HACETTEPE UNIVERSITY

Department of Electrical and Electronics Engineering

ELE 409: Digital Signal Processing Laboratory

EXPERIMENT 1 - THE DFT & ITS PROPERTIES

I. PURPOSE

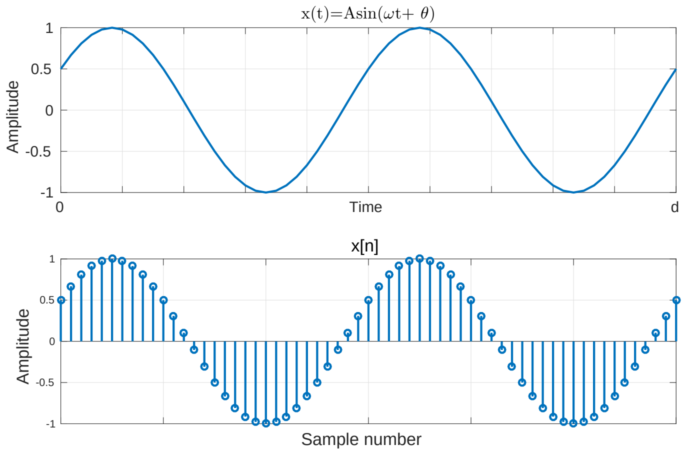
The main purpose of this experiment is to study the generation of discrete-time signals, reviewing some properties of the Discrete-Fourier Transform (DFT) and using it as a tool for signal analysis. Another aim of this experiment is to study the effects of changing the sampling rate in discrete-time systems.

- Learn how to use the following built-in functions and the reason why do we use: *audioread, fft, linspace, fftshift, sortrows, downsample, upsample, interp, plot, stem, subplot.*
- Pay particular attention on:
  - the relationship between sample number, sampling frequency, time, frequency,
  - conversion from sample number to sample (not sampling) frequency.
- Frequency range should be $[-\frac{f_s}{2}, \frac{f_s}{2}]$ in your frequency domain figures.

II. PRELIMINARY WORK

Note on the MATLAB function fft : The fft function computes the Discrete-Fourier Transform (DFT) of a vector x. If the length of the vector x is an integer power of 2, then it is possible to compute the DFT using faster algorithms, which are called Fast-Fourier Transform (FFT) algorithms. MATLAB has a built-in-function named fft, which depending on the data length, computes the DFT either directly or using the Fast-Fourier algorithm.

1. Assume that we have a continuous time sinusoidal signal segment $x(t)$ defined in the interval $0 \le t < d$ such that $x(t) = Asin(\omega t + \theta)$ where A is amplitude, ω is frequency in radians/seconds and θ is phase in radians. We sample this continuous time segment starting at time $t = 0$ with sampling frequency of ω_s to obtain discrete time sequency $x[n]$.



   Write a MATLAB function, $x = SinSamples(A, \omega, \theta, d, \omega_s)$, that generates a discrete-time sinusoidal signal x[n] obtained by sampling $x(t)$, (Assume that $x(t)$ is a sinusoidal segment with amplitude A , frequency ω , phase θ and duration d, with sampling frequency ω_s).

2. Run your function $SinSamples()$ with the following parameters:
   (a) $A = 1,\ \omega = 2\pi 1000,\ \theta = \pi/6,\ d = 2ms,\ \omega_s = 2\pi 16000$, and call the result $x_1[n]$.
   (b) $A = 1,\ \omega = 2\pi 1000,\ \theta = 0,\quad d = 2ms,\ \omega_s = 2\pi 16000$, and call the result $x_2[n]$.

   Using MATLAB's *fft* function obtain the Discrete-Fourier Transform (DFT) of $x_1[n]$ and $x_2[n]$, name them $X_1[k]$ and $X_2[k]$. Plot the signals $x_1[n]$ and $x_2[n]$, and their magnitude spectra, comment on the results.

3. Run your function $SinSamples()$ with the following parameters:

   (a) $A = 1,\ \omega = 2\pi 1000,\ \theta = 0,\ d = 10ms,\ \omega_s = 2\pi 4000$, and call the result $x_3[n]$.

   (b) $A = 1,\ \omega = 2\pi 5000,\ \theta = 0,\ d = 10ms,\ \omega_s = 2\pi 4000$, and call the result $x_4[n]$.

   Using MATLAB's *fft* function obtain the Discrete-Fourier Transform (DFT) of $x_3[n]$ and $x_4[n]$, name them $X_3[k]$ and $X_4[k]$. Plot the signals $x_3[n]$ and $x_4[n]$, and their magnitude spectra, comment on the results. What is the problem?

4. Run your function $SinSamples()$ with the following parameters:

   (a) $A = 1,\ \omega = 2\pi 1000,\ \theta = 0,\ d = 1ms,\quad \omega_s = 2\pi 16000$, and call the result $x_5[n]$.

   (b) $A = 1,\ \omega = 2\pi 1000,\ \theta = 0,\ d = 1.5ms,\ \omega_s = 2\pi 16000$, and call the result $x_6[n]$.

   (c) $A = 1,\ \omega = 2\pi 1000,\ \theta = 0,\ d = 2ms,\quad \omega_s = 2\pi 16000$, and call the result $x_7[n]$.

   Using MATLAB's *fft* function obtain the Discrete-Fourier Transform (DFT) of $x_5[n]$, $x_6[n]$ and $x_7[n]$, name them $X_5[k]$, $X_6[k]$ and $X_7[k]$. Plot the signals and their magnitude spectra, comment on the results. Although these three signals are sampled versions of same continuous time signal with same sampling frequency, their magnitude spectra seems to be different. What is the problem?

5. Run your function $SinSamples()$ with the following parameters: $A = 3,\ \omega = 2\pi 5000,\ \theta = \pi/6,\ d = 2ms,\ \omega_s = 2\pi 16000$, and call the result $x_8[n]$. Now generate the signal $x_s[n] = x_1[n] + x_8[n]$. Check numerically that the DFT is a linear transform. Plot the magnitude spectrum of $x_s[n]$ and comment on the resulting spectrum.

6. Generate the signal $x_m[n] = x_1[n]x_8[n]$. Plot the magnitude spectrum of $x_m[n]$ and comment on the resulting spectrum.

7. Generate the signal $x_9[n] = x_5[n-1]$ which is the one sample shifted version of $x_5[n]$. Assuming $x_5[n]$, $0 \le n \le N-1$, generate $x_9[n]$ such that $x_0[n] = x_5[n-1]$ (do a circular shifting). Observe numerically the effect of this operation on the *fft* of both signals.

8. Load the file **sound1.wav** (You would use MATLAB command audioread to load this file. Use MATLAB help to learn the usage of audioread). This file contains a portion of speech waveform. Take the first 512 point the signal, plot the waveform and its magnitude spectrum.

Up- and Down-sampling

In some cases, signals have to be processed at a sampling rate other than they have been obtained. This is the subject of Multirate-Signal Processing. Below we consider two examples on this topic in which the effect of reducing the sampling rate is studied.

9. Generate signals,

$$\begin{aligned} \mathbf{x_{10}[n]} &= sinc(0.2(n-128)), \quad n = 0, ..., 255 \\ \mathbf{x_{11}[n]} &= sinc(0.8(n-128)), \quad n = 0, ..., 255 \end{aligned}$$

   Downsample $x_{10}[n]$ by 2 such that $x_{down1}[0] = x_{10}[0]$. Plot $x_{10}[n]$, $x_{down1}[n]$ and their magnitude spectra. Comment on the plots. Using MATLAB's **interp** function, interpolate x_{down1} by 2 to obtain $x_{interp1}$. Compare the signals x_{10} and $x_{interp1}$ both in time and frequency domain.

10. Downsample $x_{11}[n]$ by 2 such that $x_{down2}[0] = x_{11}[0]$. Plot $x_{11}[n]$, $x_{down2}[n]$ and their magnitude spectra. Comment on the plots. Interpolate x_{down2} by 2 to obtain $x_{interp2}$. Compare the signals x_{11} and $x_{interp2}$ both in time and frequency domain.