

ECS730 Digital Audio Effects

Assignment 1: Audio Effect Analysis

Instructions:

Analyse **four** audio effects from the VST examples provided. An archive containing all examples can be downloaded from the ECS730 QMplus page (in “Code and Other Materials”).

The goal of your analysis should be straightforward: to demonstrate that the effect works properly through the use of audio examples and visualisations. Run each effect on an audio sample of your choice. The output should be saved as an audio file (*listen* to make sure it works!) that can then be analysed to generate figures (in MATLAB or any other method of your choosing).

Analysis should be in the form of visualizations that show that the effect produces the desired modification of the audio signal, plus some text explanation. The visualisations may be time domain waveforms of the input and output, spectrograms, power spectrums, time domain envelopes, etc. Choose the appropriate input signals and visualizations to clearly show the effect.

You are free to modify the example code however you like provided you document your changes. Since the primary purpose is analysis rather than creation, this is not expected however.

Turn In:

Using the online submission system, submit a ZIP archive containing:

- **Audio files** before and after the effect for each of the four audio effects (8 total unless the same audio is used as input for multiple effects).
- **Code** used to run the effects. Include any modifications you made to the code.
- **PDF Report**, 4 sides of A4 (one for each effect), containing visualisations and a one to three paragraph explanation of why your choice of figures demonstrates the effect's proper functioning. The explanation should make clear that you understand the basic operation of the effect and any particularly salient parameters. A detailed mathematical analysis is not necessary.

Helpful MATLAB Functions: *(use the 'help' function to learn more)*

`[y, fs, nbits] = wavread('filename')` *Read a WAV file from disk*

`wavwrite(y, fs, nbits, 'filename')` *Write a WAV file to disk*

`sound(y, fs)` *Play a vector as sound with sample rate fs*

`plot()`, `semilogx()`, `semilogy()`, `loglog()` *Linear and log-scale plot functions*

`fft(x, n)` *Calculate the FFT of length n. You might want to window the signal first. Also note that the fft is complex. Use `abs(fft(...))` to get the magnitude. A discussion of using `fft()` to generate spectra can be found [here](http://www.mathworks.co.uk/support/tech-notes/1700/1702.html):*

<http://www.mathworks.co.uk/support/tech-notes/1700/1702.html>

`spectrogram(x, window, 'yaxis')` *Plot the spectrogram of x. If window is an integer, x is divided into that many evenly-spaced segments. 'yaxis' places frequency on the y axis and time on the x axis, as it often appears in audio analysis*

`(0:(length(x)-1))/fs` *For a signal x and sample rate fs, go from sample number to time. Useful for the x-axis of time-domain plots.*