Lab 1: Sampling and Quantization (Audio)

Introduction

The purpose of this lab is for you to understand the principles of sampling an audio signal and image. For audio, the wave file format can be used. The effect of sampling rate conversion and quantization will be investigated. By comparing the sound quality obtained with different filters and quantizers at the same sampling frequency and quantization level, you should have a better appreciation of the differences in performances by using pre-filters and quantizers.

Preliminary

In this section you will need to setup the following schematic diagram using SIMULINK. The objective is to simulate lower sampling rates using the downsampling operator, given that the audio source was initially captured at a relatively high sampling rate. In addition, you will investigate the effect of companding vs uniform quantization.

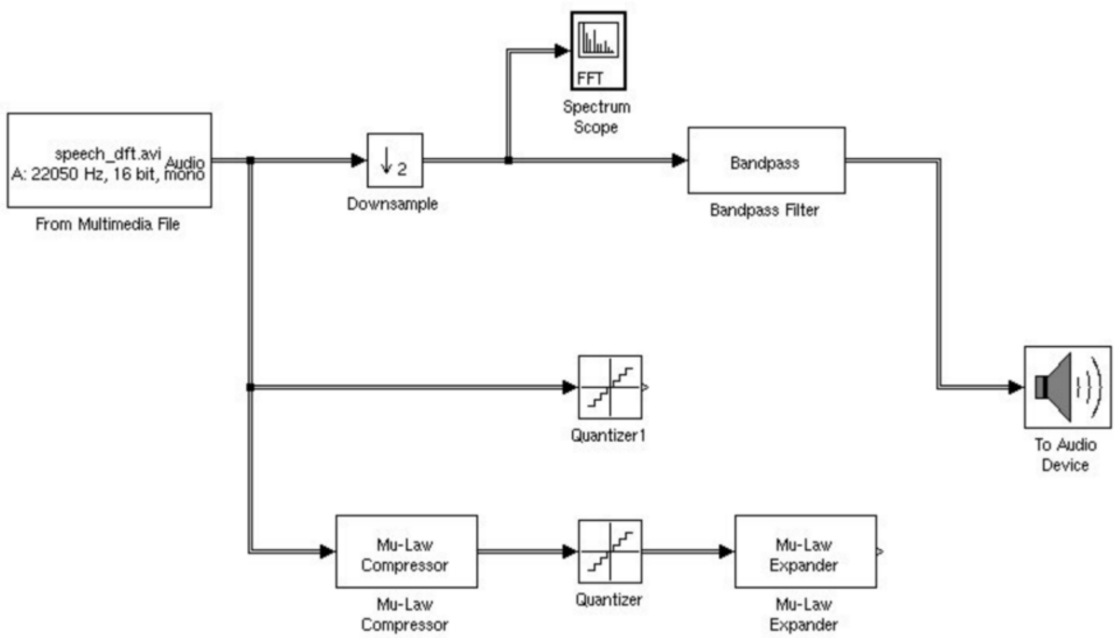

Figure 1: Audio Sampling & Filtering in SIMULINK

1. Observe the audio over the audio device or from the audio file.
2. Observe the FFT; Down sample the audio and observe FFT again
3. Setup the circuit, and listen the audio signal from the highpass filter.
4. Additional blocks show typical quantization steps for uniform/nonlinear quantization.
5. Listen to the effect on audio when using mu-law compounding

   [Note: mu-law blocks only accept input as double; remember to set parameters]

Laboratory

PART I: Sampling

In this part of the lab, you will simulate sampling rate conversion with/without the use of pre-filters and/or interpolation. [use sample audio files or record your own speech using the available webcam].

1. Construct a function `DownSample(inFile,N,pf)` that loads an audio sequence from *inFile*, then down-samples and plays it back
   - *N* is the downsampling factor (e.g. keep every N^{th} sample).
   - *pf* is a boolean flag that indicates whether or not a pre-filter should be used (e.g. using `decimate()` to down-sample. Read matlab help for filter options)

2. Create a reconstructed version of the original waveform using `interp()`.

3. Compute and display the spectra of the original, down-sampled and reconstructed signals for different values of N (e.g. N=2,3,4,8). [Hint: you may use functions such as *fft*(), *specgram*() and *spectrum*() here - see matlab help].

4. Playback and listen to original, down-sampled and reconstructed versions (using *sound*() or similar - see matlab help) and comment on the differences.

PART II: Quantization

You are asked to build two MATLAB programs to implement <u>uniform</u> quantization and <u>nonlinear</u> mu-law companding respectively:

```
UniformQuant (inFile, N, rFcn)
MulawQuant (inFile, N, Mu)
```

`UniformQuant` quantizes an input sequence using a uniform quantizer with a user-specified number of levels, where *N* = number of bits, and *rFcn* specifies the type of rounding operator, i.e. [*round()* | *ceil()* | *floor()*]. Assume the signal is rescaled to its original amplitude before playback.

`MulawQuant` performs the mu-law scaling operation <u>prior</u> to applying `UniformQuant`, and then reverses the process prior to playback (usually at a remote location). You may choose to use the `compand()` function in matlab for this part. Again assume the signal is rescaled and expanded to its original amplitude before playback.

1. Apply `UniformQuant` function to a sound recorded, with quantization level set to 8 bits (N=8), 6 bits (N=6), and 4 bits (N=4) respectively.

2. Compare the original sequence and quantized sequences with different levels, in terms of perceptual sound quality and the waveform. Also record and compare the quantization (MSE) error between the original and quantized samples.

3. Apply `MulawQuant` function to a sound le recorded, with quantization level set to 8 bits (N=8), 6 bits (N=6), and 4 bits (N=4), respectively, with N=100. Compare the original sequence and quantized sequences with different levels

4. Compare the quantized sequence obtained with the uniform quantizer and that with the mu-law quantizer with the <u>same number of quantization levels</u>. Which quantizer yields smaller (MSE) error for signal values that are small, and larger error for signal values that are large? (you should observe this on a zoomed in version the waveform plots).

In your report, your should include the waveform plots; also include a comment on the difference between different sampled & quantized sequences (in terms of perceptual sound quality, waveform and mean square error MSE):

- *Which method gives better sound quality for the same number of levels?*
- *At what point is there very little difference between observed quality?*

Submission:

Copy report (PDF in IEEE format as per blackboard), and source files (matlab *.m) into a new folder named according to the following format:

```
ele725_lab1_username
```

Now zip and submit this folder using the following commands (in the terminal):

```
zip -r ele725_lab1_username.zip ele725_lab1_username
submit ele725 lab1 ele725_lab1_username.zip
```

DEMO DUE: **week 4 lab session**
REPORT DUE: **<u>12 midnight (Friday 26 September, 2014)</u>**

RUBRIC	2 (pts)	3 (pts)	5 (pts)
	Prelim (week 3 in lab)	Demo (week 4 in lab)	Report (week 4 friday)