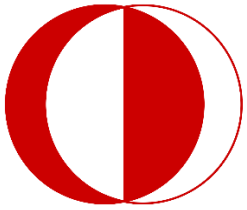

EE430 Digital Signal Processing

Project Part-2

Berkay Yaldız-2232940

Uğur Demirörs-2231819

Linear Chirp Pulse as the Emitted Signal

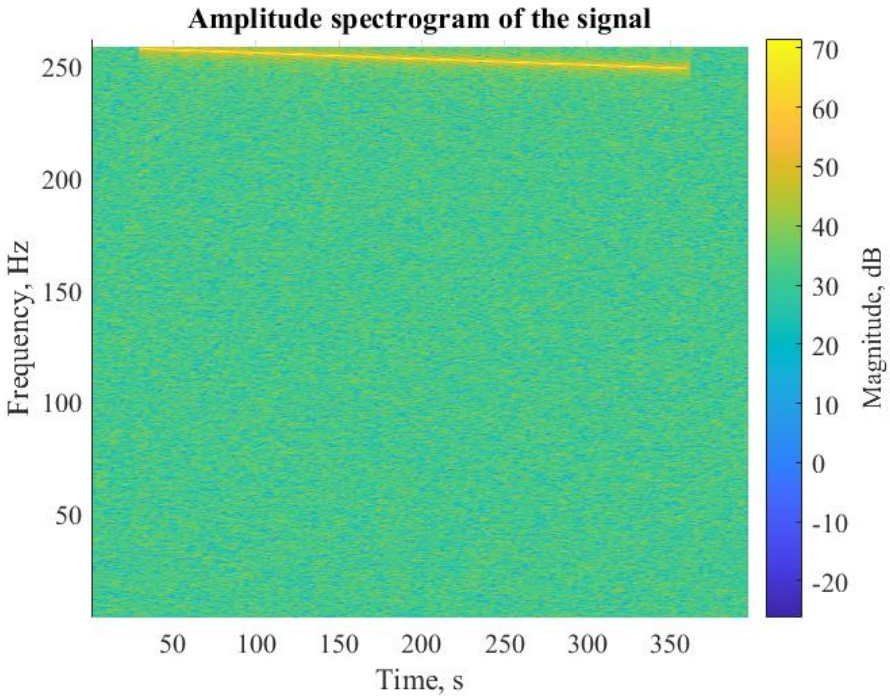

Figure 1.Spectrogram of the emittid signal with frequency 300 Hz.

Windows size: 2000 (Rect(2000))

Overlap : 500

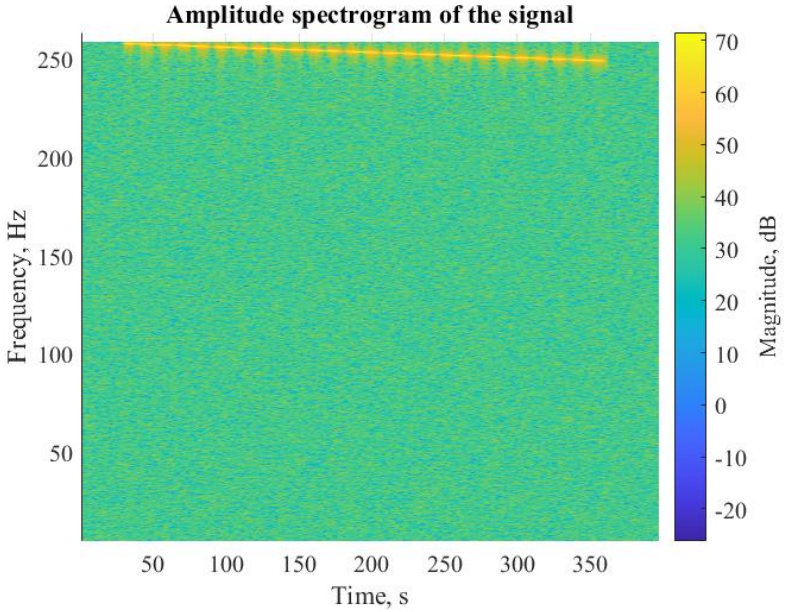

Figure 2. Spectrogram of the emittid signal with frequency 300 Hz.

Windows size: 1500 (Rect(1500))

Overlap: 500

Hence, it can be seen from the Figure 1 and Figure 2, increasing window length increases the resolution and the quality of detection.

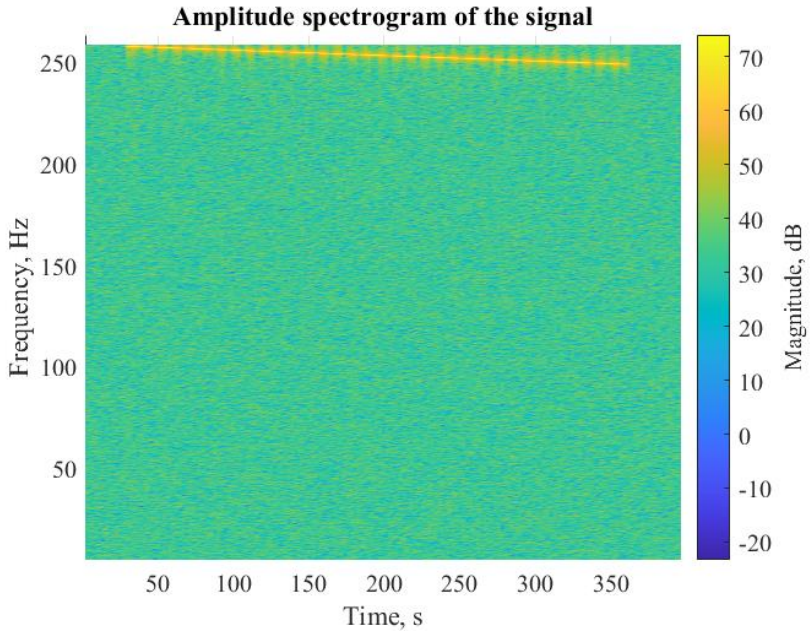

Figure 3.Spectrogram of the emittid signal with frequency 300 Hz.

Window size: 2000 (Rect(2000))

Overlap: 800

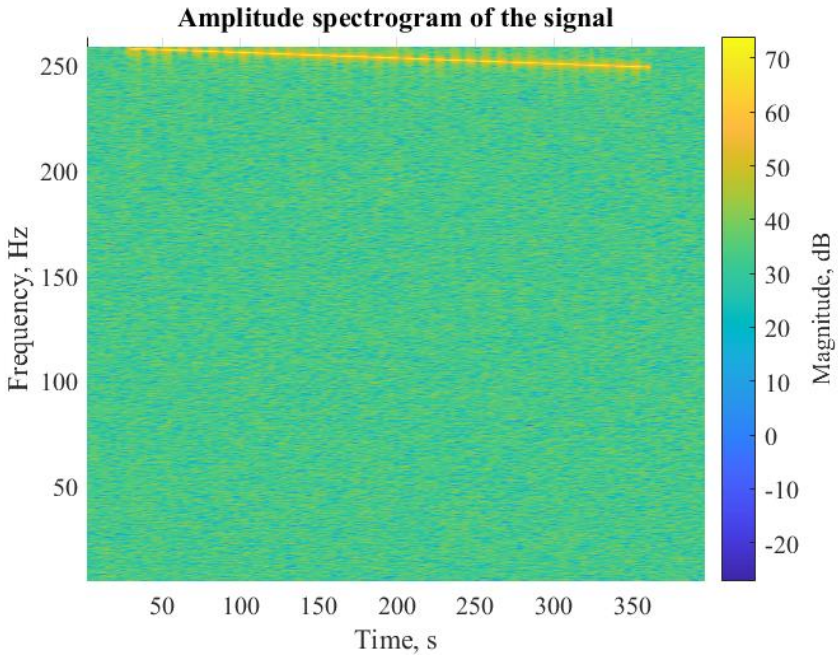

Figure 4. Spectrogram of the emittid signal with frequency 300 Hz.

Window size: 2000 (Rect(2000))

Overlap: 300

Hence, the overlap length and the quality of the spectrum output are proportional. When we inrease the overlap length, the resolution and quality increases.

Linear Chirp Pulse as the Emitted Signal

In order to determine the range of the source and the speed of the source, we decided to design a system which detects the signal with Gaussian noise and determine the frequency of the emitted signal by analyzing the spectrogram of the signal and the STFT values. With this frequency, we can determine the speed of the source. Also, since we know the speed of the signal and the source, we can determine the range using the time delay and the speed of the signal. In order to find the frequency of the detected signal, we take the mean value of the STFT results for each time index. After comparing these mean values, we take the maximum one. Then, we extract the frequency of the maximum STFT mean value. After obtaining the frequency, we used the Doppler formula to find the speed of the source. Furthermore, since we determine the speed of the source, we used the time delay to find the range.

Firstly, to test the system, we needed to determine the range of values Δ,Ω_0,m,t_0,v. We thought that six values for each range are enough to test our system. After some research, for the speed of the source, we decided to choose the regular, basic speed values , so we decided the source velocity range as [40 50 60 70 80 90] km/s. Also, the range for the m values was set as [10 20 30 40 50 60] since as m values increases, the signal becomes more immuned to the noise but its bandwidth increases so it is a trade off and the time duration Δ was set to 400. Moreover, to test the system with different time delays, we set ranges for the n and k values in the equation $t_0=(n+k*0.1)*Ts$. n values are chosen bigger than zero, so the t_0 values can take multiples of sampling period. Furthermore, the range of the emitted signal's frequency is determined as [250 500 750 900 1100 1400] Hz. These values have been chosen, because the frequency of the signals emitted by cars on the strees is around 300 Hz.

To present the performance of our method, we firstly generated the emitted signals with different variables. Then, we analyzed the spectrogram and the STFT values of the

emitted signal. Then, comparing the signal frequency and the detected frequency, we could find the speed of the source using Doppler equation seen in Figure 5.

$$f' = \frac{(v + v_o)}{(v - v_s)} f$$

f' = observed frequency
f = actual frequency
v = velocity of sound waves
v_o = velocity of the observer
v_s = velocity of the source

Figure 5. Dopple Frequency Formula

To find the range:

[c+v(speed of the source)]*t_0(time delay) is used.

Test Results:

For all signals, sampling frequency adjusted as Nyquist Rate Fs = c/(c-v) * 2 * f_o.

Also, to obtain the spectrogram, we used the code that we designed for the Project Part-1. The detail about the codes can be found in Appendix.

First signal:

Frequency of the emitted signal = 250 Hz

Speed of the source = 40 km/h

The spectrogram seen in FigureX, the detected signal's frequency is 252.5 Hz. According to the Doppler formula, the frequency should be 258 Hz which is very close to our value.

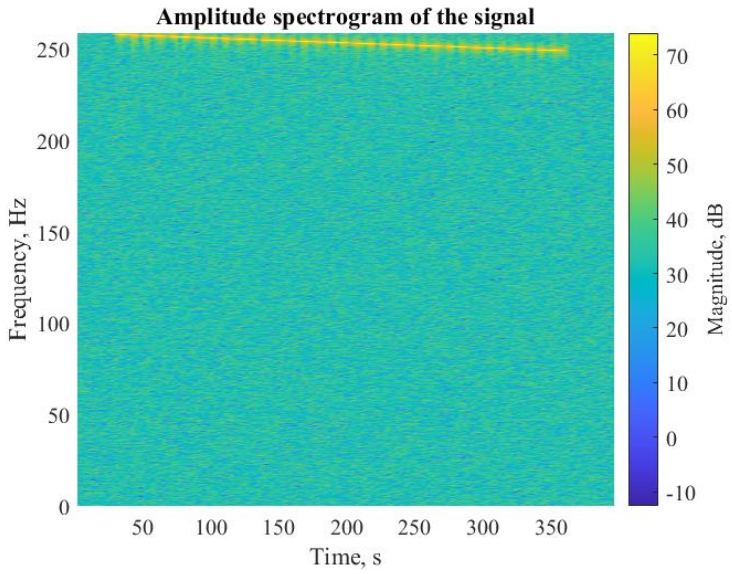

Figure 6: Spectrogram of the first signal

Speed estimation error = -7.7088 m/s

Range estimation error = -96.9407 m

Second signal:

Frequency of the emitted signal= 500 Hz

Speed of the source = 50 km/h

The spectrogram seen in FigureX, the detected signal's frequency is 509.8 Hz. According to the Doppler formula, the frequency should be 521 Hz which is very close to our value.

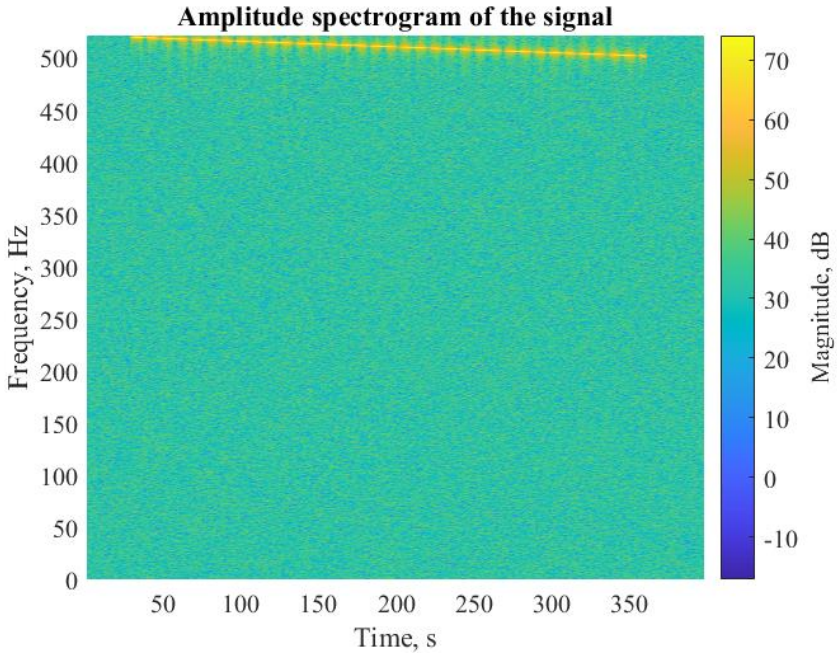

Figure 7:The spectrogram of the second signal

Speed estimation error = -7.207 m/s

Range estimation error = 308.3148 m

Third signal:

Frequency of the emitted signal = 750 Hz

Speed of the source = 60 km/h

The spectrogram seen in FigureX, the detected signal's frequency is 772 Hz. According to the Doppler formula, the frequency should be 788 Hz which is very close to our value.

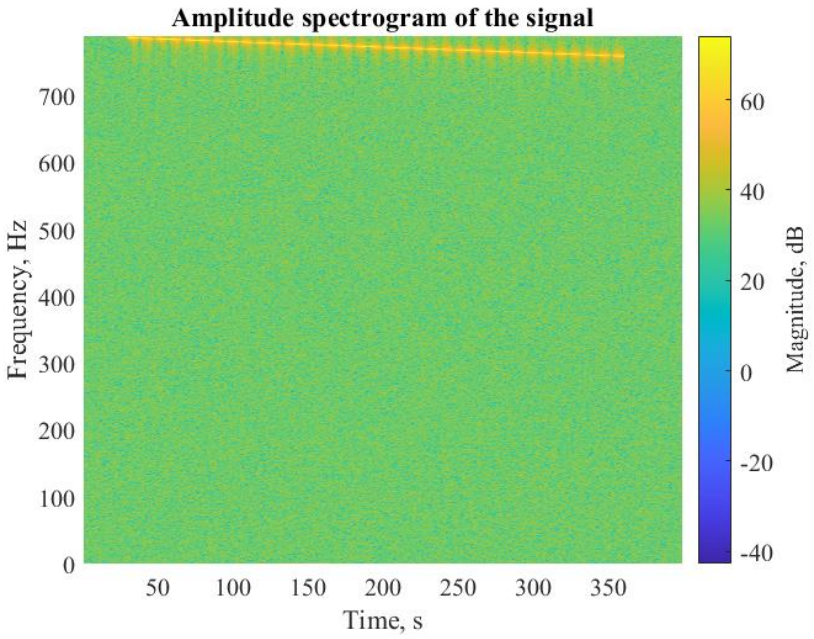

Figure 8:The spectrogram of the third signal

Speed estimation error = -6.6489 m/s

Range estimation error = 159.3005 m

Fourth signal:

Frequency = 900 Hz

Speed of the source = 70 km/h

The spectrogram seen in FigureX, the detected signal's frequency is 935.5 Hz. According to the Doppler formula, the frequency should be 954 Hz, so the difference is just 19 Hz.

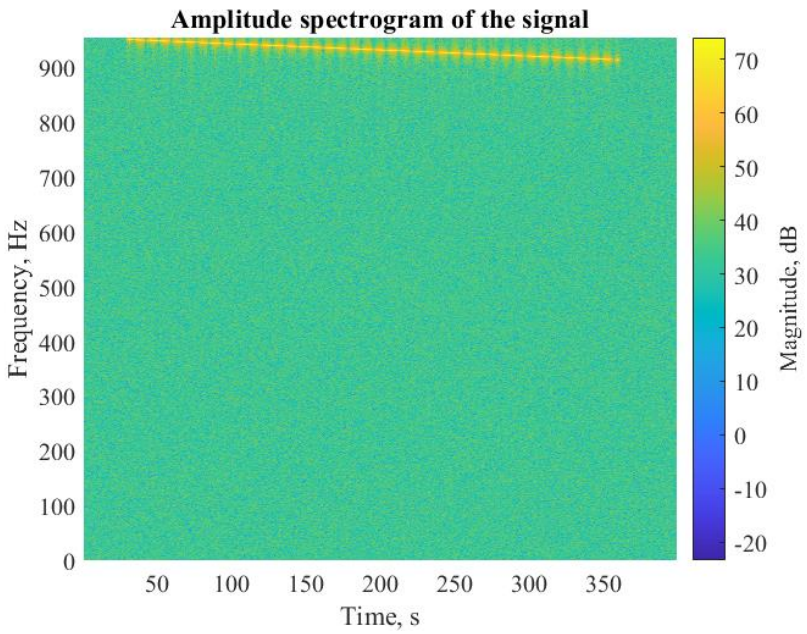

Figure 9: The spectrogram of the fourth signal

Speed estimation error = -6.033 m/s

Range estimation error = 46.7828 m

Fifth signal:

Frequency = 1.1 kHz

Speed of the source = 80 km/h

The spectrogram seen in FigureX, the detected signal's frequency is 1151 Hz. According to the Doppler formula, the frequency should be 1176 Hz, so the difference is just 25 Hz.

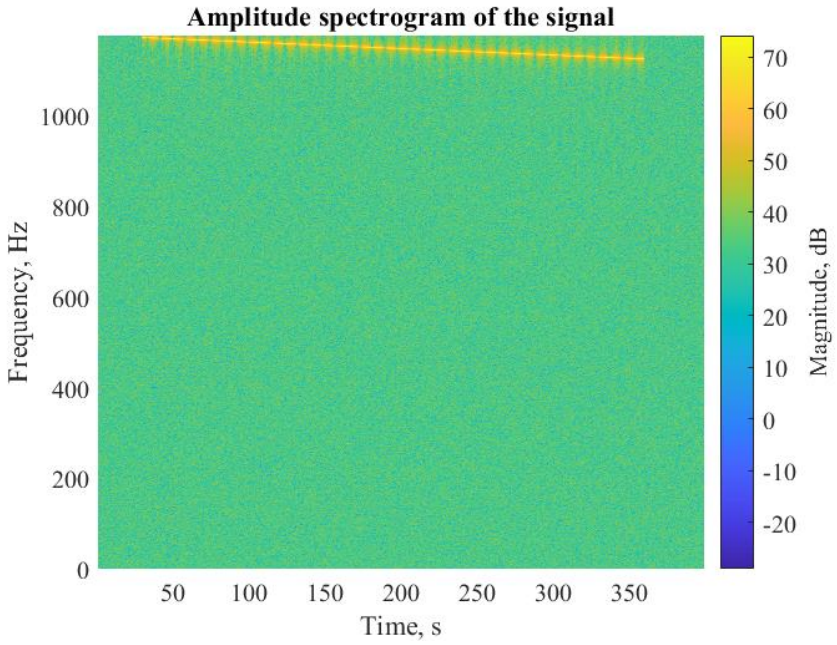

Figure 10: The spectrogram of the fifth signal

Speed estimation error = -6.4491 m/s

Range estimation error = 286.2718 m

Sixth signal:

Frequency = 1.4 kHz

Speed of the source = 90 km/h

The spectrogram seen in FigureX, the detected signal's frequency is 1476 Hz. According to the Doppler formula, the frequency should be 1511 Hz, so the difference is 35 Hz.

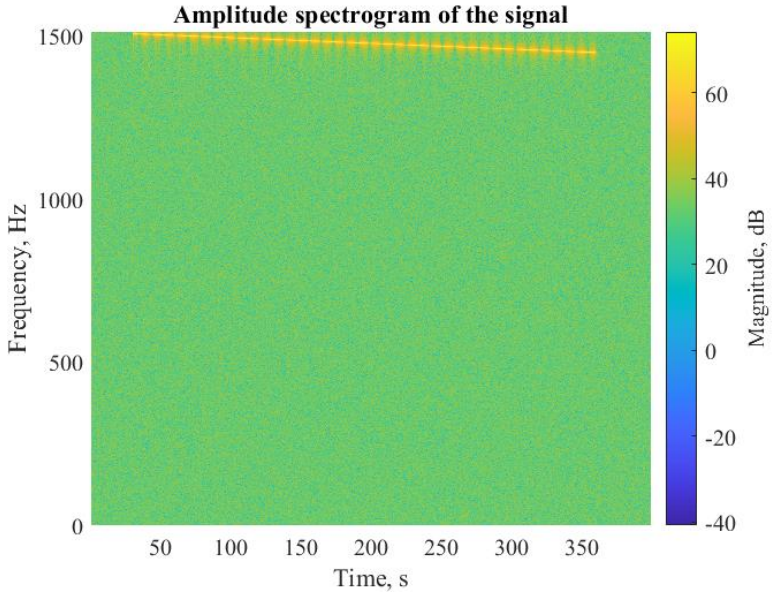

Figure 11.The spectrogram of the fifth signal

Speed estimation error = -6.4565 m/s

Range estimation error = 154.0695 m

Sinusoidal Pulse as the Emitted Signal

The method, procedure and the values are the same as the linear chirp part.

The first signal:

Frequency = 250 kHz

Speed of the source = 40 km/h

The spectrogram seen in FigureX, the detected signal's frequency is 258.4 Hz. According to the Doppler formula, the frequency should be 258.44 Hz, so they can be thought as equal. We zoomed on the spectrogram to see the detected signal better.

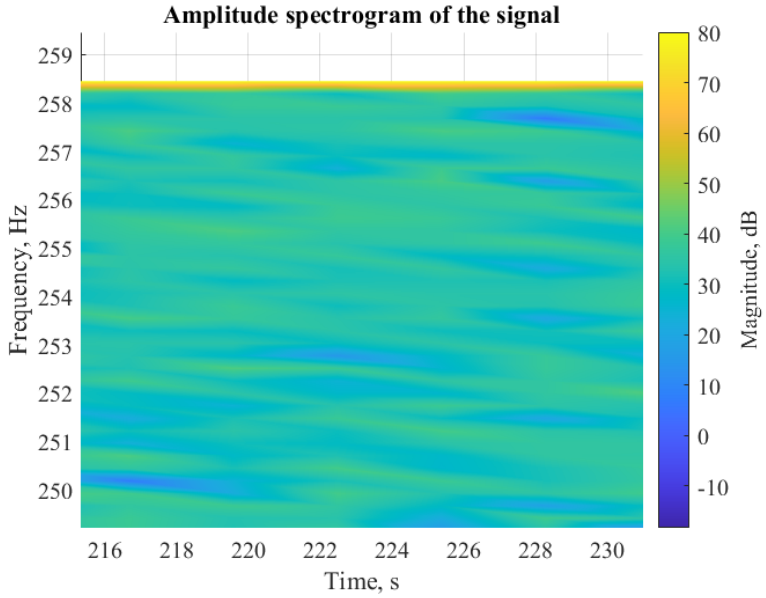

Figure 12. Obtained spectrogram of the sinusoidal signal with f=250 Hz.

Speed estimation error = 0.37538 m/s (Since the detected frequency is almost equal to the signal frequency, the speed error is too small.)

Range estimation error = -347.1807 m

The second signal:

Frequency = 500 kHz

Speed of the source = 50 km/h

The spectrogram seen in Figure 13, the detected signal's frequency is 521.29 Hz. According to the Doppler formula, the frequency should be 521.29 Hz, so they can be thought as equal. We zoomed on the spectrogram to see the detected signal better.

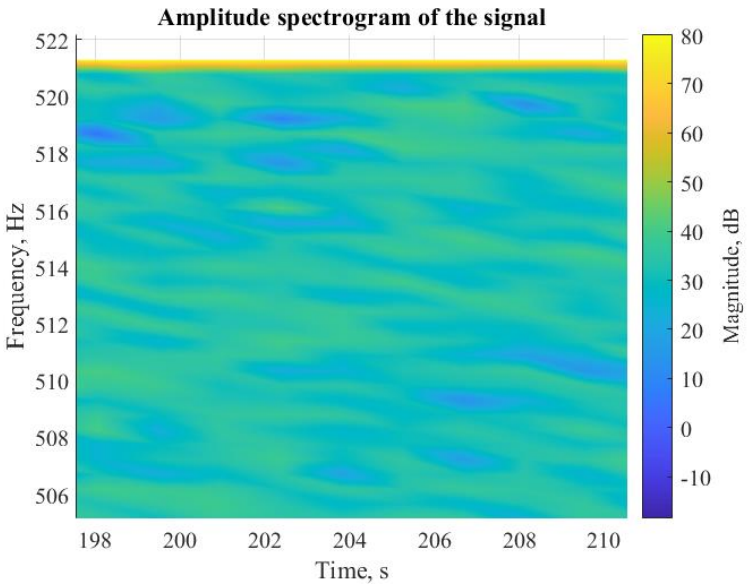

Figure 13

Speed estimation error = 0.59152 m/s

Range estimation error = 76.4348 m

The third signal:

Frequency = 750 Hz

Speed of the source = 60 km/h

The spectrogram seen in Figure 13, the detected signal's frequency is 788.65 Hz. According to the Doppler formula, the frequency should be 788.65 Hz, so they can be thought as equal. We zoomed on the spectrogram to see the detected signal better.

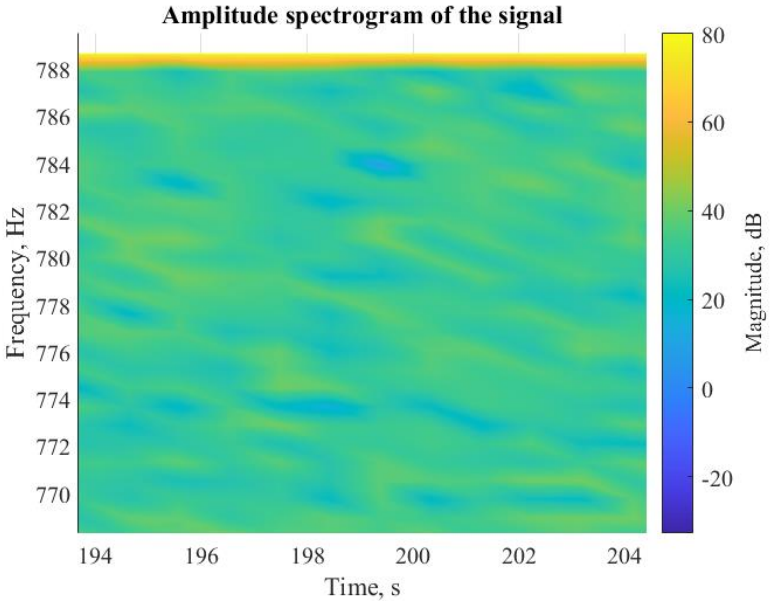

Figure 14

Speed estimation error = 0.85911 m/s

Range estimation error = -66.7995 m

The fourth signal:

Frequency = 900 Hz

Speed of the source = 70 km/h

The spectrogram seen in Figure 13, the detected signal's frequency is 954.59 Hz. According to the Doppler formula, the frequency should be 954.59 Hz, so they can be thought as equal. We zoomed on the spectrogram to see the detected signal better.

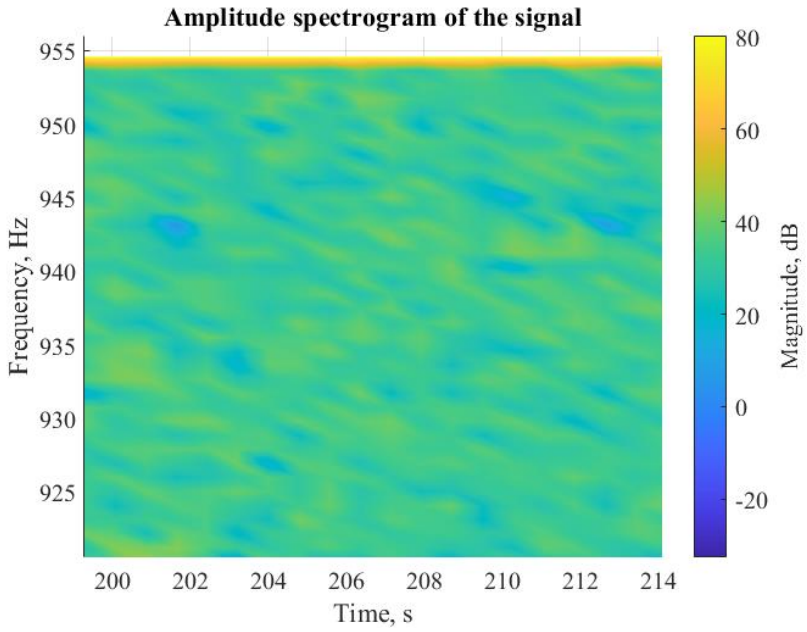

Figure 15

Speed estimation error = 1.1795 m/s

Range estimation error = 111.0051 m

The fifth signal:

Frequency = 1.1 kHz

Speed of the source = 80 km/h

The spectrogram seen in Figure 13, the detected signal's frequency is 1176.9 Hz. According to the Doppler formula, the frequency should be 1176.9 Hz, so they can be thought as equal. We zoomed on the spectrogram to see the detected signal better.

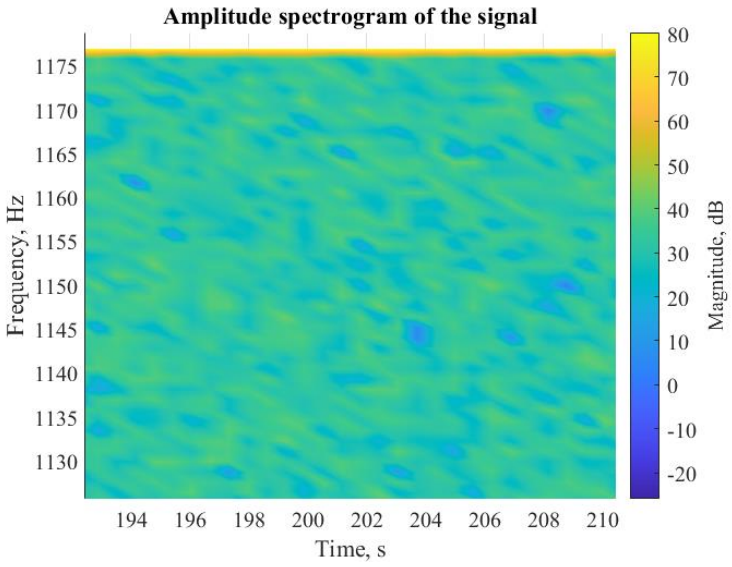

Figure 16

Speed estimation error = 1.554 m/s

Range estimation error = 48.2718 m

The sixth signal:

Frequency = 1.4 kHz

Speed of the source = 90 km/h

The spectrogram seen in Figure 13, the detected signal's frequency is 1511.1 Hz. According to the Doppler formula, the frequency should be 1511.1 Hz, so they can be thought as equal. We zoomed on the spectrogram to see the detected signal better.

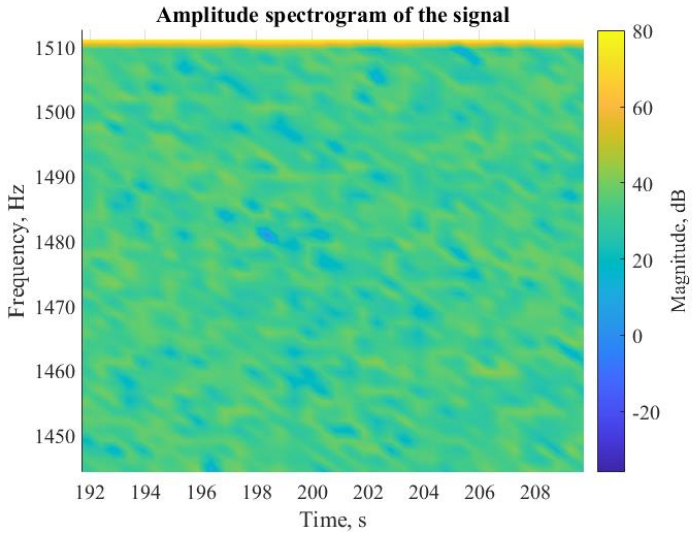

Figure 17

Speed estimation error = 1.9841 m/s

Range estimation error = -100.0805 m

As a result, our system works really good for sinusoidal signal, but for linear chirp signal errors in range and speed estimation can be important. The main reason is that frequency of the sinusoidal signal is constant through out time but frequency of the linear chirp signal is varying.

Appendix

Our code for the range and speed estimation.

```
clear

close all
N = 6; % Monte carlo trial number
c = 340; % Signal speed.
u = 10/36; % km/h to m/s convertion constant
v = [40 50 60 70 80 90] *u; % Source speed values. (Average car speed values.)
delta  = 400; % delta value ( maximum time value, so max range is 340*2000 = 680km)
f_o = [250 500 750 900 1100 1400]; % Frequency values, baseband signals (low
frequency)
% We inspected very low frequencies because of the performance and ram
% issues.
doppler_multiplier = c ./ (c-v);
Fs =2* f_o .*doppler_multiplier; % Fs equals to Nyquist rate of Doppler shifted
f_o.
Ts = 1./Fs;
n = [ 4 5 6 7 8 9]; % n values for t_0
k = [0 1 2 3 4 5]; % k values for t_0
A =5;
m = [ 10 20 30 40 50 60]; % m values. In FM modulation when beta value(proportional
with m) increases, signal becomes more
% noise immuned, but its bandwidth increases (which is bad, it is a trade-off).
first_index = zeros(2*N,1);
last_index = zeros(2*N,1);
real_range = zeros(N,1);
s_r = cell(6,1);
s_r_sin = cell(6,1);
disp('First six signals are linear chirp and last six signals are sin')
for ii=1:N
t = (0:1/Fs(ii):delta-1/Fs(ii)).'; % Time vector
t_0 = 30; % Shifting amount
t_last =360; % range is between t_last and t_0
first_index(ii) = round(t_0*Fs(ii));
last_index(ii) = round(t_last*Fs(ii));
real_range(ii) = (c+v(ii) )* t(first_index(ii)) ; % Determine the real range
noise = 1.2*randn(length(t),1);
```

```
s_r{ii} = [zeros(first_index(ii)-1,1);A * cos( doppler_multiplier(ii)* 2 * pi  *(
f_o(ii) * t(first_index(ii):last_index(ii)) + (doppler_multiplier(ii)*m(ii) ./ (2 *
delta) * (t(first_index(ii):last_index(ii)).^2) ) ) ) ;zeros(length(t)-
last_index(ii),1)] + noise;%Linear chirp signal
s_r_sin{ii} = [zeros(first_index(ii)-1,1);A * cos( doppler_multiplier(ii)* 2 * pi
*( f_o(ii) * t(first_index(ii):last_index(ii))  ) ) ;zeros(length(t)-
last_index(ii),1)] + noise; % Sin signal
end
all_datas = [s_r; s_r_sin]; % This is for estimation only in single for loop.
Calculation_Fs = [Fs Fs]; % We used same Fs values for sin and linear chirp
signals, so Fs values are same.
Calculation_f_o = [f_o f_o];
Calculation_v = [v v];
Calculation_real_range = [ real_range; real_range];
% Part II
PSD_matrix= cell(2*N,1); % To calculate maximum values stft at each frequency
Freq_axis_vector = cell(2*N,1);
Time_axis_vector = cell(2*N,1);
freq_index = zeros(2*N,1);
time_index1 = zeros(2*N,1); % Two indexes for each signal.
estimated_range = zeros(2*N,1);
f_output = zeros(2*N,1);
extracted_speed = zeros(2*N,1);
for ii=1:2*N % We have 2*N signals (sin and linear chirp)
    figure
    [PSD_matrix{ii},Freq_axis_vector{ii},Time_axis_vector{ii}] =
myspectrogram1(all_datas{ii},Calculation_Fs(ii),rectwin(2000),2000,500);
    temp_psd_vector = max(20*log10(abs(PSD_matrix{ii}))); % Find maksimum
frequencies at each time instant
    %and determine starting and ending indexes for the signal to  find the range.
    temp_freq_axis_vector = Freq_axis_vector{ii};
    temp_time_vector = Time_axis_vector{ii};
    temp = 0;
    tt = 1;
    while true
        difference = temp_psd_vector(tt+2)- temp_psd_vector(tt);
        if difference >17.2 % We determine starting and ending time indexes
according to psd value difference in the vector.
            time_index1(ii) = tt+2;
            break;
        end
        tt = tt+1;
    end
    [~,freq_index(ii)] = max(mean(abs(PSD_matrix{ii}),2));
    f_output(ii) = temp_freq_axis_vector(freq_index(ii));
```

```
    extracted_speed(ii) = c*(f_output(ii)-
Calculation_f_o(ii))./Calculation_f_o(ii);
    % Error between estimated speed and real speed.
    disp(['Speed estimation error for ', num2str(ii), 'th signal = ',
num2str(extracted_speed(ii)-Calculation_v(ii) ),' m/s' ])
    estimated_range(ii) = (c+extracted_speed(ii))
*temp_time_vector(time_index1(ii)); % Range of signals.
    disp(['Range estimation error for ', num2str(ii), 'th signal = ', num2str(
Calculation_real_range(ii)-estimated_range(ii) ), ' m'] )
end

% This part is with different window length and overlap length calculations.
figure
myspectrogram1(all_datas{1},Calculation_Fs(1),rectwin(1500),1500,500); % Less
length

figure
myspectrogram1(all_datas{1},Calculation_Fs(1),rectwin(1500),2000,500); % More
length

figure
myspectrogram1(all_datas{1},Calculation_Fs(1),rectwin(2000),2000,300); % Less
overlap length

figure
myspectrogram1(all_datas{1},Calculation_Fs(1),rectwin(2000),2000,800); % More
overlap length
```

Our code for the spectrogram.

```
function [S1,f,t]= myspectrogram1(signal, fs, window, length,overlap)

%win=@window;

[S1, f,
t]=stft(signal,fs,'Window',window,'OverlapLength',overlap,'FFTLength',length);

Row=ceil(numel(f)/2);
f=f(Row:end,1);%Take positive frequencies like spectrogram.
S1=S1(Row:end ,:);
surf(t, f, 20*log10(abs(S1)))
shading interp
axis tight
view(0, 90)
```

```
set(gca, 'FontName', 'Times New Roman', 'FontSize', 12)
xlabel('Time, s')
ylabel('Frequency, Hz')
title('Amplitude spectrogram of the signal')
hcol = colorbar;
set(hcol, 'FontName', 'Times New Roman', 'FontSize', 12)
ylabel(hcol, 'Magnitude, dB')

end
```