# Part IIA Project - SF1: Data Analysis - Second Interim Report

Bailey Brookes — Corpus Christi — bdb31

May 25, 2018

## 1 overlap_add.m

Below are answers to the questions posed in the handout about the `overlap_add.m` file.

(a) This script is a simple attempt at the Wiener filter but with constant filter gains of 0.1 and 0.2. Also, this filter use the overlap add technique.

(b) Weiner filters can be implemented in time or frequency domain. The multiplication of gains in frequency is the same as convolving them in time, meaning filtering in frequency or time produces the same results.

(c) Direct convolution filtering has complexity of $O(N^2)$ while FFT convolution has complexity of $O(N \log N)$. When the number of samples, N, per frame is 516 this means FFT convolution is:

$$\frac{N^2}{N \log N} = \frac{516^2}{516 \log 516} \approx 57.3 \text{ faster}$$

(d) `conj` takes the complex conjugate of a complex number so as the FFT is symmetric, only one half needs to be filters (0 to $\pi$) and then the second half ($\pi$ to $2\pi$) is the complexity conjugate of the first half.

## 2 Noise Reduction

### 2.1 Basic de-noising: bdb31_noiseReduction.m

`bdb31_noiseReduction.m` is the script that de-noises the required 4 audio files using a basic overlap add Weiner filter. It predicts the power spectrum of the noise using the initial silent portion of the audio, and zero pads each frame to increase precision without adding too much run time. Zero padding reduces the digital distortion resulting from filter gains of zero, which result in sharp jumps in the time domain which is head as digital noise.he resulting filtered audio is good, with minimal digital noise and almost all the random noise from the file removed.

The mean squared error (MSE) is calculated and presented after each run. The MSE however doesn't fully convey the quality of the filtered audio. For example, before zero padding there is a lot of digital distortion but the MSE result is very small, even though the audio is of a very poor quality. Furthermore, smoothing the signal using Matlab `smooth` also decreases the MSE but the audio becomes considerably quieter and 'merky', like it's being heard through a wall. Hence, zero padding is used without smoothing.

In deciding the parameters, changing the frame length has an interesting effect on the MSE, as show in Figure BLANK. MSE is almost periodic and overall lower at shorter frame lengths. However to smaller frame length result in audible distortion so a trade off is need. For my filter, a frame length of 256 and overlap 128 is used a hann. window

### 2.1.1 Time varying noise

For the two time varying noise files, the basic Weiner filter works well. This is because the power spectrum of the noise is calculated from the variance of the initial silent bit of the audio, which takes into the account the changing noise level. This is very basic and a better filter would calculate the power spectrum of the noise for each frame and use that for the filter coefficients for each frame.

## 2.2 De-noising files without a silent portion: `bdb31_noSilenceReduction.m`

Some files, such as `Fast_N.wav` don't have a silent portion from which the variance of the noise can be estimated. A different method therefore must be used.

One method is to smooth the original data and then subtract this smoothed data from the original data to get an estimate of the noise. This is done for each individual frame and produces a file with the noise removed but more digital noise is added than in the simple de-noising case that estimates the noise level using the silent portion.

### 2.2.1 Time-varying noise

The time varying noise files where also filtered using this method for comparison with the silent estimate case. While this method estimates the noise better, there is more digital distortion caused using the smoothed technique. Both are presented so methods can be compared.

# A How audio files where filtered

The file are saved using the following naming convention so the filter used is clear:

`orginal_file_name.wav_filtered_method.wav`

Where `method` tells you how the noise was estimated either using the silent portion method, denoted `silent`, or smoothing each frame and subtracting the smoothed version from the original to estimate the noise, denoted `smooth`.

# B Plots

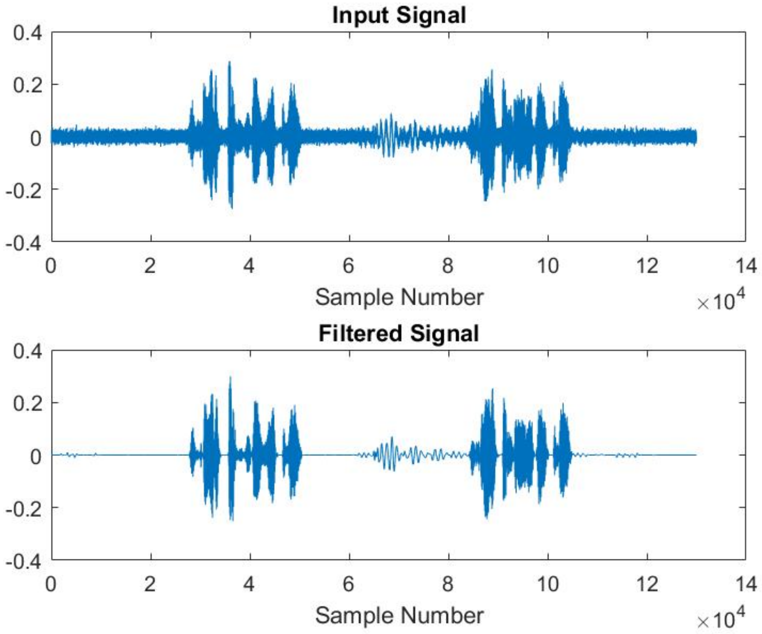

Figure 1: Plot of Input and filtered signal of male_speech_noisy_loud.wav. The script also produces plots for all files it filters.

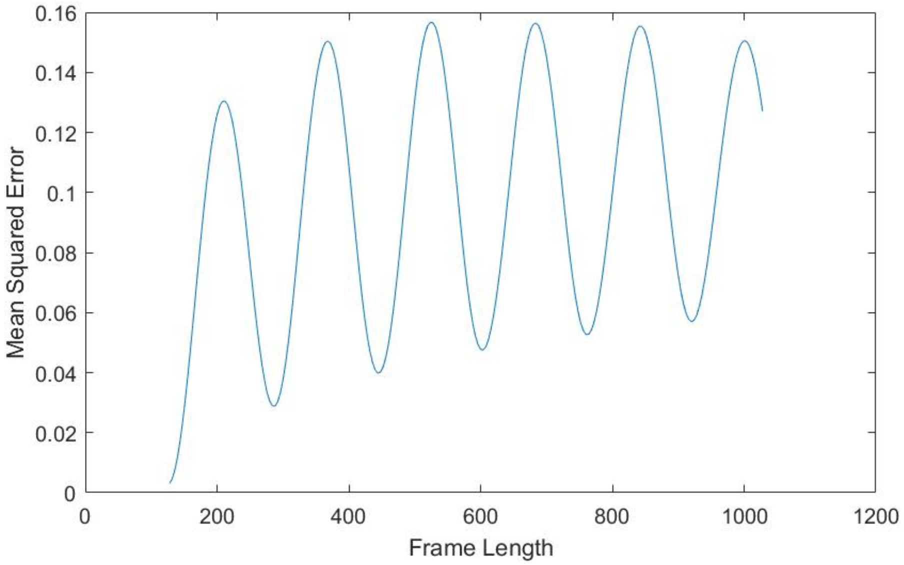

Figure 2: Plot of Mean squared error vs frame length for the overlap add Weiner filter