ECE 348: Digital Signal Processing Lab, Spring 2023

Instructor: Dr.Yuqian Zhang

TA: Ashwini Subramanian, Spilios Evmorfos

General Instructions: Please submit a written report in the pdf format within 2 weeks of your lab session. The report must describe the purposes of the experiments, the methods used, including all graphs and Matlab code. The reports must be uploaded to Canvas within the allotted time frame.

LAB – 4

---

1 Relationship between the DTFT and the DFT

In this part, we consider the analytical and numerical computation of the DTFT of a periodic signal, the computation of its DFT, and its representation as a sum of sinusoids.

Problem 1. Consider the periodic square-wave of period 10,

$$s(n) = [\cdots, \underbrace{1, 1, 1, 1, 1, -1, -1, -1, -1, -1}_{\text{one period}}, \cdots], \tag{1}$$

where the dots represent repetition of the basic period. Show analytically that the DTFT of one basic period is given by

$$S(w) = \sum_{n=0}^{9} s(n) e^{-jwn} = \frac{(1 - e^{-5jw})^2}{1 - e^{-jw}}, \tag{2}$$

Problem 2. By evaluating (2) at the 10 DFT frequencies, $\omega_k = 2\pi k/10$, $k = 0, 1, 2 \cdots, 9$, show analytically that the 10-point DFT of one basic period, i.e., the values $S_k = S(e^{jw_k})$, are given by

$$S_0 = 0, \quad and, \quad S_k = 2\frac{1 - (-1)^k}{1 - e^{-j\pi k/5}}, \quad k = 1, 2, \cdots, 9. \tag{3}$$

Problem 3. By inserting the result of Problem 2 into the inverse DFT formula,

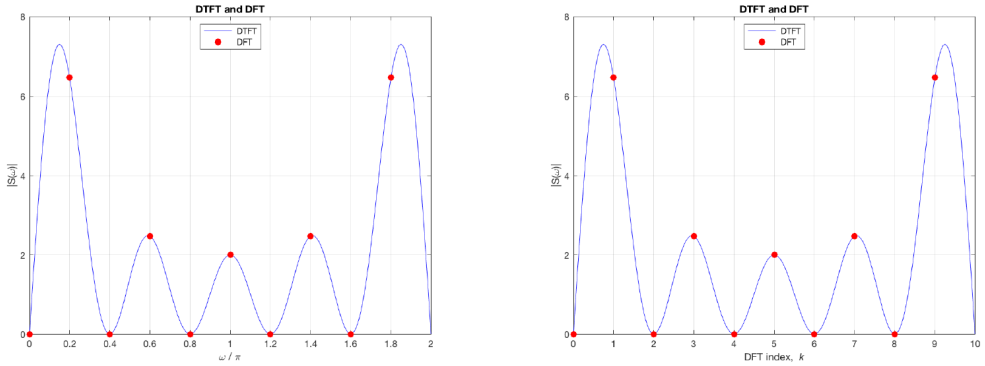
$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S_k e^{jw_k n}, \quad n = 0, 1, \cdots, N-1 \tag{4}$$

show analytically that the square wave in Problem 1 admits the following expansion in terms of sinusoids:

$$s(n) = \frac{0.4}{sin(\frac{w_1}{2})} sin(w_1 n + \frac{w_1}{2}) + \frac{0.4}{sin(\frac{w_3}{2})} sin(w_3 n + \frac{w_3}{2}) + 0.2cos(w_5 n) \tag{5}$$

and verify that this expression agrees with the square-wave given in (1) for n = 0, 1, . . . , 9.

Problem 4. Make a plot of the DTFT magnitude, $|S(w)|$, versus digital frequency w at 201 equally-spaced frequencies spanning the interval $0 \leq w \leq 2\pi$. Superimpose on the same graph the corresponding 10-point DFT values plotted with dot-markers (see example graph in Fig. 1(a)).



(a) (b)

Figure 1: Plots for Problem 4.

Make the same plot, but now plot the spectra versus the DFT index, that is, the normalized frequency variable, $k = Nw/(2\pi)$, which takes the integer values $k = 0, 1, \ldots, N-1$, at the N DFT frequencies w_k (see example graph in Fig. 1(b)). Based on this plot and the fact that s(n) is real-valued, can you explain the presence of only w_1, w_3, w_5 in the expression in (5)?

Problem 5. Verify numerically, using the same set of frequencies w of Problem 4, that $S(w)$ computed from (2) agrees with that computed via **freqz**.

Moreover, verify numerically that S_k computed by (3) agrees with that computed using the Matlab function **fft**.

2 Spectral analysis with the DFT

A real-valued signal consisting of the sum of two sinusoids of unknown frequencies and unknown amplitudes and phases was sampled at a rate of 10 kHz and 128 samples were collected. Next,

128-point DFT of the collected samples was computed and saved (as a complex-valued array) in the attached MAT file, **X.mat**.

Problem 6. Load the stored DFT vector, $X_k, k = 0, 1, \ldots, 127$, into MATLAB using the command load **X**. Make a stem plot of the magnitude $|X_k|$ versus the DFT index k. As you observe, there are two dominant peaks falling in the first half of the k-range, corresponding to the positive half of the Nyquist interval.

Using the function **sort**, determine the DFT indices, say, k_1, k_2, of these two peaks, and their corre- sponding frequencies in KHz, say, f_1, f_2. [Hint: Remember that MATLAB indices are DFT indices shifted by 1.]

Determine also the DFT indices and the frequencies of the two peaks that fall in the second half of the k range and argue that they correspond to the negative frequencies $-f_2$, $-f_1$. Explain your reasoning in detail.

Problem 7. Reconstruct the time-domain signal $x(n), n = 0, 1, \ldots, 127$, using the full DFT X_k that was provided. Reconstruct also an estimate of it, say, $x_{est}(n)$, using only the two frequency components, $\pm f1$, $\pm f2$, that you estimated in Problem 6. Make a plot of both $x(n)$ and $x_{est}(n)$ versus n (see example plots in Fig. 2).

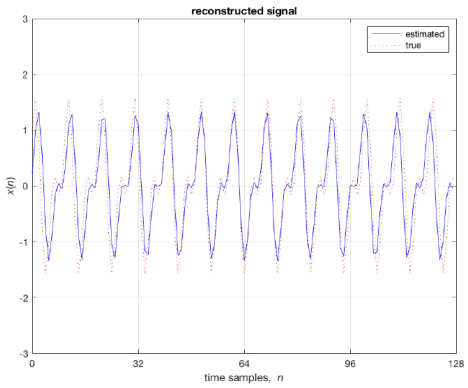


Figure 2: Plot for Problem 7.

3 Computational speed of FFT and DFT

The N-point DFT X_k of a length-N signal x_n can be expressed in matrix form as:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi jkn/N}, \quad k = 0, 1, \ldots, N-1 \Rightarrow X = Ax, \tag{6}$$

where $\mathbf{x}$, $\mathbf{X}$ are the N-dimensional column vectors of the time samples and DFT values, and $\mathbf{A}$ is the DFT matrix of size $N \times N$, with matrix elements:

$$A_{kn} = e^{-2\pi jkn/N}, \quad 0 \le k \le N-1, \quad 0 \le n \le N-1. \tag{7}$$

Problem 8. Write a single-line anonymous MATLAB function of N,called dftmat,that defines the DFT matrix of (7), and has syntax,

```
A = dtfmat(N);
```

where your anonymous definition should begin with,

```
dftmat = @(N) ...
```

Note that MATLAB has a built-in function dftmtx, but you should not use that. Instead your definition must implement (7) directly.

Problem 9. or each of the following values of N = [512, 1024, 2048, 4096], do the following:

1. Generate a length-N column vector x of normally-distributed random numbers using the function randn.
2. Compute the corresponding DFT matrix A using your function dftmat and save the computation time in seconds using the tic, toc functions.
3. Compute the matrix form of the DFT by performing the operation X = Ax, and save the computation time in seconds.
4. Compute the DFT using the function fft, and save its computation time.
5. Compute the error between the matrix and FFT computations, E = norm (Ax - fft(x)).

Using **fprintf** commands, present your results in a table form as shown below, where your numbers will be different, depending on your random signal realization, and your PC's speed:

```
  N       Tmat        Tdft        Tfft        Error
-------------------------------------------------------
 512    0.015566    0.000817    0.004119    4.3036e-11
1024    0.043848    0.000983    0.001647    1.7957e-10
2048    0.214558    0.003769    0.007031    6.6818e-10
4096    0.485770    0.011877    0.007412    2.6608e-09
```

As you can see, the matrix DFT calculation itself is very efficient, but the time it takes to actually generate the matrix A is more substantial. The FFT is by far more efficient, even though it computes internally both the required multiplier coefficients and the DFT values.

4 Filtering of periodic signals via DFT

The most important property of linear systems is the sinusoidal response, that is, the output due to a double- sided complex sinusoidal input is also a sinusoid, modified by the frequency rersponse of the filter:

$$x(n) = e^{jwn} \quad \Rightarrow \quad y(n) = H(w)e^{jwn}. \tag{8}$$

The sinusoid e^{jwn} is not necessarily a periodic signal in the discrete-time variable n unless the frequency has the form $w = 2\pi k/N$ for integer values of k, N and, in this case, it is periodic with period N, that is, $e^{jw(n+N)} = e^{jwn}$. A more general periodic discrete-time signal $s(n)$ with period N is specified by its sample values of one period,

$$s(n) = [\ldots, \underbrace{s_0, s_1, s_2, \ldots, s_{N-1}}_{\text{one period}}, \underbrace{s_0, s_1, s_2, \ldots, s_{N-1}}_{\text{one period}}, \ldots] \tag{9}$$

and it can be represented by the inverse DFT formula as a sum of sinusoids at the DFT frequencies $w_k = 2\pi k/N, k = 0, 1, \ldots, N-1$,

$$s(n) = \frac{1}{N}\sum_{k=0}^{N-1} S(k)e^{jw_k n}, \quad -\infty<n<\infty \tag{10}$$

where $S(k)$ is the N-point DFT of one period of the periodic signal:

$$S(k) = \sum_{k=0}^{N-1} s(n)e^{jw_k n}, \quad k = 0, 1, \ldots, N-1. \tag{11}$$

If a periodic signal such as $s(n)$ is sent to the input of a stable filter, then after the filter transients die out, the steady-state output signal will also be periodic with the same period N and given in its sinusoidal form by the inverse DFT:

$$s_{out}(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(e^{jw_k})S(k)e^{jw_k n}. \tag{12}$$

This follows by applying the sinusoidal response of (8) to the individual terms of (10). Thus, the fol- lowing computational steps allow one to determine the samples of the output period:

1. compute FFT of one input period: S = fft(s,N)
2. evaluate filter at DFT frequencies: H = $[H(e^{jw_0}), H(e^{jw_1}), \ldots, H(e^{jw_{N-1}})]$
3. element-wise multiplication: $S_{out} = H. * S$
4. compute inverse FFT: s_{out} =ifft(S_{out}, N)

where we assumed that all vectors are rows.

Problem 10. Write a MATLAB function that implements the steps in above, with syntax:

```
sout = periodic_output(b,a,s);
```

where b,a are the numerator and denominator coefficient vectors of the filter, and s,s_{out} represent one period of the input and output signals. All the operations inside this function must be

vectorized. To help you debug your program, the following answer is given:

$$s = [3, 6, 3], \quad H(z) = \frac{2 + z^{-1}}{1 + 0.5z^{-3}}, \quad b = [2, 1], \quad a = [1, 0, 0, 0.5] \quad \Rightarrow \quad S_{out} = [6, 10, 8]$$

Problem 11. Apply your function to the period-8 square wave:

$$s(n) = [\underbrace{1, 1, 1, 1, -1, -1, -1, -1}_{\text{one period}}, \cdots] \tag{14}$$

which is sent into the filter

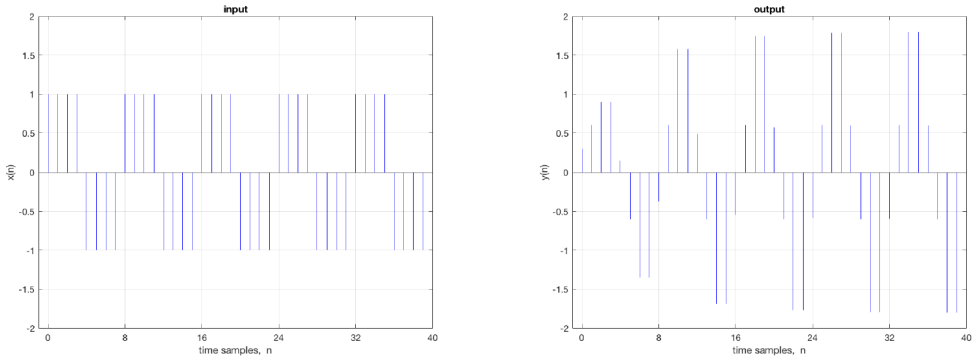
$$H(z) = \frac{1 + z^{-1} + z^{-2}}{1 + 0.5z^{-4}}$$

and compute the corresponding length-8 output period s_{out}.

Problem 12. Generate an input signal $x(n)$ from (14) consisting of 5 periods only and filter it through $H(z)$ using the function filter, i.e.,

```
y = filter(b,a,x);
```

On two separate graphs, make stem plots of the signals $x(n)$ and $y(n)$. Observe how the output signal converges to the computed output period as the transients die out.



(a) (b)

Figure 3: Plots for Problem 4.