Adaptive Filters

Tutorial 1

TECHNISCHE UNIVERSITÄT DARMSTADT

Prof. Dr.-Ing. H. Puder
Adaptive Systems for Processing of Speech and Audio Signals
Graduate School Computational Engineering

sobre overlap: quanto menor for o frameshift, mais frequentemente vc calcula os coeff do seu filtro
sem overlap at all, o resultado ficaria uma porcaria - so o janelamento ia fazer ficar lixoso

Problem 1 Wiener Filter

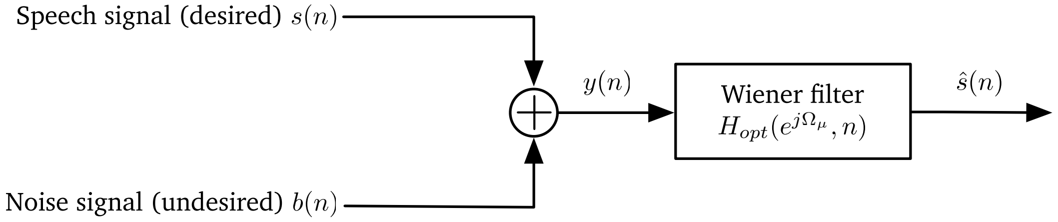

Figure 1: Noise reduction scheme with Wiener filter.

Consider a noise reduction scheme as depicted in Figure 1, where $s(n)$ denotes the speech signal and $b(n)$ the noise signal. The clean speech signal $s(n)$ is estimated based on the noisy signal $y(n)$ by the Wiener Filter $H_{\text{opt}}(e^{j\Omega_\mu}, n)$ where $\hat{s}(n)$ denotes the estimate. All signals have the length N.

a) Write a function `generate_input` which loads the speech signal `speech.wav` and the noise signal `car_noise.wav`. The function should also generate the noisy signal $y(n)$ given as

$$y(n) = s(n) + c \cdot 10^{-\frac{\text{SNR}}{20}} \cdot b(n),$$

where SNR denotes the signal-to-noise ratio in dB. The constant c normalizes the SNR and is given by

$$c = \sqrt{\frac{\sum_{n=1}^{N} s^2(n)}{\sum_{n=1}^{N} b^2(n)}}.$$

The function should have the SNR as input and should return $y(n)$, $b(n)$, $s(n)$ and the sampling frequency of the signals.
For comparison, plot $s(n)$ and $y(n)$ with an SNR at 10 dB and listen to the signals.
Hint: Wav-files can be loaded using the MATLAB function `audioread`. You can listen to signals (vectors) by using the MATLAB function `soundsc`, which expects an input sound vector and its corresponding sampling frequency.

b) Estimate the spectra $Y(e^{j\Omega_\mu}, n)$ and $B(e^{j\Omega_\mu}, n)$ of the noisy signal $y(n)$ and the noise signal $b(n)$ using the given function `estimate_spectrogram`. Then write a function `estimate_PSD` which returns the estimated short term power spectral density (PSD) of a signal. As a reminder the short term estimate of the PSD is the periodogram of a signal $x(n)$, given as

$$\hat{S}_{xx}(\Omega_\mu, n) = |X(e^{j\Omega_\mu}, n)|^2.$$

Use the function `estimate_PSD` to estimate the PSDs $\hat{S}_{yy}(\Omega_\mu, n)$ and $\hat{S}_{bb}(\Omega_\mu, n)$ of $y(n)$ and $b(n)$.
With the given function `plot_PSD` you are able to visualize a PSD. Use this function to plot $\hat{S}_{yy}(\Omega_\mu, n)$ and $\hat{S}_{bb}(\Omega_\mu, n)$ again with an SNR $= 10$ dB and interpret what you observe.

c) Write a function `wiener_filtering` which calculates the Wiener filter given by

$$\hat{H}_{\text{opt}}(e^{j\Omega_\mu}, n) = \max\left\{1 - \frac{\hat{S}_{bb}(\Omega_\mu, n)}{\hat{S}_{yy}(\Omega_\mu, n)}, H_{\text{min}}\right\},$$

where H_{min} is a constant and denotes the maximum attenuation of the filter. Additionally, perform the filtering

$$\hat{S}(e^{j\Omega_\mu}, n) = \hat{H}_{\text{opt}}(e^{j\Omega_\mu}, n) \cdot Y(e^{j\Omega_\mu}, n).$$

Your function should have the spectrum $Y(e^{j\Omega_\mu}, n)$, the two short term PSDs $\hat{S}_{bb}(\Omega_\mu, n)$, $\hat{S}_{yy}(\Omega_\mu, n)$ and the maximum attenuation H_{min} in dB as inputs. Return the result of the filtering $\hat{S}(e^{j\Omega_\mu}, n)$, i.e. the estimated spectrum of the clean signal.
Hint: Use a for-loop in order to calculate $\hat{H}_{\text{opt}}(e^{j\Omega_\mu}, n)$ for each time instance n.

d) The given function `calculate_output` transforms $\hat{S}(e^{j\Omega_\mu}, n)$ into the time domain and calculates the output signal $\hat{s}(n)$. Use this function and all the previous functions to write a main function, which estimates the spectra and short term PSDs, calculates the Wiener filter, performs the filtering and calculates the estimate $\hat{s}(n)$. As inputs use the SNR and H_{min} (both in dB). Mind that the maximum attenuation in dB corresponds to the negative minimum amplification in dB. The function should return the noisy signal $y(n)$, the estimated speech signal $\hat{s}(n)$ and the sampling frequency of the signals.
Use an SNR again of 10 dB and try three different maximum attenuation values H_{min}, namely 5 dB, 15 dB and 50 dB. Plot $\hat{s}(n)$ and listen to it. What can you observe? Which is the best choice for H_{min}?

Until now we considered that the noise signal is known. However, in reality this is not the case and now we need to estimate the PSD of the noise based on the noisy input signal. To estimate the short term PSD of the noise, we use the minima tracking of the short term PSD of the noisy input:

1) Smoothing:

$$\overline{S_{yy}(\Omega_\mu, n)} = \beta \overline{S_{yy}(\Omega_\mu, n-1)} + (1-\beta)\hat{S}_{yy}(\Omega_\mu, n)$$

2) Minimum value, with a slight increase to avoid a freezing of the estimate:

$$\hat{S}_{bb}(\Omega_\mu, n) = \min\{\overline{S_{yy}(\Omega_\mu, n)}, \hat{S}_{bb}(\Omega_\mu, n-1)\}\,(1+\epsilon) \qquad \text{with } \epsilon \ll 1.$$

As initialization, use $\overline{S_{yy}(\Omega_\mu, n)}$ as estimate for the first 10 samples of $\hat{S}_{bb}(\Omega_\mu, n)$.

3) By inserting a fixed overestimation K_{over} musical noise can be reduced:

$$\hat{S}_{bb}(\Omega_\mu, n) \longrightarrow K_{over}\hat{S}_{bb}(\Omega_\mu, n)$$

e) Write a function `estimate_noise_PSD` which calculates the estimated PSD $\hat{S}_{bb}(\Omega_\mu, n)$ according to the previous steps 1)-3). The function should use as inputs $\hat{S}_{yy}(\Omega_\mu, n)$, β, ϵ and K_{over}.
Set $\beta = 0.95$ and $K_{over} = 6$, for ϵ use 0.1 and 0.001. SNR is still 10 dB. Plot $\overline{S_{yy}(\Omega_\mu, n)}$ and the resulting short term noise PSD estimates $\hat{S}_{bb}(\Omega_\mu, n)$ for a frequency value of choice. If `estimate_noise_PSD` is implemented correctly, the estimates $\hat{S}_{bb}(\Omega_\mu, n)$ are always smaller than or equivalent to $\overline{S_{yy}(\Omega_\mu, n)}$. Additionally plot the estimated PSD of the noise for each choice of ϵ. What happens for the two choices and which ϵ would you choose for the estimation of the noise PSD?

f) Modifiy your main function of d) such that the noise PSD estimation of e) is applied where $b(n)$ is not assumed to be known and perform the Wiener filtering again. Use the following setting: SNR $= 10$ dB, $H_{min} = 15$ dB, $\beta = 0.95$, $\epsilon = 0.001$ and $K_{over} = 1$. Listen to the result. Now set $K_{over} = 6$ and listen again. Then use an overestimation of $K_{over} = 50$. Comment on what you hear for the different choices. Which is the best choice for K_{over} in this example?