Homework example problem:

Design an acoustic horn with the following conditions:

a) The diameter at the source end is 1 cm
b) The diameter at the outlet is 20 cm
c) The total length is 2 m

Plot the amplification (inverse of noise reduction) for a frequency range of 0–5000 Hz. Experiment with different shapes of horn. How do different shapes affect the response?

Solution

Note: it is not necessary to reproduce the prompt in your solution.

The response of the horn was predicted by breaking the length into 20 segments, each 0.1 m long. In this way, a gradual change in shape over the length of the horn could be made out of short pieces of constant cross section. The transition matrix for each segment is

$$\begin{bmatrix} \cos kL & \frac{j\rho_0 c}{S}\sin kL \\ \frac{jS}{\rho_0 c}\sin kL & \cos kL \end{bmatrix}$$

where L is the legnth of each section and S is the cross-sectional area of the segment. k, ρ_0, and c are the wavenumber, density, and speed of sound in air. Assuming the outlet is unflanged, the outlet impedance is

$$Z_{\text{out}} = j\frac{\rho_0 c}{S_1}\tan\big(k(L + l_o)\big)$$

where $l_o = 0.61a$ and a is the outlet radius (10 cm). Different segments produce the total transition matrix via

$$\begin{Bmatrix} p_{in} \\ S_{in}u_{in} \end{Bmatrix} = [T_N] \dots [T_2][T_1] \begin{Bmatrix} p_{out} \\ S_{out}u_{out} \end{Bmatrix}$$

with individual segment matrices numbered from outlet (1) to inlet ($N = 20$). Different area profiles were explored, including

- Constant (no increase in area)
- Linear (diameter increases linearly)
- Big mouth linear (diameter increases up to 40 cm, even though this was not in the prompt)
- Quadratic (diameter increases quadratically, so area increases linearly)
- Quarter quadratic (same as quadratic, but constant until the last quarter of the length

The amplification was calculated via

$$A = -\mathrm{NR} = -10 \log_{10} \left| T_{11} + \frac{T_{12}}{Z_{\mathrm{out}}} \right|^2$$

where T_{11} and T_{12} are entries of the combined transition matrix. Amplifications for the different shapes were calculated in MATLAB and shown in Figure 1. MATLAB code is included in Appendix A.

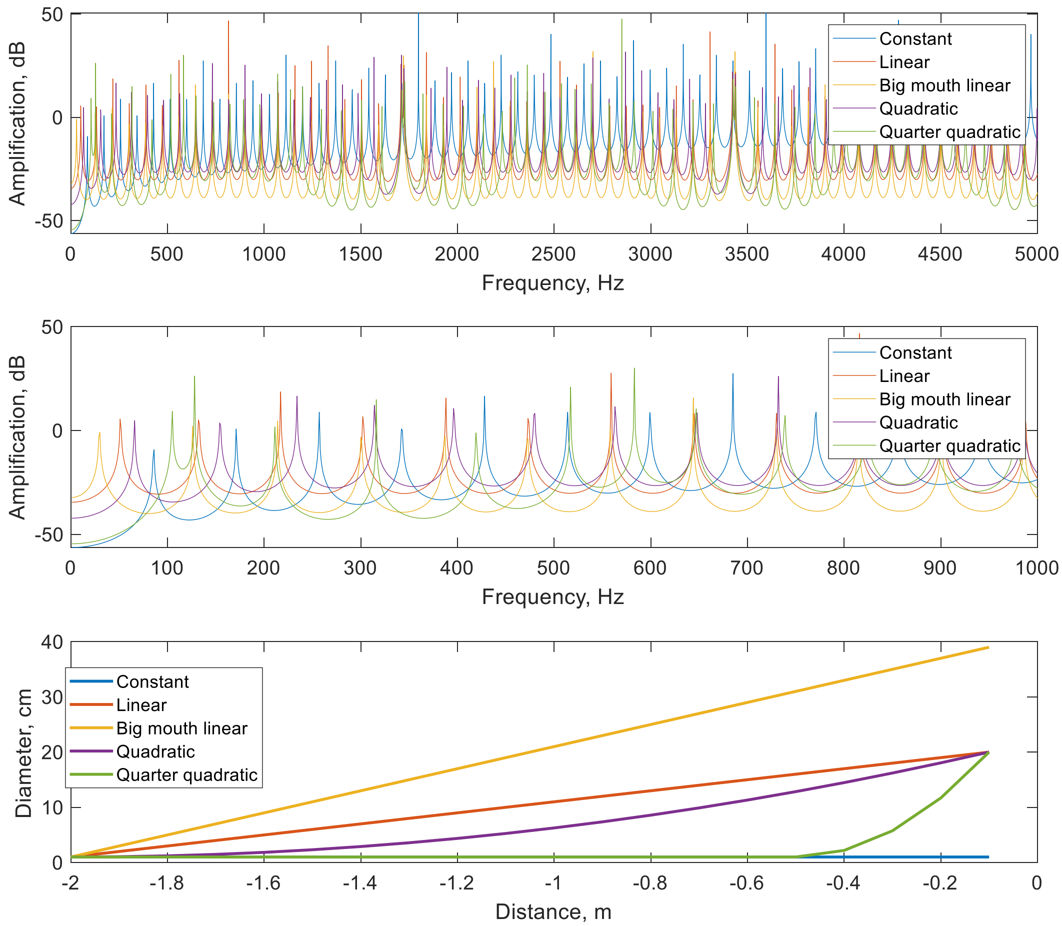

Figure 1: Responses and horn shapes. Top: Amplfication from 0–5000 Hz. Middle: Zoomed in to 0–1000 Hz. Bottom: Horn shape profiles.

The horn shapes appear to amplify the lowest frequencies (below 100 Hz) compared to the constant cross section. Additionally, the first resonance frequency shifts down, with the largest opening shifting the lowest. The “quarter quadratic” profile had some peaks move closer together, resulting in non-integer harmonics. Perhaps this is the cause of the differences in timbre in woodwind (no horn) and brass (horn) instruments.

Appendix A: MATLAB code

Note: if using MS Word, Paste→Keep source formatting, if using LaTeX, use a [code block](). Try to break out the actual math bits of code into separate functions that are just called by a "run and plot" script. I am mostly concerned with your implementation of the equations.

```
function A = horn_amplification(f, d_i, L_i, rho0, c)

% d_i is vector segment diameters (outlet to inlet)
% L_i is vector of segment lengths (outlet to inlet)
% A is amplification (inlet to outlet)

S_i = pi*d_i.^2/4;

flanged = false;

nFreq = length(f);
nSegments = length(L_i);

A = zeros(nFreq,1);

for iFreq = 1:nFreq
    T_total = [1 0; 0 1]; % Start with identity matrix
    for iSegment = 1:nSegments
        L = L_i(iSegment);
        S = S_i(iSegment);
        T_segment = duct_segment_transfer_matrix(f(iFreq), rho0, c, L, S);
        T_total = T_segment*T_total; % New segment goes on the left
    end
    T11 = T_total(1,1);
    T12 = T_total(1,2);

    Z = open_end_impedance(f(iFreq), rho0, c, 0, S(1), flanged);

    A(iFreq) = -10*log10(abs(T11+T12/Z)^2);
end
```

```
function Z = open_end_impedance(f, rho0, c, L, S, flanged)
% Only for circular cross sections

a = sqrt(S/pi);
k = 2*pi*f/c;

if flanged
    L_0 = 8*a/(3*pi);
else
    L_0 = 0.61*a;
end

L_e = L+L_0;

Z = 1j*rho0*c/S*tan(k*L_e);



function T = duct_segment_transfer_matrix(f, rho0, c, L, S)

k = 2*pi*f/c;

T = [cos(k*L), 1j*rho0*c/S*sin(k*L);...
    1j*S/(rho0*c)*sin(k*L), cos(k*L)];
```