Lab 1: Audio Signal Processing using Matlab

Report Due: 21:00, 3/22, 2019

1 Overview

In this course, we would like to learn the mechanism of communication systems, from the lab exercises. For this purpose, Matlab is commonly used to simulate these systems. The goal of Lab 1 is to familiarize yourself with Matlab by processing audio signals.

In this lab, we will learn how to read audio files, to display results, and to implement some basic systems. We will experiment with linear time-invariant systems, amplitude distortion, quantization, Fourier transforms, and other operations covered in the lecture.

2 Experiments

In this section, we want to experiment with audio signals and to *listen* to the output signal after processing. Therefore, manipulating audio files is introduced first. Once the signals are loaded, we may modify them by using several systems introduced in the lecture. Finally, we will analyze the spectrum of sinusoids through DFT and DTFT.

1. **Manipulating audio files:** Here are sample codes for reading, playing, and writing a wav file in Matlab:

   ```
   clear x fs
   [x, fs] = audioread('input_file.wav'); % Read audio files
   sound(x, fs); % Play audio files
   audiowrite('output_file.wav', x, fs); % Write to audio files
   ```

   In this lab, the input file is `handel.ogg`. The input signal is denoted by a column vector $\mathbf{x}$ of length N:

   $$\mathbf{x} = \begin{bmatrix} x(0) & x(T) & x(2T) & \dots & x((N-1)T) \end{bmatrix}^T = \begin{bmatrix} x[0] & x[1] & x[2] & \dots & x[N-1] \end{bmatrix}^T,$$

   where the sampling interval $T = 1/f_s$ and f_s is the sampling frequency in Hertz.

   (a) When *playing the audio files*, fix `x` and choose different f_s. Describe what the audio sounds like.

   (b) Choose different data types of the reading file and describe what the audio sounds like.

   (c) Show these waveforms in one figure with proper axis labels.

Note: In the following experiments, please use `handel.ogg` as your audio signal, if not specified.

2. **Redistributing the time index:** In this lab, we will implement some operations that redistribute the time indices in signals, such as circular shift and time-reversal.

   (a) Perform **circular shift** by 100000 on your audio samples. You may use the function `circshift`. Describe what these signals sound like, and show these waveforms in one figure.

   (b) Perform **time-reversal** on your audio samples. You may use the functions `fliplr` or `flipud`. Describe what these signals sound like, and show these waveforms in one figure.

   (c) **(Bonus)** Generate a sound clip which first plays the original clip in the forward direction and then in the backward direction. Explain how to achieve this goal.

   (d) **(Bonus)** Implement **upsampling** and **downsampling** for your audio file and explain how to achieve this goal.

3. **Amplitude distortion:** In the part, we will experiment with the amplitude distortion, defined as

$$y(t) = d(x(t)), \tag{1}$$

   where $d(\cdot)$ is a mapping function.

   A **hard limit** clips your data at a certain threshold T. In this lab, we consider the clipping operation on a real-valued signal $x(t)$.

$$y(t) = d(x(t)) = \begin{cases} x(t), & \text{if } -T < x(t) < T, \\ T, & \text{if } x(t) \geq T, \\ -T, & \text{otherwise.} \end{cases} \tag{2}$$

   Another operation is **squaring** the signal amplitude, defined as

$$y(t) = d(x(t)) = x(t)^2. \tag{3}$$

   The **negation** is multiplying the signal by a minus sign, and it can be describe as

$$y(t) = d(x(t)) = -x(t). \tag{4}$$

   (a) Apply hard limit with $T = 0.1$, squaring, and negation to audio files. Describe what these signals sound like. Plot waveforms and save your results to files.

   (b) Adjust the threshold T in (2). Describe what these signals sound like. Plot waveforms and save your results to files.

4. **Quantization:** In this part, we will use an L-level quantizer to quantize the waveform, as introduced in the lecture notes. The relationship between the level of a quantizer and the quality of data will then be observed.

(a) Find an expression of the mapping function $d(\cdot)$ of the L-level quantizer.
Hint: See page 18 of `Comm_Lab_Week_02_Slides_SS_ver_20190227.pdf`. You could use the floor function $\lfloor \cdot \rfloor$ in your expression.

(b) Write a Matlab function `quantizer_L_level` with the following arguments

```
y = quantizer_L_level(x, x_max, L);
```

Here `x` is the unquantized signal, `x_max` is the maximum of `x`, `L` is the number of levels, and `y` is the quantized signal.

(c) Apply your quantizer to audio files with 4-bit quantization. Describe what these signals sound like. Plot waveforms and save your results to files.

(d) Change the number of levels and repeat the previous experiment.

5. **Modulation:** The modulation operator is multiplying the signal by a complex exponential $e^{j2\pi f_c t}$, defined as

$$y(t) = x(t)e^{j2\pi f_c t}. \tag{5}$$

Assume that $x(t)$ is the audio signal in Part 1. For $f_c = 100$Hz, compute the output signal $y(t)$ after modulation. Describe what the audio sounds like, and show the magnitude and the phase of these waveforms in one figure.
Note: Take the real part of the signals when playing and writing audio samples.

6. **Noise addition:** Add additive white Gaussian noise (AWGN) to all of your audio samples. In particular, for each sample $x[n]$, we compute

$$y[n] = x[n] + w[n]. \tag{6}$$

Here $w[n]$ is a Gaussian random variable with mean 0 and variance `var`. In Matlab, you may use the command `sqrt(var)*randn` to obtain one realization of $w[n]$. Set `var=0.01`, describe what $y[n]$ sounds like, and show its waveform in one figure.

7. **Filtering:**

In this part, we will filter the audio signal in Problem 1 with an FIR filter.

(a) Design an **FIR** filter to select a voice of men (Hint: The frequency range of a male is from 94Hz to 142Hz). You can use the `fir1` function to obtain the impulse response `h`:

```
h = fir1(W, [lower_freq, higher_freq]);
```

Here `[lower_freq, higher_freq]` denotes the lower and the higher edges of the passband. We will set `W = 50` to be the order of the FIR filter. Plot the impulse response of the filter.
Hint: It is *incorrect* to set `lower_freq=94` and `higher_freq=142` in `fir1`. Refer to the documentation of `fir1` for more details.

(b) Apply the filter in Problem 7a to the signal $x[n]$ in Problem 1. Describe what the filtered signal sounds like, and show its waveform in one figure.

8. **The Fourier Transform of sinusoids:** In this problem, you will be guided to plot the Fourier transforms of signals. We will consider the following signal:

$$x(t) = \cos(2\pi f_c t), \tag{7}$$

where $f_c = 880$Hz and $0 \le t \le 2$.

   (a) Assume that $x[n] = x(nT)$ are the discrete-time samples with $T = 1/f_s$, and $f_s = 8192$Hz. Plot $x(t)$ with respect to the time t in seconds, the horizontal axis is Describe what this signal sound like.

   (b) Let $x[n]$ be the discrete-time sequence of the above signal. Compute the **DFT** $X[k]$ and plot its magnitude and phase in one figure. The horizontal axis is the index k.

   (c) Depict the magnitude and the phase of **DTFT** of $x[n]$ in one figure. The horizontal axis is ω in the range of $[-\pi, \pi]$.

   (d) Depict the magnitude and the phase of **CTFT** of $x(t)$ in one figure. The horizontal axis is the frequency f in the range of $[-f_s/2, f_s/2]$.

   (e) Now let us assume $f_s = 1200$Hz in Problem 8a. Repeat Problems 8a to 8d.

3 Lab Report

There is no format requirements for your lab report. In the report, you should address the results of the exercises mentioned above. You should also include your simulation program in the appendix of the report. Include whatever discussions about the new findings during the lab exercise, or the problems encountered and how are those solved. Do not limit yourself to the exercises specified here. You are highly encouraged to play around with your simulation program on self-initiated extra lab exercises/discussions. For example, you can record your own voice or search for sound clips with no copyright at Freesound.org (`https://freesound.org/`).