Project 3 – EECS 152B

Cameron Peterson-Zopf

57999719

Procedure 1: Data Adaptive Beamforming

For the first procedure we assume the desired signal to be at 5 degrees and the two interfering signals to be at -25 degrees and 15 degrees.

**Question 1:**

In this first question students generate 100,000 samples of data with 16 antennas using the signals in ABF.mat. The code for the generation of said data is displayed below in figure 1.

A screenshot of a computer

Description automatically generated

Figure 1 – Code for Generation of Data

**Question 2:**

Using the first 100 samples of data from question 1, students find the optimal adaptive beamformer and plot its spatial frequency response. The code for the optimal adaptive beamformer is shown in figure 2.

A screenshot of a computer program

Description automatically generated

Figure 2 – Code for Adaptive Beamformer w/ T = 100

The plot of the spatial frequency response for the above beamformer is shown in figure 3.

A graph of a graph with a red line

Description automatically generated

Figure 3 – Adaptive Beamformer w/ T = 100

As seen in figure 3, the beamformer successfully attenuates the signal at -25 degrees and 15 degrees. Furthermore, the gain at the desired signal is unity as desired. Thus, this filter has achieved its goals.

In terms of listening to the output to verify that the beamformer has a clean signal, I first played the desired signal to know what the desired output should be. Then, I listened to the output of the first antenna, to hear the noisy signal. Lastly, I played the output signal y. As expected, the output of the first antenna was filled with noise. The output signal y matched the original signal, with some slight background noise that was much lower in volume than the desired signal. Therefore, the beamformer was a success.

**Question 3:**

The next question asks students to repeat the prior experiment, with the exception of moving the interfering angle that was originally at 15 degrees to a location closer to the desired signal at 5 degrees. Thus, I first moved the signal to 10 degrees, and found no issue with the output. I then moved to 7 degrees, or 2 degrees away from the desired signal. At this point, the jet noise begins to increase in loudness; however, I can still make out the desired sound. As I get closer to the 5 degree mark, the jet noise becomes greater and greater. At 5.5 degrees, or 0.5 degrees away from the desired signal, the jet noise is louder than the desired sound, but the desired sound is still recognizable. At around 5.1 degrees, or 0.1 degrees away from the signal, the jet noise becomes so loud, that the original desired signal is no longer recognizable. Figure 4 gives an example of the code employed to create the shifted interfering angle.

A screenshot of a computer program

Description automatically generated

Figure 4 – Second Interfering Angle at 7 Degrees

As seen above, I simply changed the second interfering angle to different values, closer and closer to 5 degrees, listening to the output sound at each change. The above figure shows this for an interfering angle of 7 degrees.

The interfering angle of 5.1 degrees is the point of failure. Per direction, the amount of antenna is increased to 50. When I added the 50 antennas the result became much better, and I was once again able to recognize my desired signal. The code for this process is shown in figure 5.

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

Figure 5 – Code for Beamformer with 50 Antennas

The spatial frequency response for figure 5 is plotted in figure 6.

A graph of a graph with a red line

Description automatically generated

Figure 6 - Spatial Frequency Response of Beamformer with 50 Antennas

Analyzing the graph, we see that the interference signal at -25 degrees is suppressed. At the 5 degree mark, the black line representing the desired signal and the red line representing the interference signals are nearly on top of each other. Thus, figure 7 provides a zoomed in graph of this region.

A graph of a graph

Description automatically generated

Figure 7 – Zoomed in Spatial Frequency Response

From figure 7, the 5 degree desired signal will experience a gain of -2.5 dB, which is lower than the wanted 0 dB gain. Furthermore, the interference signal at 5.1 degrees experiences a gain of -3.4 dB, which is not that much different than the gain of the desired signal and does not attenuate the unwanted signal very much. However, because we have 50 antennas we are still able to obtain the desired signal, albeit with some interference as heard in the recording.

Procedure 2 – Time and Data Adaptive Beamforming

**Question 1:**

Utilizing the same scenario as before, but with an additional interference signal beginning at 50,001 samples, or halfway between total 100,000 samples, we add a new interference signal at zero degrees. The code for this process is displayed in figure 8.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Figure 8 – Code for Beamformer with Additional Interference at 0 Degrees

As seen above, I plot the spatial frequency response at 0 degrees with respect to time. This plot is shown below in figure 9.

A graph showing a graph

Description automatically generated with medium confidence

Figure 9 – Plot of Spatial Frequency at 0 degrees vs Time

Zooming in on the 50,001 sample point where the interference kicks in we see the following.

A graph with blue lines and white text

Description automatically generated

Figure 10 – Zoomed in Plot of Spatial Frequency at 0 degrees vs Time

In Figure 10 we see that the first trough occurs at sample 50,007, which is 6 samples after the interference is introduced. After this point, the response stays at or below -57 dB, which is roughly zero for all intensive purposes. Thus, the beamformer takes 6 samples to adjust to the new interference. Furthermore, upon listening to the output recording, the desired signal can be cleared heard and recognizable.

Notice that I have my epsilon as 0.0005. The choice of this value for epsilon was the best I found given the trade offs of having big and small epsilon values. This analysis of epsilon values will be revisited in question 4.

**Question 2:**

Question 2 prompts students to write a function that will generate the 100,000 samples of data with time-varying angles for the interference. We assume this variation is linear. The code for the function is shown in figure 11.

A screenshot of a computer

Description automatically generated

Figure 11 – Function Code for Time-Varying Angles

The function name is “GetAng” and there are two inputs: the slope and the initial angle.

**Question 3:**

For this question we assume two interfering signals, which begin at -25 degrees and 15 degrees. These two signals will then have their angles vary over time. The first 100 samples are used to initialize the beamformer to cancel the signals at -25 and 15 degrees. Thus, will the beamformer is initializing, I will not move the interfering signals. The variation of angle will begin at the 101 sample. After initializing, the interfering signals will move and the beamformer will adapt to their movement. The code for this process is displayed in figure 12.

A screenshot of a computer

Description automatically generated

A computer screen shot of a program

Description automatically generated

Figure 12 – Code for Time and Data Adaptive Beamformer

Notice that I have written this procedure as a function with 5 parameters: desired signal angle, interference angle #1, interference angle #2, rate of change of angles, and epsilon utilized. I then tested the performance of the beamformer and found the following results for the following scenarios:

1. Holding epsilon constant at 0.0005, and changing the slope from 1/100,000 to 10/100,000 to 20/100,000 to 25/100,000 to 35/100,000 to 100/100,000.

For this scenario, as the slope increased, the interference became louder, especially near the end of the recording for the slopes of 25/100,000 and 35/100,000. This is because by moving 25 and 35 total degrees, the -25 degree signal is now at 0 degrees and 10 degrees respectively. For the 25/100,000 slope, the desired signal is still recognizable near the end of playtime. However, the 35/100,000 slope is not recognizable near the end of playtime. This is for two reasons: the interference signal beginning at -25 degrees passes over the desired signal, and both of the interference signals change at a fast rate. Lastly, when the slope is increased to 100/100,000, the slope is changing much faster and there is more interference in the output as a result. When the interference signal beginning at -25 degrees passes over the desired signal, the output signal is unrecognizable; but as interference signal gets past the desired signal, we begin to recognize the desired signal once more with some interference.

1. Holding epsilon constant at 0.000000001, and changing the slope from 1/100,000 to 10/100,000 to 20/100,000 to 25/100,000 to 35/100,000 to 100/100,000.

With the epsilon decreased, the beamformer relies more heavily on the previous values. Thus, as the slope increases, the beamformer cannot change quickly enough. For all of slopes tested, the resulting output quality is less than the first scenario for each respective slope, with the slopes of 20/100,000, 25/100,000, 35/100,000, and 100/100,000 being very poor.

At the two scenarios tested above, the faster the interference could move while maintain good cancelation was 35/100,000 for the epsilon of 0.0005 and 20/100,000 for the epsilon of 0.000000001. Figure 13 displays the code for calling the function created in this question.

A screenshot of a computer

Description automatically generated

Figure 13 – Generation Output Recordings for Question 3

In the example code above, I am employing a slope of 20/100,000 and a epsilon of 0.000000001.

**Question 4:**

This question discusses the impact of the various parameters of the adaptive beamformer.

1. Rate of Change of Interfering Signals

As the rate of change is increased, the adaptive beamformer must change more rapidly to keep up with the increasing rate of change of the interfering signals. Thus, holding epsilon constant, as we increase the rate of change, the sound quality will worsen.

1. Value of Epsilon

As the value of epsilon decreases the beamformer relies more on the previous values instead of the current one. This leads to greater stabilization and less oscillation in the waveform as additional data points have less effect. However, the downside to this is that the beamformer cannot adjust very quickly. For example, with the first question of this procedure I have my epsilon as 0.0005. This was chosen to minimize the number of samples I needed to be under -55 dB for the remainder of signal. If I increased my epsilon to 0.05, I would reach the mark slightly faster; but then I will have greater oscillations that go above the -55 dB threshold. If I decreased the epsilon to 0.00005, then the oscillations will be greatly suppressed, but then the number of samples needed to reach the mark would have been greater.

1. When the interfering signals move closer to each other, the total interference in that location is very high.
2. When the interfering signals move across the desired signal, the output recording is very poor around when this happens; but is much better at the start and end of the recording when the signals are not near the desired signal.