

Homework 4

Due: November, 4, 2011

- Question 1.** Textbook 8.23
- Question 2.** Textbook 8.26
- Question 3.** Textbook 8.39
- Question 4.** Textbook 8.43
- Question 5.** Textbook 9.25
- Question 6.** Textbook 9.36
- Question 7.** Textbook 9.42
- Question 8.** Matlab Question 1
- Question 9.** Matlab Question 2

8.23. Consider the real finite-length sequence $x[n]$ shown in Figure P8.23.

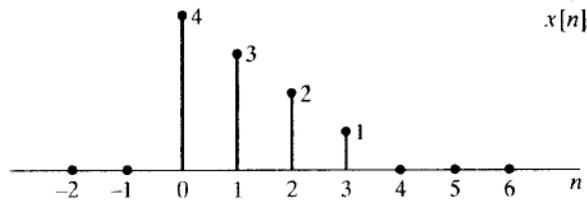


Figure P8.23

(a) Sketch the finite-length sequence $y[n]$ whose six-point DFT is

$$Y[k] = W_6^{5k} X[k],$$

where $X[k]$ is the six-point DFT of $x[n]$.

(b) Sketch the finite-length sequence $w[n]$ whose six-point DFT is

$$W[k] = \text{Im}\{X[k]\}.$$

(c) Sketch the finite-length sequence $q[n]$ whose three-point DFT is

$$Q[k] = X[2k + 1], \quad k = 0, 1, 2.$$

8.26. We have shown that the DFT $X[k]$ of a finite-length sequence $x[n]$ is identical to samples of the DTFT $X(e^{j\omega})$ of that sequence at frequencies $\omega_k = (2\pi/N)k$; i.e., $X[k] = X(e^{j(2\pi/N)k})$ for $k = 0, 1, \dots, N-1$. Now consider a sequence $y[n] = e^{-j(\pi/N)n}x[n]$ whose DFT is $Y[k]$.

- Determine the relationship between the DFT $Y[k]$ and the DTFT $X(e^{j\omega})$.
- The result of part (a) shows that $Y[k]$ is a differently sampled version of $X(e^{j\omega})$. What are the frequencies at which $X(e^{j\omega})$ is sampled?
- Given the modified DFT $Y[k]$, how would you recover the original sequence $x[n]$?

8.39. In the system shown in the Figure P8.39, $x_1[n]$ and $x_2[n]$ are both causal, 32-point sequences, i.e., they are both zero outside the interval $0 \leq n \leq 31$. $y[n]$ denotes the linear convolution of $x_1[n]$ and $x_2[n]$, i.e., $y[n] = x_1[n] * x_2[n]$.

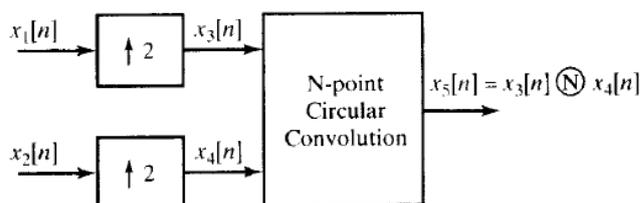


Figure P8.39

- Determine the values of N for which all the values of $y[n]$ can be completely recovered from $x_5[n]$.
- Specify explicitly how to recover $y[n]$ from $x_5[n]$ for the *smallest* value of N which you determined in part (a).

- 8.43. In Figure P8.43, $x[n]$ is a finite-length sequence of length 1024. The sequence $R[k]$ is obtained by taking the 1024-point DFT of $x[n]$ and compressing the result by 2.

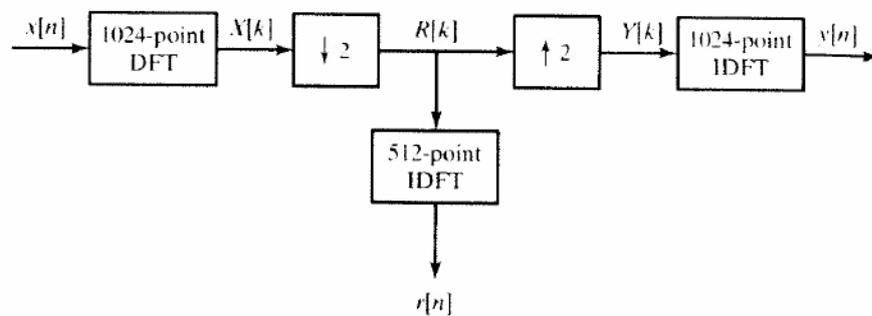


Figure P8.43

- (a) Choose the most accurate statement for $r[n]$, the 512-point inverse DFT of $R[k]$. Justify your choice in a few concise sentences.

- (i) $r[n] = x[n]$, $0 \leq n \leq 511$
- (ii) $r[n] = x[2n]$, $0 \leq n \leq 511$
- (iii) $r[n] = x[n] + x[n + 512]$, $0 \leq n \leq 511$
- (iv) $r[n] = x[n] + x[-n + 512]$, $0 \leq n \leq 511$
- (v) $r[n] = x[n] + x[1023 - n]$, $0 \leq n \leq 511$

In all cases $r[n] = 0$ outside $0 \leq n \leq 511$.

- (b) The sequence $Y[k]$ is obtained by expanding $R[k]$ by 2. Choose the most accurate statement for $y[n]$, the 1024-point inverse DFT of $Y[k]$. Justify your choice in a few concise sentences.

- (i) $y[n] = \begin{cases} \frac{1}{2} (x[n] + x[n + 512]), & 0 \leq n \leq 511 \\ \frac{1}{2} (x[n] + x[n - 512]), & 512 \leq n \leq 1023 \end{cases}$
- (ii) $y[n] = \begin{cases} x[n], & 0 \leq n \leq 511 \\ x[n - 512], & 512 \leq n \leq 1023 \end{cases}$
- (iii) $y[n] = \begin{cases} x[n], & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$
- (iv) $y[n] = \begin{cases} x[2n], & 0 \leq n \leq 511 \\ x[2(n - 512)], & 512 \leq n \leq 1023 \end{cases}$
- (v) $y[n] = \frac{1}{2} (x[n] + x[1023 - n])$, $0 \leq n \leq 1023$

In all cases $y[n] = 0$ outside $0 \leq n \leq 1023$.

9.25. The DFT is a sampled version of the DTFT of a finite-length sequence; i.e.,

$$\begin{aligned} X[k] &= X(e^{j(2\pi/N)k}) \\ &= X(e^{j\omega_k}) \Big|_{\omega_k=(2\pi/N)k} \\ &= \sum_{n=0}^{N-1} x[n]e^{-j(2\pi/N)kn} \quad k = 0, 1, \dots, N-1. \end{aligned} \quad (\text{P9.25-1})$$

Furthermore, an FFT algorithm is an efficient way to compute the values $X[k]$. Now consider a finite-length sequence $x[n]$ whose length is N samples. We want to evaluate $X(z)$, the z -transform of the finite-length sequence, at the following points in the z -plane

$$z_k = re^{j(2\pi/N)k} \quad k = 0, 1, \dots, N-1,$$

where r is a positive number. We have available an FFT algorithm.

- Plot the points z_k in the z -plane for the case $N = 8$ and $r = 0.9$.
- Write an equation [similar to Eq. (P9.25-1) above] for $X(z_k)$ that shows that $X(z_k)$ is the DFT of a modified sequence $\tilde{x}[n]$. What is $\tilde{x}[n]$?
- Describe an algorithm for computing $X(z_k)$ using the given FFT function. (*Direct evaluation is not an option.*) You may describe your algorithm using any combination of English text and equations, but you must give a step-by-step procedure that starts with the sequence $x[n]$ and ends with $X(z_k)$.

9.36. Consider a 1024-point sequence $x[n]$ constructed by interleaving two 512-point sequences $x_e[n]$ and $x_o[n]$. Specifically,

$$x[n] = \begin{cases} x_e[n/2], & \text{if } n = 0, 2, 4, \dots, 1022; \\ x_o[(n-1)/2], & \text{if } n = 1, 3, 5, \dots, 1023; \\ 0, & \text{for } n \text{ outside of the range } 0 \leq n \leq 1023. \end{cases}$$

Let $X[k]$ denote the 1024-point DFT of $x[n]$ and $X_e[k]$ and $X_o[k]$ denote the 512-point DFTs of $x_e[n]$ and $x_o[n]$, respectively. Given $X[k]$ we would like to obtain $X_e[k]$ from $X[k]$ in a computationally efficient way where computational efficiency is measured in terms of the total number of complex multiplies and adds required. One not-very-efficient approach is as shown in Figure P9.36:

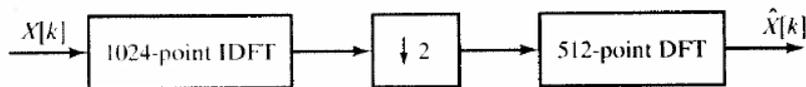


Figure P9.36

Specify the most efficient algorithm that you can (certainly more efficient than the block diagram of Figure P9.36) to obtain $X_e[k]$ from $X[k]$.

- 9.42. Consider a class of DFT-based algorithms for implementing a causal FIR filter with impulse response $h[n]$ that is zero outside the interval $0 \leq n \leq 63$. The input signal (for the FIR filter) $x[n]$ is segmented into an infinite number of possibly overlapping 128-point blocks $x_i[n]$, for i an integer and $-\infty \leq i \leq \infty$, such that

$$x_i[n] = \begin{cases} x[n], & iL \leq n \leq iL + 127, \\ 0, & \text{otherwise,} \end{cases}$$

where L is a positive integer.

Specify a method for computing

$$y_i[n] = x_i[n] * h[n]$$

for any i . Your answer should be in the form of a block diagram utilizing only the types of modules shown in Figures PP9.42-1 and PP9.42-2. A module may be used more than once or not at all.

The four modules in Figure P9.42-2 either use radix-2 FFTs to compute $X[k]$, the N -point DFT of $x[n]$, or use radix-2 inverse FFTs to compute $x[n]$ from $X[k]$.

Your specification must include the lengths of the FFTs and IFFTs used. For each "shift by n_0 " module, you should also specify a value for n_0 , the amount by which the input sequence is to be shifted.

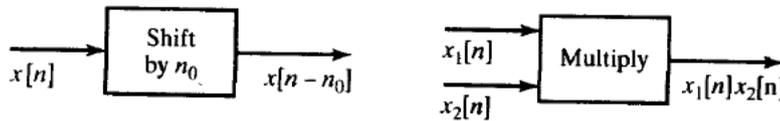


Figure P9.42-1

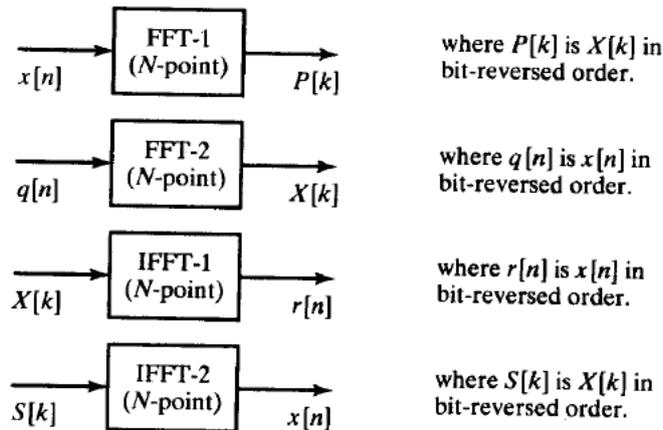


Figure P9.42-2

MATLAB Problem 1

You are given a speech file which has been distorted by passing through the following filter:

$$H(z) = 1 + 1.2z^{-199}$$

The original speech file was sampled at 8 kHz. The resulting speech file is on the web page accompanying this document. Listen to the file to hear the echo effect. The following Matlab code was used to generate the distorted speech file:

```
N = 200;  
b = zeros (N, 1);  
b(1) = 1;  
b(N) = 1.2;  
[x, Fs] = wavread (SpeechFileIn);  
y = filter (b, 1, x);  
wavwrite (y, Fs, SpeechFileOut);
```

- What is the echo delay in seconds?
- Where are the poles and zeros of $H(z)$?
- Is $H(z)$ stable?
- Is the inverse filter $A(z) = 1/H(z)$ stable? Elaborate, considering the two cases: $a[n]$ is left-sided and $a[n]$ is right-sided.
- Implement a direct inverse, i.e.

```
yq = wavread ('EchoSpeech.wav');  
xI = filter (1, b, yq);  
wavwrite (y, Fs, SpeechFileOut);
```

Listen to the resulting speech. Theoretically, the pole in the inverse filter should be cancelled by the zero in the original filter. Explain what you hear. You may want to consider the fact that the values you read from the file have been quantized to 16-bits in the speech file, i.e. you may consider the values you read to be the true filtered signal plus a small amount of "noise" due to quantization.

- Implement a scheme in Matlab to properly "inverse filter" the distorted speech file to undo the echo. The procedure should read in the distorted file, do the necessary processing and then write a "cleaned up" WAVE file. Hint: A filter with a left-sided impulse response can be implemented by time reversing the filter, time reversing the signal, filtering and then time-reversing the output.
- Explain the technique you used to remove the echo. Is there a time shift associated with your file relative to the original file (which you don't have access to)?

The answer to this problem should include a listing of your Matlab program.

Matlab Problem 2

This problem examines the relationship between sampling in the frequency domain and time-aliasing (wrapping). Consider the signal $x[n]$. Form the wrapped signal as

$$\tilde{x}[n] = \sum_{k=-\infty}^{\infty} x[n - kP]. \quad (2.1)$$

The signal $\tilde{x}[n]$ is periodic with period P .

Let $X(\omega)$ be the DTFT of $x[n]$. Then it can be shown that the DFT of the time-aliased signal $\tilde{x}[n]$ can be determined from samples of $X(\omega)$,

$$\tilde{x}[n] \leftrightarrow X\left(\frac{2\pi k}{P}\right). \quad (2.2)$$

1. Write a Matlab routine that takes a causal signal $x[n]$ of arbitrary length and returns one period of $\tilde{x}[n]$. The function should be of the form
`xt = wrapP (x, P).`
The routine `wrapP` should determine the length of the vector `x` internally (using the `length` function) and should work properly if the length of the vector is smaller or larger than `P`.
2. Write a Matlab routine that calculates $\tilde{x}[n]$ using frequency-domain techniques. The function should be of the form
`xt = wrapF (x, P).`
The routine `wrapF` should determine the length of the vector `x` internally and should work properly if the length of the vector is smaller or larger than `P`. Internally, the routine should evaluate the DTFT at the sample points and then use an inverse transform to return the time-aliased signal.
3. Form the test signal $x[n]$,

$$x[n] = \frac{1}{2} \left(1 - \cos\left(\frac{\pi(2n+1)}{N}\right)\right), \quad 0 \leq n \leq N-1.$$

Use $N = 32$. Plot the output of the two routines for $P = 24$ and $P = 64$.

Notes

- a) Equation (2.1) can be simplified for finite-length causal signals. If n is evaluated between 0 and $P-1$,

$$\tilde{x}[n] = \sum_{k=0}^{M-1} x[n - kP], \quad 0 \leq n \leq P-1,$$

where an appropriate value of M has to be determined for each value of n .

- b) The equations above are based on indices which start at 0. In Matlab, indices start at 1. One way to handle the difference is to use variables which are zero-based and then refer to vector elements with an offset of one. For instance if the variable

`n` is zero-based, one can refer to the Matlab vector element `x(n+1)` corresponding to $x[n]$ in the mathematical analysis.

- c) The `freqz` function can be used to determine samples of the DTFT.