ECSE 512 – Digital Signal Processing
McGill University, Fall 2019
Term Project

Overview:
Working alone or in team of two, students will be required to complete a term project on a selected topic/algorithm as described below. The aim of this activity is to help students expand their knowledge/understanding of DSP and gain practical experience through the solution of an engineering problem. Realization of the project will involve the following steps:

- Reading related scientific/engineering literature;
- Developing and implementing algorithm in MATLAB;
- Testing and evaluating algorithm;
- Writing a short technical report.

A detailed project description for this year, along with important related information on report format, deadlines, etc., are provided below.

Project description:
This project involves implementing an adaptive filter for suppressing a narrowband interference signal from a wideband signal.

Assume that a discrete-time signal $x[n]$ consists of a desired wideband signal, say $s[n]$, contaminated by an additive narrowband interference signal $i[n]$, where in theory the two signals are uncorrelated. Our goal is to extract the desired $s[n]$ from the observation of $x[n]$ without explicit knowledge of $i[n]$, a task well-suited for an adaptive filter. This type of problems occurs in various signal processing applications, including audio filtering, digital communications, biomedical signal enhancement, etc.

From a filtering viewpoint, the aim is to design an FIR filter that suppresses the narrowband interference without affecting the desired signal. The desired filter should place a notch in the narrow frequency band occupied by the interference. In practice however, the exact location and width of this band might be unknown, or even change over time. Hence, it is not possible to design a fixed filter for this application and one should use an adaptive filter that can learn and adjusts its parameters in real-time based on the observed signals.

The general system configuration for the suppression of narrowband interference from a wideband signal with an adaptive filter is illustrated in Fig. 1. Due to its narrowband nature, the interference signal $i[n]$ is highly predictable and can be estimated from its past samples using the adaptive filter. The estimated value is then subtracted from the observed signal $x[n]$ to generate the desire estimate. In contrast to the interference, the desired signal $s[n]$ will rapidly decorrelate over time due to its wideband nature. Hence, by introducing an adequate processing delay prior to the adaptive filter, we can prevent the latter from cancelling the desired signal. In practice, the delay n_D should be sufficiently large so that $s[n]$ and $s[n-n_d]$ are uncorrelated.

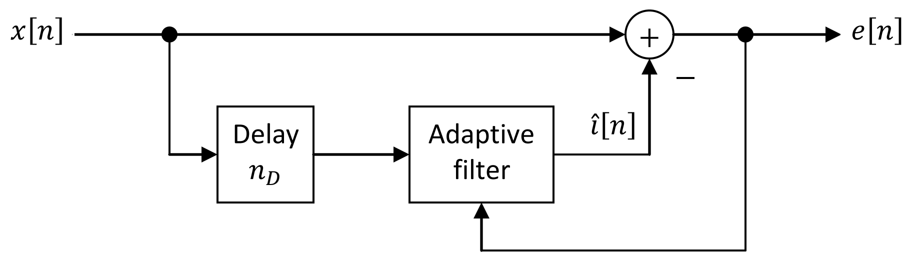

Fig. 1 Adaptive filtering structure for narrowband interference suppression.

The output of the adaptive filter, which provides the estimated interference signal, is given by

$$\hat{\imath}[n] = \sum_{k=0}^{M} h[k,n]x[n-k]$$

where $h[k,n]$ is the filter time-varying impulse response. The latter is adjusted in real-time according to a certain algorithm based on the error signal $e[n]$. In essence, the goal of an adaptive filter is to minimize the power of the error signal, but this can be done in different ways, leading to different adaptation algorithms. For instance, in the case of the least-mean-square (LMS) algorithm, the filter update takes the form

$$h[k,n] = h[k,n-1] + \mu e[n]x[n-k-n_D], \qquad k = 0,\cdots,M$$

where $\mu > 0$ is the so-called step size which controls the memory of the algorithm. For the recursive least-squares (RLS) algorithms, the update takes a more complex form but also involve a kind of memory paremter.

The main goals of the project include the following:

- Implementing the above structure in Matlab using at first the LMS algorithm and subsequently, RLS algorithms whose operation will be discussed in class (see also [1]).
- Demonstrating the correct operation of the adaptive filters in a stationary environment. To this end, you will need to run the algorithm on simulated test signals and plot the so-called learning curves, i.e. power of error signal versus time averaged over a number of independent runs. From these curves, you should be able to observe the convergence rate of the adaptive filter as a function of its internal memory parameter. For these experiments, you could generate the required test signals as follows, but you are welcome to investigate other options:
  - Desired signal $s[n]$: zero-mean Gaussian white noise.
  - Narrow-band interference $i[n]$: deterministic sin wave.
- Demonstrating the operation of the adaptive filters in a non-stationary environment by separately considering the following scenarios:
  - Use of a speech signal for $s[n]$ along with deterministic sin wave $i[n]$ with fixed parameters.
  - Use of a white Gaussian noise for $x[n]$ along with sin wave $i[n]$ with time-varying frequency.
- In your discussion, you should investigate the effects of various parameters (memory factor, SNR) on the convergence behavior of the adaptive filters, and also compare the results obtained with the LMS and RLS algorithms.

Report:

Your report should contain the following elements:

- A cover page with a short abstract.
- A table of content
- Section 1: An introduction providing (2 pages max):
    - A high level overview of the project topic (problem, applications, existing solutions) along with references.
    - Specific goals of the project and brief overview of the report's content.
- Section 2: Explanation of background theory with necessary equations (3-4 pages).
- Section 3: Description of your specific approach, algorithms and/or design choices (2-3 pages).
- Section 4: Results (figures, data/audio files, etc.) and relevant discussions (3-4 pages).
- Section 5: A brief conclusion (.5 page max).
- List of references, presented professionally.
- An appendix that briefly explains how to use your program (1 page max). See also Important note below.

You should follow the guidelines below to ensure a professional and uniform presentation:

- Use 8.5 x 11 paper, single column (portrait) mode, with black print.
- Set top margin to 1.25in and side/bottom margins to 1in.
- Use Times New Roman, size 12 pt font, with 1.5 line spacing.
- Your report should include a cover page, 10-12 pages of descriptive text (including appendix as explained above) and listing of your **Matlab code** with adequate comments.
- Except for cover page, include page number at bottom.
- On cover page, include: project title; names and ID; brief abstract (5-6 lines), date, etc.
- Use a professional word processor, with equation editing and referencing capabilities.

Due date:

The report and related information are due **Wednesday, Dec. 4th, before 4h00pm**. This includes:

- Final printed report (see formatting info above).
- USB containing: report file in pdf, Matlab code of your algorithm, selected test (input) and processed (output) files to support discussion in Section 4 of your report.

Important note:

I should be able to open your main Matlab file, run it on a provided data file, and observe or listen to the processed outputs flawlessly (common mistake includes forgetting required sub-directory, missing .m files, etc.). Please test on a different computer with a friend before submitting. If I cannot run your code, you will lose some marks.

[1] S. Haykin, Adapttive Filter Theory, 4^{th} Ed., Prentice Hall, 2002.