

# Solution: Problem Set 5

EECS123: Digital Signal Processing

Prof. Ramchandran  
Spring 2008

1. (a) Overlap Add:

If we divide the input into sections of length  $L$ , each section will have an output length:

$$L + 100 - 1 = L + 99.$$

Thus, the required length is,

$$L = 256 - 99 = 157.$$

If we had 63 sections,  $63 \times 157 = 9891$ , there will be a remainder of 109 points. Hence, we must pad the remaining data to 256 and use one more DFT computation.

Therefore, we require 64 DFTs and 64 IDFTs. Since  $h[n]$  also requires a DFT, the total is:

65 DFTs and 64 IDFTs.

(b) Overlap save:

We require 99 zeros to be padded in front of the sequence. The first 99 points of the output of each section will be discarded. Thus the length after padding is 10099 points. The length of each section overlap is  $256 - 99 = 157 = L$ .

We require  $65 \times 157 = 10205$  to get all 10099 points. Because  $h[n]$  also requires a DFT:

66 DFTs and 65 IDFTs.

(c) Ignoring the transients at the beginning and end of the direct convolution, each output point requires 100 multiplications and 99 additions.

Overlap add:

$$\begin{aligned} \# \text{multiplications} &= 129(1024) = 132096, \\ \# \text{additions} &= 129(2048) = 264192. \end{aligned}$$

Overlap save:

$$\begin{aligned}\#\text{multiplications} &= 131(1024) = 134144, \\ \#\text{additions} &= 131(2048) = 268288.\end{aligned}$$

Direct convolution:

$$\begin{aligned}\#\text{multiplications} &= 100(100000) = 1000,000, \\ \#\text{additions} &= 99(100000) = 990,000.\end{aligned}$$

**Note:** The number of additions can be accurately estimated, but it does not cause much difference in the answers. Give yourself full points if your answer is within 5%.

**Note:** Give yourself full points even if you missed to notice that  $h[n]$  should contribute only 1 to the number of FFT. But be careful about this mistake in the future.

2. (a) The length of the sequence is  $L + P - 1$ .
- (b) In evaluating  $y[n]$  using the convolution sum, each non-zero value of  $h[n]$  is multiplied once with every non-zero value of  $x[n]$ . This can be graphically seen using the flip and slide view of convolution. The total number of real multiplications is therefore  $LP$ .
- (c) To compute  $y[n] = h[n] * x[n]$  using the DFT, we use the procedure described below.
 

**step 1:** Compute  $N$  point DFTs of  $x[n]$  and  $y[n]$ .

**step 2:** Compute  $Y[k] = H[k]X[k]$  for all values of  $k$ .

**step 3:** Compute  $y[n]$  by inverse DFT.

Since  $y[n]$  has length  $L + P - 1$ , therefore,  $N$  must be greater than or equal to  $L + P - 1$  so the circular convolution implied by step 2 is equivalent to linear convolution.
- (d) For these signals,  $N$  is large enough so that the circular convolution of  $x[n]$  and  $h[n]$  and their linear convolution produce the same result. Counting the number of complex multiplications for the procedure in part (b), we get,

DFT of $x[n]$	$(N/2) \log_2 N$
DFT of $h[n]$	$(N/2) \log_2 N$
$Y[k] = X[k]H[k]$	$N$
Inverse DFT of $Y[k]$	$(N/2) \log_2 N$
Total	$(3N/2) \log_2 N + N$

Since there are 4 real multiplications for every complex multiplication, we see that the procedure takes  $6N \log_2 N + 4N$  real multiplications. Using the answer from part (a), we see that the direct method requires  $(N/2)(N/2) = N^2/4$  real multiplications.

The direct method is superior to the FFT method till about  $N = 128$ . For  $N = 256$  and beyond, the FFT method is superior. (Use calculator).

3. The structure is an input followed by three stages; here are the inputs, intermediate values, and outputs to check against your results. Refer to the length-8 decimation in time FFT diagram in the book for the structure of operations and the twiddle factors.

Decimation in Time			
1	6	16	36
5	-4	-4+j4	-4+j9.657
3	10	-4	-4+j4
7	-4	-4-j4	-4+j1.657
2	8	20	-4
6	-4	-4+j4	-4-j1.657
4	12	-4	-4-j4
8	-4	-4-j4	-4-j9.657

4. (a) It is interesting to note that (linear) convolution and polynomial multiplication are the same operation. Here, we replace

$$p(x) = \sum_{i=0}^{L-1} a_i x^i, \quad q(x) = \sum_{i=0}^{M-1} b_i x^i,$$

with

$$p[n] = \sum_{i=0}^{L-1} a_i \delta[n-i], \quad q[n] = \sum_{i=0}^{M-1} b_i \delta[n-i].$$

Then,

$$r[n] = p[n] * q[n].$$

the coefficients in  $r[n]$  will be identically equal to those of  $r(x)$ . We can compute  $r[n]$  with circular convolution, instead of linear convolution, by zero padding  $p[n]$  and  $q[n]$  to a length  $N = L + M - 1$ . This zero padding ensures that linear convolution and circular convolution will give the same result.

**Note:** Exact proof can be done as follows (sketch):

$$r(x) = \sum_{i=0}^{L-1} \sum_{j=0}^{M-1} a_i b_j x^{i+j},$$

$$\text{coeff of } x^k \text{ in } r(x) = \sum_{i=0}^{L-1} \sum_{j=0}^{M-1} a_i b_j \delta[i+j-k] = \sum_{i=0}^{L-1} a_i \cdot b_{k-i}.$$

- (b) We can implement the circular convolution of  $p[n]$  and  $q[n]$  using the following procedure:

**step 1:** Compute  $N$  point FFTs of  $p[n]$  and  $q[n]$ . This gives  $P[k]$  and  $Q[k]$ .

**step 2:** Compute  $R[k] = P[k]Q[k]$  for all values of  $k$ .

**step 3:** Compute  $r[n]$  by inverse DFT.

The direct computation of  $r[n]$  requires  $L \cdot M$  real multiplications, since  $a_i$  and  $b_i$  are real. We know that  $L + M = 2^\nu$ . Subject to this sum-constraint,  $L \cdot M$  is maximized when  $L = M = 2^{\nu-1}$ . Thus, the maximum number of multiplications needed in direct computation is  $(L + M)^2/4$ .

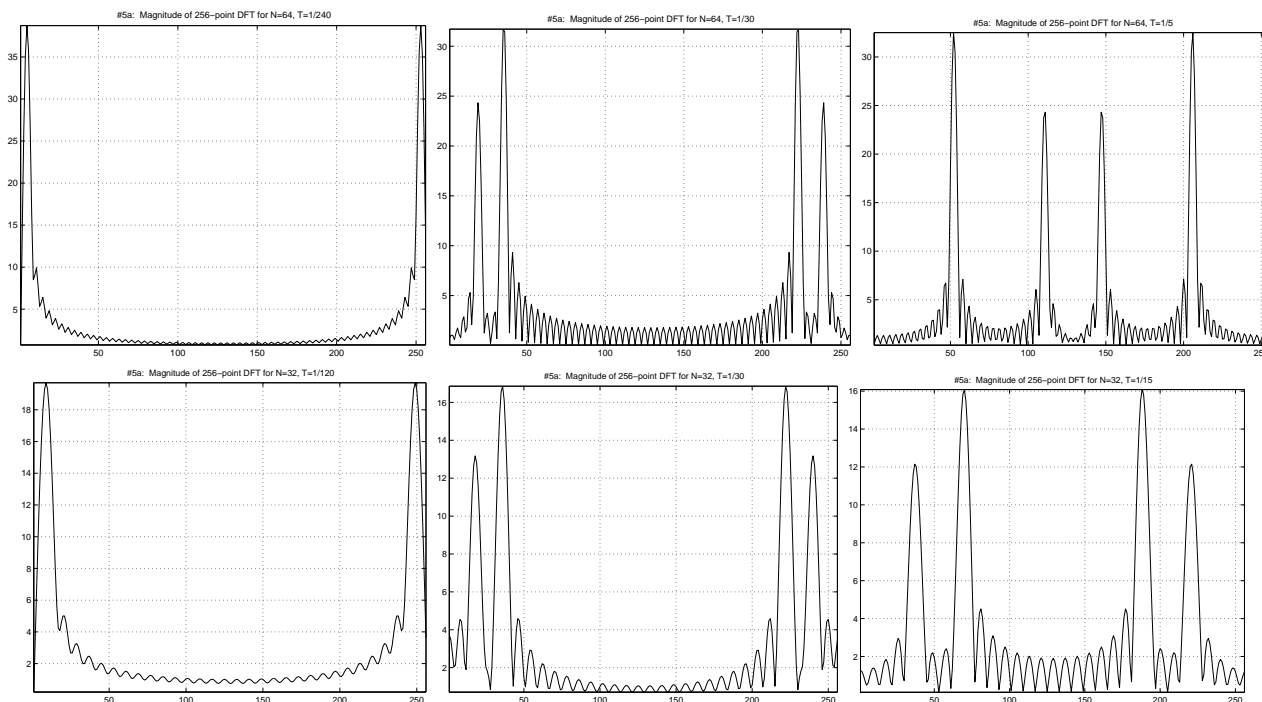
**Note:** Give yourself full point, even if you counted  $LM = 2^{2\nu}$ . But watch out for this mistake in the future.

Assuming that a length  $L + M$  FFT computation takes  $[(L + M)/2] \log_2(L + M)$  complex multiplications, we count the complex multiplications required in the procedure described above to be:

FFTs of $p[n]$ and $q[n]$	$2((L + M)/2) \log_2(L + M) = (L + M) \log_2(L + M)$
$R[k] = P[k]Q[k]$	$L + M$
Inverse FFT of $R[k]$	$((L + M)/2) \log_2(L + M)$
Total	$(3(L + M)/2) \log_2(L + M) + (L + M)$

Since a complex multiplication is computed using 4 real multiplications, the number of real multiplications required by this technique is  $6(L + M) \log_2(L + M) + 4(L + M)$ . For  $L + M = 2^\nu$ , we find that for  $(L + M) \geq 256$  the FFT approach is more efficient.

5. **Note for grading: Grades parts (a) and (b) as separate problems.**



- (a) i. Changing  $T$  has the effect of shifting the position of the sinc function along the  $\lambda$  axis of the DTFT. As  $T$  increases, the frequency resolution improves but the total bandwidth of analog frequencies spanned by the DTFT decreases. Changing  $N$  has the effect of widening or narrowing the sinc functions (scaling them about their centers). As  $N$  increases, the widths of all of the lobes decreases and frequency resolution is improved.
- ii. The best frequency resolution will be obtained when  $T$  is chosen to sample the signal at or near the Nyquist frequency (that is, let  $T \approx 1/(2f)$  where  $f$  is the highest frequency present in the signal). This results in the maximum possible resolution

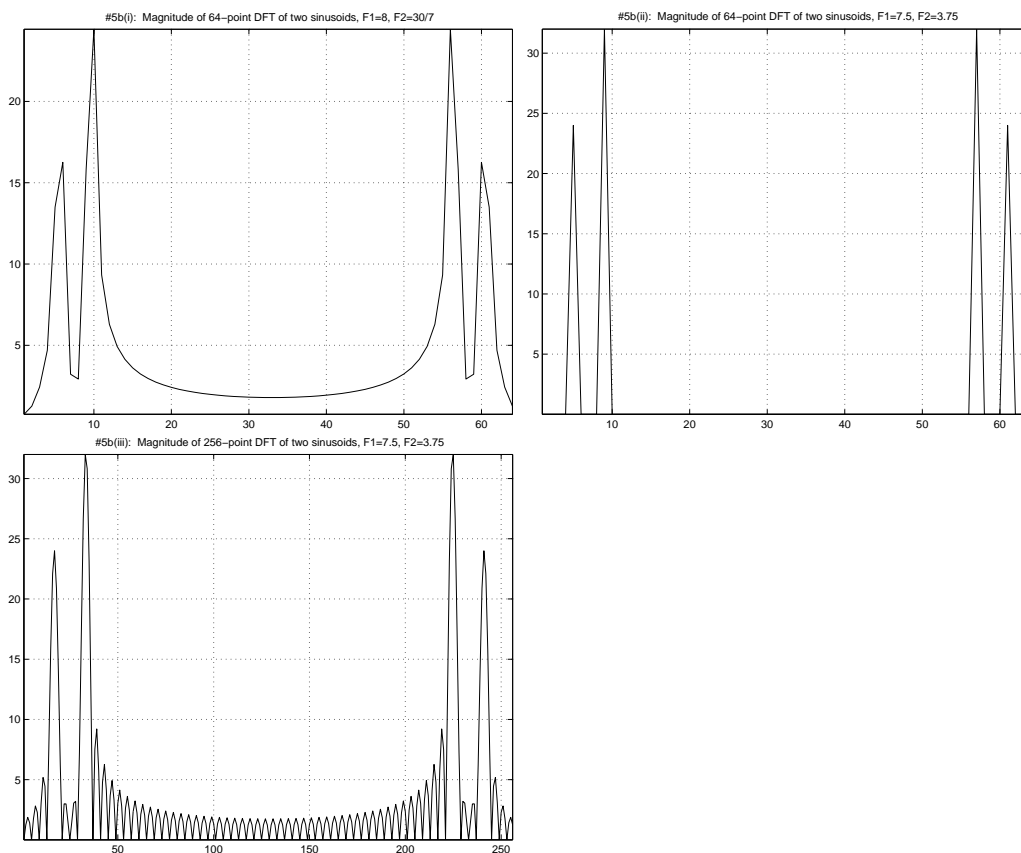
of frequencies less than the highest frequency in the signal, assuming of course that the highest frequency in the signal is known *a priori*. Another potential problem would be that the highest frequency sinusoid would alias with itself in the middle of the DFT, but if resolution of the various components is the only goal, this won't matter). In this example,  $NT$  is constant, equal to 2 seconds, so after  $T$  is chosen,  $N = 2/T$ .

- iii. The conditions on  $T$  will be similar; choose  $T$  for a sampling frequency near the Nyquist rate, with a safety factor as necessary depending on what is known of the analog signal.  $N$  is fixed at 128.
- iv. Various approaches are possible here. If clean separation of the peaks is desired, we could require that the peaks are separated by at least the width of one main lobe,  $4\pi/N$ :

$$\begin{aligned}
 (\Omega_2 - \Omega_1)T &\geq \frac{4\pi}{N_{\min}} \\
 (f_2 - f_1)T &\geq \frac{2}{N_{\min}} \\
 N_{\min} &\geq \frac{2}{(f_2 - f_1)T}
 \end{aligned}$$

Experimentally, at least 12 samples are required to distinguish the two sinusoids (although 11 could be argued). The equation above gives  $N_{\min} = 32$  or 33 samples, but it was a conservative estimate intended to separate the the components completely.

(b)



- i. Since the DFT is a set of samples of the DTFT,  $\omega = \frac{2\pi m}{N}$  where  $m$  is the sample number of the DFT output (*Caution:* must subtract one from the Matlab array indices to get  $m$  since Matlab starts at 1 instead of 0!). The analog frequency is then  $\frac{\omega}{2\pi T} = \frac{m}{NT}$ . The analog amplitude of each component is (see the course notes p. 48.3)  $A = \frac{2}{N} \cdot (\text{height of } i\text{th peak})$ .

The frequency estimates are

$$\frac{6-1}{64/30} \approx 2.34 \text{ Hz (actual } \approx 2.1429 \text{ Hz) and}$$

$$\frac{10-1}{64/30} \approx 4.22 \text{ Hz (actual = 4.00 Hz).$$

The amplitude estimates are

$$\frac{2}{64} \cdot 16.2539 \approx 0.51 \text{ (actual = 0.75) and}$$

$$\frac{2}{64} \cdot 24.4390 \approx 0.76 \text{ (actual = 1.0).$$

- ii. The frequency estimates are

$$\frac{5-1}{64/30} = 1.875 \text{ Hz (exact) and}$$

$$\frac{9-1}{64/30} = 3.75 \text{ Hz (exact).$$

The amplitude estimates are

$$\frac{2}{64} \cdot 24.0 = 0.75 \text{ (exact) and}$$

$$\frac{2}{64} \cdot 32.0 = 1.00 \text{ (exact).$$

- iii. The estimates in (i) are in error because none of the samples in the DFT line up perfectly with the peaks of the sinc functions in the DTFT. Thus, the true peaks are higher and located between the DFT samples.

The estimates in (ii) are exact because the sinusoid frequencies were such that they fall squarely on the DFT sampling grid, i.e. the peaks of the sinc spectra in the DTFT line up perfectly with the DFT samples.

- iv. From the reasoning above, the requirement is that  $f_1 NT$  and  $f_2 NT$  are integers (obtain this by solving the equation  $f = \frac{m}{NT}$ , derived above in (i), for  $m$  and requiring that  $m$  is an integer).