# ASE 389P-7
# Exam 1

Alejandro Moreno

October 4, 2022

## 1 Problem 1

### 1.1 Instruction

Write a function in Matlab that simulates the train-horn-Doppler scenario discussed in lecture. Assume that the train tracks are rectilinear.

Be sure to account for the nonzero time of flight $\delta t_{TOF}$ as discussed in lecture. The effect of $\delta t_{TOF} > 0$ is that the stationary observer will discern an $f_D$ at time $t_k$ that relates to the train's line-of-sight velocity at time $t_k - \delta t_{TOF}$. More precisely, the apparent frequency of the train horn at the location of the observer at time $t_k$ is given by

$$f_r(t_k) = \frac{f_c}{1 + \frac{v_{los}(t_k)}{v_s}} \tag{1}$$

where $f_c$ is the nominal horn frequency, $v_{los}(t_k)$ is the line-of-sight velocity at $t_k$, and $v_s$ is the speed of the signal in the medium. Note that the line-of-sight geometry used to calculate $v_{los}(t_k)$ is between the observer at time $t_k$ and the horn at time $t_k - \delta t_{TOF}$.

Download the audio file trainout.wav from Canvas. This file was created with the following input argument values:

```
fh = 440;
vTrain = 20;
t0 = 0;
x0 = 0;
delt = 0.01;
N = 1000;
vs = 343;
```

Set up your simulator with these same values. Estimate the values of xObs and dObs by adjusting them in your simulation until you get an apparent received frequency profile that matches the one in the audio file.

### 1.2 MATLAB code

```
function [fDVec,tVec] = ...
simulateTrainDoppler(fc, vTrain, t0, x0, xObs, dObs, delt, N, vs)
% simulateTrainDoppler : Simulate the train horn Doppler shift scenario.
%
% INPUTS
%
% fc ------- train horn frequency, in Hz
%
% vTrain -- constant along-track train speed, in m/s
%
% t0 ------- time at which train passed the along-track coordinate x0, in
%            seconds
%
% x0 ------- scalar along-track coordinate of train at time t0, in meters
%
% xObs ----- scalar along-track coordinate of observer, in meters
%
% dObs ----- scalar cross-track coordinate of observer, in meters (i.e.,
%            shortest distance of observer from tracks)
%
% delt ----- measurement interval, in seconds
%
% N -------- number of measurements
%
% vs ------- speed of sound, in m/s
%
%
% OUTPUTS
%
% fDVec ---- N-by-1 vector of apparent Doppler frequency shift measurements as
%            sensed by observer at the time points in tVec
%
% tVec ----- N-by-1 vector of time points starting at t0 and spaced by delt
%            corresponding to the measurements in fDVec
%
%+------------------------------------------------------------------------------+
% References:
%
%
% Author:
%+==============================================================================+
  tVec = zeros(N,1);
  fDVec = zeros(N,1);

  t = t0;
```

```matlab
    for h = 1:N
        % Find rRX(t) and vRX(t)
        rRX = positionRX(t, t0, xObs, dObs);
        vRX = velocityRX(t, t0);

        % Calculate TOF
        TOF = calculateTOF(t, rRX, t0, x0, vTrain, vs);

        % Find rTX(t-TOF) and vTX(t-TOF)
        rTX = positionTX(t-TOF, t0, x0, vTrain);
        vTX = velocityTX(t-TOF, t0, vTrain);

        % Find the line-of-sight
        los = (rRX - rTX) / norm(rRX - rTX);

        % Project the velocity of TX to the line-of-sight
        vTX_los =  dot(vTX, los);
        % Project the velocity of RX to the line-of-sight
        vRX_los =  dot(vRX, los);
        % Calculate the line-of-sight velocity
        v_los = vRX_los - vTX_los;

        % Calculate the apparent Doppler frequency shift as sensed by the
        % observer
        beta = v_los / vs;
        fr = fc / (1 + beta);
        fDVec(h) = fr - fc;
        tVec(h) = t;
        t = t + delt;
    end
end

function [rRX] = positionRX(t, t0, x0, y0)
    rRX = [x0, y0];
end

function [vRX] = velocityRX(t, t0)
    vRX = [0, 0];
end

function [rTX] = positionTX(t, t0, x0, v0)
    x = x0 + v0*(t-t0);
    rTX = [x, 0];
end
```

```
function [vTX] = velocityTX(t, t0, v0)
    vTX = [v0, 0];
end

function [TOF] = calculateTOF(t, rRX, t0, x0, vTrain, vs)
    % Initialize with a guess
    TOF = norm(rRX - positionTX(t, t0, x0, vTrain))/vs;
    error = vs*TOF - norm(rRX - positionTX(t-TOF, t0, x0, vTrain));

    % Iterate until convergence
    while ( abs(error) > 1e-3 )
        TOF = TOF - error/vs;
        error = vs*TOF - norm(rRX - positionTX(t-TOF, t0, x0, vTrain));
    end
end

% topSimulateTrainDoppler.m
%
% Top-level script for train Doppler simulation


clear; clc; close all;
%----- Setup
fc = 440;
vTrain = 20;
t0 = 0;
x0 = 0;
delt = 0.01;
N = 1000;
vs = 343;
xObs = 56.8;
dObs = 10;

%----- Simulate
[fDVec,tVec] = simulateTrainDoppler(fc,vTrain,t0,x0,xObs,dObs,delt,N,vs);
fApparentVec = fDVec + fc;

%----- Plot
plot(tVec,fDVec + fc, 'r');
xlabel('Time (seconds)');
ylabel('Apparent horn frequency (Hz)');
grid on;
shg;

%----- Generate a sound vector
T = delt*N;                     % simulation time (sec)
```

```
fs = 22050;                       % sample frequency (Hz)
deltSamp = 1/fs;                  % sampling interval (sec)
Ns = floor(T/deltSamp);           % number of samples
tsamphist = [0:Ns-1]'*deltSamp;
Phihist = zeros(Ns,1);
fApparentVecInterp = interp1(tVec,fApparentVec,tsamphist,'spline');
for ii=2:Ns
  fii = fApparentVecInterp(ii);
  Phihist(ii) = Phihist(ii-1) + 2*pi*fii*deltSamp;
end
soundVec = sin(Phihist);

% %----- Play the sound vector
%sound(soundVec, fs);

%----- Write to audio file
audiowrite('my_trainout.wav',soundVec,fs);

%----- Write frequency time history to output file
save trainData fApparentVec tVec

%% Extra Points
close all
% I use an spectrogram to better understand the original audio signal
[y, fs] = audioread('trainout.wav');

Nspec = 2^(nextpow2(length(y)) - 7);
wspec = hamming(Nspec);
Noverlap = Nspec/2;
figure()
% spectrogram(y, 512, 128, 128, fs, 'yaxis');
spectrogram(y, wspec, Noverlap, Nspec, fs, 'yaxis');
colormap jet
title("original signal")

set(gca,'Linewidth',1.2,'FontSize',36)
set(gcf,'Position',[2500 100 1550 800])

% Then, I manually analyze my simulated audio signal to make it look as
% similar to the original as posible by changing the position of the
% receiver
% [y, fs] = audioread('my_trainout.wav');
%
% figure()
% spectrogram(y, 512, 120, 128, fs, 'yaxis');
```

```
% colormap jet
% title("My signal")
%
% set(gca,'Linewidth',1.2,'FontSize',36)
% set(gcf,'Position',[2500 100 1550 800])
```

### 1.3 Result

Analizing the given signal it's possible to manually tag the moment in which the train passed in front of the receiver. Figure 1 shows how the power density is concentrated at higher frequency before 3.069 sec and drops rapidly after that. This indicates the exact moment at which the train passed infront of the receiver.

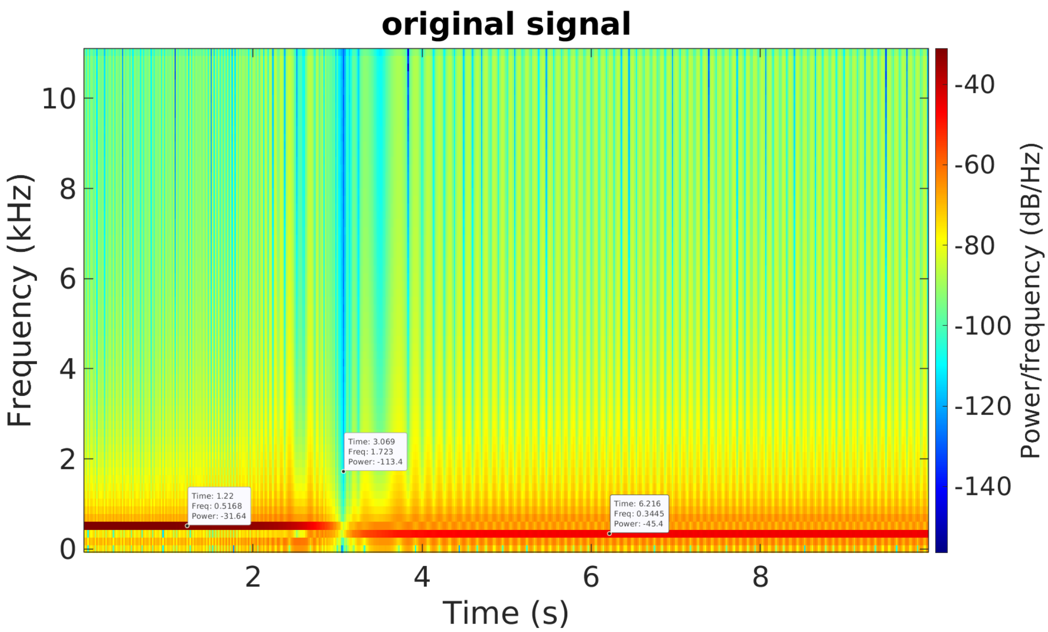


Figure 1: Spectrogram of the given signal.

This, approach lead to $xObs = 56.8$ and $dObs = 10$.

## 2 Exercise 2

### 2.1 Instruction

In this problem, we'll examine the effects of clock frequency bias on Doppler derived from frequency downconversion and sampling.

Usually, we don't measure signal properties directly at the signal's incoming frequency. Instead, we convert incoming signals to a lower center frequency by a process called frequency conversion. To understand frequency conversion, recall that

$$cos(x)cos(y) = \frac{1}{2}[cos(x-y) + cos(x+y)] \tag{2}$$

Hence, by multiplying an incoming signal $x(t) = cos(2\pi f_c t)$ by a local signal $x_l(t) = 2cos(2\pi f_l t)$ and then low-pass filtering the result to eliminate the high frequency $f_c + f_l$ component, we convert the high frequency signal down to a lower frequency $f_b = f_c - f_l$. Processes such as amplification, transmission, filtering, delaying, recording, and sampling are easier to do at lower frequencies. Figure 2 shows a diagram of the frequency conversion process.

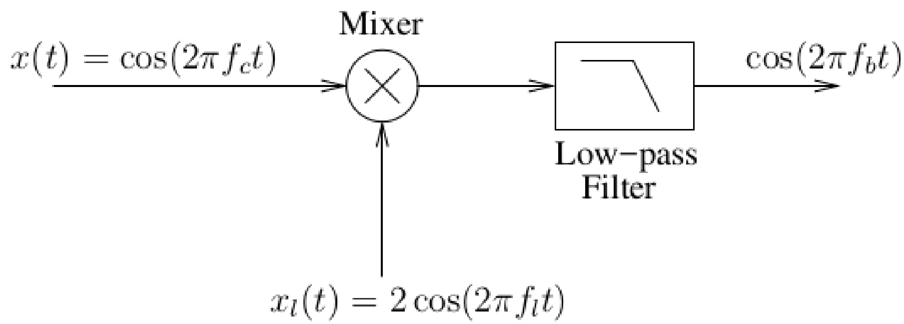


Figure 2: A single-stage frequency conversion (mixing) operation.

Assume that some oscillator with a perfect clock produces a pure sinusoid of the form $x(t) = cos(2\pi f_c t)$. Suppose that you receive this signal and make a measurement $f_m$ of the signal's frequency by (1) performing frequency conversion to translate the signal to a nominal center frequency $f_{b,nom} = 0Hz$ and then (2) sampling the signal with nominal sampling interval $\Delta t$ and observing the number of samples per period. Assume there is no motion between your receiver and the signal transmitter. Suppose that the local clock you're using to perform frequency conversion and sampling has a fractional frequency error of $\Delta f/f < 0$.

Derive an expression for the apparent Doppler $f_D$ resulting from frequency conversion and sampling with a clock that has this fractional frequency error. Express your result in terms of $f_c$ and $\Delta f/f$. How does this expression differ from the one derived in problem 3?

Note that if $f_l < f_c$, as will be the case for our problem because $\Delta f/f < 0$, then $f_b > 0$. This is an example of low-side mixing. But if $f_l > f_c$, then $f_b < 0$; this is high-side mixing. What would be the meaning of a signal for which $f_b < 0$? Upon sampling a signal with $f_b < 0$ to determine the period, would we measure a negative frequency?

## 2.2 Result

$$x(t) * x_l(t) = 2cos(2\pi f_c t)cos(2\pi f_l t)) \tag{3}$$

$$x(t) * x_l(t) = cos(2\pi(f_c - f_l)t) + cos(2\pi(f_c + f_l)t) \tag{4}$$

After applying the low-pass filter we get

$$x_{LPF}(t) = cos(2\pi(f_c - f_l)t) \tag{5}$$

Let's clarify:

$$f_b = f_c - f_l = f_c - (f_{l,nom} + \Delta f) \tag{6}$$

We know that $f_{b,nom} = 0Hz$. Therefore, $f_{l,nom} = f_c$.

Now, it's necessary to focus on the last part of the processing (sampling). For that, lets write the following relationships.

$$\frac{\Delta f}{f_{l,nom}} = \frac{f_l - f_{l,nom}}{f_{l,nom}} = \frac{f_l}{f_{l,nom}} - 1 = \frac{\Delta t_{l,nom}}{\Delta t_l} - 1 \tag{7}$$

$$\Delta t_{l,nom} = (\frac{\Delta f}{f_{l,nom}} + 1)\Delta t_l \tag{8}$$

Since the objective at this stage is to measure the frequency of the signal that comes from the low-pass filter, it is safe to say that we are trying to measure $f_b$. Thus, $f_b = \frac{1}{N*\Delta t_l}$.

$$f_m = \frac{1}{N * \Delta t_{l,nom}} = \frac{1}{N * (\frac{\Delta f}{f_{l,nom}} + 1)\Delta t_l} = \frac{f_b}{\frac{\Delta f}{f_{l,nom}} + 1} \tag{9}$$

Since it was stablished that $f_{l,nom} = f_c$. Then, $\Delta f = f_c \frac{\Delta f}{f_{l,nom}}$

$$f_m = \frac{\frac{\Delta f}{f_{l,nom}} f_c}{\frac{\Delta f}{f_{l,nom}} + 1} \tag{10}$$

Finally, it is possible to reason that the measured frequency is the very apparent Doppler $f_D$ (with a changed sign). This is because the frequency conversion already applied the following substraction $f_b = f_c - f_l$, which is almost the same substraction needed for $f_D = f_l - f_c$.

$$f_D = -\frac{\frac{\Delta f}{f_{l,nom}} f_c}{\frac{\Delta f}{f_{l,nom}} + 1} \tag{11}$$