Experiment No.-6: DTFT, DFT and DFT Spectral Analysis

1 Overview

In this lab we will perform frequency domain analysis of Discrete-Time signals using MATLAB. The objective of this lab is to teach you how interpret and analyze the frequency spectra of digital signals with fixed frequency contents and others with time varying frequency contents such as speech signals.

2 DTFT

The DTFT is used to represent discrete-time signals in terms of complex exponential signals $e^{j\omega n}$. The DTFT of a discrete-time signal $x[n]$ is given by,

$$X_d(w) = \sum_{-\infty}^{\infty} x[n]e^{-j\omega n},$$

which is called the DTFT analysis equation. The synthesis equation is given by,

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_d(\omega)e^{j\omega n}d\omega$$

Note the DTFT is periodic with period 2π. The two major limitations for any numerical implementation of DTFT are that $x[n]$ must have finite length and $X(e^{j\omega})$ can only be computed at a finite number of samples of the continous frequency variable ω.

3 DFT and IDFT

In order to compute the DTFT of a signal $x[n]$ using MATLAB the signal $x[n]$ must be first truncated to form a finite length signal. This is not necessarily a limitation since in real life applications, we only have finite recordings of sampled and quantized (i.e. digitized) continuous time signals. Also, $X(e^{jw})$ can be computed only at a discrete set of frequency samples, w_k. This is also not necessarily a limitation: if enough samples are chosen, the discrete time signal could be perfectly reconstructed using the inverse DFT (IDFT) synthesis equation. In addition, in the case when enough samples are chosen, the plot of these frequency samples will be a good representation of the actual DTFT. For computational efficiency, the best set of frequency samples is the set of equally spaced points in the interval $0 \le \omega \le 2\pi$ given by $\omega_k = \frac{2\pi k}{N}$ for $k = 0, 1, \dots, N-1$. For a signal $x[n]$ which is nonzero only for $0 \le n \le M-1$ the DFT is defined by,

$$X[k] = \sum_{n=0}^{M-1} x[n]e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1$$

The Inverse DFT (IDFT) is given by,

$$x[n] = \sum_{k=0}^{N-1} x[n]e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, M-1$$

The function `fft` implements the DFT operation in a computationally efficient manner. FFT stands for Fast Fourier Transform and will be studied in greater details towards the end of the course. If `x` is a vector containing $x[n]$ for $0 \leq n \leq M-1$ and $N \geq M$, then `X=fft(x,N)` computes N evenly spaced samples of the DTFT of `x` and stores these samples in the vector `X`. If $N < M$, then the MATLAB function `fft` truncates `x` to its first N samples before computing the DTFT, thus yielding incorrect values for the samples of the DTFT. The function `ifft` can be used to compute the IDFT efficiently. Also note that the `fft` function computes the DFT in the interval $[0, 2\pi]$. To reorder the samples in the interval $[-\pi, \pi]$ you can use the function `fftshift`.

4 Example

Consider the following signal,

1.
$$x[n] = u[n] - u[n-11]$$

   - What is the DTFT of the Signal?
   - Compute the DFT using the `fft` function. Plot the magnitude of the spectrum.
   - One can improve the resolution of the DFT by *zero padding* the input and then computing its DFT. This can be done using the `fft(x,N)` command by choosing $N > M$, where M is the sequence length. Choose $N = 100, 500, 1000$. Does the DFT look similar to the DTFT?

2. A MATLAB code to plot the DFT

```
N = 8;
n = 0:(N-1);
x = (0.7).^n;
k = 0:(N-1);
Xdft_8 = fft(x,N);
mag_Xdft_8 = abs(Xdft_8);
phase_Xdft_8 = angle(Xdft_8);
figure, subplot(2,1,1);
stem(k, mag_Xdft_8), grid on;
title(``Magnitude of 8-point DFT of x[n]''), xlabel(``k''), ylabel(``| X[k]|'');
xlim([-0.2 8.2]);
subplot(2,1,2);
stem(k, phase_Xdft_8), grid on;
title(``Phase of 8-point DFT of x[n]''), xlabel(``k''), ylabel(``< X[k] (radians)'');
xlim([-0.2 8.2]);
```

5. Exercise

Let $x[n]$ be a discrete time sequence:

$$x[n] = \begin{cases} (0.7)^n, & 0 \leq n \leq 7 \\ 0, & else \end{cases}$$

(a) Determine the analytical expression for the DTFT of $x[n]$ and plot the magnitude and phase of the DTFT.

(b) Compute in MATLAB the 8-point DFT of $x[n]$, $0 \leq n \leq 7$ using the DFT function.
Plot the magnitude and phase. You can use the `stem`, `abs`, and `angle` commands.

(c) Compute, in MATLAB, the 16-point DFT of $x[n]$, $0 \leq n \leq 15$ and stem plot its magnitude and phase. This can be accomplished by modifying the commands provided in part (b). Comment on the effect of zero-padding the signal on its DFT.

(d) Compute, in MATLAB, the 128-point DFT of $x[n]$, $0 \leq n \leq 127$ and plot its magnitude and phase. Note: the `plot` command is used instead of stem when many dense points exist to avoid appearance of a black blob.

(e) Compare the results from part (d) to the plots of part (a). How does this relate to the relationship between digital frequency ω and DFT index k?

Source: https://web.stanford.edu/~kairouzp/teaching/ece311/secure/lab2/lab2.pdf