Computer Project for EEE 509

Frequency-Domain Adaptive Noise Cancellation

MATLAB project for the EEE 509 class

Due by: June 18, 2024, 11:59 pm AZ Time

Save your report in PDF format.

Save the program code (.m files), a read-me text file with the instructions on how to run the code, and your report (pdf) in a zip file (containing code and readme) should be uploaded to Canvas. Also, upload your report (pdf) separately to Canvas.

Late submissions will be accepted up to June 20, 2024, 11:59 pm AZ Time. There will be a penalty of 20% for late submissions after June 18, 2024.

Frequency Domain Adaptive Noise Cancellation

Introduction

The purpose of this project is to familiarize the students with simulations of adaptive noise cancellation and the use of the FFT in a real-life application. Adaptive noise cancellation involves a configuration with two sensors (microphones) shown below.

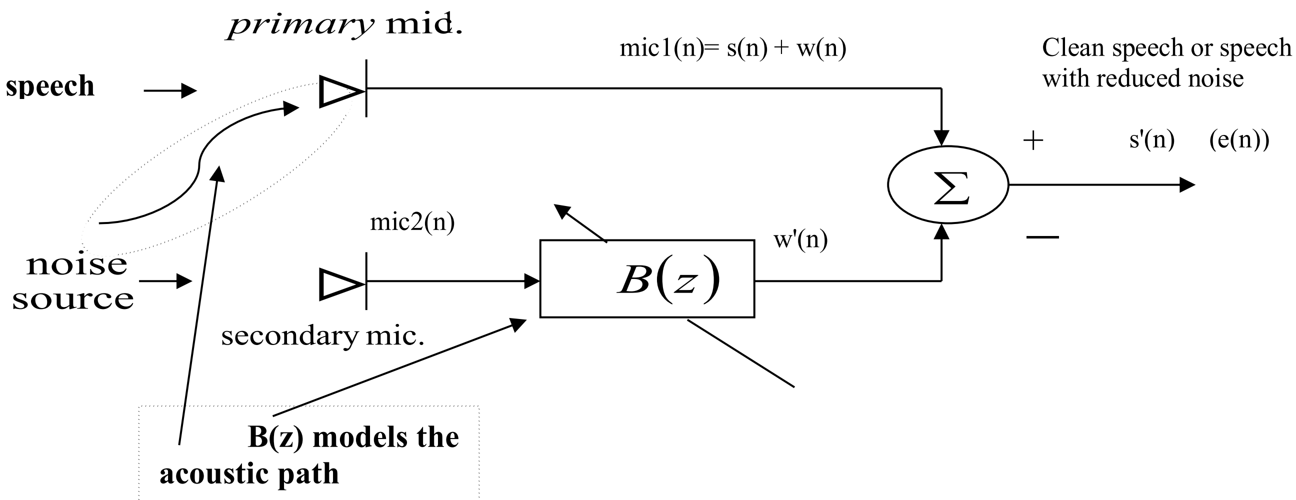


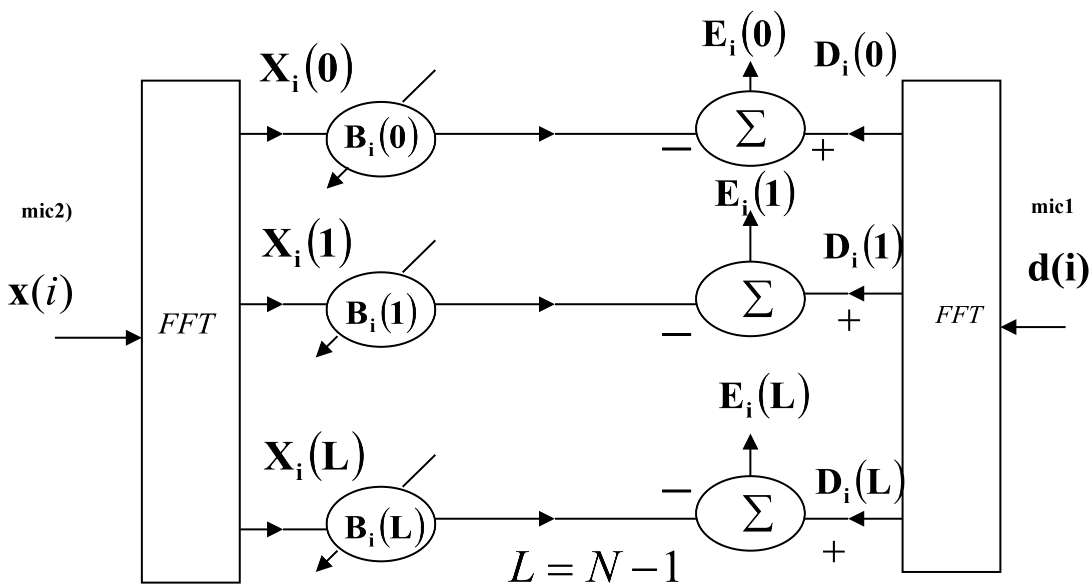
Fig. 1 Adaptive Noise Canceller

The following describes how this system works. The primary microphone is picking up speech and background noise from a certain noise source. Background noise is acquired by a secondary microphone that is located close to the noise source but sufficiently far from the primary microphone. The idea is to acquire the signal from the noise source and process it by an FIR filter and produce w'(n) which will then be subtracted from the primary microphone signal (s(n)+w(n)). If w'(n) is equal to w(n) then the noise reduction is perfect and the output of this configuration will be clean speech. Because of room acoustics (reflections, delay, etc), w(n) can not be cancelled perfectly, i.e., w'(n) approximates w(n). The FIR filter, B(z), is needed to model the reflections and delays in the acoustic path between the noise source and the primary microphone. This FIR filter can be made very long (64 or 128 coefficients or more).

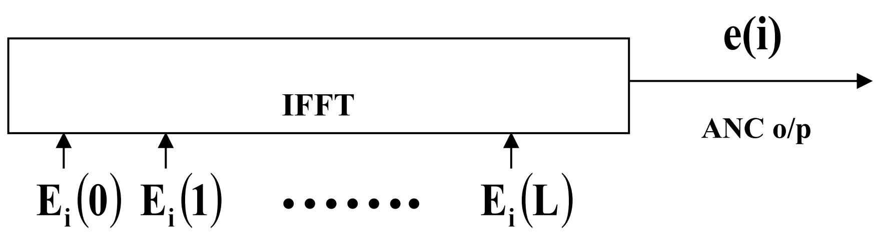
The acoustic path is usually time varying because of movement in the room etc. Since the acoustic path is time-varying it needs to be continuously estimated. For this reason , an adaptive (time-varying) FIR filter, B(z), is needed. The coefficients of B(z) can be estimated using a time or a frequency-domain adaptive algorithm.

We need to develop a simulation of this system with MATLAB. The signals from primary microphone (mic 1) and secondary microphone (mic 2) are be given to you as .wav files in the mfiles folder. Use MATLAB to read these files in vectors called mic1 and mic2. These two files must then be segmented into two sequences of N-point vectors (frames). Let us call the vectors (frames) formed as follows; frames from mic1, **d**(i), and the frames formed from mic 2, **x**(i). All the frames are to be processed one-by-one by the adaptive filter. Because the filter is processing the data frame-by-frame its output and the overall output of the noise canceller will be a sequence of frames. These frames will have to be concatenated at the end so that you can playback and listen to the output signal .
Because adaptive noise cancellation requires high-order FIR filters, the FFT is used to perform fast convolution. In addition the entire adaptive filtering algorithm operates in the frequency domain with FFTs.

Here is how the FFT-based adaptive filter works. Each frame **d**(i) and **x**(i) is "FFTed" and frequency-domain frames are formed, namely **D**(i) and **X**(i) respectively (i is the frame index). The filter operates in the frequency domain therefore **X**(i) is multiplied (frequency-by-frequency) with the frequency response of the filter, **B**(i). The picture below shows how the signals are processed



In order to playback the ANC output signal you need to transform **E**(i) back to the time domain and then concatenate successive time-domain frames (vectors) in a large output vector.



The adaptive algorithm adapts the coefficients of the filter in the frequency domain. In order to take advantage of the speed of MATLAB use vectorized operations. You will need to read the two data files mic1.wav and mic2.wav from the folder m files. Use the MATLAB commands:

[s1, fs1] = audioread('mic1.wav'); reads file mic1.wav file.
[s2, fs2] = audioread('mic2.wav'); reads mic2.wav file.

MAKE SURE THAT THE DEFAULT DIRECTORY OF YOUR MATLAB m file IS THE SAME
AS THE DIRECTORY where you stored the two wav files. Use the following commands to listen to the files (keep the volume down they are loud and noisy!):

sound(s1)
sound(s2)

Use the command length(s1) to read the size of the vector. This will help you figure out the number of frames. The algorithm is described as follows

ALGORITHM DESCRIPTION

Note that the process below describes the algorithm and is not a MATLAB program. You will need to develop your own MATLAB program in order to simulate the ANC as follows

; Form data frames **d**(i) and **x**(i) from the s1 and s2 signals
; Find the total number of N-point frames Nframes (Nframes is different than N)
;initialize **B**(Nframes, 1)=**0**

i=1,2,…,Nframes

X(i)=FFT(**x**(i)) ; take N-point FFT
D(i)=FFT(**d**(i)) ; take N-point FFT
Xdiag(i)=diag(**X**(i)); read the frequency-domain data into a diagonal matrix

E(i)=**D**(i)-**Xdiag**(i)*__B__(i)
B(i+1)=**B**(i)+2*mu* **Xdiag**^{H}(i)*__E__(i) ; (adapts the coefficients of the filter, H is complex conjugate transpose)
e(i)=IFFT(**E**(i))
next i

CALCULATIONS OF PERFORMANCE METRICS

We will evaluate the ANC using objective and subjective performance measures. In addition, we will be plotting graphs to help us visualize the adaptation process. We will use signal to noise ratios (SNRs) for an objective measure. For a subjective measure we will try to rate the resultant signal on a scale of 1 to 5 (1-bad/5-excellent). The signal to noise ratios to be used are defined as:

$$SNR_{dB}^{Before} = 10\log_{10}\left(\frac{\sum_{n=1}^{NSAMPL} s^2(n)}{\sum_{n=1}^{NSAMPL}(s(n)-mic1(n))^2}\right) \qquad SNR_{dB}^{After} = 10\log_{10}\left(\frac{\sum_{n=1}^{NSAMPL} s^2(n)}{\sum_{n=1}^{NSAMPL}(s(n)-e(n))^2}\right)$$

where *NSAMPL* is the total number of samples in the sound files, *s(n)* is the original signal (clean signal file is the file 'cleanspeech.wav' in the folder mfiles, *e(n)* is the time domain output of the noise canceller.
We will use two signal-to-noise ratios in order to evaluate the improvement after noise cancellation in dB. To measure the SNR before the adaptive noise canceller is used, calculate $SNR_{Before.}$ To measure the SNR after the adaptive noise canceller is used calculate $SNR_{After.}$ The Improvement in the SNR is given by

$$SNR_{dB}^{\textbf{Improvement}} = SNR_{dB}^{After} - SNR_{dB}^{Before}$$

We can tabulate results as follows:

N	$SNR^{Improvement}{}_{dB}$	Subjective Rating (1-5)	Best mu
4			
16			
64			
128			
256			

SIMULATIONS:

The step size mu controls the speed (gain) of adaptation of the adaptive algorithm. Large *mu* implies quick adaptation small mu implies slow adaptation. Very large mu may result in instability of the algorithm and the frequency domain filter coefficients **B**(i) can become unbounded (overflow). You need to experiment with different values of mu. Start with a very small value (say 0.001 or smaller) and test several mu's large ones and small ones. Choose one small that gives noise reduction but with slow adaptation and one large one that gives quicker adaptation. For each set of files and different values of N you have to run the software again and find a new mu.

1. Do all simulations to fill up the table above. For all cases give your best simulation result (best mu).

2. Plot error convergence curves, i.e., |**E**(i)|^2 in dB as a function of the iteration i. These will show you how fast the algorithm converges. Plotting convergence curves for N=64 and 256

3. For N=128 give a 3-D plot of |B(i,k)| vs i (iteration) and k (frequency up to N/2)). (|B(i,k)| in dB will give you a better picture). Use the mesh function.

REMARKS

Interpreting results. Give specific answers to the questions below in your Results section.

1. What do the frequency components B(k) represent regarding filter properties?
2. What is the effect of N on the SNR?
3. How is the size of the FFT (N) related to the order of the filter?
4. What is the effect of mu on the quality of speech (small mu vs big mu)?
5. How does the SNR correlate with speech quality?
6. Did you observe any artifacts in your subjective evaluation?
7. Why does the algorithm minimize the e(n)?
8. How do you determine the impulse response of the filter from **B**(i)

HINTS

- The norm |**E**(i)|^2 of the vector **E**(i) is a <u>scalar</u> and computed by

$$|\mathbf{E}(i)|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |E_i(k)|^2 = \frac{1}{N}\mathbf{E^H}(i)\mathbf{E}\ (i)$$

You need to calculate and plot |**E**(i)|^2 in dB as a function of i as per items 2 and 3 in the previous page. You can compute |**E**(i)|^2 in dB as follows

$$\varepsilon_{dB}(i) = 10\log_{10}\left(\frac{1}{N}\mathbf{E^H}(i)\,\mathbf{E}(i)\right)$$

- The frequency domain filter parameters are contained in a vector **B**(i). The elements **B**(i) are both a function of time and frequency, hence **B**(i) contains complex elements B(i,k). You are to plot the magnitude of these parameters as a function of i and k, i.e., |B(i,k)| vs i,k as per item 4 in the previous page. Use the 'mesh' function of MATLAB to do so. Use 'help mesh' and 'demo mesh' to see how to use this 3D plot function.

DELIVERABLES – SHORT REPORT

- A brief typed report that contains all simulations, explanations of results, and your concluding remarks.

- This report must have two convergence curve figures and a table of results

  table. Give all results as specified in the table.

- **Organize your report with section headings as follows**

1. **Abstract;** 50 words explaining what the project is about and 1-2 sentences commenting on your best and worst SNR results
2. **Results;** this section has all the results of the simulations; it has the two convergence figures, one table with SNRs, and the mesh plot. Label them Fig. 1, 2 etc. (about 1 page)
3. **Explanation of results; Must have** the necessary text to explain the results and figures. Answer the questions in the sequence stated before. All figures must be labeled. (1.5-2 pages is appropriate - depending on the font spacing and how you format the results and figures). Give concisely your remarks on the convergence curves and the selection of the parameter - mu. (1 Paragraph), Give comments to explain the mesh plot of **B**
4. **Conclusion;** Explain about what you have learned from this project. (1 Paragraph max ½ page).
5. **Appendix;** listing of your MATLAB program

- Please submit your .m file code for the grader to verify the results.

Include your program files (.m files) along with a readme text file (with instructions on how to run your code) must be saved in one zip file.

Your report (pdf) and the zip file (containing code and readme file) should be uploaded on Canvas for the TAs to check.

The zip file and the pdf report must have the format: E509_Sum24_lastname_pjct

THE PROJECT COUNTS FOR 15% OF YOUR FINAL GRADE