Garmin Challenge 2021.md

Garmin Challenge 2021

by Nkateko Senoamadi

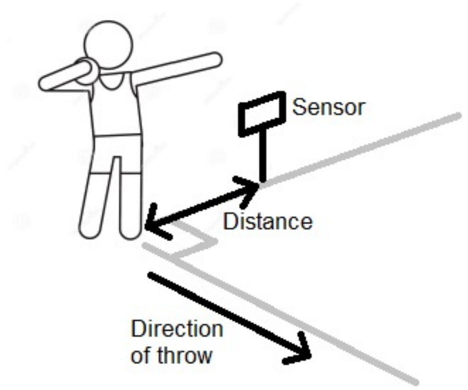
Introduction

The following document outlines my approach to the Garmin Stellenbosch Challenge 2021.

The Problem

During a hackathon at Garmin Stellenbosch, you and your teammates are experimenting with sonic sensors for athletic events like shot-put and javelin. You've managed to put together a system that can transmit a sound signal at the projectile being thrown, and record the return signal that reflects off it.

Attached are the transmitted signal that is used for all projectiles, and the received signals for a shot-put and a javelin throw. Can you calculate the speed at which each projectile was thrown?



Given:

- Transmit_1.wav (audio file)
- Shotput_receive_1.wav (audio file)
- Javelin_receive_1.wav (audio file)

My Approach

Determining speed

What we are looking for is the speeds of a shot-put & a javelin given the audio files mentioned above. The solution involves manipulating the Doppler Effect equation in terms of the velocity of the projectile(either shot-put or javelin). The Doppler effect is the change in frequency of a wave in relation to an observer who is moving relative to the wave source. The general formula is as follows:

$$f_o = \frac{(v \pm v_o)}{(v \mp v_s)} \cdot f_s$$

Where:

$*f_o = observer frequency of sound$

$*v_o = observer\ velocity$

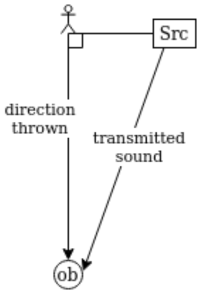
$*v = speed of soundwaves$

$*f_s = actual frequency of sound$

$*v_s = source\ velocity$

We need to identify the **observer** and the **source** but in order to that we must separate the problem into two parts path 1 and path 2. Path 1 is the path the sound travels from the sensor to the projectile and path 2 is the path the echoed sound travels from the projectile to the sensor:

Path 1:



The sound waves leave the sensor and hit the projectile. In this case we have a stationery source and an observer moving away from the source. The modified formula is as follows:

$$f_{o_1} = \frac{v - v_{proj}}{v} \cdot f_{tr}$$

Where:

$*f_o = f_{o1}$

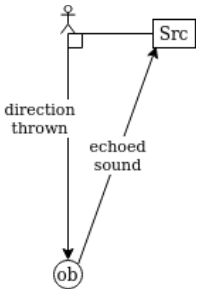
$*v_o = v_{proj}\ velocity\ of\ projectile$

$*v = speed of soundwaves$

$*f_s = f_{tr}\ transmitted$

$*v_s = 0$

Path 2:



The sound waves echos off the projectile and hits the sensor. In this case we have a stationery observer (the sensor) and source that's moving away from the observer. The modified formula is as follows:

$$f_{recv} = \frac{v}{v + v_{proj}} \cdot f_{o1}$$

Where:

$*f_o = f_{recv}\ frequency\ of\ audio\ received$

$*v_o = 0$

$*v = speed of soundwaves$

$*f_s = f_{o1} = \frac{(v - v_{proj})}{v} \cdot f_{tr}$

$*v_s = v_{proj} = velocity\ of\ projectile$

We then substitute and simplify a bit as seen below:

$$f_{recv} = \frac{v}{v + v_{proj}} \cdot \frac{v - v_{proj}}{v} \cdot f_{tr}$$

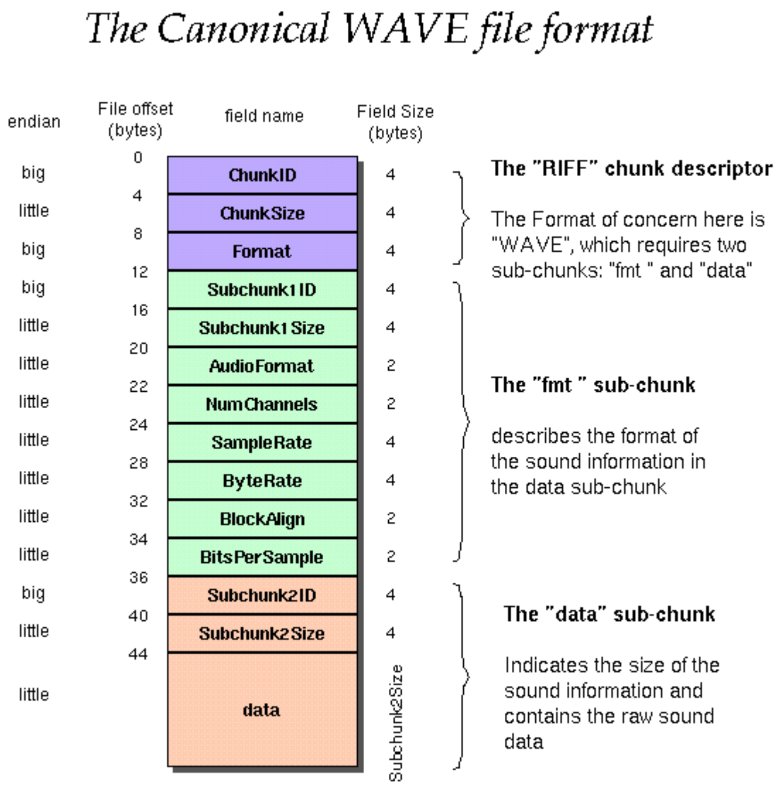
$$f_{recv} = \frac{v - v_{proj}}{v + v_{proj}} \cdot f_{tr}$$

From this point we make the velocity of the projectile the subject of the formula and we get the following:

$$v_{proj} = v \cdot \frac{(1 - (\frac{f_{recv}}{f_{tr}}))}{(1 + (\frac{f_{recv}}{f_{tr}}))}$$

Processing audio files

Given the derivation it is clear that we must find the the frequency of the transmitted frequency and the received frequency for each projectile. To do this we will use a Discrete Fourier Transform (DFT) to help us identify the frequencies of the audio files. However, before we can do that we need to process the header of the audio file as seen in the figure below.



The data is then processed using C++ paying close attention to the changing endians (with the help of online resources[1]) and the data segment of each audio file was stored in a text file as sample points separated by newline characters. The output below shows the data extracted from the headers of each audio file.

```
Chunk ID: RIFF
Chunk Size:100038
format: WAVE
1st subchunk ID: fmt
1st Subchunk size: 16
Format type: PCM
Channels: 1
Sample rate: 100000
Byte rate: 200000
Block align: 2
Bits per sample: 16
2nd subchunk ID: data
2nd subchunk size: 100002
Approx.Duration in seconds=0.50019
Error reading file. 0 bytes
```

Getting Frequencies

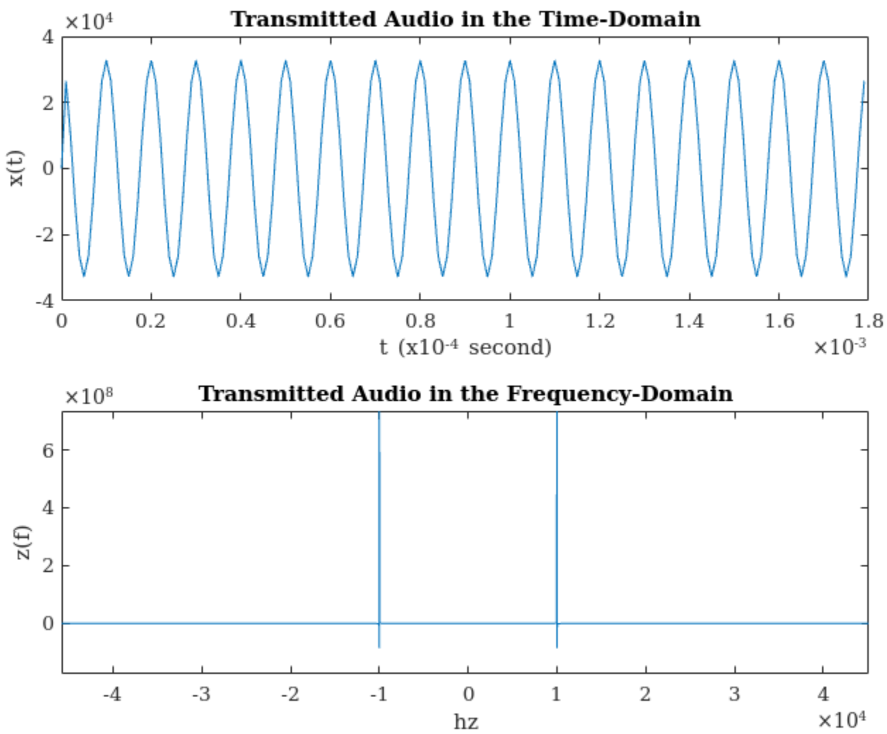
The aim is to represent function in the time domain and as functions frequency domain. This is done with the Discrete Fourier Transform (DFT) the data is loaded onto MATLAB and for the time domain its processed as follows:

```
% load processes .wav
files = dir('*.txt');
for i=1:length(files)
    eval(['load ' files(i).name ' -ascii']);
end
% Transmitted Audio time
x_tr = Transmit_1;         % Data sample points amplitudes
fs_tr = 100000;            % sample frequency/ sample rate
ts_tr = 1/fs_tr;           % time step
N_tr  = 50001;             % number of smaples
tmax_tr = (N_tr-1)*ts_tr;% max time (duration of audio)
t_tr = 0:ts_tr:tmax_tr;  % time points for each data point

```

Using the function of the data *x(t)* (a function of t) we use the Fourier Transform so that i could be represented as *z(f)* (a function of frequency) .

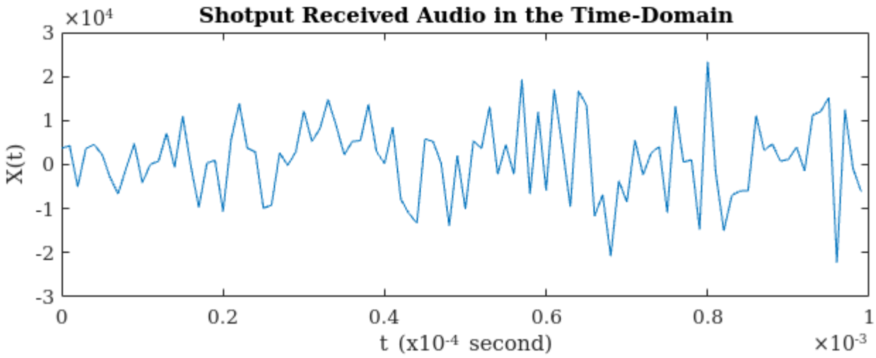
```
% Transmitted Audio frequency
f_tr= -fs_tr/2:fs_tr/(N_tr-1):fs_tr/2; % frequncy steps
z_tr= fftshift(fft(x_tr));             % amplitude

```

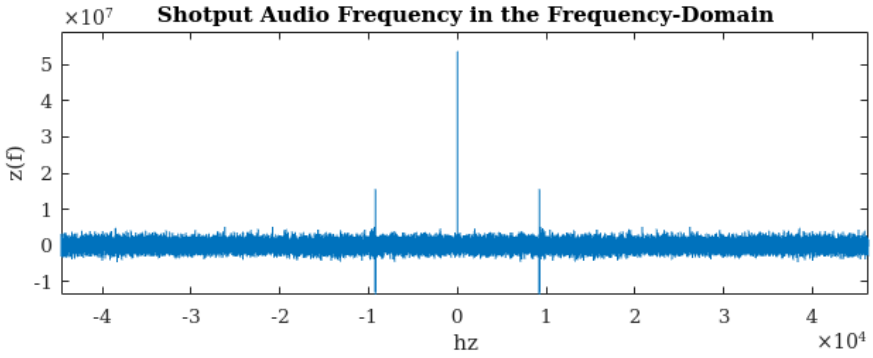
Repeating this for all audio the audio files and graphing *x(t)* and *z(f)* for the Transmit_1 , Shotput_receive_1 and Javelin_receive_1.

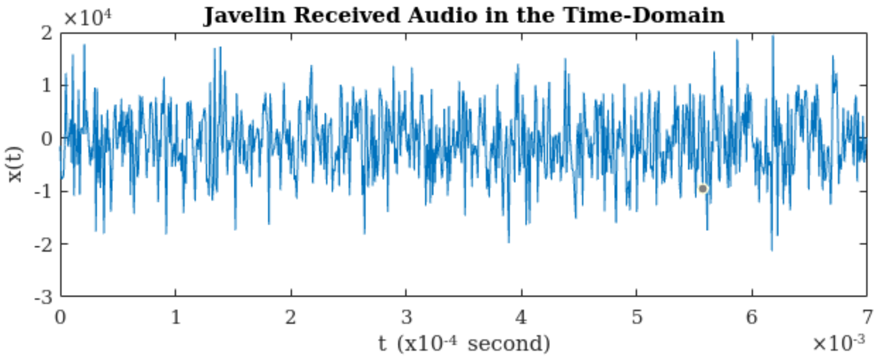


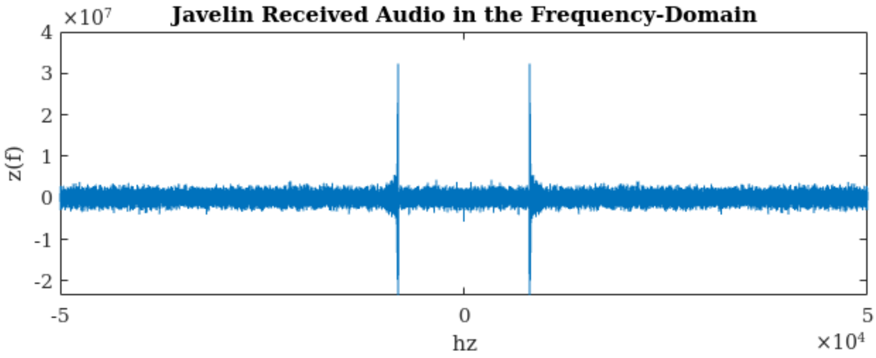
The frequency of the transmitted file is found through getting the absolute value of the frequency with the largest amplitude (*z(f)* value).

We find that the transmitted frequency is 10 000 Hz.









Similarly, we can find the the frequencies of the received audios for each projectile.

We find the received frequency for the shot-put and javelin is 8132 Hz and 9212 Hz respectively.

Applying the formula (derived earlier) in MATLAB we find the speeds of each projectile **relative to the sensor**. Taking the speed of sound as 343 m/s.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
v = 343;
ratio_frequencies_sh = F_obs_sh/F_act_tr;
ratio_frequencies_ja = F_obs_ja/F_act_tr;

V_sh = v *((1-ratio_frequencies_sh)/(1+ratio_frequencies_sh));

V_ja = v *((1-ratio_frequencies_ja)/(1+ratio_frequencies_ja));
```

Which gives us the following speeds relative to the sensor :

- *Speed of the shot-put = 14.07 m/s*
- *Speed of the javelin= 35.34 m/s*

Bibliography

1. File reading : http://truelogic.org/wordpress/2015/09/04/parsing-a-wav-file-in-c/