EE 123 University of California, Berkeley
Anant Sahai March 15, 2007

# Digital Signal Processing

# Midterm 2 Solutions

**Instructions**

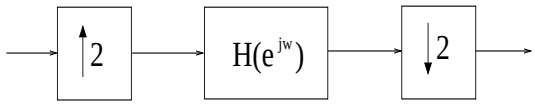
- Total time allowed for the exam is 80 minutes
- Please write your name and SID on every page of the exam
- Some useful formulas:
  - $N$ point Discrete Fourier Transform (DFT)

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

  - Inverse Discrete Fourier Transform (IDFT)

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

**1.** (*40 points*) Let $H(e^{j\omega})$ be the frequency response of a discrete time LTI filter. The filter is arranged in the following cascade combination given below:




**a.** (*10 pts*) Is the system given in the above figure linear? Justify your answer by giving a brief proof or a counterexample.

*Solution:* Since each of the individual blocks in the above cascade configuration are linear, the whole system is also linear.

**b.** (*10 pts*) Is the system given in the above figure time-invariant? Justify your answer by giving a brief proof or a counterexample.

*Solution:* Let $x[n]$ denote the input to the system, $w[n]$ the output after upconversion, $g[n]$ the output after passing through the filter $H(e^{j\omega n})$ and $y[n]$ denote the overall output of the cascade system. Also, let $h[n]$ be the impulse response of the filter. We already know that the system is linear. So, we only need to verify time-invariance when the input to the system is a delta function.
Let $x[n] = \delta[n]$. Then, $w[n] = \delta[n]$ and $g[n] = h[n]$. Therefore, $y[n] = h[2n]$.
Now, let the input to the system be a time-shifted delta function, i.e., $\tilde{x}[n] = \delta[n - n_0]$. Then,

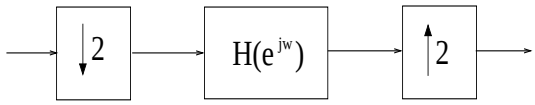
$$\begin{aligned} \tilde{w}[n] &= \left\{ \begin{array}{ll} 1 & \text{if } n = 2n_0 \\ 0 & \text{otherwise} \end{array} \right\} \\ &= \delta[n - 2n_0] \end{aligned}$$

This implies that $\tilde{g}[n] = \tilde{h}[n - 2n_0]$. Finally,

$$\begin{aligned} \tilde{y}[n] &= \tilde{g}[2n] \\ &= \tilde{h}[2n - 2n_0] \\ &= \tilde{h}[2(n - n_0)] \\ &= y[n - n_0] \end{aligned}$$

This proves that the system is time-invariant.

Now consider the following alternate cascade formation:




**c.** (*10 pts*) Is the system given in the above figure linear? Justify your answer by giving a brief proof or a counterexample.

*Solution:* Since each of the individual blocks in the above cascade configuration are linear, the whole system is also linear.

**d.** (*10 pts*) Is the system given in the above figure time-invariant? Justify your answer by giving a brief proof or a counterexample.

*Solution:* Let $x[n]$ denote the input to the system, $w[n]$ the output after decimation, $g[n]$ the output after passing through the filter $H(e^{j\omega n})$ and $y[n]$ denote the overall output of the cascade system. Also, let $h[n]$ be the impulse response of the filter.
We already know that the system is linear. So, we only need to verify time-invariance when the input to the system is a delta function.
Let $x[n] = \delta[n]$. Then, $w[n] = \delta[n]$ and $g[n] = h[n]$. Therefore,

$$y[n] = \left\{ \begin{array}{cc} h[n/2] & \text{if } n \text{ is even} \\ 0 & \text{if } n \text{ is odd} \end{array} \right\}$$

Now, let the input to the system be a time-shifted delta function, i.e., $\tilde{x}[n] = \delta[n-1]$. Then, the output after decimation is given by $\tilde{w}[n] = 0$. Hence, $\tilde{g}[n] = 0$ and $\tilde{y}[n] = 0$.
Since, $\tilde{y}[n] \neq y[n-1]$ the system is not time-invariant.

**2.** (*80 points*) Let $x[n]$ be a given time sequence. Assume that the length of the sequence is $M = 3^\nu$. The goal is to compute its $M$ point DFT coefficients $X[k]$'s.

**a.** (*10 pts*) Find the total number of complex multiplications and additions required to compute $X[k]$'s directly from the definition of the DFT.

*Solution:* From the definition of the DFT,

$$X[k] = \sum_{n=0}^{M-1} x[n] W_M^{nk}, \qquad k = 0, 1, \cdots, M-1$$

To compute $X[k]$ for a single $k$, we need

- $M - 1$ complex additions
- $M$ complex multiplications

Therefore, to compute the $M$ point DFT of $x[n]$ we need a total of

- $M(M-1)$ complex additions
- $M^2$ complex multiplications

**b.** (*20 pts*) Describe in detail a fast algorithm to compute the DFT of $x[n]$.
*Hint:* Think of a recursive algorithm similar to the decimation in time FFT algorithm.

*Solution:* From the definition of the DFT,

$$\begin{aligned}
X[k] &= \sum_{n=0}^{M-1} x[n]W_M^{nk} \\
&= \sum_{n=0 \bmod 3} x[n]W_M^{nk} + \sum_{n=1 \bmod 3} x[n]W_M^{nk} + \sum_{n=2 \bmod 3} x[n]W_M^{nk} \\
&= \sum_{r=0}^{\frac{M}{3}-1} x[3r]W_M^{3rk} + \sum_{r=0}^{\frac{M}{3}-1} x[3r+1]W_M^{(3r+1)k} + \sum_{r=0}^{\frac{M}{3}-1} x[3r+2]W_M^{(3r+2)k} \\
&= \sum_{r=0}^{\frac{M}{3}-1} x[3r]W_{\frac{M}{3}}^{rk} + W_M^k \sum_{r=0}^{\frac{M}{3}-1} x[3r+1]W_{\frac{M}{3}}^{rk} + W_M^{2k} \sum_{r=0}^{\frac{M}{3}-1} x[3r+2]W_{\frac{M}{3}}^{rk} \\
&= G[k] + W_M^k H[k] + W_M^{2k} L[k] \qquad (1)
\end{aligned}$$

Where $G[k]$, $H[k]$ and $L[k]$ are the $\frac{M}{3}$ point DFT coefficients corresponding to the sequences $x[3r]$, $x[3r+1]$ and $x[3r+2]$ respectively. Therefore, the problem of computing a $M$ point DFT has been reduced into computing three $\frac{M}{3}$ point DFTs and combining them according to Eqn. (1). This process can be repeated recursively until a stage where you need to compute 3 point DFTs.

**c.** (*10 pts*) Find the total number of complex multiplications and additions required to compute $X[k]$'s using the algorithm from part $(b)$.

*Solution:* From Eqn. (1), it is clear that we need $2M$ complex multiplications and $2M$ complex additions for computing $X[k]$ given the $G[k]$, $H[k]$ and $L[k]$ for $k = 0, 1, \cdots, M-1$.
The number of such recursive stages in the algorithm is $\log_3 M$. Therefore, the algorithm in part $(b)$ requires a total of

- $2M \log_3 M$ complex multiplications
- $2M \log_3 M$ complex additions

**d.** (*20 pts*) Now assume that the length of the sequence $x[n]$ is $M = 2^{\nu_1}3^{\nu_2}$. For this case, describe a fast algorithm to compute the DFT. Also, find the total number of complex multiplications and additions required under this algorithm.

*Solution:* There are several possible fast algorithms to compute the DFT of a sequence of length $M = 2^{\nu_1}3^{\nu_2}$. We give one possible such algorithm. One can easily modify this algorithm to come up with similar algorithms having the same computational complexity.

**Algorithm:**

- Use the radix-2 'decimation in time' FFT algorithm discussed in class to write

$$X[k] = G[k] + W_M^k H[k] \qquad k = 0, 1, \cdots, M-1$$

  where $G[k]$ and $H[k]$ are the DFT coefficients of two $\frac{M}{2} = 2^{\nu_1 - 1}3^{\nu_2}$ point sequences, $x[2r]$ and $x[2r+1]$ respectively. So, we have reduced the problem of computing a $M$ point DFT into the problem of computing two $\frac{M}{2}$ point DFTs by partitioning the original sequence.

- 'Step 1' can be repeated recursively as long as the partitioned sequences have lengths divisible by two. So, the radix-2 decimation in time recursion will continue until the partitioned sequences are each of length $3^{\nu_2}$

- At this stage, we have represented the DFT coefficients $X[k]$'s in terms of DFT coefficients of sequences of length $3^{\nu_2}$. The number of such sequences are $2^{\nu_1}$.

- Finally, to complete the algorithm we need to compute the DFT coefficients of $3^{\nu_2}$ point sequences. This can be done by applying the algorithm in part $(b)$ for each of these sequences.

**Computational Complexity of the Algorithm:**
Our algorithm has two distinct phases. First we divide using the radix-2 decimation in time algorithm, then we divide using the radix-3 decimation in time algorithm from part $(b)$. We compute the number of computations required for each of the two phases separately and add them up to give the total number of computations.
We have $\nu_1$ radix-2 decimation in time stages. This requires

- $M\nu_1$ complex multiplications

- $M\nu_1$ complex additions

After the radix-2 decimation in time stages, we need to compute the DFT of $3^{\nu_2}$ point sequences. The number of such sequences is $2^{\nu_1}$.
From part $(c)$ of the question, we know that to compute the DFT of a $3^{\nu_2}$ point sequence we need

- $2 \cdot 3^{\nu_2}\nu_2$ complex multiplications

- $2 \cdot 3^{\nu_2}\nu_2$ complex additions

Since there are a total of $2^{\nu_1}$ such sequence, we need

- $2^{\nu_1}(2 \cdot 3^{\nu_2}\nu_2) = 2M\nu_2$ complex multiplications

- $2^{\nu_1}(2 \cdot 3^{\nu_2}\nu_2) = 2M\nu_2$ complex additions

Therefore, the total number of computations needed are

- $M\nu_1 + 2M\nu_2 = M(\nu_1 + 2\nu_2)$ complex multiplications
- $M\nu_1 + 2M\nu_2 = M(\nu_1 + 2\nu_2)$ complex additions

Let $x[n]$ and $y[n]$ be two $M = 3^\nu$ point sequences. The goal is to compute the linear convolution of $x[n]$ and $y[n]$, i.e., $z[n] = x[n] * y[n]$.

**e.** (*10 pts*) Find the total number of complex multiplications and additions required to compute $z[n]$ using the DFT algorithm in part $(b)$.

*Solution:* $z[n] = x[n] * y[n]$. Since both $x[n]$ and $y[n]$ are $M$ point sequences, the length of the sequence $z[n]$ is atmost $2M - 1$. Let $N \geq (2M - 1)$. To compute $z[n]$ we need to perform the following operations

1. Compute the $N$ point DFT $X[k]$ and $Y[k]$.
2. Compute the product $Z[k] = X[k] \cdot Y[k]$.
3. Compute the IDFT of $Z[k]$, whose output will give us $z[n]$.

Now, in order to use the algorithm in part $(b)$ to compute the DFT and IDFT (in steps 1 and 3) we must choose $N$ to be the smallest power of three greater than $(2M - 1)$. Since $M = 3^\nu$, the smallest power of three greater than $(2 \cdot 3^\nu - 1)$ is $3^{\nu+1}$. Hence, we choose $N = 3^{\nu+1}$
The computational complexity is given by

- **Step 1:** To compute two $N$ point DFTs we need
  - $4N \log_3 N$ complex multiplications
  - $4N \log_3 N$ complex additions
- **Step 2:** To compute the product of $X[k]$ and $Y[k]$ we need
  - $N$ complex multiplications
- **Step 3:** To compute an $N$ point IDFT we need
  - $2N \log_3 N$ complex multiplications
  - $2N \log_3 N$ complex additions

Therefore, to compute $z[n]$ using the DFT algorithm in part $(b)$ we need a total of

- $6N \log_3 N + N$ complex multiplications
- $2N \log_3 N$ complex additions

**f.** (*10 pts*) Find the total number of complex multiplications and additions required to compute $z[n]$ using the FFT algorithm discussed in class, i.e., the FFT algorithm for powers of two. **Hint:** The question asks for a linear convolution, which is different from a circular convolution.

*Solution:* $z[n] = x[n] * y[n]$. Since both $x[n]$ and $y[n]$ are $M$ point sequences, the length of the sequence $z[n]$ is atmost $2M - 1$. Let $N \geq (2M - 1)$. To compute $z[n]$ we need to perform the following operations

1. Compute the $N$ point DFT $X[k]$ and $Y[k]$.
2. Compute the product $Z[k] = X[k] \cdot Y[k]$.
3. Compute the IDFT of $Z[k]$, whose output will give us $z[n]$.

In this question we are specifically asked to use the radix-2 FFT algorithm discussed in class to compute the DFT and IDFT in steps 1 and 3. So, we must choose $N$ to be the smallest power of two greater than $(2M - 1)$, i.e., $N = 2^{\beta}$, where $\beta$ is the smallest number such that $2^{\beta} \geq (2 \cdot 3^{\nu} - 1)$.
In this case, the computational complexity is given by

- **Step 1:** To compute two $N$ point DFTs using the radix-2 FFT algorithm, we need
  - $2N \log_3 N$ complex multiplications
  - $2N \log_3 N$ complex additions
- **Step 2:** To compute the product of $X[k]$ and $Y[k]$ we need
  - $N$ complex multiplications
- **Step 3:** To compute an $N$ point IDFT using the radix-2 IFFT algorithm, we need
  - $N \log_3 N$ complex multiplications
  - $N \log_3 N$ complex additions

Therefore, to compute $z[n]$ using the radix-2 FFT algorithm we need a total of

- $2N \log_3 N + N$ complex multiplications
- $2N \log_3 N$ complex additions

**3.** (*40 points*) Let $h[n]$ be the impulse response of a LTI filter. Assume that $h[n] = 0$ for $n < 0$ and $n \geq L$. Let $d[n]$ be an $N$ ($N > L$) point data sequence, i.e., $d[n] = 0$ for $n < 0$ and $n \geq N$. Denote this sequence by the vector

$$\mathbf{d} = [d[0], d[1], \cdots, d[N-1]]$$

Create a sequence $x[n]$ by adding a $L-1$ point *cyclic prefix* to $d[n]$. This can be represented by the following vector

$$\mathbf{x} = [d[N-L+1], d[N-L+2], \cdots, d[N-1], d[0], d[1], \cdots, d[N-1]]$$

Let $w[n]$ denote the output of the filter when the input is $x[n]$. We ignore the first $L-1$ symbols of the output sequence $w[n]$ and collect the next $N$ symbols. Let $y[n]$ denote this output of length $N$, which can be represented as

$$\mathbf{y} = [w[L], w[L+1], \cdots, w[N+L-1]]$$

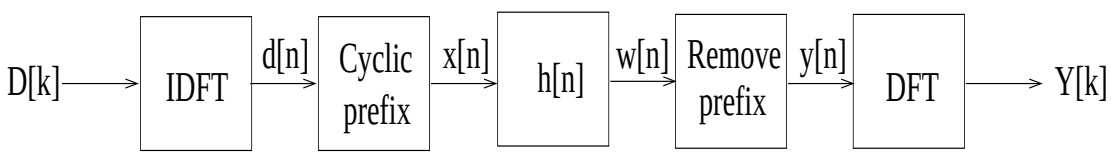



Figure 1: Overall system

**a.** (*10 pts*) Compute the sequence $y[n]$ as a function of $d[n]$ and the filter impulse response $h[n]$.

*Solution:* From Fig. 3, we have

$$\begin{aligned}
w[n] &= x[n] * h[n] \\
&= \sum_{l=0}^{L-1} h[l]x[n-l], \qquad n = 1, 2, \cdots, N+L-1 \\
\Rightarrow y[n] &= \sum_{l=0}^{L-1} h[l]x[n-l], \qquad n = L, 2, \cdots, N+L-1 \\
&= \sum_{l=0}^{L-1} h[l]d[((n-L-l))_N] \\
\Rightarrow y[n] &= h[n] \otimes_N d[n]
\end{aligned}$$

Here $h[n]$ is zero-padded so as to make the length of the sequence $N$, that is, $h[n] = [h[0], h[1], \cdots, h[L-1], 0, 0, \cdots, 0]$.

**b.** (*10 pts*) Let $Y[k]$, $k = 0, 1, \cdots, N-1$ denote the $N$ point DFT of the sequence $y[n]$. Determine the sequence $Y[k]$ as a function of $H[k]$ and $D[k]$, the $N$ point DFTs of $h[n]$ and $d[n]$ respectively (See Fig. 1).

*Solution:* From part $(a)$ of the question we know that

$$y[n] = d[n] \otimes_N h[n]$$

i.e., the $N$ point circular convolution of $d[n]$ and $h[n]$. Taking the DFT on both sides of the above equation, we have

$$Y[k] = D[k] \cdot H[k], \qquad k = 0, 1, \cdots, N-1$$

**c.** (*20 pts*) Let $N$ be a multiple of $L$, i.e., $N = mL$ for some integer $m > 0$. Suppose that the DFT coefficients $D[mk]$, $k = 0, 1, \cdots, L-1$ are known to be equal to 1. In this case can you recover the filter coefficients $h[n]$ from the $y[n]$ sequence. How would you do so?

*Solution:* From part ($b$) of the question, we know that

$$Y[k] = D[k] \cdot H[k], \qquad k = 0, 1, \cdots, mL-1$$

Since $D[mk] = 1$, for $k = 0, 1, \cdots, L-1$, we have

$$Y[mk] = H[mk], \qquad k = 0, 1, \cdots, L-1$$

By the definition of the DFT,

$$\begin{aligned} Y[mk] = H[mk] &= H(e^{j\omega})|_{\omega=\frac{2\pi mk}{N}} \\ &= H(e^{j\omega})|_{\omega=\frac{2\pi mk}{mL}} \\ &= H(e^{j\omega})|_{\omega=\frac{2\pi k}{L}} \qquad k = 0, 1, \cdots, L-1 \\ &= L \text{ point DFT of } h[n] \end{aligned}$$

Note, that the last equality is true only because $h[n]$ is an $L$ point sequence. Finally, in order to recover the filter coefficients $h[n]$ we just need to take the IDFT of the sequence $Y[mk]$, $k = 0, 1, \cdots, L-1$.