ECE 113 HW 6

Lawrence Liu

November 28, 2022

Problem 1

(a)

We have $10000 \cdot 5$ samples for our time domain signal, each of these will be 4 bytes, so in total we would have that we would need 200kB to naively represent the signal. For our DFT signal we would have $10000 \cdot 5$ samples, each of which would take 8 bytes to represent, so in total we would need 400kB to represent the signal.

(b)

Since it is a real value signal we would have that

$$X_k = X^*_{N-k}$$

Thus we can only send half of the DFT signal, which would be $10000 \cdot 5/2 + 1$ samples, each of which would take 8 bytes to represent, so in total we would need 200kB to represent the signal.

(c)

Then we would have that we would only need to send the values of X_k between $k = 2000$ and $k = 4000$, which would be 2001 samples, each of which would take 8 bytes to represent, so in total we would need 16008 bytes to represent the signal.

Problem 2

(a)

From nyquist we would need to sample at least $2 \cdot 8000$ Hz, so then we would need $2{\cdot}8000{\cdot}5 = 80000$ bytes to represent the signal in discrete time. Likewise we would need 80001 bytes to represent the signal in the frequency domain since the signal is discrete.

(b)

We have that the sampled signal's frequency domain is

$$X_s = \sum_{k=-\infty}^{k=\infty} X(f - kf_s)$$

Where $X(f)$ is the frequency domain of the original signal, then since we have that $X(f) > 0$ only for $f \in [4000, 8000]$, then we get that if we just apply a bandpass filter, we can sample at 4000 Hz and still reconstruct the signal. so therefore we would only need $4000 \cdot 8 = 32000$ bytes to represent the signal in discrete time.

Problem 3

(a)

We need to multiply every term by every term in the polynomial and summing so it will take $O(n^2)$ time.

(b)

We have that the coefficeint of x^k, c_k is

$$c_k = \sum_{i=0}^{i=k} a_i b_{k-i}$$

Where a_i and b_i are the coefficeints of the polynomials a and b for x^i. So we can see that c_k is the circular convolution of a and b.

(c)

Therefore we can do it with FFT by first taking the FFT of the signals made up of the coefficents of the two polynomials, then multiplying the two FFTs, and then taking the inverse FFT of the result. This will take $O(n \log n)$ time.

Problem 4

(a)

We have that

(b)

$$[0,0,1,0] \rightarrow [1,j,-1,-j]$$

Applying property that convolution in frequency domain is multiplication in time domain, we get that

$$[0,1,0,0] \circ x[n] \rightarrow [0,1-j,-1,1-j]$$

(c)

$$[0,0,1,0] \rightarrow [1,-j,-1,j]$$

(d)

Applying the time shift property we get

$$x[n-1] \rightarrow e^{-j\frac{\pi}{2}} DFT(x[n]) = [0,-j-1,-1,j-1]$$

Problem 5