logMMSE

# Statistical speech enhancement systems

Below is a general system of a statistical speech enhancement system which is transform-based and single microphone.

For the outer system, the time domain signal, it is first framed and windowed. In general these frames are overlapping. Then a transform, typically FFT is applied to each and every frame. After the processing system, the gain is applied to the transform coefficients and the inverse transform is performed followed by an overlap-and add method that is to compensate for the windowing and framing.

The observed signal is given by

MMSESTSA(1)

Fourier with time and sub-bin index

Where k denotes fourier expansion index and l denotes time index.

And the problem here is to estimate the fourier expansion coefficients of the speech given our noisy speech sequence.

We also assume that voice is not always present, but that noise is almost always present.

The main problems to be solved in this type of system are that of estimating the target PSD, i.e a priori SNR, estimating the noise PSD estimate and to find a suitable statistical assumption for the speech and noise.

# logMMSE

In our case, we assume that the fourier expansion coefficients can be modelled as statistically independent Gaussian variables. We assume that they are both independent in time and in fourier expansion index. This is motivated by the fact that the fourier expansion coefficients are coefficients that comes from weighted random time samples and through the central limit theory, these should approach the Gaussian variables. These assumptions might not be entirely correct as the windows are overlapping and that there will be dependencies across fourier expansion coefficients due to windowing etc.

The problem here is to estimate the fourier expansion coefficients using the logarithmic Minimum Mean Square error function

logMMSE(1)

This estimator is given by

logMMSE(2)

logMMSE(3)

And through the Gaussian model our probability distribution functions are:

logMMSE(8)

logMMSE(9)

and finally through a whole lot of calculations, we end up with the gain function as follows

logMMSE(20)(divided by Rk)

The remaining problem is that of estimating the *a priori* SNR and estimating the noise variance in each frequency bin.

For estimating the noise variance, we use a so-called Voice Activity Detection that tells us when there is no voice present, and that enables us to estimate the only the noise variance as it is only the noise that is present. During moments when there is speech and noise, the noise variance of the past when there is only noise present is used. This is valid if the noise does not change to dramatically.

The Voice Activity Detection works by comparing the spectral distance between the noise and the speech.

For estimating the *a priori* SNR, we use a popular technique which is called *decision-directed a priori SNR estimation*.

logMMSE(51), logMMSE(52)

And in our case alpha is 0.99

This is one of the simpler ways to estimate the a priori SNR and a very effective one.