EN3551 Assignment 1

Detecting Harmonics in Noisy Data and Signal Interpolation using DFT

Due on or before: Friday, September 12, 2025

---

1 Introduction and General Guidelines

This programming assignment is concerned with two important applications of the *discrete Fourier transform (DFT)*, namely, detecting sinusoidal signals from noisy discrete-time signal and interpolating a discrete-time signal using the DFT.

Section 2 contains a general overview of the theory used in this assignment, followed by Section 3 and Section 4, which contains the specific tasks for the assignment.

Submit a report containing your solutions, diagrams, and explanations for each task. Briefly explain/justify how you obtained each of your answers. This will help us determine your understanding of the problem whether or not you got the correct answer. Moreover, in the event of an incorrect answer, we can still try to give you partial credit based on the explanation you provide. You may append your MATLAB codes as an appendix of this report.

2 Theoretical Background

2.1 Preliminaries

The DFT of a length-N discrete-time signal $x[n]$ is defined as;

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

If we denote

$$W_N = e^{-j2\pi/N}, \tag{2}$$

then

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad 0 \leq k \leq N-1. \tag{3}$$

If the signal $x[n]$ is real valued, then $X[N-k] = X^*[k]$ for $k = 0, 1, ..., N/2$, where a $X^*[k]$ is the complex conjugate of $X[k]$. Consequently, in this case, only a half of the DFT components are independent, and $|X[k]|, k = 0, 1, ..., N-1$ is a mirror-image plot.

If the discrete-time signal $x[n]$ is obtained by sampling a continuous time signal $x(t)$ at a rate f_s (in Hz), then the index range $0 \leq l \leq N-1$ for the DFT sequence $X[k]$ corresponds to the frequency range $[0, f_s]$. Hence, for a real-valued signal $x[n]$, the plot of the first $(N/2+1)$ values of $|X[k]|$ is associated with the frequency region $[0, f_s/2]$. (Here we have assumed N is an even integer).

The *inverse discrete Fourier transform (IDFT)*, which synthesize the discrete-time signal $\{x[n], n = 0, 1, ..., N-1\}$ using $\{X[k], k = 0, 1, .., N-1\}$ is given by

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \quad 0 \leq n \leq N-1. \tag{4}$$

2.2 DFT Averaging for Harmonic Detection

Say we have an N-point discrete-time signal $\{x[n]\}$ which contains several sinusoidal components and additive white Gaussian noise. The discrete signal is therefore, $x[n] = s[n] + w[n]$. Here $s(\cdot)$ is the signal, and $w(\cdot)$ is the additive white Gaussian noise. A straightforward way is to apply the DFT to the discrete-time signal $\{x[n]\}$, and taking the frequency domain sequence $\{X[k]\}$. If $\{x[n]\}$ is obtained by sampling a continuous time signal at a sampling rate greater than the Nyquist sampling rate, $(f_s \geq f_{Nyquist})$, and if the noise is not severe, then by plotting the magnitude of $\{X[k]\}$, the sinusoidal requencies $f_1, f_2, ...f_M$ can be identified where there are spikes in the DFT magnitude plot. But this approach fails when the nosie is high, so as to effectively hids/bury the DFT of the sinusoids within the noise spectrum.

The solution for the above problem is called *DFT averaging* [1] which works when a relatively large number of signal samples are available (large N). The steps for this approach are;

1. Partition the input signal $\{x[n], n = 0, 1, 2, ..., N-1\}$ into L subsets, with each subset having K samples. $(N = LK)$

2. Apply DFT to each subset of the samples. Let the DFT of the ith subset be $\{X^{(i)}[k], k = 0, 1, ..., K-1\}$.

3. Calculate the arithmetic mean of the L sets of DFT sequences and note as $\{X^{(a)}[k], k = 0, 1, ..., K-1\}$. Here $X^{(a)}[k] = \frac{X^{(1)}[k]+X^{(2)}[k]+...+X^{(L)}[k]}{L}$.

4. Find the frequencies corresponding to the peaks of the plot $\{|X^{(a)}[k]|\}$

5. Pay attention to the conditions under which this method is more effective. You may need to increase the number of partitions, if the noise effect is too high. The aim is to cancel out the noise spectrum by averaging.

2.3 Frequency-Domain Interpolation by Zero Insertion

First, let us consider the case when zeros are inserted in the time domain. Inserting K zeros after each sample in a signal $x[n]$ of length N, is called *upsampling* by $K+1$. The result is a signal $y[n]$ of length $(K+1)N$,

$$y[n] = \{x[0], \underbrace{0, 0, 0...}_{K\ zeros}, x[1], \underbrace{0, 0, 0...}_{K\ zeros}, ..., x[N-1]\, \underbrace{0, 0, 0...}_{K\ zeros}\}, \tag{5}$$

where $0 \leq n \leq (K+1)N - 1$. The DFT of $y[n]$ is found to be

$$Y[k] = \sum_{n=0}^{(K+1)N-1} y[n] e^{-j\frac{2\pi kn}{(K+1)N}} = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}} = X[k], \tag{6}$$

where $0 \leq k \leq (K+1)N - 1$.

Because $X[k]$ is periodic with a period N, it follows from above that $Y[k]$ is a succession of $(K+1)$ DFTs of the discrete-time signal $x[n]$ $(0 \leq n \leq N-1)$. Furthermore, we note that, because the number of samples in $y[n]$ are $(K+1)$ times as many as that in $x[n]$, the sampling frequency used to get $y[n]$ is $(K+1)$ times higher than the sampling frequency f_s used to get $x[n]$. Therefore, if an ideal lowpass filter contains only the components of the first of the sequence of DFTs in equation 6, this creates an interpolated version of $x[n]$.

The process described above can be realized only with an ideal lowpass filter, that is not realisable in practice. Therefore, we use a trick in frequency domain, and expand $X[k]$ with zeros, effectively creating a zero-padded version of $Y[k]$ without the frequencies higher than $f_s/2$. This is done by inserting KN zeros in the middle of $X[k]$, but we have to be careful how the DFT sequence $X[k]$ is split.

- If N, the length of $X[k]$ is odd, then the first $N_1 = (N+1)/2$ points of $X[k]$ are placed to the left end and the rest $(N-1)/2$ are placed to the right end of the sequence with a total of KN zeros inserted in between.

$$X_z = [X(1 : N_1); \text{zeros}(KN, 1); X((N_1 + 1) : N)]. \tag{7}$$

- If N is even, then the first $N_2 = \frac{N}{2}$ points of $X[k]$ are placed to the left end, then the sample $\frac{X(N_2+1)}{2}$, followed by $(KN-1)$ zeros, then again $\frac{X(N_2+1)}{2}$, and the remaining $(N_2 - 1)$ points of $X[k]$.

$$Xz = [X(1 : N_2); \frac{X(N_2+1)}{2}; \text{zeros}(KN-1, 1); \frac{X(N_2+1)}{2}; X((N_2+2) : N)]. \tag{8}$$

After we form X_z, inverse DFT is applied and scaled by $(K+1)$. Then, we take the first $(K+1)(N-1)+1$ samples as an interpolation of the N-point input sequence $x[n]$. Intuitively, the inverse DFT is multipled by $(K+1)$ to compensate the amplitude loss in the signal due to the zero-insertion step.

3 Procedure

3.1 Harmonic Detection

1. Download the noise corrupted signal (xn_test) of 1793 samples, that correspond to your index number (Refer the Readme file in the signals folder to find your signal.). Each signal contains 4 harmonics and severe white Gaussian noise. Consider the samples were collected at a sampling rate of $f_s = 128$ Hz over the period from 0 to 14 s. The four harmonics are no greater than 64 Hz.

2. Construct several subsets by taking the first $128, 256, 512, 1024$ and 1792 samples from the sequence $\{x[n]\}$ and denote them by $S1, S2, S3, S4$ and $S5$.

3. Apply DFT to each subset of samples and display the magnitude of the reulting DFT sequences so as to identify the harmonics. Observe and comment of the results obtained.

4. Now apply the DFT averaging method described in the Section 2 above, where the length of each subset is taken to be $K = 128$, and the number of subsets is taken to be $L = 14$.

5. What is the smallest value of L such that the four peaks that correspond to the four harmonics not greater than 64 remain clearly visible?

6. Can one use other values for K (say $K = 100$ or $K = 135$)?. Explain why.

3.2 Interpolation

1. The test signal in this part of the experiment is the first 20,000 samples of a music clip ”Hallelujah” by Handel, which is available from MATLAB. To get it, execute ‘load handel’ in MATLAB. Let us denote the loaded signal as $y[n]$.

2. The signals to be used below are generated from the first 20,000 samples of $y[n]$ as follows:

   - $N = 20,000;$
   - $x = y(1 : N);$
   - $x_2 = x(1 : 2 : N);$
   - $x_3 = x(1 : 3 : N);$
   - $x_4 = x(1 : 4 : N);$

3. Apply the DFT-based method to interpolate the signals x_2, x_3 and x_4 (note whether they are odd or even) and make some comparisons as follows:

   (a) Interpolate the signal x_2 with $K = 1$. Compute the difference between the interpolated signal and the original signal x in 2-norm (notice that the length of the two signals involved might be different). Compare the waveform of the interpolated signal with that of the original signal x by plotting the first 50 samples of both signals in the same figure.

   (b) Repeat above for signal x_3 with $K = 2$.

(c) Repeat the first part for signal x_4 with $K = 3$.

(d) Observe and comment on the results obtained for each of the above tasks.

3.3 Useful MATLAB commands

- plot: plots a function
- fft: calculates the DFT
- fftshift: shift zero-frequency component to the centre of the spectrum
- load: load contents from a file into a MATLAB workspace.
- norm: returns the norm of a vector input.

4 References

[1] E.O. Brigham, The Fast Fourier Transform and its applications, Prentice Hall, 1988.