Subject: Noise Control Applications - Module 1 Assignement

Date: February 9, 2025 (Submitted)

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,

$$f_{cut-on} = 0.5 \cdot \frac{c}{L} \tag{1}$$

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

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

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

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,

$$f_{cut-on} = 0.568 \cdot \frac{c}{d} \tag{2}$$

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

The cross-sectional area of the rectangular duct is,

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

The corresponding diameter for this area is,

$$\text{diameter} = \sqrt{\frac{0.24\ m^2}{\pi}} \cdot 2 = 0.17\ m$$

Using Eq. 2, the lowest cut-on frequency for this circular duct with air flow is,

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

Problem 1c

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

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

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,

$$c = \sqrt{\gamma \cdot R \cdot T_K} \tag{3}$$

where $\gamma = 1.4$ is the ratio of specific heats, $R = 287\ \frac{J}{kg \cdot K}$ 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.

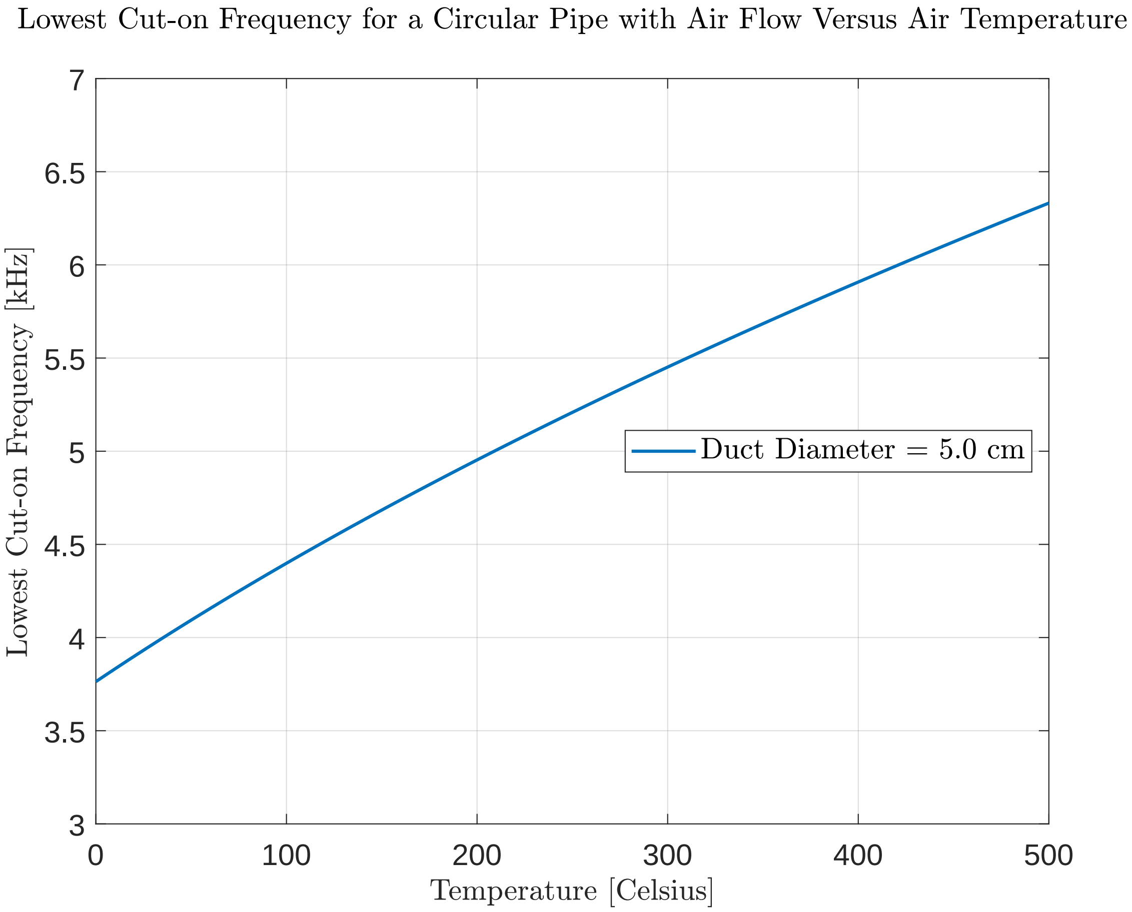

Figure 1: Lowest cut-on frequency for a circular 5 cm diameter duct versus air temperature.

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 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.

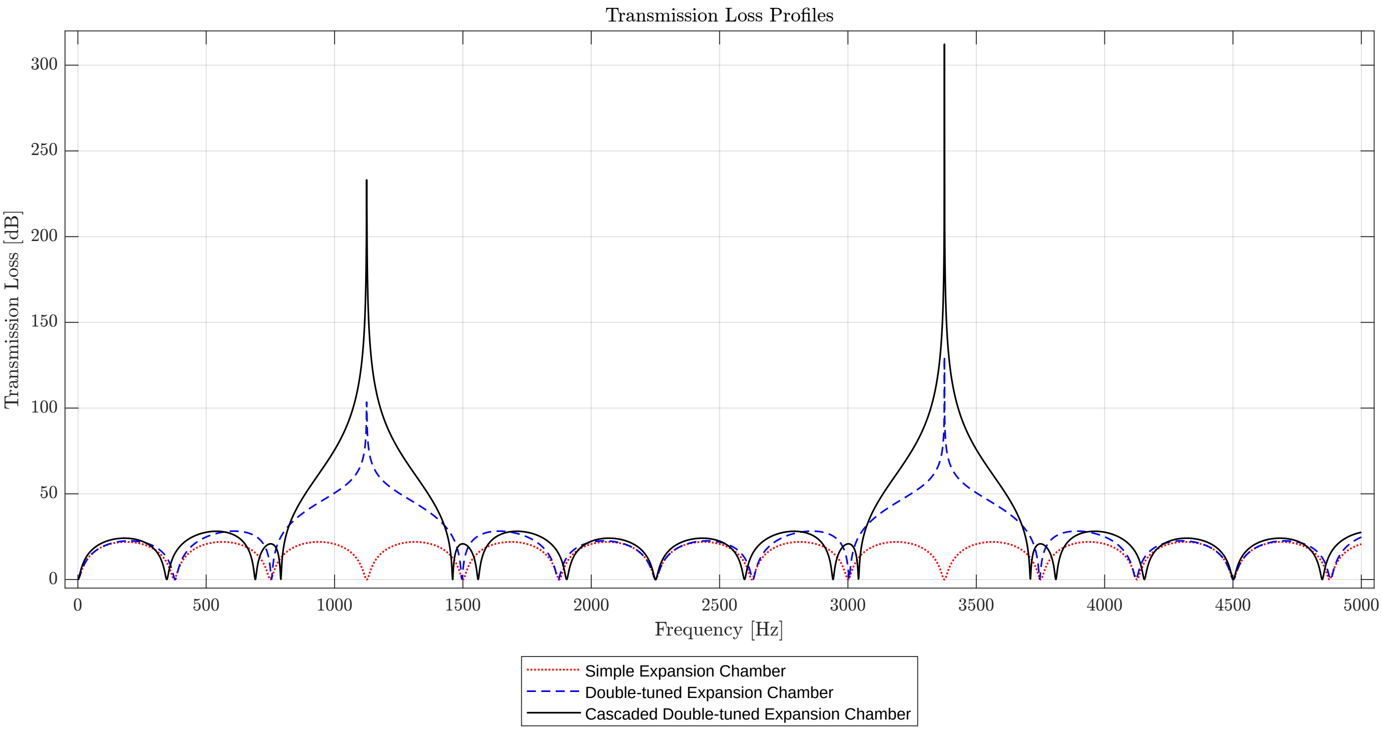

Figure 2: Transmission loss profiles for a simple expansion chamber, a double-tuned expansion chamber, and a cascaded double-tuned expansion chamber mufflers.

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.

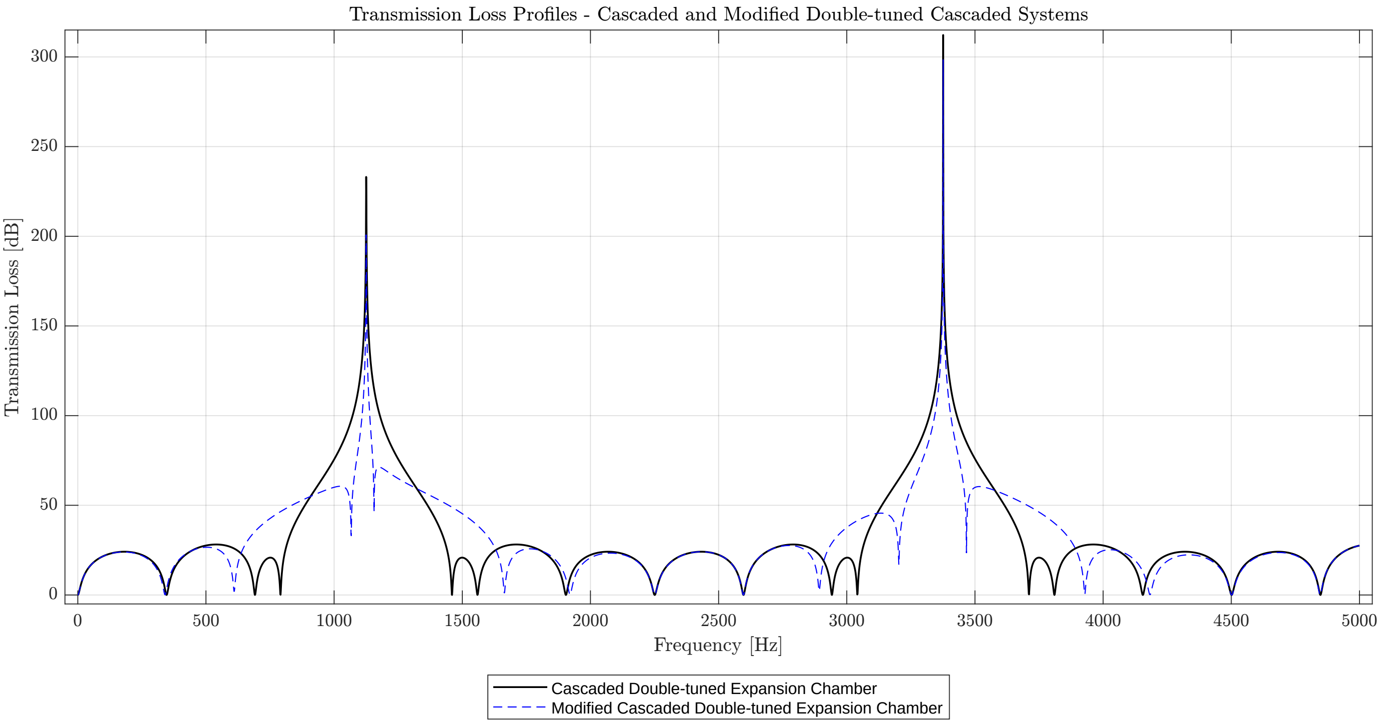

Figure 3: Transmission loss profiles for a cascaded double-tuned expansion chamber muffler and a modified version of this muffler.

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.

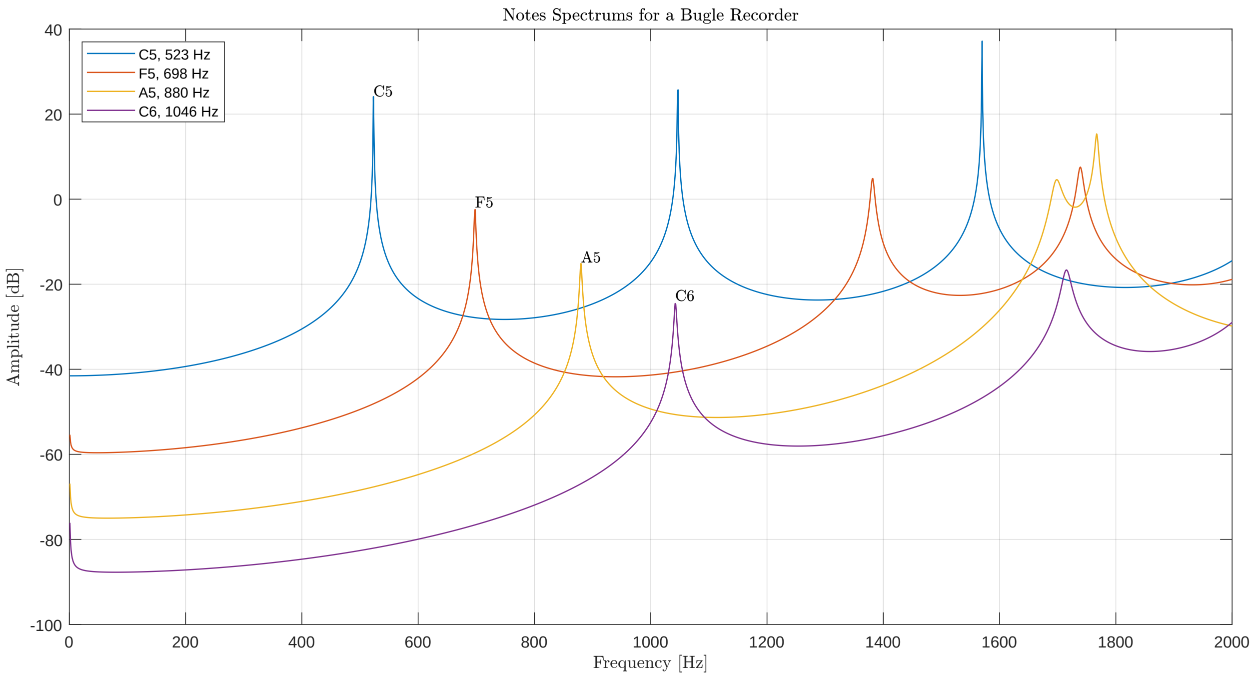

Figure 4: Spectrum for the C5, F5, A5, and C6 bugle recorder notes.

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

```
%% Synopsis

% Question 1 - Cut-on Frequencies in Ducts and Pipes



%% 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', 1.5 );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );

PRINT_FIGURES = 0;



%% Define Constants and Anonymous Functions

c_air = 343; % The speed of sound in air (meters per second).
c_water = 1500; % The speed of sound in water (meters per second).

gamma = 1.4; % The ratio of specific heats [unitless].
R = 287; % The gas constant [Joules per ( kilogram * Kelvin)].


h_f_cut_on_rectangular_duct = @( c, L )  0.5 .* c ./ L;
%
% c - The speed of sound.
% L - The largest cross-section dimension of the rectangular duct.


h_f_cut_on_circular_duct = @( c, d )  0.568 .* c ./ d;
%
% c - The speec of sound.
% L - The diameter of the circular duct.


h_speed_of_sound_in_air = @( gamma, R, temperature_Kelvin )  sqrt( gamma .* R .*
    temperature_Kelvin );



%% Problem 1a

% The cross-sectional dimensions for the rectangular duct are:  Lx = 12 cm and Ly = 20 cm.

% The largest dimension is Ly = 20 cm or 0.2 m.

% The cut-on frequency is,
h_f_cut_on_rectangular_duct( c_air, 0.2 ); % 857.5 Hz (shown in class 858 Hz)
    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 ) );



%% Problem 1b

% The cross-sectional dimensions for the rectangular duct are:  Lx = 12 cm and Ly = 20 cm.

% The cross-sectional area of the rectangular duct is 12 cm * 20 cm = 240 cm^2 or 0.024 m^2.
```

```matlab
rectangular_duct_cross_sectional_area = 0.12 * 0.20;  % 0.024 m^2

% The diameter of the circulat pipe is,
circular_duct_diameter = sqrt( 0.024 / pi ) * 2;  % 0.17481 meters
%
% Check:
    % pi * ( circular_duct_diameter / 2 )^2  CHECKED


% The cut-on frequency for the circular duct is,
h_f_cut_on_circular_duct( c_air, circular_duct_diameter );  % 1,114.5 Hz
    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 ) );



%% Problem 1c

% The cut-on frequency for the circular duct with water is,
h_f_cut_on_circular_duct( c_water, circular_duct_diameter );  % 4,873.9 Hz
    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 ) );

% The cut-on frequency should be higher because it is proportional to the
% speed of sound in a given medium.



%% Problem 1d

fprintf( 1, '\n Problem 1d:  See the figure.\n' );

temperature_range_celsius = 0:0.1:500;  % Celsius
    temperature_range_kelvin = temperature_range_celsius + 273.15;  % Kelvin


FONT_SIZE = 14;

figure( ); ...
    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;
        legend( 'Duct Diameter = 5.0 cm', 'Location', 'East', 'FontSize', FONT_SIZE, 'Interpreter
    ', 'Latex' );
        set( gca, 'FontSize', FONT_SIZE );
    %
    xlabel( 'Temperature [Celsius]', 'FontSize', FONT_SIZE );
        % xl = get( gca, 'xlabel' );    pxl = get( xl, 'position' );  pxl( 2 ) = 1.1 * pxl( 2 );
        %     set( xl, 'position', pxl );
    %
    ylabel( 'Lowest Cut-on Frequency [kHz]', 'FontSize', FONT_SIZE );
        % yl = get( gca, 'ylabel' );  pyl = get( yl, 'position' );  pyl( 1 ) = 1.2 * pyl( 1 );
        %     set( yl, 'position', pyl );
    %
    caption = sprintf( 'Lowest Cut-on Frequency for a Circular Pipe with Air Flow Versus Air
    Temperature\n' );
        title( caption, 'FontSize', FONT_SIZE );
    %
    ylim( [ 3  7 ] );



%% Problem 1e

fprintf( 1, '\n Problem 1e:  See Section Problem 1e of the Matlab script for the answers.\n\n' );


% 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 the cut-on
```

```
% frequencies are approximately equal.


% Question:  What about in air versus water?

% The lowest cut-on frequency is larger with water than air.  This due to
% the fact that the cut-on frequency is proportional to the speed of sound
% and the speed of sound in water is greater than it is in air.


% Question:  What about cold versus hot air?

% For a circular pipe, the cut-on frequency is higher in warm air than cold
% air.



%% 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

if ( PRINT_FIGURES == 1 )
    saveas( gcf, 'Cut-on Frequency Versus Temperature - Sunday, January 19, 2025.pdf' );
end

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

2 Appendix - Matlab Code for Problem 2

```
%% Synopsis

% Question 2 - Muffler Design Comparison



%% 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 );

PRINT_FIGURES = 0;



%% Constants

rho0 = 1.21;  % Air density (kg per m^3).
c = 343;  % Speed of sound in air (meters per second).

frequency_set = 0:1:5e3;  % Hertz



%% Dimensions

convert.inches_to_meters = 0.0254;
convert.foot_to_meters = 0.3048;

dimensions.inlet_diameter_meters = 2 * convert.inches_to_meters;  % 0.0508 meters
dimensions.inlet_length_meters = 6 * convert.foot_to_meters;  % 1.82 meters

dimensions.muffler_diameter_meters = 10 * convert.inches_to_meters;  % 0.254 meters
dimensions.muffler_length_meters = 18 * convert.inches_to_meters;  % 0.4572 meters

dimensions.outlet_diameter_meters = 2 * convert.inches_to_meters;  % 0.0508 meters
dimensions.outlet_length_meters = 1 * convert.foot_to_meters;  % 0.3048 meters

outlet_flanged = false;

dimensions.overhang = 3 *convert.inches_to_meters;  % 0.0762 meters

segment_diameters = [ ...
    dimensions.outlet_diameter_meters, ...
    dimensions.muffler_diameter_meters, ...
    dimensions.inlet_diameter_meters, ...
    ].';
%
h_area_from_diameter = @( d )  pi .* d.^2 ./ 4;
%
segment_areas = h_area_from_diameter( segment_diameters );

segment_lengths = [ ...
    dimensions.outlet_length_meters, ...
    dimensions.muffler_length_meters, ...
    dimensions.inlet_length_meters, ...
    dimensions.overhang, ...
    ].';

```

```
%% Part a - Simple Expansion Chamber

nFreq = length( frequency_set );
    TL = zeros( nFreq, 1 );

for frequency_index = 1:1:nFreq

    f = frequency_set( frequency_index );

    T_total = [ 1 0; 0 1 ];

    T1 = duct_segment_transfer_matrix( f, rho0, c, 0.3048, 0.0020268 );  % Duct - Outlet
    T2 = duct_segment_transfer_matrix( f, rho0, c, 0.4572, 0.050671 );  % Duct
    T3 = duct_segment_transfer_matrix( f, rho0, c, 1.8288, 0.0020268 );  % Duct - Inlet

    T_net = T3 * T2 * T1 * T_total;
    % T_net = T_inlet *  T_total;  % Zero transmission loss for a straight duct.

    T11 = T_net(1, 1);  T12 = T_net(1, 2);  T21 = T_net(2, 1);  T22 = T_net(2, 2);
        TL( frequency_index ) = 10 * log10( abs( ( T11  +  0.0020268*T12/(rho0*c)  +  (rho0*c)*T21/0.0020268  +  T22 ) / 2 )^2 );
        %
        % The transmission loss calculation does not require a load impedance.

end

TL_parta = TL;  % The maximum peak value should be about 22 (21.952) dB.
%
% Expected behaviour:
%
%   1.)  0 dB at 0 Hz.
%   2.)  The transmission loss of a straight duct section is zero;  energy out equals energy in.

max( TL_parta );  % 22 dB



%% Part b - Double-tuned Expansion Chamber

annulus_area_squared_meters = pi/4 * ( 0.254^2 - 0.0508^2 );

L_o = 0;  % Assume that the Lo extension is neglible.


nFreq = length( frequency_set );
    TL = zeros( nFreq, 1 );

for frequency_index = 1:1:nFreq

    f = frequency_set( frequency_index );

    T_total = [ 1 0; 0 1 ];

    T1 = duct_segment_transfer_matrix( f, rho0, c, (0.3048 + 0.0762), 0.0020268 );
    T3 = duct_segment_transfer_matrix( f, rho0, c, (0.4572 - 2*0.0762), 0.050671 );
    T5 = duct_segment_transfer_matrix( f, rho0, c, (1.8288 + 0.0762), 0.0020268 );

    k = 2*pi*f/c;
        Z_A = -1j*rho0*c/annulus_area_squared_meters*cot(k * ( 0.0762 + L_o ) );
            T2 = [ 1  0;  1/Z_A  1 ];
                T4 = T2;

    T_net = T5 * T4 * T3 * T2 * T1 * T_total;


    T11 = T_net(1, 1);  T12 = T_net(1, 2);  T21 = T_net(2, 1);  T22 = T_net(2, 2);
        TL( frequency_index ) = 10 * log10( abs( ( T11  +  0.0020268*T12/(rho0*c)  +  (rho0*c)*T21/0.0020268  +  T22 ) / 2 )^2 );
        %
        % The transmission loss calculation does not require a load impedance.

end

TL_partb = TL;
%
% Expected behaviour:
```

```
%
%    1.)   0 dB at 0 Hz.
%    2.)   0 dB at same locations as a simple expansion chamber.
%    3.)   Peaks at 1,125 Hz and 3,376 Hz;

% Frequency at which the quarter-wavelength is 0.0762 meters.
% 343 / ( 4 * 0.0762 );  % 1,125 Hz.

% Also work at three-quarter-wavelength.
% 3 * 1125;  % 3,375 Hz



%% Part c - Cascaded, Double-tuned Expansion Chamber

annulus_area_squared_meters = pi/4 * ( 0.254^2 - 0.0508^2 );

L_o = 0;  % Assume that the Lo extension is neglible.


nFreq = length( frequency_set );
    TL = zeros( nFreq, 1 );

for frequency_index = 1:1:nFreq

    f = frequency_set( frequency_index );
        k = 2*pi*f/c;
            Z_A = -1j*rho0*c/annulus_area_squared_meters*cot(k * ( 0.0762 + L_o ) );

    T_total = [ 1 0; 0 1 ];

    T1 = duct_segment_transfer_matrix( f, rho0, c, (0.3048 + 0.0762), 0.0020268 );  % Duct - Outlet
    T2 = [ 1   0;   1/Z_A   1 ];  % Straight Side Branch
    T3 = duct_segment_transfer_matrix( f, rho0, c, (0.2286 - 2*0.0762), 0.050671 );  % Duct
    T4 = T2;  % Straight Side Branch
    T5 = duct_segment_transfer_matrix( f, rho0, c, 2*0.0762, 0.050671 );  % Duct
    T6 = T2;  % Straight Side Branch
    T7 = T3;  % Duct
    T8 = T2;  % Straight Side Branch
    T9 = duct_segment_transfer_matrix( f, rho0, c, (1.8288 + 0.0762), 0.0020268 );  % Duct - Inlet

    T_net = T9 * T8 * T7 * T6 * T5 * T4 * T3 * T2 * T1 * T_total;

    T11 = T_net(1, 1);  T12 = T_net(1, 2);  T21 = T_net(2, 1);  T22 = T_net(2, 2);
        TL( frequency_index ) = 10 * log10( abs( ( T11  +  0.0020268*T12/(rho0*c)  +  (rho0*c)*T21/0.0020268  +  T22 ) / 2 )^2 );

end

TL_partc = TL;



%% Part d - Cascaded, Double-tuned Expansion Chamber

nFreq = length( frequency_set );
    TL = zeros( nFreq, 1 );

for frequency_index = 1:1:nFreq

    f = frequency_set( frequency_index );
        k = 2*pi*f/c;
            Z_A = -1j*rho0*c/annulus_area_squared_meters*cot(k * ( 0.0762 + L_o ) );

    T_total = [ 1 0; 0 1 ];

    T1 = duct_segment_transfer_matrix( f, rho0, c, (0.3048 + 0.0762), 0.0020268 );  % Duct - Outlet
    T2 = [ 1   0;   1/Z_A   1 ];  % Straight Side Branch
    T3 = duct_segment_transfer_matrix( f, rho0, c, 0.0076, 0.050671 );  % Duct
    T4 = T2;  % Straight Side Branch
    T5 = duct_segment_transfer_matrix( f, rho0, c, 2*0.0762, 0.050671 );  % Duct
    T6 = T2;  % Straight Side Branch
    T7 = duct_segment_transfer_matrix( f, rho0, c, (0.29718 - 2*0.0762), 0.050671 );  % Duct
    T8 = T2;  % Straight Side Branch
    T9 = duct_segment_transfer_matrix( f, rho0, c, (1.8288 + 0.0762), 0.0020268 );  % Duct -
```

```
    Inlet

    T_net = T9 * T8 * T7 * T6 * T5 * T4 * T3 * T2 * T1 * T_total;

    T11 = T_net(1, 1);  T12 = T_net(1, 2);  T21 = T_net(2, 1);  T22 = T_net(2, 2);
        TL( frequency_index ) = 10 * log10( abs( ( T11 + 0.0020268*T12/(rho0*c) + (rho0*c)*T21/0.0020268 + T22 ) / 2 )^2 );

end

TL_partd = TL;



%% Plot Transmission Loss Profiles

Y_LIMITS = [ -5 320 ];

h_figure_1 = figure( ); ...
    plot( frequency_set, TL_parta, 'LineWidth', 1.0, 'LineStyle', ':', 'Color', 'r' );  hold on;
    plot( frequency_set, TL_partb, 'LineWidth', 0.9, 'LineStyle', '--', 'Color', 'b' );
    plot( frequency_set, TL_partc, 'LineWidth', 0.9, 'LineStyle', '-', 'Color', 'k' );  grid on;
        legend( ...
            'Simple Expansion Chamber', ...
            'Double-tuned Expansion Chamber', ...
            'Cascaded Double-tuned Expansion Chamber', ...
            'Location', 'SouthOutside' );
    xlabel( 'Frequency [Hz]' );  ylabel( 'Transmission Loss [dB]' );
    title( 'Transmission Loss Profiles' );
    %
    Ax = gca;
        Ax.XAxis.TickLabelInterpreter = 'latex';
        Ax.YAxis.TickLabelInterpreter = 'latex';
    %
    axis( [ -50 5e3+50 Y_LIMITS ] );


Y_LIMITS = [ -5 315 ];

h_figure_2 = figure( ); ...
    plot( frequency_set, TL_partc, 'LineWidth', 1.0, 'LineStyle', '-', 'Color', 'k' );  hold on;
    plot( frequency_set, TL_partd, 'LineWidth', 0.6, 'LineStyle', '--', 'Color', 'b' );  grid on;
        legend( ...
            'Cascaded Double-tuned Expansion Chamber', ...
            'Modified Cascaded Double-tuned Expansion Chamber', ...
            'Location', 'SouthOutside' );
    xlabel( 'Frequency [Hz]' );  ylabel( 'Transmission Loss [dB]' );
    title( 'Transmission Loss Profiles - Cascaded and Modified Double-tuned Cascaded Systems' );
    %
    Ax = gca;
        Ax.XAxis.TickLabelInterpreter = 'latex';
        Ax.YAxis.TickLabelInterpreter = 'latex';
    %
    axis( [ -50 5e3+50 Y_LIMITS ] );



%% 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

if ( PRINT_FIGURES == 1 )
        exportgraphics( h_figure_1, 'Assignment 1 - Question 2 Figure All TL Profiles.pdf', 'Append', true );
        exportgraphics( h_figure_2, 'Assignment 1 - Question 2 Figure Comparison TL Plot For Cascaded Systems.pdf', 'Append', true );
end

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

3 Appendix - Matlab Code for Problem 3

```matlab



%% Synopsis

% Question 3 - Bugle Recorder



%% 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', 0.8 );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );

PRINT_FIGURES = 0;



%% Constants and Anonymous Functions

rho0 = 1.21; % Density of air (kg per cubic-meter).
c = 343; % Speed of sound in air (meters per second).

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);
%
% See Equation 8.34 on page 479 of Bies et al (2024).



%% Part a

% The estimated total length of the recorder is 0.325 meters.

% Estimation of hole location was done by trial-and-error.

pipe_net_length = 0.325;
pipe_area = pi*0.009^2/4;

flanged = false;


frequency_set = 1:1:2e3;

nFreq = length( frequency_set );
    A = zeros( nFreq, 1 );

for frequency_index = 1:1:nFreq

    f = frequency_set( frequency_index );

    T_total = [ 1 0; 0 1 ];

    T_segment = duct_segment_transfer_matrix( f, rho0, c, pipe_net_length, pipe_area );

    T_total = T_segment * T_total;

    Z = open_end_impedance( f, rho0, c, 0, pipe_area, flanged );

```

```
    T11 = T_total(1, 1);  T12 = T_total(1, 2);
        A( frequency_index ) = -10*log10( abs( T11 + T12 / Z )^2 );

end

A_parta = A;  clear A;

% figure( ); ...
%     plot( frequency_set, A_parta, 'LineWidth', 0.8 );  grid on;
%     xlabel( 'Frequency [Hz]' );  ylabel( 'Amplitude [dB]' );
%         legend( 'C5' );

% return

%% Part b

epsilon = 0.006 / 0.004;  % Diameter of the hold divided by diameter of pipe section (1.5).

switch ( 3 )

    case 0  % Original Value

        L_o = 0.00001;  % Estimate
            L_e = 0.004 + 2*L_o;

        fprintf( 1, '\nZero - Percentage change in pipe thickness:  %3.1f%%.\n\n', ( L_e - 0.004
    ) / 0.004 * 100 );

    case 1  % Ingard (2010)

        a = 0.006 / 2;

        L_o = ( 0.6*a + 0.85*a ) / 2;
            L_e = 0.004 + 2*L_o;

        fprintf( 1, '\nIngard (2010) - Percentage change in pipe thickness:  %3.1f%%.\n\n', ( L_e
     - 0.004 ) / 0.004 * 100 );

    case 2  % Kurze and Riedel (2013)

        e = epsilon^2;

        a = 0.006 / 2;

        L_o = pi*a*( 1 - 1.47*e^0.5 + 0.47*e^1.5 );
            L_e = 0.004 + 2*L_o;

        fprintf( 1, '\nKurze and Riedel (2013) - Percentage change in pipe thickness:  %3.1f%%.\n
    \n', ( L_e - 0.004 ) / 0.004 * 100 );

    case 3  % Ji (2005)

        a = 0.006 / 2;

        L_o = a*( 0.9326 - 0.6196*epsilon);
            L_e = 0.004 + 2*L_o;

        fprintf( 1, '\nJi (2005) - Percentage change in pipe thickness:  %3.1f%%.\n\n', ( L_e -
    0.004 ) / 0.004 * 100 );

    otherwise
        error ( '*** Invalid SWITCH Index ***' );

end


duct_lengths = 0.235/4 * ones( 4, 1 );  % 523 Hz, all holes covered.


frequency_set = 0:1:2e3;

nFreq = length( frequency_set );
    A = zeros( nFreq, 1 );

for frequency_index = 1:1:nFreq

    f = frequency_set( frequency_index );

```

```
    T_total = [ 1 0; 0 1 ];


    Z_A = 1j * rho0 * (2 * pi * f) * L_e / ( pi*0.006^2/4 );
    %
    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 );
    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 );
    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 );
        R_A = term_1 + term_2 + term_3;
    %
    Z_A = Z_A + R_A;
        T_Hole = [ 1  0;  1/Z_A  1 ];



    switch ( 0 )

        case 0  % All holes covered.  523 Hz
            T1 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area );  % Duct - Outlet
            T2 = [ 1 0;  0 1 ];  % Hole
            T3 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area );  % Duct
            T4 = [ 1 0;  0 1 ];  % Hole
            T5 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area );  % Duct
            T6 = [ 1 0;  0 1 ];  % Hole
            T7 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area );  % Duct

            % Check
            % 4 * 0.05875 = 0.235;

        case 1  % Hole 1 uncovered.  698 Hz
            T1 = duct_segment_transfer_matrix( f, rho0, c, 0.08775, pipe_area );  % Duct - Outlet
            T2 = T_Hole;
            T3 = duct_segment_transfer_matrix( f, rho0, c, 0.02975, pipe_area );  % Duct
            T4 = [ 1 0;  0 1 ];  % Hole
            T5 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area );  % Duct
            T6 = [ 1 0;  0 1 ];  % Hole
            T7 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area );  % Duct

            % Check
            % 0.08775 + 0.02975 + 2*0.05875 = 0.235;

        case 2  % Hole 2 uncovered.  880 Hz

            offset = 0.00625;  % 874

            T1 = duct_segment_transfer_matrix( f, rho0, c, 0.08775, pipe_area );  % Duct - Outlet
            T2 = T_Hole;
            T3 = duct_segment_transfer_matrix( f, rho0, c, 0.0505, pipe_area );  % Duct
            T4 = T_Hole;
            T5 = duct_segment_transfer_matrix( f, rho0, c, 0.038, pipe_area );  % Duct
            T6 = [ 1 0;  0 1 ];  % Hole
            T7 = duct_segment_transfer_matrix( f, rho0, c, 0.05875, pipe_area );  % Duct

            % Check
            % 0.08775 + 0.0505 + 0.038 + 0.05875 = 0.235

        case 3  % Hole 3 uncovered.  1,046 Hz

            T1 = duct_segment_transfer_matrix( f, rho0, c, 0.08775, pipe_area );  % Duct - Outlet
            T2 = T_Hole;
            T3 = duct_segment_transfer_matrix( f, rho0, c, 0.0505, pipe_area );  % Duct
            T4 = T_Hole;
            T5 = duct_segment_transfer_matrix( f, rho0, c, 0.02975, pipe_area );  % Duct
            T6 = T_Hole;
            T7 = duct_segment_transfer_matrix( f, rho0, c, 0.067, pipe_area );  % Duct

            % Check
            % 0.08775 + 0.0505 + 0.02975 + 0.067 = 0.235 m

    end

        T8 = duct_segment_transfer_matrix( f, rho0, c, 0.09, pipe_area );  % Duct - Inlet

        T_total =  T8 * T7 * T6 * T5 * T4 * T3 * T2 * T1 * T_total;

```

```
        Z = open_end_impedance( f, rho0, c, 0, pipe_area, flanged );

        T11 = T_total(1, 1);  T12 = T_total(1, 2);
            A( frequency_index ) = -10*log10( abs( T11 + T12 / Z )^2 );

end


% figure( ); ...
%     plot( frequency_set, A );  grid on;
%     xlabel( 'Frequency [Hz]' );  ylabel( 'Amplitude [dB]' );



%% Plot Note Set

load( 'A_C5_Data.mat' );  % Variable(s):  A_C5
load( 'A_F5_Data.mat' );  % Variable(s):  A_F5
load( 'A_A5_Data.mat' );  % Variable(s):  A_A5
load( 'A_C6_Data.mat' );  % Variable(s):  A_C6


h_figure_1 = figure( ); ...
    plot( frequency_set, A_C5 );  hold on;
        text( 523, 25, 'C5' );
    plot( frequency_set, A_F5 );
        text( 698, -1, 'F5' );
    plot( frequency_set, A_A5 );
        text( 880, -14, 'A5' );
    plot( frequency_set, A_C6 );  grid on;
        text( 1042, -23, 'C6' );
        %
        legend( 'C5, 523 Hz', 'F5, 698 Hz', 'A5, 880 Hz', 'C6, 1046 Hz', 'Location', 'NorthWest'
    );
    xlabel( 'Frequency [Hz]' );  ylabel( 'Amplitude [dB]' );
    title( 'Notes Spectrums for a Bugle Recorder' );



%% 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

if ( PRINT_FIGURES == 1 )
        exportgraphics( h_figure_1, 'Assignment 1 - Question 3 Bugle Recorder Note Spectrums.pdf'
    , 'Append', true );
end

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



%% Reference(s)
```

4 Appendix - Matlab Code for Problem 4

```
%% Synopsis

% Question 4 - Intake Duct



%% 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', 1.5 );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );

PRINT_FIGURES = 0;



%% Constants

rho0 = 1.21;  % Density of air (kg per cubic-meter).
c = 343;  % Speed of sound in air (meters per second).

h_area = @( diameter )  pi * diameter^2 / 4;



%% Define Shape

% Source
duct_1.diameter_meters = 0.0254;  % meters
    duct_1.area = h_area( duct_1.diameter_meters );  % squared-meters
duct_1.length_meters = 0.1524;  % meters


% Outlet
duct_2.diameter_meters = 0.1016;  % 4 inches
    duct_2.area = h_area( duct_2.diameter_meters );
duct_2.length_meters = 0.127;  % 5 inches

% Flanged.

flow_rate_cubic_meters_per_second = 1.04772 / 60;  % or 37 cubic-feet per minute



%% Part a

% Calculate the Mach number of the flow in both pip sections.

duct_1.Mach = -1.0 * flow_rate_cubic_meters_per_second / ( (pi * duct_1.diameter_meters^2 / 4 ) *
    c );  % 0.100 unitless
duct_2.Mach = -1.0 * flow_rate_cubic_meters_per_second / ( (pi * duct_2.diameter_meters^2 / 4 ) *
    c );  % 0.00628 unitless

return

%% Part b

% No flow.

outlet_flanged = true;  % Flanged end.
```

```
TEST_FLAG = 1;  % 1: right-to-left.

frequency_set = 0:0.1:2.5e3;
    nFreq = length( frequency_set );
        TL = zeros( nFreq, 1 );


for frequency_index = 1:1:nFreq

    f = frequency_set( frequency_index );


    T_total = [ 1 0; 0 1 ];


    if ( TEST_FLAG == 1 )

        % Right-to-left.
        %
        T_outlet = duct_segment_transfer_matrix( f, rho0, c, duct_2.length_meters, duct_2.area );
        T_contraction = [ 1   0; 0   duct_2.area/duct_1.area ];
        T_inlet = duct_segment_transfer_matrix( f, rho0, c, duct_1.length_meters, duct_1.area );

        Z = open_end_impedance( f, rho0, c, duct_2.length_meters, duct_2.area, outlet_flanged );

    else

        % Left-to-right.
        %
        T_outlet = duct_segment_transfer_matrix( f, rho0, c, duct_1.length_meters, duct_1.area );
        T_contraction = [ 1   0; 0   duct_1.area/duct_2.area ];
        T_inlet = duct_segment_transfer_matrix( f, rho0, c, duct_2.length_meters, duct_2.area );

        Z = open_end_impedance( f, rho0, c, duct_1.length_meters, duct_1.area, outlet_flanged );

    end


    T_net = T_inlet * T_contraction * T_outlet * T_total;


    T11 = T_net(1, 1);   T12 = T_net(1, 2);   T21 = T_net(2, 1);   T22 = T_net(2, 2);

    if ( TEST_FLAG == 1 )
        TL( frequency_index ) = 10 * log10( abs( ( T11  +  duct_1.area*T12/(rho0*c)  +  (rho0*c)*T21/duct_2.area  +  T22 ) / 2 )^2 );
    else
        TL( frequency_index ) = 10 * log10( abs( ( T11  +  duct_2.area*T12/(rho0*c)  +  (rho0*c)*T21/duct_1.area  +  T22 ) / 2 )^2 );
    end


end  % End:  for f = frequency_set

TL_part_b = TL;

% return

%% Part c

% Flow present (use Mach numbers).

outlet_flanged = true;  % Flanged end.


frequency_set = 0:0.1:2.5e3;
    nFreq = length( frequency_set );
        TL = zeros( nFreq, 1 );


for frequency_index = 1:1:nFreq

    f = frequency_set( frequency_index );


    T_total = [ 1 0; 0 1 ];
```

```
    T_outlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_2.length_meters, duct_2.area, duct_2.Mach );

    T_expansion = duct_expansion_connection_transfer_matrix( rho0, c, duct_2.area, duct_1.area, duct_1.Mach );

    T_inlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_1.length_meters, duct_1.area, duct_1.Mach );

    T_net = T_inlet * T_expansion * T_outlet * T_total;


    Z = open_end_impedance( f, rho0, c, duct_2.length_meters, duct_2.area, outlet_flanged );

    T11 = T_net(1, 1);  T12 = T_net(1, 2);  T21 = T_net(2, 1);  T22 = T_net(2, 2);
        TL( frequency_index ) = 10 * log10( abs( ( T11  +  duct_1.area*T12/(rho0*c)  +  (rho0*c)*T21/duct_2.area  +  T22 ) / 2 )^2 );


end  % End:  for f = frequency_set

TL_part_c = TL;

% return

%% Part d

% Flow present (use Mach numbers).
% Helmholtz resonator in place (between lefthand duct and expansion).

outlet_flanged = true;  % Flanged end.


% Resonance
w_o = 2*pi*136.6;  % Estimated from plot.


% helmholtz_diameter_cavity =
% helmholtz_diameter_neck = 1e-3;
%     helmholtz_L01 = 0.82 * ( 1 - 1.33*(helmholtz_diameter_neck/helmholtz_diameter_cavity ) );
%     %
%     epsilon = helmholtz_diameter_cavity / duct_1.length_meters;
%     % helmholtz_L02 =
%
% helmholtz_volume = 1e-3;
%
% % keyboard
%
% helmholtz_L_neck = 1e-3;
% Q = 2;
%
% R_A = rho0*c / Q * sqrt( L_e / ( pi*helmholtz_diameter_neck^2/4 * helmholtz_volume ) );



frequency_set = 0:0.1:2.5e3;
    nFreq = length( frequency_set );
        TL = zeros( nFreq, 1 );


for frequency_index = 1:1:nFreq

    f = frequency_set( frequency_index );


    T_total = [ 1 0; 0 1 ];

    T_outlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_2.length_meters, duct_2.area, duct_2.Mach );

    T_expansion = duct_expansion_connection_transfer_matrix( rho0, c, duct_2.area, duct_1.area, duct_1.Mach );


    % 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;
        % T_Helmholtz = [ 1   0;   1/Z_A   1 ];
```

```
    T_inlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_1.length_meters, duct_1.area,
    duct_1.Mach );


    % T_net = T_inlet * T_Helmholtz * T_expansion * T_outlet * T_total;
    T_net = T_inlet * T_expansion * T_outlet * T_total;


    Z = open_end_impedance( f, rho0, c, duct_2.length_meters, duct_2.area, outlet_flanged );

    T11 = T_net(1, 1);  T12 = T_net(1, 2);  T21 = T_net(2, 1);  T22 = T_net(2, 2);
        TL( frequency_index ) = 10 * log10( abs( ( T11  +  duct_1.area*T12/(rho0*c)  +  (rho0*c)*
    T21/duct_2.area  +  T22 ) / 2 )^2 );


end  % End:  for f = frequency_set

TL_part_d = TL;

% return

%% Plot

Y_LIMITS = [ 0  50 ];

figure( ); ...
    plot( frequency_set, TL_part_b );  hold on;
    plot( frequency_set, TL_part_c, '-' );
    plot( frequency_set, TL_part_d, '-.' );  grid on;
    legend( ...
        'No Flow - Part b', ...
        'Flow - Part c', ...
        'Flow and Resonator - Part d', ...
        'Location', 'SouthOutside' );
    xlabel( 'Frequency [Hz]' );  ylabel( 'Amplitude [dB]' );
    title( 'Transmission Loss Profiles' );
    %
    Ax = gca;
        Ax.XAxis.TickLabelInterpreter = 'latex';
        Ax.YAxis.TickLabelInterpreter = 'latex';
    %
    % axis( [ -50  5e3+50  Y_LIMITS ] );
    %
    if ( PRINT_FIGURES == 1 )
        exportgraphics( gcf, 'Figure TL All Profiles.pdf', 'Append', true );
    end

return

%% Part c

return


%% Part d

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)
```

5 Appendix - Matlab Code for Problem 5

```
%% Synopsis

% Question 4 - Intake Duct



%% To Do

% Check loss profile.



%% Note(s)

% Search for FIXMEs.


% In class note, the areas for the impedance might have bee wrong;  switch them?


% Use negative Mach numbers in the equations.  The analysis for this case
% is the same as for the horn example.  Inlet on the left, outlet on the
% right.


% The S in the diagram (pink area) for the Helmholtz resonator is the
% cross-sectional area of the resonator neck connecting it to the tube.


% For Lo2, use the value for a quarter-wavelength side tube.


% The area expansion ratio is determined using the original duct diameters.



%% 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', 1.5 );
set( 0, 'DefaultTextInterpreter', 'Latex' );

format ShortG;

pause( 1 );

PRINT_FIGURES = 0;



%% Define Constants and Anonymous Functions

rho0 = 1.21;  % Density of air (kg per cubic-meter).
c = 343;  % Speed of sound in air (meters per second).


h_area = @( diameter )  pi * diameter^2 / 4;



%% Define Shape

% Source
duct_1.diameter_meters = 0.0254;  % 1 inch
duct_1.length_meters = 0.1524;  % 6 inches
```

```
duct_1.area = h_area( duct_1.diameter_meters );

% Outlet
duct_2.diameter_meters = 0.1016;  % 4 inches
duct_2.length_meters = 0.127;  % 5 inches
duct_2.area = h_area( duct_2.diameter_meters );
%
% Flanged.

%

flow_rate_cubic_meters_per_second = 1.04772 / 60;  % or 37 cubic-feet per minute

% return

%% Part a

% The -35 dB transmission loss dip\notch is at about 1,150 Hz.
f = 1150;

% Total attenuation required is +35 dB to make it 0 dB.

l = 0.0381;  % meters (1.5 inches)
h = 0.0127;  % meters (0.5 inches)

% The total attenuation of the lining from Figure 8.37 (Bies et. al., Fifth Edition),
%
% m = ( h_area( 0.1016 ) - h_area( 0.0254 ) ) / h_area( 0.0254 )
%
% ASSUMPTION:  Cross-section area of lined duct is the same as the inlet.
m = 1;

k = 2*pi*1.15e3;

length_of_expansion_chamber = 0.127;  % meters

% kL = k*length_of_expansion_chamber

% ASSUMPTION:
%
% Expansion ratio is m = 1;
% Assume peak difference is 10 dB.
% Total attunation of lining is 10 dB.
%
% The attenuation rate is about 10 dB / 0.127 meters or 78.7 dB per meter.

% return

%% Part b

l = 0.0381;  % meters (1.5 inches)

h = 0.011255;  % meters
    h_validate = 0.5*sqrt( ( pi*( duct_2.diameter_meters - 2*l  )^2 ) / 4 );  % Same value.



%% Part c

% The liner thickness ratio is,
l / h;  % 3.3852 unitless

% The normalized frequency is,
( 2 * h ) / ( 343 / f );  % 0.075471 unitless



%% Part d

% Assume the attenuation rate from Part a is 18 dB per meter.

% Use bottom, right subplot (16).

% The approximate resistivity parameter is 16.



%% Part e
```

```
% Calculate the flow resistivity.

R1 = 16 * rho0*c / l;  % 1.74e5 kg / m^3*s



%% Placeholder

% % Flow present (use Mach numbers).
% % Helmholtz resonator in place (between lefthand duct and expansion).
%
% outlet_flanged = true;  % Flanged end.
%
%
% % Resonance
% w_o = 2*pi*136.6;  % Estimated from plot.
%
%
% % helmholtz_diameter_cavity =
% % helmholtz_diameter_neck = 1e-3;
% %     helmholtz_L01 = 0.82 * ( 1 - 1.33*(helmholtz_diameter_neck/helmholtz_diameter_cavity ) );
% %     %
% %     epsilon = helmholtz_diameter_cavity / duct_1.length_meters;
% %     % helmholtz_L02 =
% %
% % helmholtz_volume = 1e-3;
% %
% % % keyboard
% %
% % helmholtz_L_neck = 1e-3;
% % Q = 2;
% %
% % R_A = rho0*c / Q * sqrt( L_e / ( pi*helmholtz_diameter_neck^2/4 * helmholtz_volume ) );
%
%
%
% frequency_set = 0:0.1:2.5e3;
%     nFreq = length( frequency_set );
%         TL = zeros( nFreq, 1 );
%
%
% for frequency_index = 1:1:nFreq
%
%     f = frequency_set( frequency_index );
%
%
%     T_total = [ 1 0; 0 1 ];
%
%     T_outlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_2.length_meters, duct_2.area, duct_2.Mach );
%
%     T_expansion = duct_expansion_connection_transfer_matrix( rho0, c, duct_2.area, duct_1.area, duct_1.Mach );
%
%
%     % 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;
%         % T_Helmholtz = [ 1  0;  1/Z_A  1 ];
%
%
%     T_inlet = duct_segment_transfer_matrix_flow( f, rho0, c, duct_1.length_meters, duct_1.area, duct_1.Mach );
%
%
%     % T_net = T_inlet * T_Helmholtz * T_expansion * T_outlet * T_total;
%     T_net = T_inlet * T_expansion * T_outlet * T_total;
%
%
%     Z = open_end_impedance( f, rho0, c, duct_2.length_meters, duct_2.area, outlet_flanged );
%
%     T11 = T_net(1, 1);  T12 = T_net(1, 2);  T21 = T_net(2, 1);  T22 = T_net(2, 2);
%         TL( frequency_index ) = 10 * log10( abs( ( T11 + duct_1.area*T12/(rho0*c) + (rho0*c)*T21/duct_2.area + T22 ) / 2 )^2 );
%
%
% end  % End:  for f = frequency_set
%
```

```matlab
% TL_part_d = TL;

% return

%% Plot

% Y_LIMITS = [ 0   50 ];
%
% figure( ); ...
%      plot( frequency_set, TL_part_d );   hold on;
%      % plot( frequency_set, TL_part_c, '-' );
%      % plot( frequency_set, TL_part_d, '-.' );   grid on;
%      % legend( ...
%      %      'No Flow - Part b', ...
%      %      'Flow - Part c', ...
%      %      'Flow and Resonator - Part d', ...
%      %      'Location', 'SouthOutside' );
%      xlabel( 'Frequency [Hz]' );   ylabel( 'Transmission Loss [dB]' );
%      title( 'Transmission Loss Profiles' );
%      %
%      Ax = gca;
%          Ax.XAxis.TickLabelInterpreter = 'latex';
%          Ax.YAxis.TickLabelInterpreter = 'latex';
%      %
%      % axis( [ -50   5e3+50   Y_LIMITS ] );
%      %
%      if ( PRINT_FIGURES == 1 )
%          exportgraphics( gcf, 'Figure TL All Profiles.pdf', 'Append', true );
%      end

% 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)
```