Indian Institute of Science
Department of Electrical Communication Engineering

E9 211: Adaptive Signal Processing

October 2020 - January 2021

Final project (deadline 1 Feb. 2021)

This project consists of two parts on implementing and studying adaptive filters for adaptive noise cancellation: (a) using single-channel microphone recordings and (b) using multi-channel microphone recordings.

1. Make a short report containing the required Matlab/Python files, plots, explanations, and answers, and turn it in by the deadline using Microsoft Teams under your name.
2. Include a .zip file containing "9 audio files" named in the format: `firstname_lastname_part_funcname.wav`, e.g., `sundeep_chepuri_A_lms.wav`
3. Create an MS Teams meeting with yourself to record a "5 min presentation" explaining your code, observations, and outputs related to each question. Don't forget to share the computer audio while creating the MS teams meeting so that the outputs are also recorded.

Part A: Adaptive interference cancellation using a single-channel microphone

In this section, we perform adaptive interference cancellation using LMS. We are interested in removing a background interference while recording a class room lecture. Using a single isotropic microphone, we record the desired speech along with other sources (e.g., someone talking in audience or outside the classroom).

Let $d(t)$ denote the output of the microphone that records the desired signal $s(t)$ and an interference $v(t)$. We have

$$d(t) = \alpha s(t) + \beta v(t) + n(t), \tag{1}$$

where $\alpha, \beta \in \mathbf{R}$ are the unknown channel gains and $n(t)$ denote the additive white Gaussian noise with variance σ^2. For convenience, let us assume that the signal, noise, and interference are mutually uncorrelated.

We are now interested in suppressing the interference in $d(t)$. To do so, we have access to recordings from a secondary microphone that is placed closer to the audience or outside the classroom and away from the desired speaker. That is, the secondary microphone picks up only the interference. Let the output of the secondary mic be given by $x(t) = v(t)$.

For Part A of the project, we will be using the dataset `data_partA.mat`, which contains two variables `x` and `d`. Both are 32000×1 vectors corresponding to a 4 second recording at the primary and secondary microphones recorded at a sampling rate of $F_s = 8$ kHz.

Consider a M tap LMS filter with a coefficient vector $\mathbf{w}_k : M \times 1$ and regressor vector

$$\mathbf{x}_k = \begin{bmatrix} x[k] & x[k-1] & \dots & x[k-M+1] \end{bmatrix}^{\mathrm{T}} : \quad M \times 1,$$

where $x(t) = v(t)$ is the interference term.

1. Explain how to suppress the interference in $d(k)$ using $\mathbf{w}_k$.
2. Load the file `data_partA.mat` and listen the variables `x` and `d` as audio. In Matlab, you can listen using `soundsc(x,Fs)`. Make sure that `x` corresponds to a laughter and `d` is a mix of laughter and the desired signal, which goes like "the discrete Fourier transform...".
3. Make a function to implement LMS to perform interference reduction

   ```
   [s w]=lms(x,d,M,w_init,mu)
   ```

4. Derive and create a function to implement the sign-error LMS algorithm to perform interference reduction

   ```
   [s w]=sgnlms(x,d,M,w_init,mu)
   ```

5. Derive and create a function to implement the normalized LMS algorithm to perform interference reduction

   ```
   [s w]=nlms(x,d,M,w_init,mu)
   ```

6. For all the LMS variants above, use `w_init`$=\mathbf{0}$. Choose an appropriate step size to ensure convergence. Which choice of M leads to a better performance and why? Is the interference suppressed for $M = 1$? What is the structure in `w` for different values of M, and why does it have this structure? Can we infer the value of β in (1)? Play the output `s` and verify if the interference (i.e., laughter) is suppressed. Comment on the performance of the different LMS algorithms. Which one would you prefer and why?

Part B: Multi-channel interference cancellation

In this part, we use a uniform linear array (ULA) with $N = 10$ microphones to perform interference cancellation by exploiting the spatial signature of different sources. We consider the same scenario as before to suppress the interference.

We assume that the desired signal (lecture in this case) and the interference (laughter from audience) are arising from different directions relative to the array. Let $s(t)$ be the desired signal and $v(t)$ be the interference. Let θ_1 and θ_2 denote the directions from which the signals arrive at the ULA. Let the inter-element spacing in the ULA be δ and the speed of sound be c. Let $x_i(t)$ denote the signal received at the i-th antenna at time t. Then the output of the ULA at time t can be written as

$$\mathbf{x}(t) = \begin{bmatrix} s(t) + v(t) \\ s(t-\tau_1) + v(t-\tau_2) \\ \vdots \\ s(t-(M-1)\tau_1) + v(t-(M-1)\tau_2) \end{bmatrix} + \mathbf{n}(t) : \quad N \times 1$$

where $\tau_1 = \delta \sin(\theta_1)/c$ and $\tau_2 = \delta \sin(\theta_2)/c$ are the delays, and $\mathbf{n}(t)$ is the noise vector. Since the speech signal is not narrowband, we cannot use the approximation $s(t-T) \approx s(t)$ that we have used in the previous assignments so far. Therefore, we split the signal into subbands such the narrowband condition holds.

To do so, let us split P samples of data into K frames such that each frame has $L = \lfloor \frac{P}{K} \rfloor$ samples. The signal at microphone i corresponding to frame k can then be written as

$$\mathbf{x}_{i,k}^T = [x_i[k,1], x_i[k,2], \ldots, x_i[k,L]]^T : \quad 1 \times L.$$

We denote the Fourier transform (obtained using an L-point FFT) $\mathbf{x}_{i,l}^T$ as

$$\tilde{\mathbf{x}}_{i,l}^T = [\tilde{x}_i[k,1], \tilde{x}_i[k,2], \ldots, \tilde{x}_i[k,L]]^T : \quad 1 \times L,$$

where each entry corresponding to a frequency f_l can now be treated as a narrowband signal and a delay of a narrowband signal $\tilde{x}_i[k,l]$ can now be treated as a phase shift. Specifically, we can write the array output at frame k and frequency f_l as

$$\tilde{\mathbf{x}}_l[k] = \begin{bmatrix} \mathbf{a}_l(\theta_1) & \mathbf{a}_l(\theta_2) \end{bmatrix} \begin{bmatrix} \tilde{s}[k,l] \\ \tilde{v}[k,l] \end{bmatrix} + \tilde{\mathbf{n}}_l[k],$$

where $\mathbf{a}_l(\theta)$ is the array response vector of the ULA towards the direction θ at frequency f_l, and is given by

$$\mathbf{a}_l(\theta) = \begin{bmatrix} 1 \\ e^{-j2\pi \frac{d\sin(\theta)}{c} f_l} \\ \vdots \\ e^{-j(M-1)2\pi \frac{d\sin(\theta)}{c} f_l} \end{bmatrix} : \quad N \times 1. \tag{2}$$

We are required to design a beamformer $\mathbf{w}_l$ for each frequency f_l to suppress the interference and recover the desired signal (i.e., lecture) as $\tilde{y}[k,l] = \mathbf{w}_l^H \tilde{\mathbf{x}}_l[k]$. Finally, the desired time-domain signal of the kth frame, denoted by $\mathbf{y}[k] : L \times 1$, is computed using an L-point IFFT of $\tilde{\mathbf{y}}[k] = [\tilde{y}[k,1], \tilde{y}[k,2], \ldots, \tilde{y}[k,L]]^T$.

Load the data-set `data_partB.mat`, which contains a single variable `X_time`. This dataset contains 4 second audio recording recorded at a sampling rate of $F_s = 8$ kHz. `X_time` is a 3D tensor of size $10 \times 64 \times 500$ containing the output of a ULA with $N = 10$ elements having an inter-element spacing of $\delta = 5$ cm. We have $K = 500$ frames with $L = 64$ samples in each frame.

1. Extract the recording related to the first microphone of the ULA. Play (the real part) recording at microphone 1 as an audio file. Verify that it is a mix of laughter and lecture, which goes like "the discrete Fourier transform...".

2. Create a function to generate the frequency domain array response vector $\mathbf{a}_l(\theta)$ given in (2)

   ```
   function a_l = gen_a_wideband(M,delta,theta,f_l)
   ```

   Use speed of sound, $c = 340$ m/s.

3. The source (speech) originates from an angle of `theta_sig` $= \theta_1 = 5°$ and the interference (laughter) from `theta_inter`$= \theta_2 = 30°$. Design a matched filter beamformer for each bin and obtain the output of the matched filter beamformer

   ```
   y = mf(X_time,F_s,theta_sig)
   ```

   Use `F_s = 8 kHz`. Play `y` as an audio file and comment on the output.

4. Derive a LCMV beamformer which produces a distortionless response towards `theta_sig` and places a null at `theta_inter`. Formulate and solve the optimization problem for each bin separately and mention the parameters of the equality constraint $\mathbf{C}$ and $\mathbf{f}$ for each bin. Also explain how can we compute the desired covariance matrices using `X_time`. Write a program to implement LCMV beamformer

   ```
   y = lcmv(X_time,F_s,theta_sig,theta_inter)
   ```

   Use `F_s = 8 kHz`. Play `y` as an audio file and comment on the output. Compare the performance to that of matched filter beamformer.

5. Derive an LMS variant of the LCMV beamformer that is referred to as Frost LMS. Note that we will be having L different beamformers corresponding to each bin. The beamformers for different bins are to be updated as we proceed from one frame to the next using an update equation of the form

   $$\mathbf{w}_l^{(k+1)} = f_l(\mathbf{w}_l^{(k)}) + \mathbf{w}_l^{\star}$$

   where that k is the iteration/frame index, l is the frequency bin index, $f_l(\cdot)$ is the function that you need to compute, and $\mathbf{w}_l^{\star}$ is the minimum norm solution. Give $f_l(\cdot)$ and $\mathbf{w}_l^{\star}$. Write a program to implement the frost LMS beamformer

   ```
   y = frost_lms(X_time,F_s,theta_sig,theta_inter,mu)
   ```

Use `F_s = 8 kHz` and an appropriate `mu`. Play `y` as an audio file and comment on the output. Compare the performance to that of the matched filter and LCMV beamformer.

6. Now assume that we made an error in estimating the true angles of the desired source. Compute the outputs of the matched filter, LCMV, and Frost LMS beamfomers with a mismatched angle `theta_sig` $= 8°$ (instead of $5°$) and comment on the performance.