 = duct_segment_transfer_matrix( f, rho0, c, (0.3048 + 0.0762), 0.0020268 );
131     T3 = duct_segment_transfer_matrix( f, rho0, c, (0.4572 - 2*0.0762), 0.050671 );
132     T5 = duct_segment_transfer_matrix( f, rho0, c, (1.8288 + 0.0762), 0.0020268 );
133
134     k = 2*pi*f/c;
135     Z_A = -1j*rho0*c/annulus_area_squared_meters*cot(k * ( 0.0762 + L_o ));
136     T2 = [ 1 0; 1/Z_A 1 ];
137     T4 = T2;
138
139     T_net = T5 * T4 * T3 * T2 * T1 * T_total;
140
141
142     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
143     TL( frequency_index ) = 10 * log10( abs( ( T11 + 0.0020268*T12/(rho0*c) + (rho0*c)*
144         T21/0.0020268 + T22 ) / 2 )^2 );
145     %
146     % The transmission loss calculation does not require a load impedance.
147 end
148
149 TL_partb = TL;
150 %
151 % Expected behaviour:

```

```

152 %
153 % 1.) 0 dB at 0 Hz.
154 % 2.) 0 dB at same locations as a simple expansion chamber.
155 % 3.) Peaks at 1,125 Hz and 3,376 Hz;
156
157 % Frequency at which the quarter-wavelength is 0.0762 meters.
158 %  $343 / (4 * 0.0762)$ ; % 1,125 Hz.
159
160 % Also work at three-quarter-wavelength.
161 %  $3 * 1125$ ; % 3,375 Hz
162
163
164
165 %% Part c - Cascaded, Double-tuned Expansion Chamber
166
167 annulus_area_squared_meters = pi/4 * ( 0.254^2 - 0.0508^2 );
168
169 L_o = 0; % Assume that the L_o extension is negligible.
170
171
172 nFreq = length( frequency_set );
173 TL = zeros( nFreq, 1 );
174
175 for frequency_index = 1:1:nFreq
176
177     f = frequency_set( frequency_index );
178     k = 2*pi*f/c;
179     Z_A = -lj*rho0*c/annulus_area_squared_meters*cot(k * ( 0.0762 + L_o ));
180
181     T_total = [ 1 0; 0 1 ];
182
183     T1 = duct_segment_transfer_matrix( f, rho0, c, (0.3048 + 0.0762), 0.0020268 ); % Duct -
184     % Outlet
185     T2 = [ 1 0; 1/Z_A 1 ]; % Straight Side Branch
186     T3 = duct_segment_transfer_matrix( f, rho0, c, (0.2286 - 2*0.0762), 0.050671 ); % Duct
187     T4 = T2; % Straight Side Branch
188     T5 = duct_segment_transfer_matrix( f, rho0, c, 2*0.0762, 0.050671 ); % Duct
189     T6 = T2; % Straight Side Branch
190     T7 = T3; % Duct
191     T8 = T2; % Straight Side Branch
192     T9 = duct_segment_transfer_matrix( f, rho0, c, (1.8288 + 0.0762), 0.0020268 ); % Duct -
193     % Inlet
194
195     T_net = T9 * T8 * T7 * T6 * T5 * T4 * T3 * T2 * T1 * T_total;
196
197     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
198     TL( frequency_index ) = 10 * log10( abs( ( T11 + 0.0020268*T12/(rho0*c) + (rho0*c)*
199     T21/0.0020268 + T22 ) / 2 )^2 );
200
201 end
202
203 TL_partc = TL;
204
205
206 %% Part d - Cascaded, Double-tuned Expansion Chamber
207
208 nFreq = length( frequency_set );
209 TL = zeros( nFreq, 1 );
210
211 for frequency_index = 1:1:nFreq
212
213     f = frequency_set( frequency_index );
214     k = 2*pi*f/c;
215     Z_A = -lj*rho0*c/annulus_area_squared_meters*cot(k * ( 0.0762 + L_o ));
216
217     T_total = [ 1 0; 0 1 ];
218
219     T1 = duct_segment_transfer_matrix( f, rho0, c, (0.3048 + 0.0762), 0.0020268 ); % Duct -
220     % Outlet
221     T2 = [ 1 0; 1/Z_A 1 ]; % Straight Side Branch
222     T3 = duct_segment_transfer_matrix( f, rho0, c, 0.0076, 0.050671 ); % Duct
223     T4 = T2; % Straight Side Branch
224     T5 = duct_segment_transfer_matrix( f, rho0, c, 2*0.0762, 0.050671 ); % Duct
225     T6 = T2; % Straight Side Branch
226     T7 = duct_segment_transfer_matrix( f, rho0, c, (0.29718 - 2*0.0762), 0.050671 ); % Duct
227     T8 = T2; % Straight Side Branch
228     T9 = duct_segment_transfer_matrix( f, rho0, c, (1.8288 + 0.0762), 0.0020268 ); % Duct -

```

```

226 Inlet
227 T_net = T9 * T8 * T7 * T6 * T5 * T4 * T3 * T2 * T1 * T_total;
228
229 T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
230 TL( frequency_index ) = 10 * log10( abs( ( T11 + 0.0020268*T12/(rho0*c) + (rho0*c)*
231 T21/0.0020268 + T22 ) / 2 )^2 );
232 end
233
234 TL_partd = TL;
235
236
237
238 %% Plot Transmission Loss Profiles
239
240 Y_LIMITS = [ -5 320 ];
241
242 h_figure_1 = figure( ); ...
243 plot( frequency_set, TL_parta, 'LineWidth', 1.0, 'LineStyle', ':', 'Color', 'r' ); hold on;
244 plot( frequency_set, TL_partb, 'LineWidth', 0.9, 'LineStyle', '—', 'Color', 'b' );
245 plot( frequency_set, TL_partc, 'LineWidth', 0.9, 'LineStyle', '—', 'Color', 'k' ); grid on;
246 legend( ...
247     'Simple Expansion Chamber', ...
248     'Double-tuned Expansion Chamber', ...
249     'Cascaded Double-tuned Expansion Chamber', ...
250     'Location', 'SouthOutside' );
251 xlabel( 'Frequency [Hz]' ); ylabel( 'Transmission Loss [dB]' );
252 title( 'Transmission Loss Profiles' );
253 %
254 Ax = gca;
255 Ax.XAxis.TickLabelInterpreter = 'latex';
256 Ax.YAxis.TickLabelInterpreter = 'latex';
257 %
258 axis( [ -50 5e3+50 Y_LIMITS ] );
259
260
261 Y_LIMITS = [ -5 315 ];
262
263 h_figure_2 = figure( ); ...
264 plot( frequency_set, TL_partc, 'LineWidth', 1.0, 'LineStyle', '—', 'Color', 'k' ); hold on;
265 plot( frequency_set, TL_partd, 'LineWidth', 0.6, 'LineStyle', '—', 'Color', 'b' ); grid on;
266 legend( ...
267     'Cascaded Double-tuned Expansion Chamber', ...
268     'Modified Cascaded Double-tuned Expansion Chamber', ...
269     'Location', 'SouthOutside' );
270 xlabel( 'Frequency [Hz]' ); ylabel( 'Transmission Loss [dB]' );
271 title( 'Transmission Loss Profiles – Cascaded and Modified Double-tuned Cascaded Systems' );
272 %
273 Ax = gca;
274 Ax.XAxis.TickLabelInterpreter = 'latex';
275 Ax.YAxis.TickLabelInterpreter = 'latex';
276 %
277 axis( [ -50 5e3+50 Y_LIMITS ] );
278
279
280
281 %% Clean-up
282
283 if ( ~isempty( findobj( 'Type', 'figure' ) ) )
284     monitors = get( 0, 'MonitorPositions' );
285     if ( size( monitors, 1 ) == 1 )
286         autoArrangeFigures( 2, 2, 1 );
287     elseif ( 1 < size( monitors, 1 ) )
288         autoArrangeFigures( 2, 2, 1 );
289     end
290 end
291
292 if ( PRINT_FIGURES == 1 )
293     exportgraphics( h_figure_1, 'Assignment 1 – Question 2 Figure All TL Profiles.pdf', '
Append', true );
294     exportgraphics( h_figure_2, 'Assignment 1 – Question 2 Figure Comparison TL Plot For
Cascaded Systems.pdf', 'Append', true );
295 end
296
297 fprintf( 1, '\n\n*** Processing Complete ***\n\n' );

```

3 Appendix - Matlab Code for Problem 3

```
1
2
3
4 %% Synopsis
5
6 % Question 3 – Bugle Recorder
7
8
9
10 %% Environment
11
12 close all; clear; clc;
13 % restoredefaultpath;
14
15 % addpath( genpath( '' ), '-begin' );
16 addpath( genpath( '../00 Support' ), '-begin' );
17
18 % set( 0, 'DefaultFigurePosition', [ 400 400 900 400 ] ); % [ left bottom width height ]
19 set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
20 set( 0, 'DefaultFigureWindowStyle', 'normal' );
21 set( 0, 'DefaultLineLineWidth', 0.8 );
22 set( 0, 'DefaultTextInterpreter', 'Latex' );
23
24 format ShortG;
25
26 pause( 1 );
27
28 PRINT_FIGURES = 0;
29
30
31
32 %% Constants and Anonymous Functions
33
34 rho0 = 1.21; % Density of air (kg per cubic-meter).
35 c = 343; % Speed of sound in air (meters per second).
36
37 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 );
38 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));
39 h_RA_term_3 = @( rho0, c, S, k, delta_mu, D, w, h ) ( rho0*c/S ) * (0.5*S*k^2)/(2*pi);
40 %
41 % See Equation 8.34 on page 479 of Bies et al (2024).
42
43
44
45 %% Part a
46
47 % The estimated total length of the recorder is 0.325 meters.
48
49 % Estimation of hole location was done by trial-and-error.
50
51 pipe_net_length = 0.325;
52 pipe_area = pi*0.009^2/4;
53
54 flanged = false;
55
56
57 frequency_set = 1:1:2e3;
58
59 nFreq = length( frequency_set );
60 A = zeros( nFreq, 1 );
61
62 for frequency_index = 1:1:nFreq
63
64     f = frequency_set( frequency_index );
65
66     T_total = [ 1 0; 0 1 ];
67
68     T_segment = duct_segment_transfer_matrix( f, rho0, c, pipe_net_length, pipe_area );
69
70     T_total = T_segment * T_total;
71
72     Z = open_end_impedance( f, rho0, c, 0, pipe_area, flanged );
73
```



```

74     T11 = T_total(1, 1); T12 = T_total(1, 2);
75     A( frequency_index ) = -10*log10( abs( T11 + T12 / Z )^2 );
76
77 end
78
79 A_parta = A; clear A;
80
81 % figure( ); ...
82 % plot( frequency_set, A_parta, 'LineWidth', 0.8 ); grid on;
83 % xlabel( 'Frequency [Hz]' ); ylabel( 'Amplitude [dB]' );
84 % legend( 'C5' );
85
86 % return
87
88 %% Part b
89
90 epsilon = 0.006 / 0.004; % Diameter of the hold divided by diameter of pipe section (1.5).
91
92 switch ( 3 )
93
94     case 0 % Original Value
95
96         L_o = 0.00001; % Estimate
97         L_e = 0.004 + 2*L_o;
98
99         fprintf( 1, '\nZero - Percentage change in pipe thickness: %3.1f%%\n\n', ( L_e - 0.004
100 ) / 0.004 * 100 );
101
102     case 1 % Ingard (2010)
103
104         a = 0.006 / 2;
105
106         L_o = ( 0.6*a + 0.85*a ) / 2;
107         L_e = 0.004 + 2*L_o;
108
109         fprintf( 1, '\nIngard (2010) - Percentage change in pipe thickness: %3.1f%%\n\n', ( L_e
110 - 0.004 ) / 0.004 * 100 );
111
112     case 2 % Kurze and Riedel (2013)
113
114         e = epsilon^2;
115
116         a = 0.006 / 2;
117
118         L_o = pi*a*( 1 - 1.47*e^0.5 + 0.47*e^1.5 );
119         L_e = 0.004 + 2*L_o;
120
121         fprintf( 1, '\nKurze and Riedel (2013) - Percentage change in pipe thickness: %3.1f%%\n
122 \n', ( L_e - 0.004 ) / 0.004 * 100 );
123
124     case 3 % Ji (2005)
125
126         a = 0.006 / 2;
127
128         L_o = a*( 0.9326 - 0.6196*epsilon );
129         L_e = 0.004 + 2*L_o;
130
131         fprintf( 1, '\nJi (2005) - Percentage change in pipe thickness: %3.1f%%\n\n', ( L_e -
132 0.004 ) / 0.004 * 100 );
133
134     otherwise
135         error ( '*** Invalid SWITCH Index ***' );
136
137 end
138
139 duct_lengths = 0.235/4 * ones( 4, 1 ); % 523 Hz, all holes covered.
140
141 frequency_set = 0:1:2e3;
142
143 nFreq = length( frequency_set );
144 A = zeros( nFreq, 1 );
145
146 for frequency_index = 1:1:nFreq
147     f = frequency_set( frequency_index );

```

```

148 T_total = [ 1 0; 0 1 ];
149
150
151 Z_A = 1j * rho0 * (2 * pi * f) * L_e / ( pi*0.006^2/4 );
152 %
153 term_1 = h_RA_term_1( rho0, c, pi*(0.006)^2/4, 2*pi*f/c, sqrt( (2 * 3.178e-5) / ( 2*pi*f *
rho0 ) ), pi * 0.006, 2*pi*f );
154 term_2 = h_RA_term_2( rho0, c, pi*(0.006)^2/4, 2*pi*f/c, sqrt( (2 * 3.178e-5) / ( 2*pi*f *
rho0 ) ), pi * 0.006, 2*pi*f, 0.3 );
155 term_3 = h_RA_term_3( rho0, c, pi*(0.006)^2/4, 2*pi*f/c, sqrt( (2 * 3.178e-5) / ( 2*pi*f *
rho0 ) ), pi * 0.006, 2*pi*f, 0.3 );
156 R_A = term_1 + term_2 + term_3;
157 %
158 Z_A = Z_A + R_A;
159 T_Hole = [ 1 0; 1/Z_A 1 ];
160
161
162
163 switch ( 0 )
164
165     case 0 % All holes covered. 523 Hz
166         T1 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct - Outlet
167         T2 = [ 1 0; 0 1 ]; % Hole
168         T3 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
169         T4 = [ 1 0; 0 1 ]; % Hole
170         T5 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
171         T6 = [ 1 0; 0 1 ]; % Hole
172         T7 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
173
174         % Check
175         % 4 * 0.05875 = 0.235;
176
177     case 1 % Hole 1 uncovered. 698 Hz
178         T1 = duct_segment_transfer_matrix( f, rho0, c, 0.08775, pipe_area ); % Duct - Outlet
179         T2 = T_Hole;
180         T3 = duct_segment_transfer_matrix( f, rho0, c, 0.02975, pipe_area ); % Duct
181         T4 = [ 1 0; 0 1 ]; % Hole
182         T5 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
183         T6 = [ 1 0; 0 1 ]; % Hole
184         T7 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
185
186         % Check
187         % 0.08775 + 0.02975 + 2*0.05875 = 0.235;
188
189     case 2 % Hole 2 uncovered. 880 Hz
190
191         offset = 0.00625; % 874
192
193         T1 = duct_segment_transfer_matrix( f, rho0, c, 0.08775, pipe_area ); % Duct - Outlet
194         T2 = T_Hole;
195         T3 = duct_segment_transfer_matrix( f, rho0, c, 0.0505, pipe_area ); % Duct
196         T4 = T_Hole;
197         T5 = duct_segment_transfer_matrix( f, rho0, c, 0.038, pipe_area ); % Duct
198         T6 = [ 1 0; 0 1 ]; % Hole
199         T7 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
200
201         % Check
202         % 0.08775 + 0.0505 + 0.038 + 0.05875 = 0.235
203
204     case 3 % Hole 3 uncovered. 1,046 Hz
205
206         T1 = duct_segment_transfer_matrix( f, rho0, c, 0.08775, pipe_area ); % Duct - Outlet
207         T2 = T_Hole;
208         T3 = duct_segment_transfer_matrix( f, rho0, c, 0.0505, pipe_area ); % Duct
209         T4 = T_Hole;
210         T5 = duct_segment_transfer_matrix( f, rho0, c, 0.02975, pipe_area ); % Duct
211         T6 = T_Hole;
212         T7 = duct_segment_transfer_matrix( f, rho0, c, 0.067, pipe_area ); % Duct
213
214         % Check
215         % 0.08775 + 0.0505 + 0.02975 + 0.067 = 0.235 m
216
217 end
218
219 T8 = duct_segment_transfer_matrix( f, rho0, c, 0.09, pipe_area ); % Duct - Inlet
220
221 T_total = T8 * T7 * T6 * T5 * T4 * T3 * T2 * T1 * T_total;
222

```

```

223     Z = open_end_impedance( f, rho0, c, 0, pipe_area, flanged );
224
225     T11 = T_total(1, 1); T12 = T_total(1, 2);
226     A( frequency_index ) = -10*log10( abs( T11 + T12 / Z )^2 );
227
228 end
229
230
231 % figure( ); ...
232 %     plot( frequency_set, A ); grid on;
233 %     xlabel( 'Frequency [Hz]' ); ylabel( 'Amplitude [dB]' );
234
235
236
237 %% Plot Note Set
238
239 load( 'A_C5_Data.mat' ); % Variable(s): A_C5
240 load( 'A_F5_Data.mat' ); % Variable(s): A_F5
241 load( 'A_A5_Data.mat' ); % Variable(s): A_A5
242 load( 'A_C6_Data.mat' ); % Variable(s): A_C6
243
244
245 h_figure_1 = figure( ); ...
246     plot( frequency_set, A_C5 ); hold on;
247     text( 523, 25, 'C5' );
248     plot( frequency_set, A_F5 );
249     text( 698, -1, 'F5' );
250     plot( frequency_set, A_A5 );
251     text( 880, -14, 'A5' );
252     plot( frequency_set, A_C6 ); grid on;
253     text( 1042, -23, 'C6' );
254     %
255     legend( 'C5, 523 Hz', 'F5, 698 Hz', 'A5, 880 Hz', 'C6, 1046 Hz', 'Location', 'NorthWest'
256 );
257     xlabel( 'Frequency [Hz]' ); ylabel( 'Amplitude [dB]' );
258     title( 'Notes Spectrums for a Bugle Recorder' );
259
260
261 %% Clean-up
262
263 if ( ~isempty( findobj( 'Type', 'figure' ) ) )
264     monitors = get( 0, 'MonitorPositions' );
265     if ( size( monitors, 1 ) == 1 )
266         autoArrangeFigures( 2, 2, 1 );
267     elseif ( 1 < size( monitors, 1 ) )
268         autoArrangeFigures( 2, 2, 1 );
269     end
270 end
271
272 if ( PRINT_FIGURES == 1 )
273     exportgraphics( h_figure_1, 'Assignment 1 - Question 3 Bugle Recorder Note Spectrums.pdf'
274 , 'Append', true );
275 end
276 fprintf( 1, '\n\n\n*** Processing Complete ***\n\n\n' );
277
278
279
280 %% Reference(s)

```

4 Appendix - Matlab Code for Problem 4

```
1
2
3
4 %% Synopsis
5
6 % Question 4 – Intake Duct
7
8
9
10 %% Environment
11
12 close all; clear; clc;
13 % restoredefaultpath;
14
15 % addpath( genpath( '' ), '-begin' );
16 addpath( genpath( '../40 Assignments/00 Support' ), '-begin' );
17
18 % set( 0, 'DefaultFigurePosition', [ 400 400 900 400 ] ); % [ left bottom width height ]
19 set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
20 set( 0, 'DefaultFigureWindowStyle', 'normal' );
21 set( 0, 'DefaultLineLineWidth', 1.5 );
22 set( 0, 'DefaultTextInterpreter', 'Latex' );
23
24 format ShortG;
25
26 pause( 1 );
27
28 PRINT_FIGURES = 0;
29
30
31
32 %% Constants
33
34 rho0 = 1.21; % Density of air (kg per cubic-meter).
35 c = 343; % Speed of sound in air (meters per second).
36
37 h_area = @( diameter ) pi * diameter^2 / 4;
38
39
40
41 %% Define Shape
42
43 % Source
44 duct_1.diameter_meters = 0.0254; % meters
45 duct_1.area = h_area( duct_1.diameter_meters ); % squared-meters
46 duct_1.length_meters = 0.1524; % meters
47
48
49 % Outlet
50 duct_2.diameter_meters = 0.1016; % 4 inches
51 duct_2.area = h_area( duct_2.diameter_meters );
52 duct_2.length_meters = 0.127; % 5 inches
53
54 % Flanged.
55
56 flow_rate_cubic_meters_per_second = 1.04772 / 60; % or 37 cubic-feet per minute
57
58
59
60 %% Part a
61
62 % Calculate the Mach number of the flow in both pip sections.
63
64 duct_1.Mach = -1.0 * flow_rate_cubic_meters_per_second / ( (pi * duct_1.diameter_meters^2 / 4 ) *
65     c ); % 0.100 unitless
66
67 duct_2.Mach = -1.0 * flow_rate_cubic_meters_per_second / ( (pi * duct_2.diameter_meters^2 / 4 ) *
68     c ); % 0.00628 unitless
69
70
71
72
73 return
74
75 %% Part b
76
77 % No flow.
78
79 outlet_flanged = true; % Flanged end.
```

```

74
75
76 TEST_FLAG = 1; % 1: right-to-left.
77
78 frequency_set = 0:0.1:2.5e3;
79 nFreq = length( frequency_set );
80 TL = zeros( nFreq, 1 );
81
82
83 for frequency_index = 1:1:nFreq
84
85     f = frequency_set( frequency_index );
86
87
88     T_total = [ 1 0; 0 1 ];
89
90
91     if ( TEST_FLAG == 1 )
92
93         % Right-to-left.
94         %
95         T_outlet = duct_segment_transfer_matrix( f, rho0, c, duct_2.length_meters, duct_2.area );
96         T_contraction = [ 1 0; 0 duct_2.area/duct_1.area ];
97         T_inlet = duct_segment_transfer_matrix( f, rho0, c, duct_1.length_meters, duct_1.area );
98
99         Z = open_end_impedance( f, rho0, c, duct_2.length_meters, duct_2.area, outlet_flanged );
100
101     else
102
103         % Left-to-right.
104         %
105         T_outlet = duct_segment_transfer_matrix( f, rho0, c, duct_1.length_meters, duct_1.area );
106         T_contraction = [ 1 0; 0 duct_1.area/duct_2.area ];
107         T_inlet = duct_segment_transfer_matrix( f, rho0, c, duct_2.length_meters, duct_2.area );
108
109         Z = open_end_impedance( f, rho0, c, duct_1.length_meters, duct_1.area, outlet_flanged );
110
111     end
112
113
114     T_net = T_inlet * T_contraction * T_outlet * T_total;
115
116
117     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
118
119     if ( TEST_FLAG == 1 )
120         TL( frequency_index ) = 10 * log10( abs( ( T11 + duct_1.area*T12/(rho0*c) + (rho0*c)*
121         T21/duct_2.area + T22 ) / 2 )^2 );
122     else
123         TL( frequency_index ) = 10 * log10( abs( ( T11 + duct_2.area*T12/(rho0*c) + (rho0*c)*
124         T21/duct_1.area + T22 ) / 2 )^2 );
125     end
126
127 end % End: for f = frequency_set
128
129 TL_part_b = TL;
130
131 % return
132
133 %% Part c
134
135 % Flow present (use Mach numbers).
136
137 outlet_flanged = true; % Flanged end.
138
139 frequency_set = 0:0.1:2.5e3;
140 nFreq = length( frequency_set );
141 TL = zeros( nFreq, 1 );
142
143
144 for frequency_index = 1:1:nFreq
145
146     f = frequency_set( frequency_index );
147
148
149     T_total = [ 1 0; 0 1 ];

```

```

150
151     T_outlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_2.length_meters, duct_2.area,
duct_2.Mach );
152
153     T_expansion = duct_expansion_connection_transfer_matrix( rho0, c, duct_2.area, duct_1.area,
duct_1.Mach );
154
155     T_inlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_1.length_meters, duct_1.area,
duct_1.Mach );
156
157     T_net = T_inlet * T_expansion * T_outlet * T_total;
158
159
160     Z = open_end_impedance( f, rho0, c, duct_2.length_meters, duct_2.area, outlet_flanged );
161
162     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
163     TL( frequency_index ) = 10 * log10( abs( ( T11 + duct_1.area*T12/(rho0*c) + (rho0*c)*
T21/duct_2.area + T22 ) / 2 )^2 );
164
165
166 end % End: for f = frequency_set
167
168 TL_part_c = TL;
169
170 % return
171
172 %% Part d
173
174 % Flow present (use Mach numbers).
175 % Helmholtz resonator in place (between lefthand duct and expansion).
176
177 outlet_flanged = true; % Flanged end.
178
179
180 % Resonance
181 w_o = 2*pi*136.6; % Estimated from plot.
182
183
184 % helmholtz_diameter_cavity =
185 % helmholtz_diameter_neck = 1e-3;
186 %     helmholtz_L01 = 0.82 * ( 1 - 1.33*(helmholtz_diameter_neck/helmholtz_diameter_cavity ) );
187 %     %
188 %     epsilon = helmholtz_diameter_cavity / duct_1.length_meters;
189 %     helmholtz_L02 =
190 %
191 % helmholtz_volume = 1e-3;
192 %
193 % % keyboard
194 %
195 % helmholtz_L_neck = 1e-3;
196 % Q = 2;
197 %
198 % R_A = rho0*c / Q * sqrt( L_e / ( pi*helmholtz_diameter_neck^2/4 * helmholtz_volume ) );
199
200
201
202 frequency_set = 0:0.1:2.5e3;
203 nFreq = length( frequency_set );
204 TL = zeros( nFreq, 1 );
205
206
207 for frequency_index = 1:1:nFreq
208
209     f = frequency_set( frequency_index );
210
211
212     T_total = [ 1 0; 0 1 ];
213
214     T_outlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_2.length_meters, duct_2.area,
duct_2.Mach );
215
216     T_expansion = duct_expansion_connection_transfer_matrix( rho0, c, duct_2.area, duct_1.area,
duct_1.Mach );
217
218
219     % Z_A = 1j*rho0*2*pi*f(frequency_index) * L_e / ( pi*helmholtz_neck_diameter^2/4 ) - 1j*
rho0*c^2/(helmholtz_volume*2*pi*f(frequency_index)) + R_A;
220     % T_Helmholtz = [ 1 0; 1/Z_A 1 ];

```

```

221
222
223     T_inlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_1.length_meters, duct_1.area,
duct_1.Mach );
224
225
226     % T_net = T_inlet * T_Helmholtz * T_expansion * T_outlet * T_total;
227     T_net = T_inlet * T_expansion * T_outlet * T_total;
228
229
230     Z = open_end_impedance( f, rho0, c, duct_2.length_meters, duct_2.area, outlet_flanged );
231
232     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
233     TL( frequency_index ) = 10 * log10( abs( ( T11 + duct_1.area*T12/(rho0*c) + (rho0*c)*
T21/duct_2.area + T22 ) / 2 )^2 );
234
235
236 end % End: for f = frequency_set
237
238 TL_part_d = TL;
239
240 % return
241
242 %% Plot
243
244 Y_LIMITS = [ 0 50 ];
245
246 figure( ); ...
247     plot( frequency_set, TL_part_b ); hold on;
248     plot( frequency_set, TL_part_c, '-.' );
249     plot( frequency_set, TL_part_d, '-.' ); grid on;
250     legend( ...
251         'No Flow - Part b', ...
252         'Flow - Part c', ...
253         'Flow and Resonator - Part d', ...
254         'Location', 'SouthOutside' );
255     xlabel( 'Frequency [Hz]' ); ylabel( 'Amplitude [dB]' );
256     title( 'Transmission Loss Profiles' );
257     %
258     Ax = gca;
259     Ax.XAxis.TickLabelInterpreter = 'latex';
260     Ax.YAxis.TickLabelInterpreter = 'latex';
261     %
262     % axis( [ -50 5e3+50 Y_LIMITS ] );
263     %
264     if ( PRINT_FIGURES == 1 )
265         exportgraphics((gcf, 'Figure TL All Profiles.pdf', 'Append', true );
266     end
267
268 return
269
270 %% Part c
271
272 return
273
274
275 %% Part d
276
277 return
278
279 %% Clean-up
280
281 if ( ~isempty( findobj( 'Type', 'figure' ) ) )
282     monitors = get( 0, 'MonitorPositions' );
283     if ( size( monitors, 1 ) == 1 )
284         autoArrangeFigures( 2, 2, 1 );
285     elseif ( 1 < size( monitors, 1 ) )
286         autoArrangeFigures( 2, 2, 1 );
287     end
288 end
289
290
291 fprintf( 1, '\n\n*** Processing Complete ***\n\n' );
292
293
294
295 %% Reference(s)

```

5 Appendix - Matlab Code for Problem 5

```
1
2
3 %% Synopsis
4
5 % Question 4 – Intake Duct
6
7
8
9 %% To Do
10
11 % Check loss profile.
12
13
14
15 %% Note(s)
16
17 % Search for FIXMEs.
18
19
20 % In class note, the areas for the impedance might have been wrong; switch them?
21
22
23 % Use negative Mach numbers in the equations. The analysis for this case
24 % is the same as for the horn example. Inlet on the left, outlet on the
25 % right.
26
27
28 % The S in the diagram (pink area) for the Helmholtz resonator is the
29 % cross-sectional area of the resonator neck connecting it to the tube.
30
31
32 % For Lo2, use the value for a quarter-wavelength side tube.
33
34
35 % The area expansion ratio is determined using the original duct diameters.
36
37
38
39 %% Environment
40
41 close all; clear; clc;
42 % restoredefaultpath;
43
44 % addpath( genpath( '' ), '-begin' );
45 addpath( genpath( '../40 Assignments/00 Support' ), '-begin' );
46
47 % set( 0, 'DefaultFigurePosition', [ 400 400 900 400 ] ); % [ left bottom width height ]
48 set( 0, 'DefaultFigurePaperPositionMode', 'manual' );
49 set( 0, 'DefaultFigureWindowStyle', 'normal' );
50 set( 0, 'DefaultLineLineWidth', 1.5 );
51 set( 0, 'DefaultTextInterpreter', 'Latex' );
52
53 format ShortG;
54
55 pause( 1 );
56
57 PRINT_FIGURES = 0;
58
59
60
61 %% Define Constants and Anonymous Functions
62
63 rho0 = 1.21; % Density of air (kg per cubic-meter).
64 c = 343; % Speed of sound in air (meters per second).
65
66
67 h_area = @( diameter ) pi * diameter^2 / 4;
68
69
70
71 %% Define Shape
72
73 % Source
74 duct_1.diameter_meters = 0.0254; % 1 inch
75 duct_1.length_meters = 0.1524; % 6 inches
```



```

76 duct_1.area = h_area( duct_1.diameter_meters );
77
78 % Outlet
79 duct_2.diameter_meters = 0.1016; % 4 inches
80 duct_2.length_meters = 0.127; % 5 inches
81 duct_2.area = h_area( duct_2.diameter_meters );
82 %
83 % Flanged.
84
85 %
86
87 flow_rate_cubic_meters_per_second = 1.04772 / 60; % or 37 cubic-feet per minute
88
89 % return
90
91 %% Part a
92
93 % The -35 dB transmission loss dip|notch is at about 1,150 Hz.
94 f = 1150;
95
96 % Total attenuation required is +35 dB to make it 0 dB.
97
98 l = 0.0381; % meters (1.5 inches)
99 h = 0.0127; % meters (0.5 inches)
100
101 % The total attenuation of the lining from Figure 8.37 (Bies et. al., Fifth Edition),
102 %
103 %  $m = (h\_area(0.1016) - h\_area(0.0254)) / h\_area(0.0254)$ 
104 %
105 % ASSUMPTION: Cross-section area of lined duct is the same as the inlet.
106 m = 1;
107
108 k = 2*pi*1.15e3;
109
110 length_of_expansion_chamber = 0.127; % meters
111
112 %  $kL = k*length\_of\_expansion\_chamber$ 
113
114 % ASSUMPTION:
115 %
116 % Expansion ratio is  $m = 1$ ;
117 % Assume peak difference is 10 dB.
118 % Total attenuation of lining is 10 dB.
119 %
120 % The attenuation rate is about 10 dB / 0.127 meters or 78.7 dB per meter.
121
122 % return
123
124 %% Part b
125
126 l = 0.0381; % meters (1.5 inches)
127
128 h = 0.011255; % meters
129 h_validate = 0.5*sqrt( ( pi*( duct_2.diameter_meters - 2*l )^2 ) / 4 ); % Same value.
130
131
132
133 %% Part c
134
135 % The liner thickness ratio is ,
136 l / h; % 3.3852 unitless
137
138 % The normalized frequency is ,
139 ( 2 * h ) / ( 343 / f ); % 0.075471 unitless
140
141
142
143 %% Part d
144
145 % Assume the attenuation rate from Part a is 18 dB per meter.
146
147 % Use bottom, right subplot (16).
148
149 % The approximate resistivity parameter is 16.
150
151
152
153 %% Part e

```

```

154
155 % Calculate the flow resistivity.
156
157 R1 = 16 * rho0*c / l; % 1.74e5 kg / m^3*s
158
159
160
161 %% Placeholder
162
163 % % Flow present (use Mach numbers).
164 % % Helmholtz resonator in place (between lefthand duct and expansion).
165 %
166 % outlet_flanged = true; % Flanged end.
167 %
168 %
169 % % Resonance
170 % w_o = 2*pi*136.6; % Estimated from plot.
171 %
172 %
173 % % helmholtz_diameter_cavity =
174 % % helmholtz_diameter_neck = 1e-3;
175 % % helmholtz_L01 = 0.82 * ( 1 - 1.33*(helmholtz_diameter_neck/helmholtz_diameter_cavity ) );
176 % %
177 % % epsilon = helmholtz_diameter_cavity / duct_1.length_meters;
178 % % helmholtz_L02 =
179 % %
180 % % helmholtz_volume = 1e-3;
181 % %
182 % % % keyboard
183 % %
184 % % helmholtz_L_neck = 1e-3;
185 % % Q = 2;
186 % %
187 % % R_A = rho0*c / Q * sqrt( L_e / ( pi*helmholtz_diameter_neck^2/4 * helmholtz_volume ) );
188 %
189 %
190 %
191 % frequency_set = 0:0.1:2.5e3;
192 % nFreq = length( frequency_set );
193 % TL = zeros( nFreq, 1 );
194 %
195 %
196 % for frequency_index = 1:1:nFreq
197 %
198 % f = frequency_set( frequency_index );
199 %
200 %
201 % T_total = [ 1 0; 0 1 ];
202 %
203 % T_outlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_2.length_meters, duct_2.area
, duct_2.Mach );
204 %
205 % T_expansion = duct_expansion_connection_transfer_matrix( rho0, c, duct_2.area, duct_1.area,
duct_1.Mach );
206 %
207 %
208 % Z_A = 1j*rho0*2*pi*f(frequency_index) * L_e / ( pi*helmholtz_neck_diameter^2/4 ) - 1j*
rho0*c^2/(helmholtz_volume*2*pi*f(frequency_index)) + R_A;
209 % % T_Helmholtz = [ 1 0; 1/Z_A 1 ];
210 %
211 %
212 % T_inlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_1.length_meters, duct_1.area,
duct_1.Mach );
213 %
214 %
215 % T_net = T_inlet * T_Helmholtz * T_expansion * T_outlet * T_total;
216 % T_net = T_inlet * T_expansion * T_outlet * T_total;
217 %
218 %
219 % Z = open_end_impedance( f, rho0, c, duct_2.length_meters, duct_2.area, outlet_flanged );
220 %
221 % T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
222 % TL( frequency_index ) = 10 * log10( abs( ( T11 + duct_1.area*T12/(rho0*c) + (rho0*c
)*T21/duct_2.area + T22 ) / 2 )^2 );
223 %
224 %
225 % end % End: for f = frequency_set
226 %

```

```

227 % TL_part_d = TL;
228
229 % return
230
231 %% Plot
232
233 % Y_LIMITS = [ 0 50 ];
234 %
235 % figure( ); ...
236 %     plot( frequency_set, TL_part_d ); hold on;
237 %     % plot( frequency_set, TL_part_c, '-' );
238 %     % plot( frequency_set, TL_part_d, '-.' ); grid on;
239 %     legend( ...
240 %         'No Flow - Part b', ...
241 %         'Flow - Part c', ...
242 %         'Flow and Resonator - Part d', ...
243 %         'Location', 'SouthOutside' );
244 %     xlabel( 'Frequency [Hz]' ); ylabel( 'Transmission Loss [dB]' );
245 %     title( 'Transmission Loss Profiles' );
246 %
247 %     Ax = gca;
248 %     Ax.XAxis.TickLabelInterpreter = 'latex';
249 %     Ax.YAxis.TickLabelInterpreter = 'latex';
250 %
251 %     axis( [ -50 5e3+50 Y_LIMITS ] );
252 %
253 %     if ( PRINT_FIGURES == 1 )
254 %         exportgraphics((gcf, 'Figure TL All Profiles.pdf', 'Append', true );
255 %     end
256
257 % return
258
259 %% Clean-up
260
261 if ( ~isempty( findobj( 'Type', 'figure' ) ) )
262     monitors = get( 0, 'MonitorPositions' );
263     if ( size( monitors, 1 ) == 1 )
264         autoArrangeFigures( 2, 2, 1 );
265     elseif ( 1 < size( monitors, 1 ) )
266         autoArrangeFigures( 2, 2, 1 );
267     end
268 end
269
270
271 fprintf( 1, '\n\n*** Processing Complete ***\n\n' );
272
273
274
275 %% Reference(s)

```