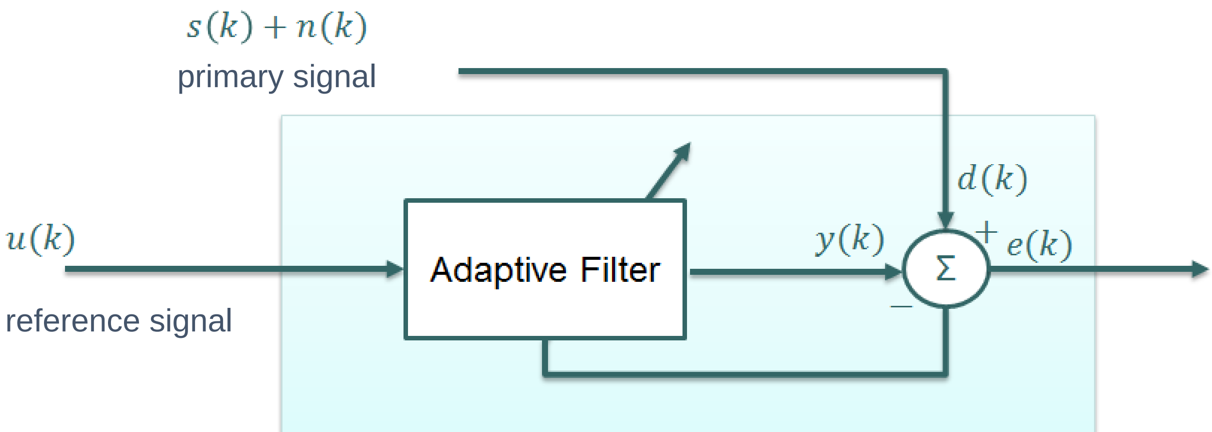
Lab Homework: Adaptive Noise Cancellation
Digital Signal Processing

Maria Amer, Concordia University, Department of ECE

Introduction:
When we communicate with digital (for example, home) assistants at some distance with a degree of sound interference, we must process not only the main (speech) signal, but also echoes and conflicting noise that deteriorate processing accuracy. Noise includes TV, kitchen appliances, and other persons talking. Noise cancellation or noise removal are techniques in signal processing that perform the removal of certain undesired audible frequencies. If the undesired frequencies are removed, we call it noise removal and if they are supressed, we call it noise cancellation. Adaptive Noise Cancellation is a variation of optimal filtering that involves producing an estimate of the noise by filtering the reference input and then subtracting this noise estimate from the primary input containing both signal and noise (see figure).



Active noise cancellation (ANC) is an active DSP topic that employs a process of using a microphone to monitor environmental noise and creating anti-noise that is then mixed in with audio playback to cancel noise entering the user's ears.

Problem:
In this work, we aim at cancelling the additive noise in a noisy audio signal, namely noise reduction, by means of *adaptive filtering* techniques. Least mean squares (LMS) is an adaptive noise reduction technique that is widely used.

In this work, you are supposed to combine the audio samples of a speech or music (***original signal)*** of your choice with the <u>**real**</u> noise (such as traffic, appliances, etc.) to generate ***noisy signal***. Alternatively, you may use audio files such as .MP3s or. WAVs. Make sure you record signals with different amounts of signal-to-noise (SNR) ranging from -10dB to +10dB. Then use **LMS** based noise cancelling techniques on the input ***noisy signal*** to obtain the de-noised **enhanced signal** as the output. The clean speech signal **(*original signal)*** can be used as the reference signal to calculate the LMS filter weights. It should be noted that the enhanced signal has to be of the same length as the noisy and clean speeches.

Tools to use:

1. Usage of Matlab (R2017 or newer version) is highly recommended. (You may use python 3.5 also.)
2. Make sure the audio samples are minimum 10 seconds in length.

PseudoCode:

1. Use Matlab's *audioread* function to read or *audiorecord* to record clean audio signal. Example (to record using Matlab's audiorecorder function):
```
recObj = audiorecorder;
disp('Start speaking.')
% Record audio from  the PC/mic.
recordblocking(recObj, 10);
disp('End of Recording.');
% Play back the recording.
play(recObj);
% Store data in double-precision array.
Original_signal = getaudiodata(recObj);
```

2. Repeat the process to obtain noisy signal (make sure your original signal is corrupted with surrounding noise or add noisy signal directly in Matlab).
3. Normalize the speech signals by the RMS of the signal, i.e.,
$$CleanSignal = CleanSignal/\sqrt{power}$$
4. Design the **LMS** filter, obtain the weights or coefficients (See [1] for details).
5. Filter the ***noisy signal*** with the **LMS** filter to obtain the denoised signal.
6. Analyze (play/plot/visualize) the output.

Lab report:

In your report, you want to plot all signals in time and frequency domains and analyze these plots. No two reports from students should be the same, since the input signals and the real noise are different. Submit, say .zip file via cloud, your report, your **source code**, and **all test signals**. More information on how to prepare your report is here
https://users.encs.concordia.ca/~amer/teach/elec6601/src/freport.html

References:

1. Adaptive noise cancellation: https://www.cs.cmu.edu/~aarti/pubs/ANC.pdf
2. Adaptive Filter Theory by Simon Haykin, 4ed,2002 Prentice Hall
3. B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1985.
4. An overview to adaptive filters is here
https://users.encs.concordia.ca/~amer/teach/elec6601/src/notes/Ch16_adaptiveFilters.pdf