MIDDLE EAST TECHNICAL UNIVERSITY

ELECTRICAL and ELECTRONICS ENGINEERING

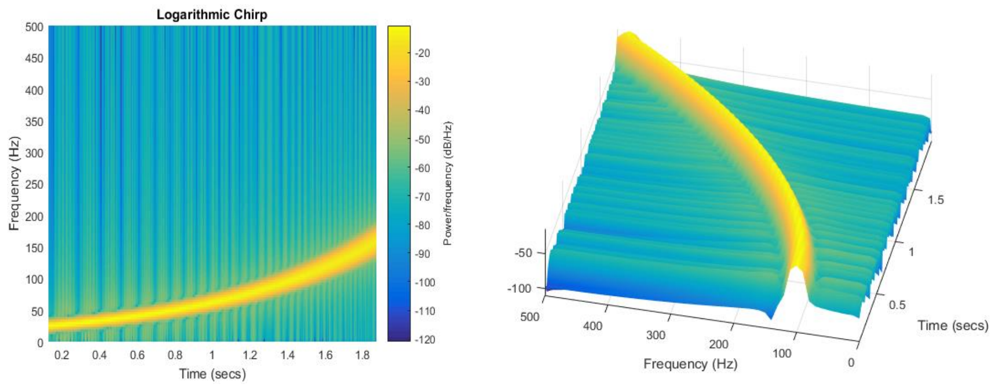
EE 430 TERM PROJECT

In this semester you will construct a system details of which are provided below.

The system should be capable of performing the following tasks:

1) **Data acquisition:** System should capture an audio (or voice) playback from another device (e.g. mobile phone) by the help of a microphone connected to the user computer. Analog-to-digital conversion sampling rate of this process should be adjusted from a user interface. The captured audio input should be played on the speaker of the computer. The system should also be able to process mp3 music files, which means that you must convert mp3 files to a workable format first.

2) **Spectrogram:** System should determine and display the spectrogram of an audio file which is previously captured by a microphone. System should be able to playback and plot the time waveform of the data, while showing the spectrogram. Spectrogram should be displayed both in 2D and 3D (Spectrogram must be obtained without using `spectrogram` command in MATLAB; everyone must write their own code). The variables (such as window size, DFT size, overlap size, etc.) related to spectrogram should be parametric and therefore adjustable by the user. The window function (rect, hamming, etc.) should also be adjustable.



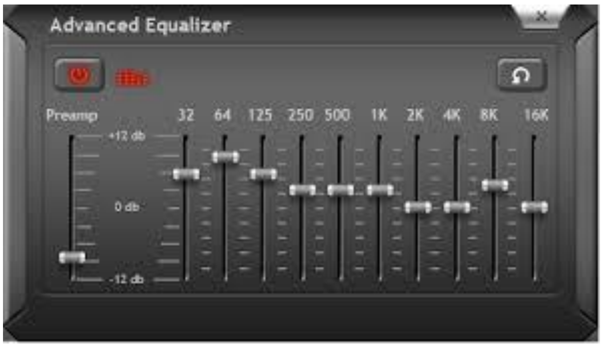
3) **Real-time spectrum:** As another feature, the system should be able to get recorded audio in frames while simultaneosly showing the DFT of the current frame in real time. The variables such as frame size, DFT size, etc. should be parametric and therefore adjustable by the user. The window function (rect, hamming, etc.) should also be adjustable.

4) **Equalizer and filtering:**

a) By the help of a user interface, system should be able amplify or suppress the following frequency bands of input data, working as an audio equalizer:

0-32 Hz
32-64 Hz
64-125 Hz
125-250 Hz
250-500 Hz
500-1000 Hz
1000-2000 Hz
2000-4000 Hz
4000-8000 Hz
8000-16000 Hz

The input and output time waveforms, as well as spectrograms should be displayed. All the frequency selective filters should be selected as linear phase to minimize distortion. Specify the filter design method you use.



b) Now use a filter design method which does not enforce linear phase to design a filter for the 250-500 Hz filter. Compare the outputs of the equalizer when this filter is linear phase and nonlinear phase.

5) **Special effects:** The system must also be capable of creating the following special audio effects. You must develop your own code.
   a) Reverberation
   b) Synthetic stereo
   c) Change (increase/decrease) the speed of the audio by an external factor

Check the following website http://www.mathworks.com/help/dsp/examples/audio-special-effects.html for explanation of these effects.

6) **Interfering tone removal:** The system should be able to add a pure cosine signal to the recorded audio. The frequency of the cosine signal should be specified by user. This will be counted as an additional noise to our original audio signal.

a) Design a filter that will suppress (or filter out) the generated noise above. The system should be able to play the noisy and the filtered audio. Moreover the spectrograms of those signals can be shown on the screen. The final filtered audio is expected to sound similar to the original one.
b) Propose and realize a method that will estimate the frequency of the above noise.

Submission Guideline:

Project submission will be in **2 phases**.

Phase 1: In this phase you are expected to complete the first two steps of the project. You should submit your results in a report and your related m-files through "odtuclass". You should submit **TWO** files:

- Your report in a **SINGLE pdf** file
- Your m-files contained in a **SINGLE zip** file.

Deadline for Phase 1 is 23:55, 11.12.2015.

Phase 2: In this phase you are expected to complete all steps. You will run your code and demonstrate your results through audio data and visual figures in the student PC lab. You may be asked to change parameters of the system during demonstration, therefore you are advised to write your code as parametric as possible. You will submit your **printed final report** containing all the steps and all the details of your work during the demonstration. Moreover again as in Phase 1, just after the demonstration, you will submit through "odtuclass":

- Your final report in a **SINGLE pdf** file
- Your m-files contained in a **SINGLE zip** file.

Demonstrations will be on the last week of January (25.01.2016-29.01.2016)

It is important that you stick to this guideline. For any questions contact:

Oktay Sipahigil (osipahigil@eee.metu.edu.tr)

Erdal Epçaçan (epcacan@metu.edu.tr)