Name: Josef Ullrich
Class: ECE148
Instructor: Hua Lee

Assignment Seven

Introduction

In this assignment, I will analyze how two acoustic receivers can identify the angle at which a signal hits them, and how to reverse engineer this angle by precisely choosing the right delay between the two receivers. I will then look at how ground penetrating radar can be used to scan the subsurface of the ground. To do so I will use step-frequency FMCW radar data, and take the inverse FFT to obtain depth profiles. Throughout the entire assignment I am manipulating time and frequency data to obtain useful physical location data.

Part A

For this part, I am using two acoustic receiver signals to estimate the direction that the sound is coming from. Since the two receivers are separated by a known distance, the sound will usually reach one receiver slightly before the other. This creates a small time delay between the two signals. By estimating this time delay, I can then estimate the bearing angle of the acoustic source.

This part was particularly interesting to me because in some ISAC (Integrated Sensing and Communication) research I'm doing, a critical component is to estimate angle of arrival of incoming rays. We do this by looking at the spatial phase difference across antenna elements with a spatial FFT to find the strongest angle. Here, the idea is similar, but instead of using many antenna elements and phase differences, I will use two acoustic receivers and cross-correlation to estimate the time delay between them.

We are given the receiver spacing:

$$d = 2.0 \text{ m}$$

The speed of sound in air:

$$c = 343.6 \text{ m/s}$$

And the sampling frequency:

$$f_s = 48 \text{ kHz} = 48000 \text{ Hz}$$

First I read in the two signals, x1(n) and x2(n), and compute the cross-correlation between them. The cross-correlation tells me how similar the two signals are for different time shifts. The value where the cross-correlation is largest is the estimated delay between the two receiver signals, in terms of samples.

$$r_{12}(k) = \sum_n x_1(n)x_2(n-k)$$

The lag value k is in samples, so I convert it into seconds using the sampling frequency:

$$\tau = \frac{k}{f_s}$$

The maximum possible time delay is limited by the receiver spacing and the speed of sound:

$$\tau_{\max} = \frac{d}{c}$$

Substituting in the values:

$$\tau_{\max} = \frac{2.0}{343.6}$$

$$\tau_{\max} \approx 5.82 \times 10^{-3} \text{ s}$$

So the time-delay profile should only include delays between approximately:

$$-5.82 \text{ ms} \leq \tau \leq 5.82 \text{ ms}$$

The time-delay profile is shown below.

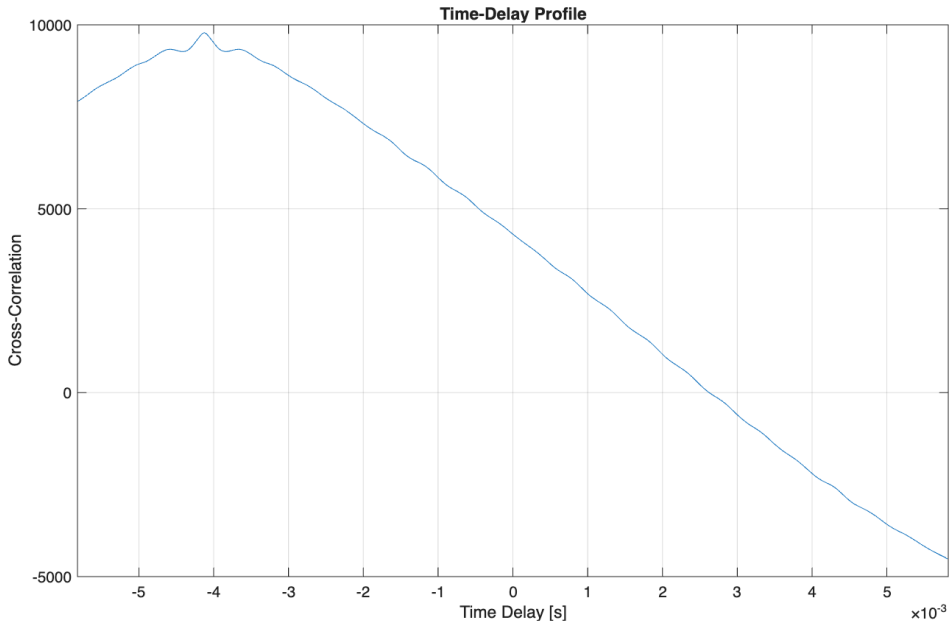

Figure 1: Time-delay profile found by cross-correlating the two acoustic receiver signals.

From the plot, the largest cross-correlation value occurs at approximately:

$$\tau \approx -4.1 \times 10^{-3} \text{ s}$$

This means that the two signals are best aligned when one signal is shifted by about -4.1 ms. Since the cross correlation peak occurs at a negative delay, that means that microphone 2 received the sound first, about 4.1 ms before microphone 1.

Next, I will convert this time-delay profile into a bearing-angle profile. The relationship between time delay and bearing angle is:

$$\tau = \frac{d \sin(\theta)}{c}$$

Solving for θ:

$$\theta = \sin^{-1}\left(\frac{c\tau}{d}\right)$$

Using the time delay we found earlier:

$$\theta = \sin^{-1}\left(\frac{343.6(-4.1 \times 10^{-3})}{2.0}\right)$$

$$\theta \approx \sin^{-1}(-0.704)$$

$$\theta \approx -45^\circ$$

So the estimated bearing angle is approximately:

$$\theta \approx -45^\circ$$

The bearing-angle profile is shown below.

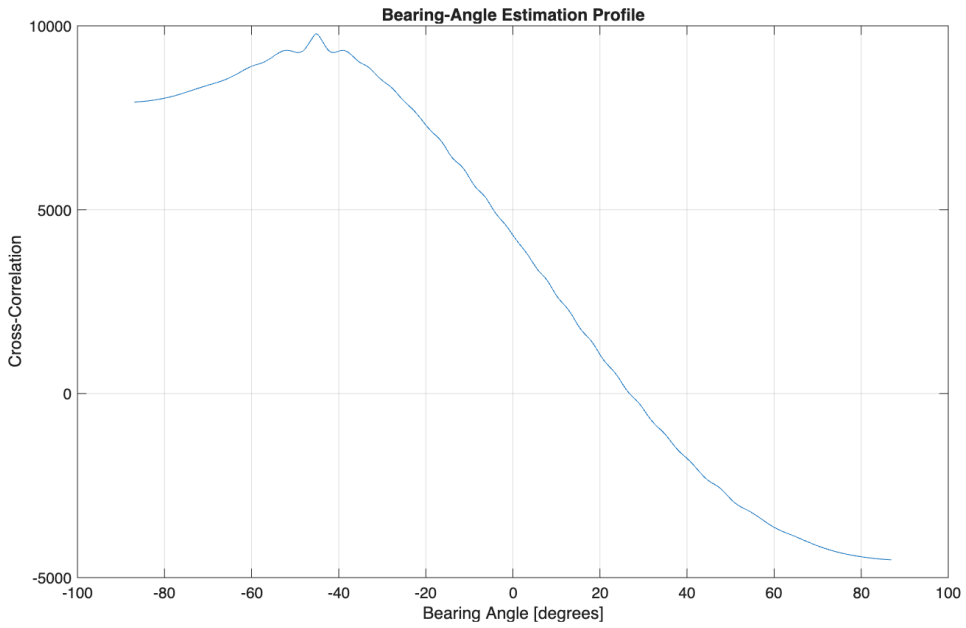

Figure 2: Bearing-angle estimation profile found by converting the time-delay profile into an angular-offset profile.

Overall, the plots seem reasonable. Since the signal is hitting microphone 2 first, this tells me that the source is on the mic 2 side, 45 degrees away from the perpendicular bisector of the line connecting the two microphones.

Part A Code

```
1
2  % Part A, bearing angle estimation
3
4  % we are provided with spacing and sampling rate
5  d = 2.0;
6  v = 343.6;
7  fs = 48000;
8
9  % load
10 [x1, fs1] = audioread('hw_7_partA_mic1.m4a');
11 [x2, fs2] = audioread('hw_7_partA_mic2.m4a');
12
13 % make sure signals are column vectors and remove offset
14 x1 = x1(:);
15 x2 = x2(:);
16 x1 = x1 - mean(x1);
17 x2 = x2 - mean(x2);
18
19 % we use matlab xcorr for cross correlation
20 [r, lags] = xcorr(x1, x2);
21
22 % convert lag samples to time delay
23 tau = lags / fs;
24
25 % make sure our values stay in the range
26 tau_max = d / v;
27 in_range = abs(tau) <= tau_max;
28 tau_plot = tau(in_range);
29 r_plot = r(in_range);
30
31 % plot
32 figure('Visible', 'off');
33 plot(tau_plot, r_plot);
34 xlabel('Time Delay [s]');
35 ylabel('Cross-Correlation');
36 title('Time-Delay Profile');
37 xlim([-tau_max, tau_max]);
38 grid on;
39 saveas(gcf, 'time_delay_profile.png');
40
41 % find the peak correlation
42 [~, idx] = max(abs(r_plot));
43 tau_est = tau_plot(idx);
44
45 % convert time delay to bearing angle
46 theta_est = asind((v * tau_est) / d);
```

```
47
48 % build angular offset for plot
49 theta = asind((v * tau_plot) / d);
50 r_valid = r_plot;
51
52 % plot
53 figure('Visible', 'off');
54 plot(theta, r_valid);
55 xlabel('Bearing Angle [degrees]');
56 ylabel('Cross-Correlation');
57 title('Bearing-Angle Estimation Profile');
58 grid on;
59 saveas(gcf, 'bearing_angle_profile.png');
60
61 fprintf('Estimated time delay: %.6f seconds\n', tau_est);
62 fprintf('Estimated bearing angle: %.2f degrees\n', theta_est);
```

Part B

I will use the first microphone recording from part A for this track. I only use the first channel of this audio file from this track.

In this section I will make the sound appear to come from five different angles:

$$-60^\circ,\ -30^\circ,\ 0^\circ,\ +30^\circ,\ +60^\circ$$

I will do what I did in part A, but in reverese. In Part A, I used the time delay between two signals to estimate the angle. In this part, I start with an angle and calculate the time delay needed between the left and right channels in order to make the perceived sound come from that angle.

The relationship between time delay and angle is:

$$\tau = \frac{d \sin(\theta)}{c}$$

where d is the receiver spacing, c is the speed of sound, and θ is the desired angle. I used:

$$d = 2.0 \text{ m}$$

and:

$$c = 343.6 \text{ m/s}$$

After finding the time delay, I converted it into a sample delay using the sampling frequency:

$$N_{\text{delay}} = |\tau| f_s$$

Since the audio signal is stored as discrete samples, this tells me how many samples one channel should be delayed relative to the other.

For each of the five angles I played the entire track once, with the delay calculated from the angle I want to emulate. For negative angles, I made the left channel lead and delayed the right channel. For positive angles, I made the right channel lead and delayed the left channel. For 0°, I used the same signal in both channels with no delay.

For example, when:

$$\theta < 0$$

the sound should appear to come from the left side, so the left channel receives the signal first. When:

$$\theta > 0$$

the sound should appear to come from the right side, so the right channel receives the signal first.

The five angle are:

$$-60^\circ \rightarrow \text{left channel leads}$$

$$-30^\circ \rightarrow \text{left channel leads}$$

$$0^\circ \rightarrow \text{no delay}$$

$$+30^\circ \rightarrow \text{right channel leads}$$

$$+60^\circ \rightarrow \text{right channel leads}$$

After creating the left and right channels, I combined them into one stereo signal:

$$\text{stereo output} = [L(n), R(n)]$$

Then I normalized the signal and saved it as a WAV file.

Now the output sound file will have 5 explosions, each of which has the effect of the sound coming in from a different angle.

Part B Code

```
1
2  % Part B
3  % 5-gun salute stereo sound track
4
5  clear; close all; clc;
6
7  % Read mono sound track
8
9  filename = '../Part A/hw_7_partA_mic1.m4a';
10
11 [x, fs] = audioread(filename);
12
13 % make sure we are only using first channel and normalize
14 x = x(:,1);
15 x = x / max(abs(x));
16
17 % speed of sound and receiver spacing
18
19 c = 343.6;
20 d = 2.0;
21
22 angles_deg = [-60 -30 0 30 60];
23
24 % now set up the output for left and right channels
25
26 N = length(x);
27 num_plays = length(angles_deg);
28 total_N = N * num_plays;
29
30 left = zeros(total_N,1);
31 right = zeros(total_N,1);
32
33 % cycle through each angle, apply time delay etc
34
35 for i = 1:num_plays
36
37     theta = angles_deg(i);
38
39     % define correct indices
40     start_idx = (i-1)*N + 1;
41     end_idx = i*N;
42
43     % start with the full original sound each time
44     section = x;
45
46     % calculate time delay for this angle
```

```
    tau = d * sind(theta) / c;

    % convert time delay to sample delay
    sample_delay = round(abs(tau) * fs);
    delayed_section = [zeros(sample_delay,1); section];

    % now we trim to original length
    delayed_section = delayed_section(1:length(section));

    % if angle is negative, left ear leads
    if theta < 0
        % left ear leads, right ear delayed
        left(start_idx:end_idx) = section;
        right(start_idx:end_idx) = delayed_section;

    elseif theta > 0
        % right ear leads, left ear delayed
        left(start_idx:end_idx) = delayed_section;
        right(start_idx:end_idx) = section;

    else
        % center, no delay
        left(start_idx:end_idx) = section;
        right(start_idx:end_idx) = section;
    end
end

% combine the two channels

stereo_out = [left right];

% again normailize
stereo_out = stereo_out / max(abs(stereo_out(:)));


sound(stereo_out, fs);
audiowrite('partB_5gun_stereo.wav', stereo_out, fs);
```

Part C

For part C, I will perform range estimation using the FMCW radar data, with the goal of reconstructing a subsurface image of the walkway pavement at Broida Hall. The radar was moved along in a straight line for 200 positions, with 2.13 centimeters between each position.

At each of these positions, the radar measured the reflected signal at 128 different frequencies. The data is stored in a matrix F, where each column corresponds to one antenna position, and each row corresponds to one frequency value. Therefore, each column of F contains the frequency response of the ground at one antenna position.

So our starting data is a frequency by position matrix and we take the ifft along the frequency dimension to get the delay by position matrix.

The frequency range is:

$$f_{\min} = 0.976 \text{ GHz}$$

$$f_{\max} = 2.00 \text{ GHz}$$

We find the bandwidth of the radar:

$$B = f_{\max} - f_{\min}$$

Which means:

$$B = 2.00 \text{ GHz} - 0.976 \text{ GHz}$$

$$B = 1.024 \text{ GHz}$$

Since the radar wave is traveling through pavement instead of air, the propagation speed needs to be adjusted using the relative permittivity. This means that the wave will travel slower in concrete than in air. The relative permittivity is given as:

$$\epsilon_r = 6.0$$

The equation for propagation speed in a material is:

$$v = \frac{c}{\sqrt{\epsilon_r}}$$

Using:

$$c = 3 \times 10^8 \text{ m/s}$$

the propagation speed becomes:

$$v = \frac{3 \times 10^8}{\sqrt{6}}$$

$$v \approx 1.225 \times 10^8 \text{ m/s}$$

The main signal processing idea is that the measured frequency response can be converted into a delay response using the inverse FFT. At the boundary between two different materials, part of an electromagnetic wave penetrates through and part of it gets reflected back. This reflected component creates a delayed echo, and that delay creates a phase change across frequency. By taking the inverse FFT across the frequency samples, I can convert each frequency response into a depth profile.

For one antenna position, the frequency response is:

$$H(f)$$

I take the inverse FFT which gives:

$$h(\tau) = \text{IFFT}\{H(f)\}$$

This is the impulse response, and the magnitude of it tells me the reflection strength at different delays:

$$|h(\tau)|$$

Then the time delay can be converted into depth. Since the radar signal has to travel down to the object and then back to the receiver, the total travel distance is twice the depth. Therefore:

$$z = \frac{v\tau}{2}$$

Intuitively, bandwidth has an inverse relationship to time-delay separation that can be resolved:

$$\Delta\tau \approx \frac{1}{B}$$

This time delay is converted into distance using the propagation speed v. Since the radar signal travels to the reflector and back, the depth is half of the round-trip distance:

$$\Delta z = \frac{v\Delta\tau}{2}$$

Substituting $\Delta\tau \approx \frac{1}{B}$:

$$\Delta z \approx \frac{v}{2B}$$

Now we plug in the values:

$$\Delta z = \frac{1.225 \times 10^8}{2(1.024 \times 10^9)}$$

$$\Delta z \approx 0.0598 \text{ m}$$

And I get a depth resolution of approximately:

$$\Delta z \approx 6.0 \text{ cm}$$

Which means that two materials would need to be separated by about 6 cm in depth to be clearly distinguished.

The maximum detectable depth is based on the spacing between frequency samples. This is because of aliasing: a deeper reflection corresponds to a faster phase change across frequencies. If the frequency spacing is too large then the phase can wrap a full cycle making the depth indistinguishable from a smaller depth. The frequency spacing is:

$$\Delta f = \frac{f_{\max} - f_{\min}}{N - 1}$$

where:

$$N = 128$$

Plugging in:

$$\Delta f = \frac{1.024 \times 10^9}{127}$$

$$\Delta f \approx 8.06 \times 10^6 \text{ Hz}$$

The maximum detectable depth is approximately:

$$z_{\max} = \frac{v}{2\Delta f}$$

Substituting in the values:

$$z_{\max} = \frac{1.225 \times 10^8}{2(8.06 \times 10^6)}$$

$$z_{\max} \approx 7.6 \text{ m}$$

So the maximum unambiguous depth is approximately:

$$z_{\max} \approx 7.6 \text{ m}$$

In the actual image, I only display the shallow region because the rebars are expected to be near the surface. The assignment says that the displayed profiles up to around 20 to 30 cm is sufficient. The top surface created a very strong reflection that was making it harder to see some of the smaller underground reflections. I removed this first reflection for clarity.

The horizontal pixel size is given by the antenna spacing:

$$\Delta x = 0.0213 \text{ m}$$

$$\Delta x = 2.13 \text{ cm}$$

The vertical pixel spacing is based on the depth bins from the inverse FFT. Using the bandwidth calculation above, the depth bin spacing is approximately:

$$\Delta z \approx 0.0598 \text{ m}$$

$$\Delta z \approx 6.0 \text{ cm}$$

So the physical pixel size is approximately:

$$\Delta x = 2.13 \text{ cm}$$

in the horizontal direction and:

$$\Delta z = 6.0 \text{ cm}$$

in the depth direction.
The reconstructed image is shown below.

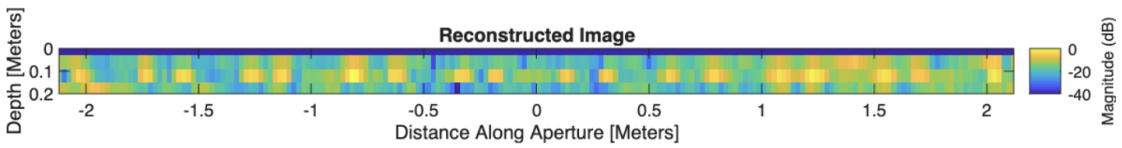

Figure 3: Reconstructed subsurface range image from the step-frequency FMCW radar data.

In the image, the x axis represents antenna position along the walkway, and the y axis represents depth into the pavement. Brighter regions correspond to stronger reflections. The strongest reflection are at 10 to 15 centimeters of depth, which suggests that this is where the rebars are located. The yellow flashes at .1 to .15 meters follows a uniform pattern in spacing, which further confirms that these are manmade rebar objects.

We used the inverse FFT to convert the FMCW radar into depth data, where each antenna position gives one depth profile, and putting all 200 depth profiles together gives us the final 2D subsurface image to analyze.

Part C Code

```
1
2  % Part C
3  % range estimation using step frequency radar data
4
5  clear; close all; clc;
6
7  % load
8  load('gpr_data.mat');
9
10 % F  = 128 x 200 frequency data
11 % f  = 128 x 1 frequency vector, so one spatial point
12 % da = antenna position
13
14 disp(size(F));
15 disp(size(f));
16 disp(size(da));
17
18 % define constants for wave speed
19 c = 3e8;
20 er = 6.0;
21 v = c / sqrt(er);
22
23 % find basic frequency info for reporting
24
25 Nf = length(f);
26 f_min = min(f);
27 f_max = max(f);
28 B = f_max - f_min;
29
30 df = mean(diff(f));
31
32 fprintf('Number of frequency samples: %d\n', Nf);
33 fprintf('f_min = %.3e Hz\n', f_min);
34 fprintf('f_max = %.3e Hz\n', f_max);
35 fprintf('Bandwidth = %.3e Hz\n', B);
36 fprintf('Frequency spacing = %.3e Hz\n', df);
37
38 % calculate resolution and maximum depth from frequency data
39 depth_resolution = v / (2 * B);
40 max_depth = v / (2 * df);
41
42 fprintf('Depth resolution = %.4f m\n', depth_resolution);
43 fprintf('Maximum detectable depth = %.4f m\n', max_depth);
44
45 % convert frequency response to depth response
46 % We take the IFFT down each column, across frequency.
```

```
depth_profiles = ifft(F, [], 1);

% magnitude tells reflection strength
depth_mag = abs(depth_profiles);

% build depth axis

depth_axis = (0:Nf-1) * depth_resolution;

% remove the strong surface reflection
gate_depth = 0.03;   % remove first 3 cm
gate_bins = depth_axis < gate_depth;

depth_mag(gate_bins, :) = 0;

% display correct shallow region
max_display_depth = 0.20;

display_bins = depth_axis <= max_display_depth;

image_data = depth_mag(display_bins, :);
depth_display = depth_axis(display_bins);

% convert to dB for visualization
image_db = 20*log10(image_data + eps);

% normalize
image_db = image_db - max(image_db(:));

% clip
range = 40;
image_db(image_db < -range) = -range;

% plot
da_plot = da - mean(da);
figure('Position', [100 100 900 220]);
imagesc(da_plot, depth_display, image_db);
axis ij;
xlim([min(da_plot), max(da_plot)]);
ylim([0, max_display_depth]);
daspect([1 1 1]);
xlabel('Distance Along Aperture [Meters]');
ylabel('Depth [Meters]');
title('Reconstructed Image');
colorbar;
ylabel(colorbar, 'Magnitude (dB)');
colormap(gca, parula);
```

```
94
95 % save
96 saveas(gcf, 'partC_range_image.png');
```

Conclusion

In conclusion, this assignment showed signal processing techniques that can be used to estimate the direction and range from time-frequency data. In part A I utilized the time delay between two acoustic receivers to estimate the bearing angle, in part B I reverse engineered the required time delays to emulate different bearing angles, and in part C I converted step-frequency radar into depth profiles that I could physically analyze. This assignment helped me connect abstract signal processing concepts like cross-correlation, time delay, phase and FFTs into practical uses where I can see how they are used in the real world.