

Lab 6: Denoising

In this lab we will use denoising to clean up some noisy recordings. We will use the provided sounds `room-speech.wav`, `aircomm.wav` and `wind-speech.wav` in the lab archive.

Part 1. Cleaning up the sounds

Let's start by cleaning up these sounds as best as we can. We will do a straightforward magnitude spectral subtraction. For all of these sounds there is only noise in the first few seconds of the recording so that you can learn a noise profile from there. Do the following:

1. Perform an STFT of the recordings
2. Estimate the magnitude spectrum of the noise from the beginning of the recording
 - It's up to you to figure out how many seconds to use (hint: look at the spectrogram)
3. Perform spectral subtraction by subtracting that spectrum from the input's magnitude STFT
 - Remember to clip any resulting negative values to zero
 - Try to find how much of the noise to subtract so that the output looks good
4. Use the original signal's phase to convert back to a time series.

Make a note of which examples sound the best and are easier to work with. Try to explain why.

At this point some of the outputs will exhibit "musical noise". To minimize its effects apply a median filter on the denoised magnitude spectrogram to make it sound better (hint: `scipy.signal.medfilt2`). How big should the median window be? Try different values and find which work best.

Part 2 (optional). Implement a Voice Activity Detector (VAD)

For the last sound we have an evolving noise profile, which causes problems since our noise description from the first two seconds isn't accurate throughout. Because we're lazy we want to automatically update the noise model and not to select it manually. To do so we need a VAD that lets us know when to gather noise statistics and when to denoise. Do the following:

1. Take the square of the input waveform and lowpass filter it (a lot) to get an energy level over time
 - Experiment with the cutoff frequency so that you get a smooth energy-looking function
2. Set a threshold over which we seem to have speech in the input
3. Implement a real-time spectral subtraction denoiser
 - If an input frame is under threshold, it is noise
 - Keep track of the last few noise frames and their average amplitude will be your noise spectrum
 - If an input is over the threshold it is speech
 - Once you encounter speech perform spectral subtraction with the current noise estimate