Signals and Systems MATLAB HW2

Deadline: 2023/4/25 23:59

Discrete Fourier Transform

The objective of this section is to learn how to use MATLAB **fft** function.

1. Background

In order to analyze the frequency domain of a finite-duration and discrete-time signal $x[n]$, $n = 1, 2, \dots, N$, its discrete Fourier transform (DFT) is defined as

$$X_k = \sum_{n=1}^{N} x[n] e^{-j\frac{2\pi}{N}(n-1)(k-1)}, k = 1, 2, \dots, N. \tag{1}$$

It is observed that DFT is the sampled Fourier transform of a finite-duration signal with frequency $\omega = \frac{2\pi(k-1)}{N}$. On the other hand, the inverse DFT (IDFT) of X_k is defined as

$$x[n] = \frac{1}{N} \sum_{k=1}^{N} X_k e^{j\frac{2\pi}{N}(n-1)(k-1)}, n = 1, 2, \dots, N. \tag{2}$$

The fast Fourier transform (FFT) is equivalent to DFT with reduced computational complexity as well as inverse FFT (IFFT) to IDFT. To calculate the DFT of the signal $x[n]$ in MATLAB function, you may type:

X = fft(x);

If you want to explicitly specify the length M, then you can type:

X = fft(x, M**);**

Furthermore, MATLAB function **fftshift** command swaps the first and the second half of the vector **X** so that the frequency range is in $\left[\frac{-N}{2}, \frac{N}{2}\right]$ (assuming N is even).

However, for signals with infinite length, we have to truncate it so that it can be computed with MATLAB. Such truncation causes *Gibbs phenomenon* (pp. 200-201 of the textbook).

2. Questions

Please write a MATLAB script (saved as **fftsinc.m**) to implement problems (a) to (f).

Part I

Let $x(t)$ be a sinc function written as

$$x(t) = \frac{\sin(2\pi t)}{2\pi t}.$$

Now, $x(t)$ is sampled at a rate $T_s = T/N_1$ so that $x[n] = x(nT_s)$, $n \in \{-N_1, -N_1+1, \ldots, 0, \ldots, N_1-1, N_1\}$ and $N = 2N_1 + 1$.
Let $N = 1001$ and $T = 100$.

(a) (10%) Use the MATLAB function **plot** to plot $x[n]$ vs n.

(b) (20%) Use the MATLAB function **fft** directly to compute DFT of $x[n]$, and use the MATLAB function **plot** to plot the magnitude of the **fft** output vs frequency ω. The zero frequency should be centered in your plot. Observe the *Gibbs phenomenon* in (b) and give some explanation for it in your report.

(c) (20%) Create a MATLAB program by yourself to compute $X_k(e^{j\omega})$ of equation (1) and use the MATLAB function **plot** to plot the magnitude of $X_k(e^{j\omega})$ vs frequency ω. You also need to rearrange $X_k(e^{j\omega})$ so that the zero frequency is centered in your plot. Verify whether the answer is the same as Problem (b).

Part II

A way of mitigating *Gibbs phenomenon* is to multiply $x(t)$ by a finite-duration signal $w(t)$, i.e., $y(t) = x(t)w(t)$. The signal $w(t)$ is called the window function. A famous one is *Hanning* window, which is specifically written as

$$w(t) = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{2\pi|t|}{T_w}\right)\right], & |t| \le \frac{T_w}{2} \\ 0 & , else \end{cases}$$

where T_w denotes the duration of the window function.
Suppose $w(t)$ is also sampled at a rate $T_s = T/N_1$ so that $w[n] = w(nT_s)$, $n \in \{-N_1, -N_1+1, \ldots, 0, \ldots, N_1-1, N_1\}$, $N = 2N_1 + 1$.
Let $N = 1001$, $T = 100$, and $T_w = T/2$.

(d) (15%) Use the MATLAB function **plot** to plot $w[n]$ vs n.

(e) (15%) Use the MATLAB function **plot** to plot $y[n]$ vs n, where $y[n] = x[n]w[n]$, and $x[n]$ is the signal plotted in (a).

(f) (20%) Use the MATLAB function **fft** directly to compute DFT of $y[n]$ in (e), and use the MATLAB function **plot** to plot the magnitude of the **fft** output vs frequency ω. The zero frequency should be also centered in your plot. Observe the *Gibbs phenomenon* here and give some explanation for comparison with (b) in your report.

Note: We expect that executing your **fftsinc.m** file will output 6 figures in order.
(One figure for each of Problems (a) to (f))

3. NTU COOL Submission

- Please upload a compressed file (.zip), which includes your **m-file** (saved as **fftsinc.m**) and a **word file** (saved as **report.doc**). Please show the figures mentioned above in the word file (report.doc) and give some explanation if needed.
- The compressed file should be named as **ID_MATLAB2.zip**.
  (e.g., B10901xxx_MATLAB2.zip)