Selected Topics in Signal Processing

442.003 Digital Signal Processing, Laboratory
Winter Term 2020/21

Signal Processing and Speech Communication Laboratory
`www.spsc.tugraz.at`

Last updated: October 28, 2020

Abstract

Noise reduction is a notorious challenging task. Throughout this laboratory we deal with noise reduction where the desired signal is a speech signal deteriorated by additive, undesired noise, e.g. somebody talks (the speech signal) at a heavy traffic place (the noise) using a mobile phone. We want to reduce the traffic noise and keep pure speech signals. This application of noise reduction is often denoted as speech enhancement. In the first experiment you are encouraged to implement a Wiener noise reduction filter using Matlab. In the second experiment you will implement an online noise reduction filter.

1 Liner noise reduction filter

In this laboratory we treat Wiener filters for noise reduction. Using time-discrete notation with n as time index, we observe a signal $y[n]$ with length N which is composed of the desired signal $s[n]$ and additive noise $w[n]$

$$y[n] = s[n] + w[n]. \tag{1}$$

In order to estimate $\hat{s}[n]$ given the observation $y[n]$ we apply the Wiener filter $g[n]$ resulting in

$$\hat{s}[n] = (g * y)[n] \tag{2}$$

which may be written in discrete Fourier domain as

$$\hat{S}[k] = G[k]Y[k] \tag{3}$$

with $k \in [0, N-1]$. The Wiener filter $G[k]$ minimizes the squared error between $S[k]$ and $\hat{S}[k]$ resulting in

$$G[k] = \frac{|S[k]|^2}{|S[k]|^2 + |W[k]|^2} \tag{4}$$

where $|S[k]|^2$ and $|W[k]|^2$ correspond to the power spectral density (PSD) of the speech and noise signal. At this point we may identify (at least) two challenges: both $|S[k]|^2$ and $|W[k]|^2$ are unknown. $|S[k]|^2$ captures speech (the spoken words) whose characterization is not straight forward. Furthermore, the PSD of speech changes within fraction of seconds. The second term $|W[k]|^2$ models the additive noise, strongly dependent on the environment (street, people talking in the background, construction place) and not simple to characterize as well.

In this laboratory unit, we shall assume that we have access to two microphones 1 and 2 which record the signals $y_1[n]$ and $y_2[n]$. Microphone 1 is located very close to the talking person and records both speech and noise, hence $y_1[n] = y[n]$, as modeled in (1). Microphone 2 is separated from the talking person and we assume that this microphone records noise only $y_2[n] = w[n]$.

Question: If Microphone 1 records speech plus noise and Microphone 2 records noise only, what are the challenges to recover $\hat{s}[n] = y_1[n] - y_2[n]$?

To calculate (4) we shall assume that $s[n]$ is stationary within a short fraction of time, say 20 ms. Utilizing the framework of short-term Fourier transform, $y_1[n]$ and $y_2[n]$ can be divided into short terms, denoted as frames. Subsequently, $Y_1[k]$ and $Y_2[k]$ are calculated for each frame individually and (4) is calculated for each frame individually as well. Finally, we use overlap-add to put the individual estimates $\hat{s}[n]$ of each frame back to a continuous time signal.

2 Experiment 1: Implement a noise suppression

Equipment: PC, SPSC iMobile
Software: MATLAB, download and unzip `Unit5.zip` from www.spsc.tugraz.at

1. Connect the Microphones of the SPSC iMobile to the Desktop Computer using the Microphone Front Input.

2. Open `WF.m` in Matlab.

3. Provide a noise source (either from a loudspeaker or somebody who 'simulates' a passing car) and execute the Matlab script. The script records 5 seconds of data. Within this 5 seconds you are required to use the SPSC iMobile (talk something). Talk very silent but do not whisper!

4. Plot both signals $y_1[n]$ and $y_2[n]$, the time domain should report time in seconds, not samples. *Question:* Can you identify your spoken words in the plots?

5. The Matlab script has implemented an overlap add procedure using an overlap of 50 %. The overlapping 'frames' are stored in `xs` and `xn`. Calculate the short-term Fourier transform (STFT) of both signals and plot the signals using `imagesc` in dB scale. *Question:* Can you identify your spoken words in the plots?

6. Replace the gain function by your own implementation of a Wiener filter based on [1], Equation (12). Visualize the STFT `Xs` and `Xr` in dB scale. Report the differences! Listen to the processed output file.

7. Constrain the gain function to a minimum value of $G_{\text{min}} = -20$ dB and a maximum value of $G_{\text{max}} = 0$ dB. Again, visualize the STFT `Xs` and `Xr` in dB scale and explain the differences.

8. Listen to the processed output file. Can you hear a difference in comparison to the previous task?

9. The outcome of the Wiener filter contains 'musical noise' which stems from sudden changes in the gain function. Reduce this sudden changes by considering a recursive smoothing applied to the gain function. Apply the filter $G[k, n] \leftarrow aG[k, n-1] + (1 -$

$a)G[k,n]$ where $G[k,n]$ denotes the STFT of the gain function and $a = 0.9$. Again, visualize the STFT `Xs` and `Xr` in dB scale and explain the differences. Listen to the processed output file. Can you hear a difference in comparison to the previous tasks?

3 Experiment 2: Online noise suppression implementation

Equipment: PC, SPSC iMobile
Software: Netbeans, download and unzip `Unit5.zip` from www.spsc.tugraz.at

1. Open Exp1_NoiseSuppression in netbeans. The overlap add procedure is implemented already. In line 68 in file main.cpp you may find that each two seconds either the input is forwarded to the output (line 69) or the input is forwarded to object WF (the Wiener filter). Thus, the Wiener filter is switch on/off each two seconds such that you can easily recognize the difference. Go to the Wiener filter object WienerFilter.cpp. Example code how to use the Fourier and inverse Fourier transform is provided. Note, the spectral length for a input signal with length N is $N/2 + 1$!

2. Implement your filter step by step. The filter object WF is executed at each frame. If you are required to save a previous frame internally, introduce a new variable in WienerFilter.h, equivalent to the variables specreal and specimag.

3. Test your filter using headphones. Ask your neighbor to 'use' the SPSC iMobile!

A Credits

This document was authored by Josef Kulmer.

References

[1] R. McAulay and M. Malpass, Speech enhancement using a soft-decision noise suppression filter, IEEE Transactions on Acoustics, Speech, and Signal Processing, 1980