Seung Jun Baek

Topics: DFT and FFT.

1. compute the DFT of each the following finite-length sequence considered to be of length N(where N is even):

   (a) $x[n] = \delta[n]$

   (b) $x[n] = \delta[n - n_0], \quad 0 \leq n_0 \leq N$-1

   (c) $x[n] = \begin{cases} 1, & n \quad even, \qquad 0 \leq n \leq N-1, \\ 0, & n \quad odd, \qquad 0 \leq n \leq N-1, \end{cases}$

   (d) $x[n] = \begin{cases} 1, & 0 \leq n \leq N/2 - 1, \\ 0, & N/2 \leq n \leq N-1, \end{cases}$

   (e) $x[n] = \begin{cases} a^n, & 0 \leq n \leq N-1, \\ 0, & \text{otherwise}, \end{cases}$

   SOL:

   (a)

$$x[n] = \delta[n]$$

$$X[k] = \sum_{n=0}^{N-1} \delta[n] W_N^{kn}, \quad 0 \leq K \leq (N-1)$$

$$= 1$$

   (b)

$$x[n] = \delta[n - n_0], \quad 0 \leq n_0 \leq (N-1)$$

$$X[k] = \sum_{n=0}^{N-1} \delta[n - n_0] W_N^{kn}, \quad 0 \leq K \leq (N-1)$$

$$= W_N^{kn_0}$$

   (c)

$$x[n] = \begin{cases} 1, & n \quad even \\ 0, & n \quad odd \end{cases}$$

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad 0 \leq K \leq (N-1)$$

$$= \sum_{n=0}^{(N/2)-1} x[n] W_N^{2kn}$$

$$= \frac{1 - e^{-j2\pi k}}{1 - e^{-j(4\pi k/N)}}$$

$$X[k] = \begin{cases} N/2, & k = 0, N/2 \\ 0, & \text{otherwise} \end{cases}$$

(d)

$$x[n] = \begin{cases} 1, & 0 \le n \le N/2 - 1, \\ 0, & N/2 \le n \le N - 1, \end{cases}$$

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad 0 \le K \le (N-1)$$

$$= \sum_{n=0}^{(N/2)-1} x[n] W_N^{kn}$$

$$= \frac{1 - e^{-j\pi k}}{1 - e^{-j(2\pi k)/N}}$$

$$X[k] = \begin{cases} N/2, & K = 0, \\ \frac{2}{1 - e^{-j(2\pi k/N)}}, & k \quad odd \\ 0, & k \quad even \qquad\qquad 0 \le k \le (N-1), \end{cases}$$

(e)

$$x[n] = \begin{cases} a^n, & 0 \le n \le N - 1, \\ 0, & \text{otherwise}, \end{cases}$$

$$X[k] = \sum_{n=0}^{N-1} a^n W_N^{kn} \quad 0 \le K \le (N-1)$$

$$= \frac{1 - a^N e^{-j2\pi k}}{1 - ae^{-j(2\pi k)/N}}$$

$$X[k] = \frac{1 - a^N}{1 - ae^{-j(2\pi k)/N}}$$

2. Let $X(e^{jw})$ denote the Fourier transform of the sequence $x[n] = (0.5)^n u[n]$. Let $y[n]$ denote a finite-duration sequence of length 10; i.e., $y[n] = 0, n < 0$, and $y[n] = 0$, $n \ge 10$. The 10- point DFT of $y[n]$, denoted by $Y[k]$, corresponds to 10 equally spaced samples of $X(e^{jw})$; i.e., $Y[k] = X(e^{j2\pi k/10})$. Determine $y[n]$. **SOL:** Fourier transform of

$$x[n] = (1/2)^n u[n] :$$

$$X(e^{jw}) = \sum_{n=-\infty}^{\infty} x[n] e^{-jwn}$$

$$= \sum_{n=0}^{\infty} (\frac{1}{2})^n e^{-jwn}$$

$$= \frac{1}{1 - \frac{1}{2}e^{-jw}}$$

Now, sample the frequency spectra of $x[n]$:

$$Y[k] = \frac{1}{1 - \frac{1}{2}e^{-j\frac{2\pi k}{10}}}, \quad 0 \leq k \leq 9$$

$$= \sum_{n=0}^{9} y[n] W_{10}^{kn}$$

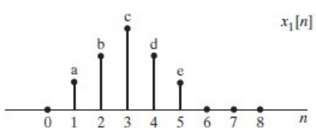
Recall:

$$\frac{1}{2}^n \rightarrow_{N-pt}^{DFT} \rightarrow \frac{1 - \frac{1}{2}^n}{1 - \frac{1}{2}e^{-j(2\pi k/N)}}$$

So, we may infer :

$$y[n] = \frac{\frac{1}{2}^n}{1 - \frac{1}{2}^{10}}, \quad 0 \leq n \leq 9$$

3. The two eight-point sequences $x_1[n]$ and $x_2[n]$ shown in Figure P8.10 have DFTs $X_1[k]$ and $X_2[k]$, respectively. Determine the relationship between $X_1[k]$ and $X_2[k]$.



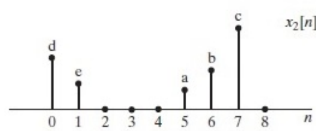


Figure 1: P8.10

SOL: From fig P8.10-1, the two 8-pt sequences are related through a circular shift. Specifically

$$x_2[n] = x_1[((n-4))_8]$$

From property 5 in Table 8.2,

$$DFTx_1[((n-4))_8] = W_8^{4k} X_1[k]$$

Thus,

$$X_2[K] = W_8^{4K} X_1[k]$$

$$= e^{-j\pi k} X_1[k]$$

$$X_2[K] = (-1)^k X_1[K]$$

4. Suppose we have two four-point sequences $x[n]$ and $h[n]$ as follows:

$$x[n] = \cos(\frac{\pi n}{2}) \quad n = 0,1,2,3$$
$$h[n] = 2^n \quad n = 0,1,2,3$$

(a) Calculate the four-point DFT X[k].
(b) Calculate the four-point DFT H[k].
(c) Calculate y[n] = x[n] ④ h[n] by doing the circular convolution directly.
(d) Calculate y[n] of part (c) by multiplying the DFTs of x[n] and h[n] and performing an inverse DFT.

SOL:

(a)

$$x[n] = \cos(\frac{\pi n}{2}), \quad 0 \leq k \leq 3$$

transforms to

$$X[k] = \sum_{n=0}^{3} \cos(\frac{\pi n}{2}) W_4^{kn}, \quad 0 \leq k \leq 3$$

The cosine term contributes only two non-zero values to the summation, giving:

$$X[k] = 1 - e^{-j\pi k}, \quad 0 \leq k \leq 3$$

$$= 1 - W_4^{2k}$$

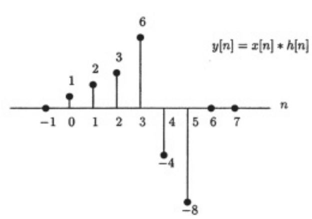
(b)

$$h[n] = 2^n, \quad 0 \leq n \leq 3$$

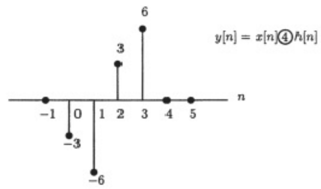
$$H[k] = \sum_{n=0}^{3} 2^n W_4^{kn}, \quad 0 \leq n \leq 3$$

$$= 1 + 2W_4^k + 4W_4^{2k} + 8W_4^{3k}$$

(c) Remember, circular convolution equals linear convolution plus aliasing. We need N $\geq$ 3+4-1=6 to avoid aliasing.Since N=4,we expect to get aliasing here. First, find $y[n] = x[n] * h[n]$:



For this problem, aliasing means the last three points(n=4,5,6) will wrap-around on top of the first three giving $y[n] = x[n] ④ h[n]$:



(d) Using the DFT values we calculated in parts (a) and (b) :
$Y[k] = X[k]H[k]$
$= 1 + 2W_4^k + 4W_4^{2k} + 8W_4^{3k} - W_4^{2k} - 2W_4^{3k} - 4W_4^{4k} - 8W_4^{5k}$
Since $W_4^{4k} = W_4^{0k}$ and $W_4^{5k} = W_4^k$
$Y[k] = -3 - 6W_4^k + 3W_4^{2k} + 6W_4^{3k}, \quad 0 \leq k \leq 3$
Taking the inverse DFT:
$y[n] = -3\delta[n] - 6\delta[n-1] + 3\delta[n-2] + 6\delta[n-3], \quad 0 \leq n \leq 3$

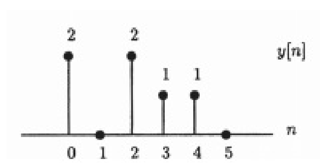
5. Consider the finite-length sequence $x[n]$ in Figure P8.13. The five-point DFT of $x[n]$ is denoted by $X[k]$. Plot the sequence $y[n]$ whose DFT is

$$Y[k] = W_5^{2k} X[k].$$

Figure 2: P8.13

SOL:

Using the properties of the DFT, we get $y[n] = x[((n-2))_5]$ that is y[n] is equal to $x[n]$ circularly shifted by 2. We get:



6. Figure P8.17 shows two finite-length sequences $x_1[n]$ and $x_2[n]$.What is the smallest N such that the N-point circular convolution of $x_1[n]$ and $x_2[n]$ are equal to the linear convolution of these sequences, i.e., such that $x_1[n] \, N \, x_2[n]$ = $x_1[n]$ * $x_2[n]$?

SOL: Looking at the sequences, we see that $x_1[n] * x_2[n]$ is non-zero for $1 \leq n \leq 8$. The smallest N such that $x_1[n] \, N \, x_2[n] = x_1[n] * x_2[n]$ is therefore N=9.

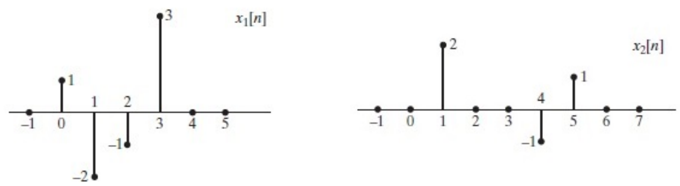


Figure 3: P8.17

7. Consider the finite-length sequence $x[n]$ in Fig P8.26. The four-point DFT of $x[n]$ is denoted $X[k]$. Plot the sequence $y[n]$ whose DFT is 4

$$Y[k] = W_4^{3k} X[k].$$

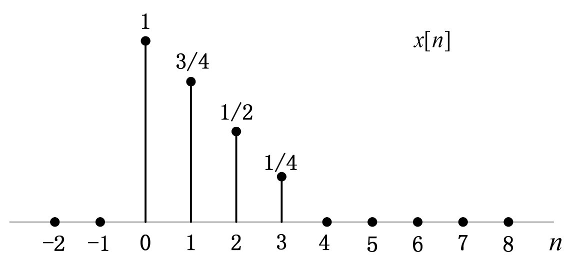


Fig 4: P8.26

SOL:

A. We know

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j\frac{2\pi}{N}kn}$$

$$= \sum_{n=0}^{N-1} x[n] e^{-j\frac{\pi}{N}n} e^{-j\frac{2\pi}{N}kn}$$

$$= \sum_{n=0}^{N-1} x[n] e^{-j\frac{\pi}{N}n(1+2k)}$$

Let

$$\omega_k = \frac{\pi(1+2k)}{N}.$$

Then

$$\tilde{X}[k] = \sum_{n=0}^{N-1} x[n] e^{-jw_k n} = X(e^{j\omega_k})$$

$$= X(e^{f(\frac{\pi + 2\pi k}{N})}), \quad k = 0, 1, ..., N-1$$

B. The frequencies of sampling are given by

$$\omega_k = \frac{\pi(1+2k)}{N}, \quad k = 0, 1, ..., N-1$$

C. Given the modified $\tilde{X}[k]$,we can use the inverse transform to find $\tilde{x}[n]$. To get $x[n]$ from $\tilde{x}[n]$ it is a simple point-by-point multiplication given by

$$x[n] = e^{j\frac{\pi}{N}n} \tilde{x}[n]$$

8. Fig. 5 shows two sequences,

$$x_1[n] = \begin{cases} 1, & 0 \le n \le 99, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$x_2[n] = \begin{cases} 1, & 0 \le n \le 9, \\ 0, & \text{otherwise}, \end{cases}$$

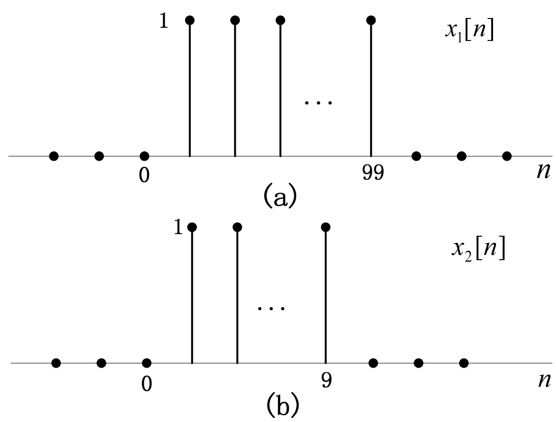


Fig 5: P8.31

(a) Determine and sketch the linear convolution $x_1[n] * x_2[n]$.

(b) Determine and sketch the 100-point circular convolution $x_1[n] \text{ (100) } x_2[n]$.

(c) Determine and sketch the 110-point circular convolution $x_1[n] \text{ (110) } x_2[n]$.

SOL:

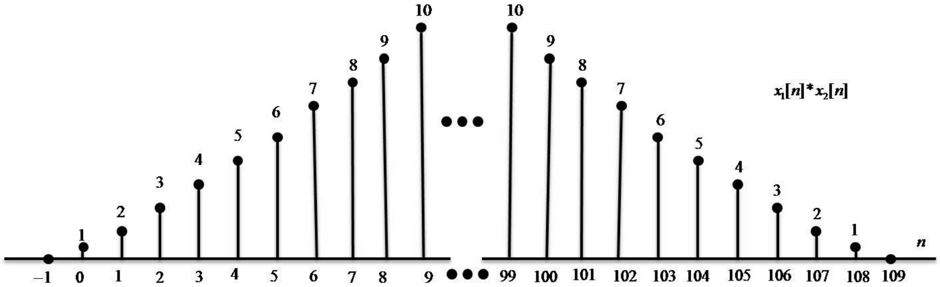


Figure 6:

(a) The linear convolution, $x_1[n] * x_2[n]$ is a sequence of length $100 + 10 - 1 = 109$.

(b) The circular convolution, $x_1[n]\langle 100\rangle x_2[n]$, can be obtained by aliasing the first 9 points of the linear convolution above.

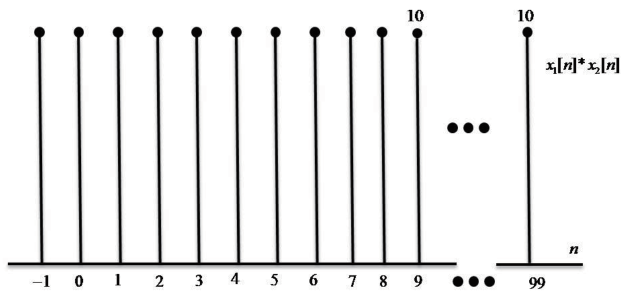


Figure 7:

(c) Since $N \geq 109$, the circular convolution $x_1[n]\langle 110\rangle x_2[n]$ will be equivalent to the linear convolution of part (a).

9. Consider two finite-length sequences $x[n]$ and $h[n]$ for which $x[n] = 0$ outside the interval $0 \leq n \leq 49$ and $h[n]$ outside the interval $0 \leq n \leq 9$.

   (a) What is the maximum possible number of nonzero values in the *linear* convolution of $x[n]$ and $h[n]$?

   (b) The 50-point *circular* convolution of $x[n]$ and $h[n]$ is

$$x[n] (50) h[n] = 10, \qquad 0 \leq n \leq 49.$$

   The first points of the *linear* convolution of $x[n]$ and $h[n]$ are

$$x[n] * h[n] = 5, \quad 0 \leq n \leq 4.$$

   Determine as many points as possible of the linear convolution of $x[n] * h[n]$.

(a) Since $x[n]$ is 50 points long, and $h[n]$ is 10 points long, the linear convolution $y[n] = x[n] * h[n]$ must be $50 + 10 - 1 = 59$ pts long.

(b) Circular convolution = linear convolutin + aliasing.
If we let $y[n] = x[n] * h[n]$, a more mathematical statement of the above is given by

$$x[n] \, (N) \, h[n] = \sum_{r=-\infty}^{\infty} y[n + rN], \quad 0 \leq n \leq (N-1)$$

For $N = 50$,

$$x[n] \, (50) \, h[n] = y[n] + y[n + 50], \quad 0 \leq n \leq 49$$

We are given: $x[n] \, (50) \, h[n] = 10$
Hence,

$$y[n] + y[n + 50] = 10, \quad 0 \leq n \leq 49$$

Also, $y[n] = 5, \quad 0 \leq n \leq 4$.
Using the above information:

$$\begin{array}{lll}
n = 0 & y[0] + y[50] = 10 \Rightarrow y[50] = 5 \\
 & \vdots \\
n = 4 & y[4] + y[54] = 10 \Rightarrow y[54] = 5 \\
n = 5 & y[5] + y[55] = 10 \Rightarrow y[55] = ? \\
 & \vdots \\
n = 8 & y[8] + y[58] = 10 \Rightarrow y[58] = ? \\
n = 9 & y[9] \qquad\qquad = 10 \\
 & \vdots \\
n = 49 & y[49] \qquad\quad = 10
\end{array}$$

3 Hence we can determine $y[n]$ for $9 \leq n \leq 54$ only (Note that $y[n]$ for $0 \leq n \leq 4$ is given.)

10. Suppose $x_1[n]$ is an infinite-length, stable (i.e., absolutely summable) sequence with DTFT given by

$$X_1(e^{j\omega}) = \frac{1}{1 - \frac{1}{3}e^{-j\omega}}.$$

Suppose $x_2[n]$ is a finite-length sequence of length N, and the N-point DFT of $x_2[n]$ is

$$X_2[m] = X_1(e^{j\omega})|_{\omega = 2\pi m/N}, \quad m = 0, 1, \cdots, N-1.$$

Determine $x_2[n]$.

SOL: $x_2[n]$ is $x_1[n]$ time aliased to have only N samples. Since

$$x_1[n] = \left(\frac{1}{3}\right)^n u[n],$$

We get

$$x_2[n] = \begin{cases} \frac{(\frac{1}{3})^n}{1-(\frac{1}{3})^N}, & n = 0, \cdots, N-1 \\ 0, & otherwise \end{cases}$$

11. Construct a flow graph for a 16-point radix-2 decimation-in-time FFT algorithm. Label all multipliers in terms of powers of W_{16}, and also label any branch transmittances that are equal to -1. Label the input and output nodes with the appropriate values of the input and DFT sequences, respectively. Determine the number of real multiplications and the number of real additions required to implement the flow graph.

    SOL:

    $N = 16$,
    number of multiplies $= \frac{N}{2}\log_2 N = 8 * 4 = 32.$
    number of additions $N\log_2 N = 16 * 4 = 64$

12. (programming) In this exercise we will practice filtering via DFT using either Python or MATLAB – choose your preferred language.

    (a) Create a random discrete signal of length 500 where each sample is a normally distributed random variable with mean 0 and variance 1. Let this signal $x[n]$. Plot its spectrum using `freqz` function for MATLAB, or `scipy.signal.freqz` in Python. (Hint: `randn` MATLAB function or `numpy.random.randn` for Python may be useful.)

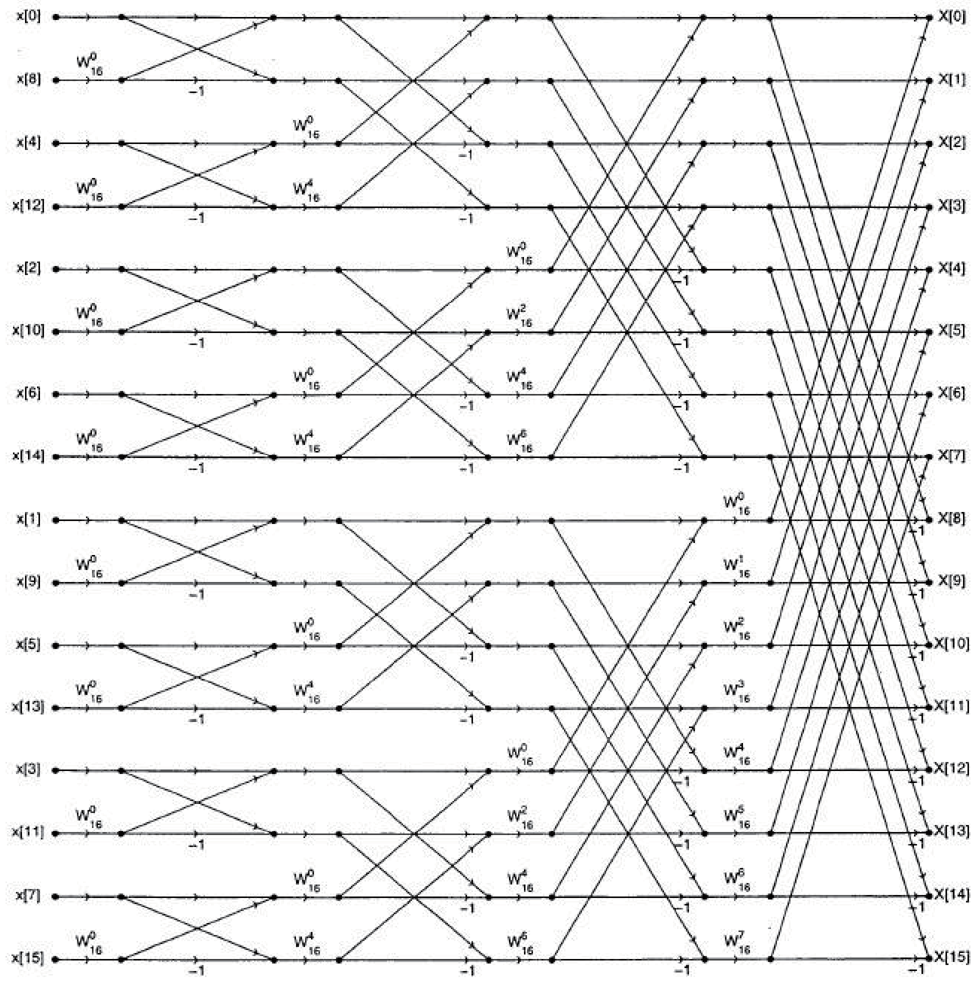
    (b) Create a lowpass FIR filter with cutoff at $\pi/4$ using Hamming window of length $2M + 1$ where $M = 250$. Let this impulse response be $h[n]$. Plot the spectrum of $h[n]$ using `freqz`.

    (c) Filter $x[n]$ in part (a) using your Hamming filter $h[n]$. Use plain linear convolution (`conv` function) for filtering. Let the filtered output be $y[n]$. Plot the spectrum of $y[n]$ using `freqz` function.

    (d) Now let's use DFT to see the spectrums. Namely use `fft` function in MATLAB (for Python put `from scipy.fftpack import fft, ifft` to use FFT or inverse FFT) to first calculate DFT of the signal created in part (a) and plot its magnitude response. Similarly apply `fft` function to calculate DFT of your filter and plot its magnitude response. Use the length of 1024-point FFT in your `fft` function. Use the log scale when you plot these magnitudes (function `semilogy` may be useful). Do these plots match the magnitude plots from (a) and (b)?

    (e) Using the results from (d), perform the filtering $x[n] * h[n]$ in *computationally efficient way*. In the class I have emphasized several times on the efficient filtering using DFT. Plot the spectrum of the filtered output $y[n]$ with `freqz` after applying this efficient method. Does this match with the plot you obtained in part (c)? (Hint: Use `ifft` function.)

9.22. The flow graph for 16 point radix-2 decimation-in-time FFT algorithm is shown below.



To determine the number of real multiplications and additions required to implement the flow graph,

Figure 8: