

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,

$$f_{\text{cut-on}} = 0.568 \cdot \frac{1,500 \frac{\text{m}}{\text{s}}}{0.17 \text{ m}} = \mathbf{1,114.5 \text{ Hz}}$$

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.

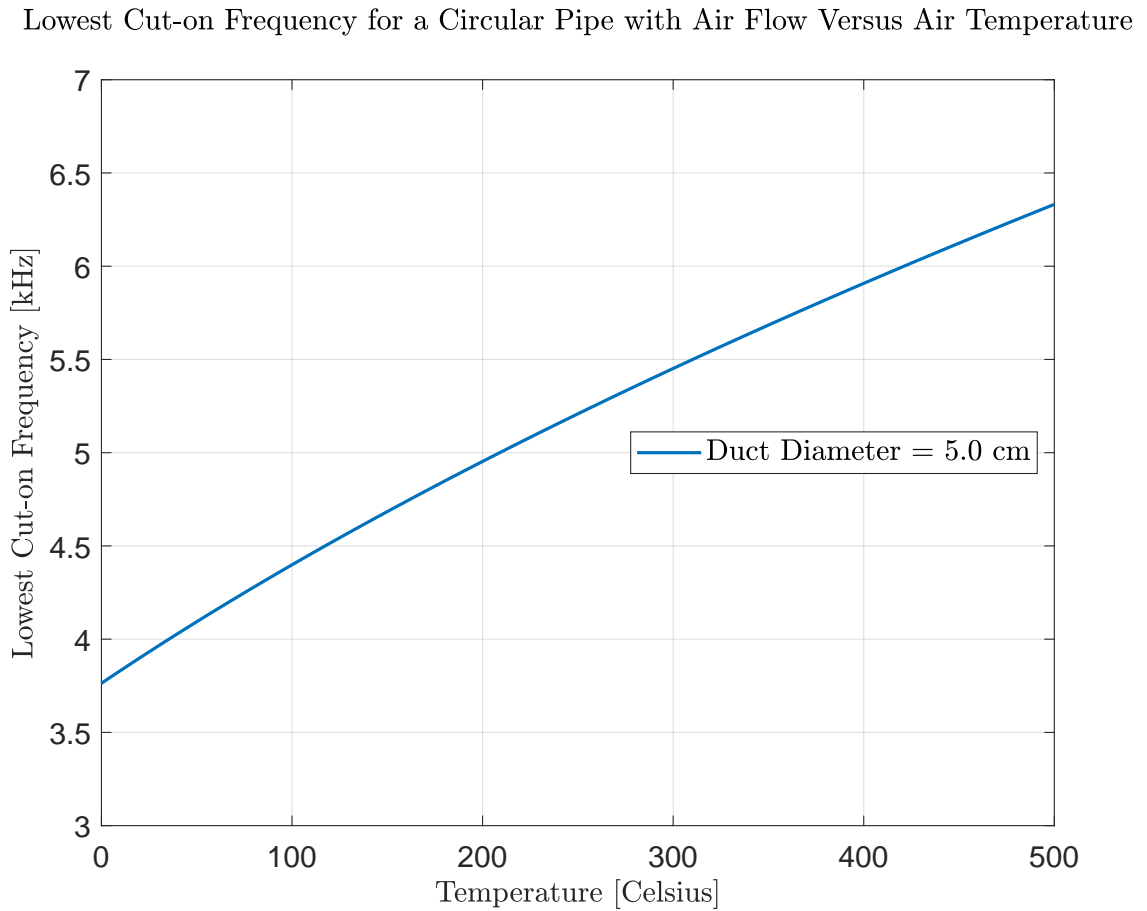


Figure 1: Lowest cut-on frequency for a circular 5 cm diameter duct versus air temperature.

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 it.

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 office hours, there is no damping which produces artificially high resonances.

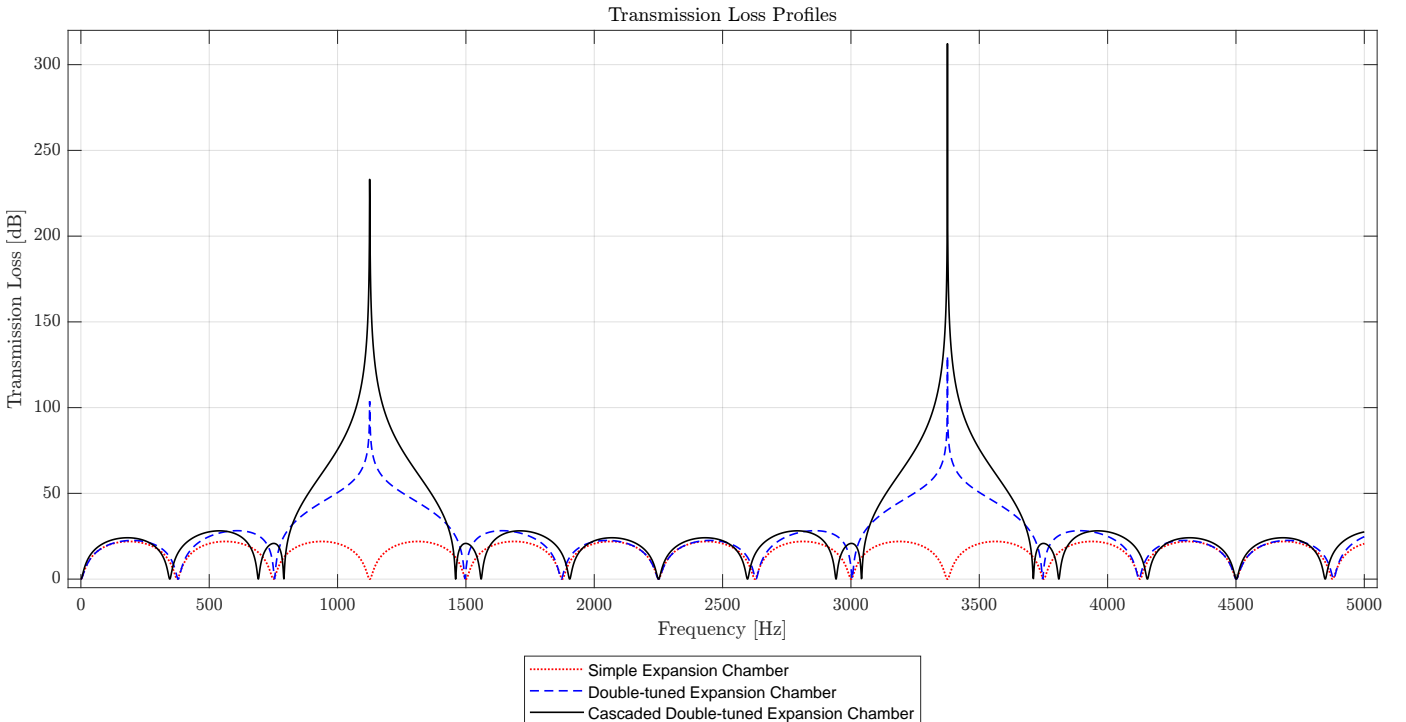


Figure 2: Transmission loss profiles for a simple expansion chamber, a double-tuned expansion chamber, and a cascaded double-tuned expansion chamber muffler.

Problem 2d

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

Two modifications were made to the original muffler:

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.

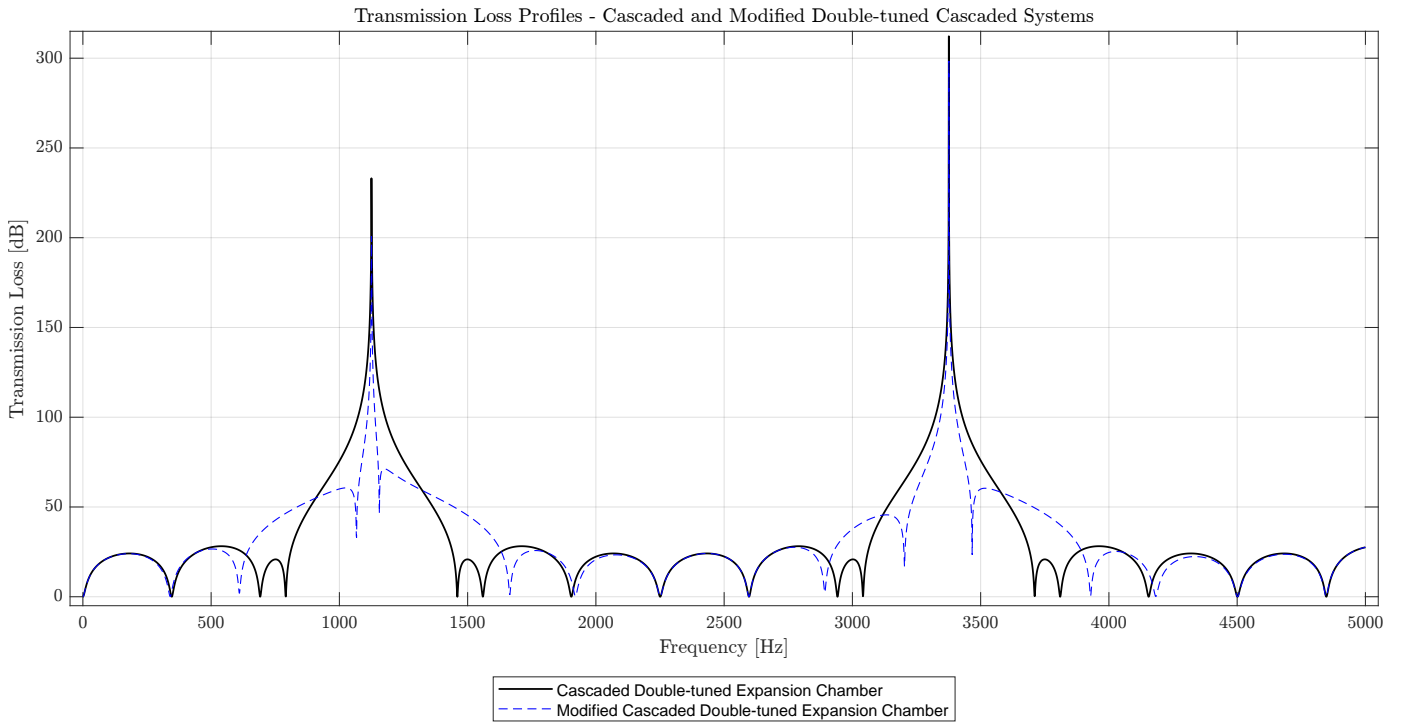


Figure 3: Transmission loss profiles for a cascaded double-tuned expansion chamber muffler and a modified version of this muffler.

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

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.

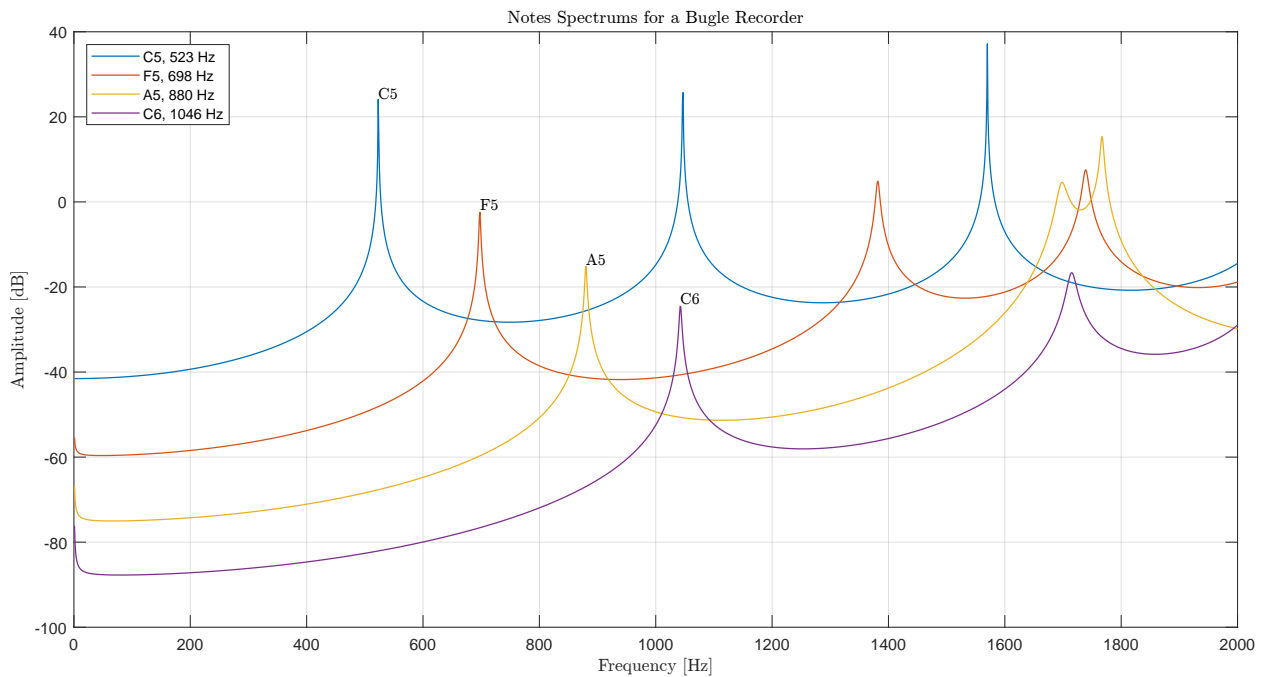


Figure 4: Spectrum for the C5, F5, A5, and C6 bugle recorder notes.

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. The flow rate is $0.017462 \frac{\text{m}^3}{\text{s}}$.

Pipe	Area [m ²]	Mach Number [unitless]
Inlet	0.000507	-0.10047
Outlet	0.00811	-0.0062795

Table 3: Calculated Mach numbers.

Problems 4b, 4c, and 4d

Figure 5 shows the transmission loss profiles.

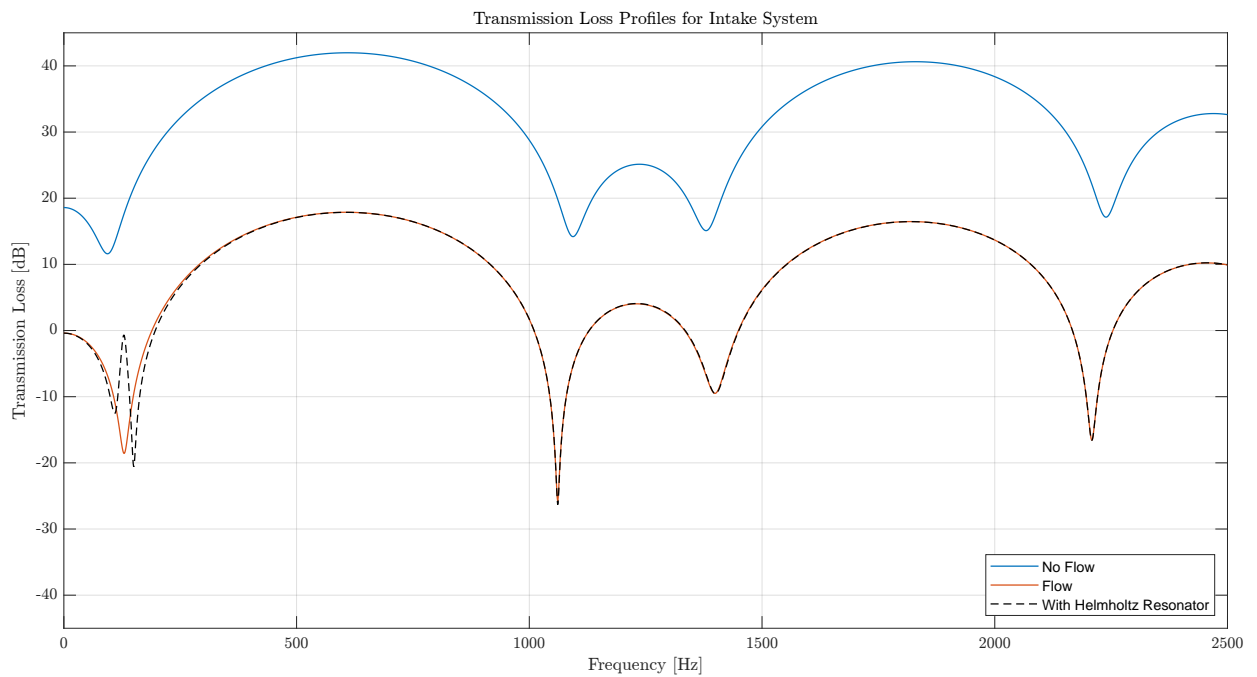


Figure 5: Transmission loss profiles for no flow, flow, and flow with a lossy Helmholtz resonator.

The addition of flow to the intake system introduces a slight phase delay, a lower overall level of loss (approximately 22 dB), and greater loss at the dips. The phase delay is easier to see at respective dips in the loss profile.

The phase delay is introduced by each of the pipe segments by the complex-exponential term in the respective transfer function. The attenuation is introduced by the expansion transfer function.

Table 4 lists the dimensions of the lossy Helmholtz resonator for the 129 Hz notch.

Item	Measure]
Cavity Diameter	0.254 m
Neck Diameter	0.0254 m
Neck Length	0.127 m
Neck Area	$0.5\text{e-}3 \text{ m}^2$
Length Correction 1, L_{o1}	0.711 m
Length Correction 2, L_{o2}	0.3 m
Cavity Volume	$8.0\text{e-}5 \text{ m}^3$
Q Factor	10

Table 4: Dimensions of the lossy Helmholtz resonator (see slide 15 of Lecture 3 notes).

The addition of the Helmholtz resonator offset the attenuation of the dip to approximately 0 dB.

Problem 5 - Intake Duct Silencer

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

The addition of the Helmholtz resonator did not create a notch at 940 Hz in the transmission loss profile, but it did offset the notch at 129 Hz as intended. The notch present in the given range was at 1,060 Hz (as noted in the discussion) and had a loss of -25 dB.

For this problem I used the 940 Hz dip requiring a 14 dB correction.

Problem 5a

As noted in the discussion, there are two cases to be considered here.

First, with a liner thickness of 0.0381 meters and a half-liner width (circular duct) of 0.0113 meters, the expansion ratio, m , is 2.7. From Figure 8.37, a 14 dB overall loss has a total liner attenuation of 10 dB. The attenuation rate across the 0.127 m long outlet is about $78.4 \frac{\text{dB}}{\text{m}}$.

Secondly, Figure 8.37 considers the combined effect of the expansion and the liner. In Problem 4, the effect of the expansion was calculated, so including it here would include its effect twice. Therefore, with an m of 1, there is no additional attenuation with the linear.

Problem 5b

l , the liner thickness, is 0.0381 m. h , the liner half width, is 0.0113 m.

Problem 5c

The liner thickness ratio is 3.385 [unitless]. The normalized frequency is 0.0617 [unitless] for 940 Hz.

Problem 5d

With a liner thickness ratio of 3.385, curve 5 is selected from the attenuation rate curve figure set. The $\frac{2 \cdot h}{\lambda}$ value is 0.06169. The target attenuation rate for the attenuation rate curves is $39.2 \frac{\text{dB}}{\text{m}}$ (half of the value from Problem 5a) multiplied by $h = 0.0113$ which gives 0.44.

From this information, the resistivity parameter plot selected is the bottom right plot with a value of 16.

Problem 5e

Using a resistivity parameter of 16, the respective flow resistivity, R_1 , is about $1.74\text{e}5 \frac{\text{kg}}{\text{m}^2 \cdot \text{s}}$ or rayl.

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', 0.8 );
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', 0.8 );
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 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 convert.inches_to_meters = 0.0254;
41 convert.foot_to_meters = 0.3048;
42
43 dimensions.inlet_diameter_meters = 2 * convert.inches_to_meters; % 0.0508 meters
44 dimensions.inlet_length_meters = 6 * convert.foot_to_meters; % 1.82 meters
45
46 dimensions.muffler_diameter_meters = 10 * convert.inches_to_meters; % 0.254 meters
47 dimensions.muffler_length_meters = 18 * convert.inches_to_meters; % 0.4572 meters
48
49 dimensions.outlet_diameter_meters = 2 * convert.inches_to_meters; % 0.0508 meters
50 dimensions.outlet_length_meters = 1 * convert.foot_to_meters; % 0.3048 meters
51
52 outlet_flanged = false;
53
54 dimensions.overhang = 3 * convert.inches_to_meters; % 0.0762 meters
55
56 segment_diameters = [ ...
57     dimensions.outlet_diameter_meters, ...
58     dimensions.muffler_diameter_meters, ...
59     dimensions.inlet_diameter_meters, ...
60     ].';
61 %
62 h_area_from_diameter = @( d ) pi .* d.^2 ./ 4;
63 %
64 segment_areas = h_area_from_diameter( segment_diameters );
65
66 segment_lengths = [ ...
67     dimensions.outlet_length_meters, ...
68     dimensions.muffler_length_meters, ...
69     dimensions.inlet_length_meters, ...
70     dimensions.overhang, ...
71     ].';
72
73
74
75 %% Problem 2a
```

```

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

```

```

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

```

```

225
226     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
227     TL( frequency_index ) = 10 * log10( abs( ( T11 + 0.0020268*T12/(rho0*c) + (rho0*c)*
228     T21/0.0020268 + T22 ) / 2 )^2 );
229 end
230
231 TL_partd = TL;
232
233
234
235 %% Plot Transmission Loss Profiles
236
237 Y_LIMITS = [ -5 320 ];
238
239 h_figure_1 = figure( ); ...
240 plot( frequency_set, TL_parta, 'LineWidth', 1.0, 'LineStyle', ':', 'Color', 'r' ); hold on;
241 plot( frequency_set, TL_partb, 'LineWidth', 0.9, 'LineStyle', '—', 'Color', 'b' );
242 plot( frequency_set, TL_partc, 'LineWidth', 0.9, 'LineStyle', '—', 'Color', 'k' ); grid on;
243 legend( ...
244     'Simple Expansion Chamber', ...
245     'Double-tuned Expansion Chamber', ...
246     'Cascaded Double-tuned Expansion Chamber', ...
247     'Location', 'SouthOutside' );
248 xlabel( 'Frequency [Hz]' ); ylabel( 'Transmission Loss [dB]' );
249 title( 'Transmission Loss Profiles' );
250 %
251 Ax = gca;
252 Ax.XAxis.TickLabelInterpreter = 'latex';
253 Ax.YAxis.TickLabelInterpreter = 'latex';
254 %
255 axis( [ -50 5e3+50 Y_LIMITS ] );
256
257
258 Y_LIMITS = [ -5 315 ];
259
260 h_figure_2 = figure( ); ...
261 plot( frequency_set, TL_partc, 'LineWidth', 1.0, 'LineStyle', '—', 'Color', 'k' ); hold on;
262 plot( frequency_set, TL_partd, 'LineWidth', 0.6, 'LineStyle', '—', 'Color', 'b' ); grid on;
263 legend( ...
264     'Cascaded Double-tuned Expansion Chamber', ...
265     'Modified Cascaded Double-tuned Expansion Chamber', ...
266     'Location', 'SouthOutside' );
267 xlabel( 'Frequency [Hz]' ); ylabel( 'Transmission Loss [dB]' );
268 title( 'Transmission Loss Profiles – Cascaded and Modified Double-tuned Cascaded Systems' );
269 %
270 Ax = gca;
271 Ax.XAxis.TickLabelInterpreter = 'latex';
272 Ax.YAxis.TickLabelInterpreter = 'latex';
273 %
274 axis( [ -50 5e3+50 Y_LIMITS ] );
275
276
277
278 %% Clean-up
279
280 if ( ~isempty( findobj( 'Type', 'figure' ) ) )
281     monitors = get( 0, 'MonitorPositions' );
282     if ( size( monitors, 1 ) == 1 )
283         autoArrangeFigures( 2, 2, 1 );
284     elseif ( 1 < size( monitors, 1 ) )
285         autoArrangeFigures( 2, 2, 1 );
286     end
287 end
288
289 if ( PRINT_FIGURES == 1 )
290     exportgraphics( h_figure_1, 'Assignment 1 – Question 2 Figure All TL Profiles.pdf', '
Append', true );
291     exportgraphics( h_figure_2, 'Assignment 1 – Question 2 Figure Comparison TL Plot For
Cascaded Systems.pdf', 'Append', true );
292 end
293
294 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 %% Define 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 %% Problem 3a
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 %% Problem 3b
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 switch ( 0 )
163
164     case 0 % All holes covered. 523 Hz
165         T1 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct - Outlet
166         T2 = [ 1 0; 0 1 ]; % Hole
167         T3 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
168         T4 = [ 1 0; 0 1 ]; % Hole
169         T5 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
170         T6 = [ 1 0; 0 1 ]; % Hole
171         T7 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
172
173         % Check
174         % 4 * 0.05875 = 0.235;
175
176     case 1 % Hole 1 uncovered. 698 Hz
177         T1 = duct_segment_transfer_matrix( f, rho0, c, 0.08775, pipe_area ); % Duct - Outlet
178         T2 = T_Hole;
179         T3 = duct_segment_transfer_matrix( f, rho0, c, 0.02975, pipe_area ); % Duct
180         T4 = [ 1 0; 0 1 ]; % Hole
181         T5 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
182         T6 = [ 1 0; 0 1 ]; % Hole
183         T7 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
184
185         % Check
186         % 0.08775 + 0.02975 + 2*0.05875 = 0.235;
187
188     case 2 % Hole 2 uncovered. 880 Hz
189
190         offset = 0.00625; % 874
191
192         T1 = duct_segment_transfer_matrix( f, rho0, c, 0.08775, pipe_area ); % Duct - Outlet
193         T2 = T_Hole;
194         T3 = duct_segment_transfer_matrix( f, rho0, c, 0.0505, pipe_area ); % Duct
195         T4 = T_Hole;
196         T5 = duct_segment_transfer_matrix( f, rho0, c, 0.038, pipe_area ); % Duct
197         T6 = [ 1 0; 0 1 ]; % Hole
198         T7 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area ); % Duct
199
200         % Check
201         % 0.08775 + 0.0505 + 0.038 + 0.05875 = 0.235
202
203     case 3 % Hole 3 uncovered. 1,046 Hz
204
205         T1 = duct_segment_transfer_matrix( f, rho0, c, 0.08775, pipe_area ); % Duct - Outlet
206         T2 = T_Hole;
207         T3 = duct_segment_transfer_matrix( f, rho0, c, 0.0505, pipe_area ); % Duct
208         T4 = T_Hole;
209         T5 = duct_segment_transfer_matrix( f, rho0, c, 0.02975, pipe_area ); % Duct
210         T6 = T_Hole;
211         T7 = duct_segment_transfer_matrix( f, rho0, c, 0.067, pipe_area ); % Duct
212
213         % Check
214         % 0.08775 + 0.0505 + 0.02975 + 0.067 = 0.235 m
215
216 end
217
218 T8 = duct_segment_transfer_matrix( f, rho0, c, 0.09, pipe_area ); % Duct - Inlet
219
220 T_total = T8 * T7 * T6 * T5 * T4 * T3 * T2 * T1 * T_total;
221
222 Z = open_end_impedance( f, rho0, c, 0, pipe_area, flanged );

```

```

223
224     T11 = T_total(1, 1); T12 = T_total(1, 2);
225     A( frequency_index ) = -10*log10( abs( T11 + T12 / Z )^2 );
226
227 end
228
229
230 % figure( ); ...
231 %     plot( frequency_set, A ); grid on;
232 %     xlabel( 'Frequency [Hz]' ); ylabel( 'Amplitude [dB]' );
233
234
235
236 %% Plot Note Set
237
238 load( 'A_C5_Data.mat' ); % Variable(s): A_C5
239 load( 'A_F5_Data.mat' ); % Variable(s): A_F5
240 load( 'A_A5_Data.mat' ); % Variable(s): A_A5
241 load( 'A_C6_Data.mat' ); % Variable(s): A_C6
242
243
244 h_figure_1 = figure( ); ...
245     plot( frequency_set, A_C5 ); hold on;
246     text( 523, 25, 'C5' );
247     plot( frequency_set, A_F5 );
248     text( 698, -1, 'F5' );
249     plot( frequency_set, A_A5 );
250     text( 880, -14, 'A5' );
251     plot( frequency_set, A_C6 ); grid on;
252     text( 1042, -23, 'C6' );
253     %
254     legend( 'C5, 523 Hz', 'F5, 698 Hz', 'A5, 880 Hz', 'C6, 1046 Hz', 'Location', 'NorthWest'
255 );
256 xlabel( 'Frequency [Hz]' ); ylabel( 'Amplitude [dB]' );
257 title( 'Notes Spectrums for a Bugle Recorder' );
258
259
260 %% Clean-up
261
262 if ( ~isempty( findobj( 'Type', 'figure' ) ) )
263     monitors = get( 0, 'MonitorPositions' );
264     if ( size( monitors, 1 ) == 1 )
265         autoArrangeFigures( 2, 2, 1 );
266     elseif ( 1 < size( monitors, 1 ) )
267         autoArrangeFigures( 2, 2, 1 );
268     end
269 end
270
271 if ( PRINT_FIGURES == 1 )
272     exportgraphics( h_figure_1, 'Assignment 1 - Question 3 Bugle Recorder Note Spectrums.pdf'
273 , 'Append', true );
274 end
275
276 fprintf( 1, '\n\n*** Processing Complete ***\n\n' );
277
278
279 %% 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', 0.8 );
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 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 % Source
41 duct_1.diameter_meters = 0.0254; % meters
42 duct_1.area = h_area( duct_1.diameter_meters ); % squared-meters
43 duct_1.length_meters = 0.1524; % meters
44
45
46 % Outlet
47 duct_2.diameter_meters = 0.1016; % 4 inches
48 duct_2.area = h_area( duct_2.diameter_meters );
49 duct_2.length_meters = 0.127; % 5 inches
50
51 % Flanged.
52
53 flow_rate_cubic_meters_per_second = 1.04772 / 60; % or 37 cubic-feet per minute
54
55
56
57 %% Problem 4a
58
59 % Calculate the Mach number of the flow in both pip sections.
60
61 duct_1.Mach = -1.0 * flow_rate_cubic_meters_per_second / ( (pi * duct_1.diameter_meters^2 / 4 ) *
62     c ); % 0.100 unitless
63
64 duct_2.Mach = -1.0 * flow_rate_cubic_meters_per_second / ( (pi * duct_2.diameter_meters^2 / 4 ) *
65     c ); % 0.00628 unitless
66
67
68 %% Problem 4b
69
70 % The flange is not considered because transmission loss is calculated.
71
72 frequency_set = 0:1:2.5e3;
73 nFreq = length( frequency_set );
74 TL = zeros( nFreq, 1 );
```

```

74
75 for frequency_index = 1:1:nFreq
76
77     f = frequency_set( frequency_index );
78
79     T_total = [ 1 0; 0 1 ];
80
81     T1 = duct_segment_transfer_matrix( f, rho0, c, duct_2.length_meters, duct_2.area );
82     T2 = [ 1 0; 0 duct_2.area/duct_1.area ];
83     T3 = duct_segment_transfer_matrix( f, rho0, c, duct_1.length_meters, duct_1.area );
84
85     T_net = T3 * T2 * T1 * T_total;
86
87     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
88     TL( frequency_index ) = 10 * log10( abs( ( T11 + duct_1.area*T12/(rho0*c) + (rho0*c)*
89         T21/duct_2.area + T22 ) / 2 )^2 );
90
91 end
92
93 TL_part_b = TL;
94
95
96
97 %% Problem 4c
98
99 % The flange is not considered because transmission loss is calculated.
100
101
102 frequency_set = 0:1:2.5e3;
103 nFreq = length( frequency_set );
104 TL = zeros( nFreq, 1 );
105
106
107 for frequency_index = 1:1:nFreq
108
109     f = frequency_set( frequency_index );
110
111     T_total = [ 1 0; 0 1 ];
112
113     T1 = duct_segment_transfer_matrix_flow( f, rho0, c, duct_2.length_meters, duct_2.area, duct_2
114         .Mach );
115     T2 = duct_expansion_connection_transfer_matrix( rho0, c, duct_2.area, duct_1.area, duct_1.
116         Mach );
117     T3 = duct_segment_transfer_matrix_flow( f, rho0, c, duct_1.length_meters, duct_1.area, duct_1
118         .Mach );
119
120     T_net = T3 * T2 * T1 * T_total;
121
122     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
123     TL( frequency_index ) = 10 * log10( abs( ( T11 + duct_1.area*T12/(rho0*c) + (rho0*c)*
124         T21/duct_2.area + T22 ) / 2 )^2 );
125
126 end
127
128 TL_part_c = TL;
129
130
131 %% Problem 4d
132
133 % Use volume to tune resonator.
134
135
136 w_o = 2*pi*129; % Target resonate frequency; estimated from plot.
137
138 d_cavity = 10*0.0254; % meters
139 d_neck = 0.0254; % meters
140 L_neck = 5*0.0254; % meters
141
142 S = d_neck^2*pi / 4;
143
144 Lo1 = 0.82 * ( 1 - 1.33*( d_neck / d_cavity ) );
145 Lo2 = 0.3;
146 Le = L_neck + Lo1 + Lo2;

```

```

147 V = S / ( (w_o/c)^2 * Le );
148
149
150 R_A = rho0*c / 10 * sqrt( Le / ( S * V ) );
151
152 h_Z_A = @( f, Le, S, V, R_A ) 1j*rho0*2*pi*f*Le/S - 1j*rho0*c^2/(V*2*pi*f) + R_A;
153
154
155 frequency_set = 0:1:2.5e3;
156 nFreq = length( frequency_set );
157 TL = zeros( nFreq, 1 );
158
159 for frequency_index = 1:1:nFreq
160
161     f = frequency_set( frequency_index );
162
163     Z_A = h_Z_A( f, Le, S, V, R_A );
164
165     T_total = [ 1 0; 0 1 ];
166
167     T1 = duct_segment_transfer_matrix_flow( f, rho0, c, duct_2.length_meters, duct_2.area, duct_2
        .Mach );
168     T2 = duct_expansion_connection_transfer_matrix( rho0, c, duct_2.area, duct_1.area, duct_1.
        Mach );
169     T3 = [ 1 0; 1/Z_A 1 ];
170     T4 = duct_segment_transfer_matrix_flow( f, rho0, c, duct_1.length_meters, duct_1.area, duct_1
        .Mach );
171
172     T_net = T4 * T3 * T2 * T1 * T_total;
173
174     T11 = T_net(1, 1); T12 = T_net(1, 2); T21 = T_net(2, 1); T22 = T_net(2, 2);
175     TL( frequency_index ) = 10 * log10( abs( ( T11 + duct_1.area*T12/(rho0*c) + (rho0*c)*
        T21/duct_2.area + T22 ) / 2 )^2 );
176
177 end
178
179
180 TL_part_d = TL;
181
182
183
184 %% Plot Transmission Loss Profiles
185
186 figure( ); ...
187 plot( frequency_set, TL_part_b ); hold on;
188 plot( frequency_set, TL_part_c );
189 plot( frequency_set, TL_part_d, 'Color', 'k', 'LineStyle', '—' ); grid on;
190 legend( 'No Flow', 'Flow', 'With Helmholtz Resonator', 'Location', 'SouthEast' );
191 xlabel( 'Frequency [Hz]' ); ylabel( 'Transmission Loss [dB]' );
192 title( 'Transmission Loss Profiles for Intake System' );
193 ylim( [ -45 45 ] );
194 %
195 Ax = gca;
196 Ax.XAxis.TickLabelInterpreter = 'latex';
197 Ax.YAxis.TickLabelInterpreter = 'latex';
198
199
200
201 %% Clean-up
202
203 if ( ~isempty( findobj( 'Type', 'figure' ) ) )
204     monitors = get( 0, 'MonitorPositions' );
205     if ( size( monitors, 1 ) == 1 )
206         autoArrangeFigures( 2, 2, 1 );
207     elseif ( 1 < size( monitors, 1 ) )
208         autoArrangeFigures( 2, 2, 1 );
209     end
210 end
211
212 if ( PRINT_FIGURES == 1 )
213     exportgraphics((gcf, 'Intake System.pdf', 'Append', true );
214 end
215
216 fprintf( 1, '\n\n*** Processing Complete ***\n\n' );
217
218
219
220 %% Reference(s)

```

5 Appendix - Matlab Code for Problem 5

```
1
2
3
4 %% Synopsis
5
6 % Question 5 – Intake Duct Silencer
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, 'DefaultFigureWindowState', '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 %% Problem 5a
33
34 % The considered dip frequency 940 Hz.
35
36 % The additional attenuation required is 14 dB.
37
38 liner_thickness = 0.0381; % meters
39 h = sqrt( ( pi*( 0.1016 - 2*liner_thickness )^2 ) / 4 ); % 0.0113 meters
40
41 m = 1 + liner_thickness/h; % 2.7
42
43 % As noted in the discussion, there are two cases to consider here.
44
45 % The first case is to consider the m value of 2.7, which provides
46 % additional attenuation. From Figure 8.37, the total attenuation of
47 % the lining is 10 dB.
48 attenuation_rate = 10 / 0.127; % 78.4 dB per meter
49
50 % The second card is that Figure 8.37 considers the combined effect of the
51 % expansion and the liner. In Problem 4, the effect of the expansion was
52 % calculated, so including it here would include its effect twice. Therefore,
53 % with an m of 1, there is no additional attenuation with the linear.
54
55
56
57 %% Problem 5b
58
59 % For a circular duct.
60
61 liner_thickness = 0.0381; % meters
62 half_diameter_of_open_orifice = 0.0127; % meters
63
64 h = sqrt(pi) * half_diameter_of_open_orifice / 2; % 0.0113 meters
65
66
67
68 %% Problem 5c
69
70 % The liner thickness ratio is,
71 liner_thickness / h; % 3.3852 unitless
72
73 % The normalized frequency is,
74 ( 2 * h ) / ( 343 / 940 ); % 0.06169 unitless
75
```



```

76
77
78 %% Problem 5d
79
80 % With a liner thickness ratio of 3.385, curve 5 is selected from the attenuation
81 % rate curve figure set. The horizontal axis value is 0.06169. The target attenuation
82 % rate for the attenuation rate curves is 39.2 dB/m (half of the value from Problem 5a)
83 % multiplied by  $h = 0.0113$ , which gives 0.44.
84
85 % From this information, the resistivity parameter plot selected is the bottom right plot
86 % with a value of 16.
87
88
89
90 %% Problem 5e
91
92 % Calculate the flow resistivity.
93
94 rho0 = 1.21; % Density of air (kg per cubic-meter).
95 c = 343; % Speed of sound in air (meters per second).
96
97 R1 = 16 * rho0*c / liner_thickness; % 1.74e5 kg / m^3*s
98
99
100 %% Clean-up
101
102 if ( ~isempty( findobj( 'Type', 'figure' ) ) )
103     monitors = get( 0, 'MonitorPositions' );
104     if ( size( monitors, 1 ) == 1 )
105         autoArrangeFigures( 2, 2, 1 );
106     elseif ( 1 < size( monitors, 1 ) )
107         autoArrangeFigures( 2, 2, 1 );
108     end
109 end
110
111
112 fprintf( 1, '\n\n*** Processing Complete ***\n\n' );
113
114
115
116 %% Reference(s)

```