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.

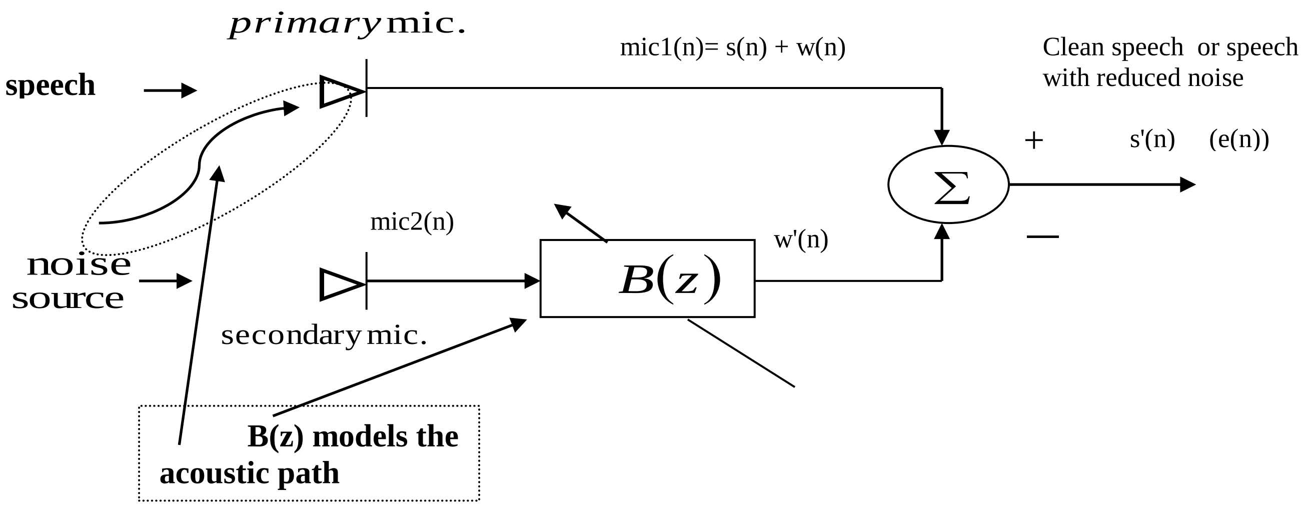


Fig. 1 Adaptive Noise Canceller

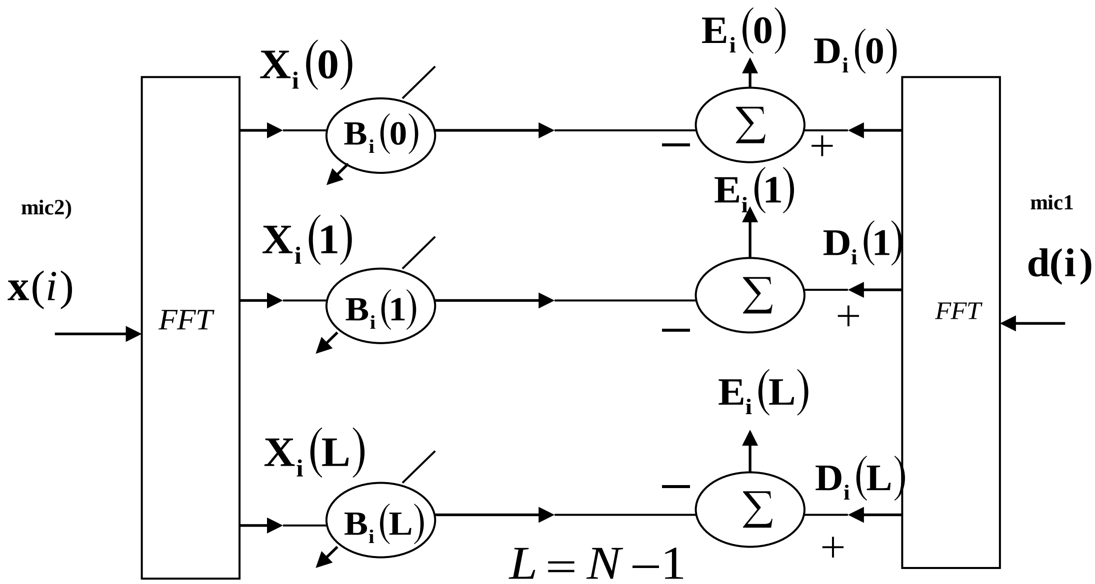
The following describes how this system works. The primary microphone is picking up speech and background noise from a certain noise source. Background noise is acquired by a secondary microphone that is located close to the noise source but sufficiently far from the primary microphone. The idea is to 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.

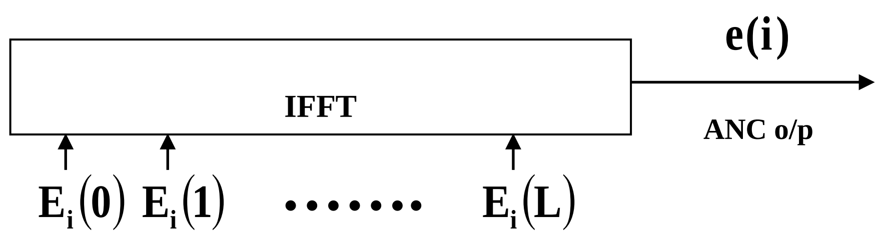
We need to develop a simulation of this system with MATLAB. The signals from primary microphone (mic 1) and secondary microphone (mic 2) are be given to you as .wav files in the mfiles folder. Use MATLAB to read these files in vectors called mic1 and mic2. These two files must then be segmented into 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:

[mic1,fs,bits]=wavread('mic1') reads file mic1.wav in the vector mic1
[mic2,fs,bits]=wavread('mic2'); reads mic2.wav in the vector mic2
(these commands may differ depending on the version of MATLAB you are using. For newer versions of MATLAB, use “audioread” function instead)

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(mic1)
sound(mic2)

Use the command size(mic1) 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 mic1 and mic2 signals
; Find the total number of N-point frames Nframes (Nframes is different than N)
;initialize **B**(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 <u>before</u> the adaptive noise canceller is used, calculate $SNR_{Before.}$ To measure the SNR <u>after</u> 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	$SNR^{\textbf{Improvement}}_{dB}$	Subjective Rating (1-5)	Best mu
4			
16			
64			
128			
256			