ELEC3104: Digital Signal Processing

PROJECT – S1, 2017

This project is an important part of the laboratory component of this course and is designed to focus on the application of the theory you learn in the lectures and tutorials to practically implement DSP systems. You can work on this project in your own time and successful completion will require independent learning on your part.

1. The project comprises of two stages, with both stages marked separately.
2. Stage 1 is due in week 9 and is worth 40% of the total project mark.
3. Stage 2 is due in week 13 and is worth 60% of the total project mark.
4. You should write your own code, you are not allowed to use third party code or built-in MATLAB commands except to interface with any hardware (if required) and for file-handling. If you are unsure ask the lecturer/demonstrators.

Stage 1 (Mark: 40%)

You choose either option A or B for this stage. Option A is easier but the maximum mark you can get for it is 25% of the total project mark. Option B is a little harder but is worth the full 40%.

Option A

Platform: MATLAB

Implement a system that analyses an audio file (.wav format) and visually shows how the energy at each frequency changes with time. You can choose the frequency resolution and the rate at which the energy change is estimated but structure your code such that these are adjustable parameters.

Option B

Platform: MATLAB

Implement a system that analyses audio *in real time* and displays energy at each frequency evolving as a function of time. Frequency resolution and rate at which energy changes are estimated should be adjustable parameters in your code. Audio input should be via the PC mic-in/line-in. In addition, your system should also be able to accept an audio file (.wav format, 16kHz sampling rate) as input and produce the required display as the audio is played in real time via loudspeakers/headphones.

Stage 2 (Mark: 60%)

You choose between options C, D or E for this stage. Option C is the simplest but is only worth 30%; option D is of intermediate difficulty and is worth 50%; while option E is the most challenging and is worth the full 60%.

Option C

Platform: MATLAB

Implement a 5-channel equaliser that can read in an audio file (.wav format, 16kHz sampling rate), split into five frequency bands (channels), apply a desired gain in each channel, reconstruct the audio signal and save it as an audio file. The system should not introduce any perceptible distortion to the audio signal except for the desired gains in each channel. You should choose appropriate frequency ranges for the five channels.

Option D

Platform: MATLAB

Implement a real-time 5-channel equaliser where you can adjust the gain of each channel at any time. The system should read an audio file (.wav format, 16kHz sampling rate) and output via speakers/headphones. The system should not introduce any perceptible distortion to the audio signal except for the desired gains in each channel. You should choose appropriate frequency ranges for the five channels.

Option E

Platform: MATLAB

Same as option D except the input .wav file can have a sampling rate of either 8kHz, 16kHz, 22.05kHz or 44.1kHz. Your system should convert the input (with known sampling rate) to a signal sampled at 16kHz and then pass it through the equaliser and output via speakers/headphones. You should ensure that any aliasing due to resampling is inaudible (to the lab demonstrators' ears) while also ensuring that you do not audibly distort the signal of interest with your anti-aliasing filter.

Assessment: Both stage 1 and stage 2 will be assessed in your scheduled tute-lab class by a panel of two or three lab demonstrators based on a demonstration and answers to questions about your design, implementation and the theory behind it. You will also be asked to submit your code prior to assessment.