LAB 5: Sampling and the Discrete Fourier Transform

Objective

In this assignment, you will learn how to down-sample multiple discrete signals including an audio signal and examine how the signal's spectrum changes. You will also observe that while zero-padding increases the frequency resolution, it does not generate new information. So, if the number of samples in the time domain is not sufficient, and do not effectively represent frequency components, zero-padding will not be helpful. Also, you will design an audio equalizer and manipulate a signal in the frequency domain and investigate how it affects the time-domain signal.

Discrete Fourier Transform (DFT)

If x_n is the nth sample from a signal $x(t)$, the DFT of x_n is defined as below

$$X_r = \sum_{n=0}^{N_0-1} x_n e^{-jr\Omega_0 n} \quad , \Omega_0 = \omega_0 T = \frac{2\pi}{N_0}$$

Where, X_r is the rth sample from its Fourier transform $X(\omega)$.

The Inverse Discrete Fourier Transform (IDFT) is

$$x_n = \frac{1}{N_0} \sum_{n=0}^{N_0-1} X_r e^{jr\Omega_0 n}$$

Both

x_n and X_r are periodic sequences with period N_0.

Preparation

- Read chapter 8, especially section 8.5 from *Linear Signals and Systems by B.P. Lathi*.
- Work through Computer Example 8.8 of the text.
- Work through Computer Example 8.7-1 of the text.

Lab Assignment

A. Discrete Fourier Transform and Zero Padding

For the signals $x_1[n]$ and $x_2[n]$ answer the following questions, (*where* **H** *and* **I** *are from your student number*):

$$x_1[n] = e^{j2\pi n(10(\mathbf{H}+\mathbf{1}))/100} + e^{j2\pi n33/100}$$

$$x_2\,[n] = 2cos\,(2\pi n(10(\mathbf{H}+1))/100) + 0.5\,cos(2\pi n(10(\mathbf{I}+1))/100)$$

1) Compute and plot the length-N DFT magnitude of signals $x_1[n]$ and $x_2[n]$ with respect to frequency f_r, where $f_r = \frac{r}{N}$ and N is the number of samples. For this purpose, take 10 samples from the signals and display the spectrum for $-1/2 \leq f_r < 1/2$.

   i. Which signal has a symmetric spectrum and why?
   ii. Is it possible to distinguish both frequency components in the plots? Explain your reasoning.
   iii. Explain why you see other frequency components in the plot for DFT($x_1[n]$).

2) Zero-pad the signals $x_1[n]$ and $x_2[n]$ with 490 zeros, then compute and plot the length-500 DFT of both signals. Do you see any improvement?
3) Repeat part (1) by taking 100 samples from each signal. Plot the signals spectrums and identify which signal has a symmetric spectrum and why?
4) Zero-pad the signals $x_1[n]$ and $x_2[n]$ with 400 zeros, then compute and plot the length-500 DFT of both signals. Do you see any improvement?

Note 1: *To zero-pad a signal, we add zeros to the beginning and end of a time-limited signal.*

Note 2*: To avoid introducing high frequencies to the signal, it is a good idea to add zeros where the signal value is closest to zero. A large discontinuity such as jumping from 7 to zero, adds high frequencies to the spectrum.*

B. Sampling

Now, you will sample an audio signal in the time domain and investigate how different sampling rates affects the frequency spectrum. First, you need an audio file. You may record your voice or use any *.wav* audio file you can find. If you decide to use these files, take the following steps to save the audio file in your project directory path:

```
filename = 'audio.wav'
```

Next, we read the audio file generated from your voice recorder or the above code.

```
y, fs = soundfile.read(filename)
```

We will use the ***sounddevice*** library in Python to play the audio files. The command ***read*** in *soundfile* reads *.wav* audio files and returns sampled data y and sampling frequency f_s. Note that you are using a computer, so any audio signal is discrete and already sampled by the frequency rate f_s. Also, if you have recorded your own voice in stereo, signal y will have two channels. For simplicity, you may choose only one channel to work with.

1) Find the number of samples (N_0), the duration of the signal (T_0), and sampling interval (T).
2) Plot signal y with respect to the time.
3) Compute and plot the DFT of signal y.
4) Generate the subsampled signal y_1 from signal y by subsampling with rate "2". Find the number of samples (N_0), the duration of the signal (T_0), and sampling interval (T).
   Note: The duration y_1 is the same as y but the sampling intervals are different.
5) Plot signal y_1 with respect to time.
6) Compute and plot DFT of y_1.
   How the signal y_1 and its spectrum have changed compared to the original signal. Enlarge a part of the plot for y and y_1 for better visual demonstration. Explain the reason.
7) To listen to the audio signal, use command ***sounddevice.play(x, fs)*** . Play both signals and see how the original audio signal has changed after subsampling.
8) Change the sampling rate in step (4) to 5 and explain how the audio signal and its spectrum change.

C. Filter design

Now we will investigate the effect of changing the frequency response. For this purpose, we use the audio signal y from above and design a simple audio equalizer. An audio equalizer is usually used in sound recording or reproduction to adjust the frequency components of the audio signal. It can boost or weaken the strength (energy) of a desired frequency band.

1) Create a **rect** filter that only passes frequencies less than 2000. Apply this filter to the signal y, and plot the response in time and frequency domain.
2) Use command ***play*** and listen to how the audio signal has changed.
3) Build a filter that cuts the bass frequencies (frequencies between 16 and 256 Hz. Plot the frequency spectrum and listen to the sound. How has it changed?
4) Build a filter that amplifies treble frequencies (frequencies between 2048 and 16384 Hz at the higher end of human hearing.) by 25%. Plot the frequency spectrum and listen to the sound. How has it changed?
5) Which property of DFT did you use to perform the task (4)?