$\boxed{\text{ECE 513 HMW4}}$

Problem 1: Solve:

(a) $X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}}, \; k=0,1,\cdots,N-1$

when k is even: $X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}} \quad k=0,2,\cdots,N-2 \; \cdots (1)$

(1) is equal to $\left[x(0)e^{-j\frac{2\pi k\cdot 0}{N}} + x(\frac{N}{2})e^{-j\frac{2\pi k\cdot\frac{N}{2}}{N}}\right] + \left[x(1)e^{-j\frac{2\pi k\cdot 1}{N}} + x(\frac{N}{2}+1)e^{-j\frac{2\pi k(1+\frac{N}{2})}{N}}\right] + \cdots$

$\cdots + \left[x(\frac{N}{2}-1)e^{-j\frac{2\pi k(\frac{N}{2}-1)}{N}} + x(N-1)e^{-j\frac{2\pi k(N-1)}{N}}\right]$

Due to $x(0) = -x(\frac{N}{2}),\; x(1) = -x(\frac{N}{2}+1), \cdots, x(\frac{N}{2}-1) = -x(N-1)$

let integer m: $0 \le m \le \frac{N}{2}-1$

(1) is equal to $X(k) = \sum_{m=0}^{\frac{N}{2}-1} \left(x(m)e^{-j\frac{2\pi km}{N}} - x(m)e^{-j\frac{2\pi k(m+\frac{N}{2})}{N}}\right)$

$= \sum_{m=0}^{\frac{N}{2}-1} \left[x(m)\left(e^{-j\frac{2\pi km}{N}} - e^{-j\frac{2\pi k(\frac{N}{2}+m)}{N}}\right)\right] \cdots\cdots (2)$ k is even

Because $e^{-j\frac{2\pi km}{N}} = \cos\frac{2\pi km}{N} - j\sin\frac{2\pi km}{N}$, $e^{-j\frac{2\pi k(\frac{N}{2}+m)}{N}} = \cos(\frac{2\pi km}{N}+\pi k) - j\sin(\frac{2\pi km}{N}+\pi k)$

due to k is even, $e^{-j\frac{2\pi km}{N}} - e^{-j\frac{2\pi k(\frac{N}{2}+m)}{N}} = 0$.

So (2) is equal to 0 $\rightarrow X(k) = 0$ when k is even

(b) $Y(m) = \sum_{n=0}^{\frac{N}{2}-1} y(n) e^{-j\frac{2\pi mn}{\frac{N}{2}}}, \; m=0,1,\cdots,\frac{N}{2}-1 \Rightarrow \sum_{n=0}^{\frac{N}{2}-1} y(n) e^{-j\frac{4\pi mn}{N}}, \; m=0,1,\cdots,\frac{N}{2}-1$

k is odd: $X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \quad k=1,3,\cdots,N-1$

due to being identical and $k=2m+1$

So $\sum_{m=0}^{\frac{N}{2}-1} y(m) e^{-j\frac{4\pi mn}{N}} = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \quad k=1,3,\cdots,N-1. \cdots (3)$

The right side of (3) is equal to $\sum_{m=0}^{\frac{N}{2}-1} \left[x(m)\left(e^{-j\frac{2\pi km}{N}} - e^{-j\frac{2\pi k(\frac{N}{2}+m)}{N}}\right)\right]$

due to k is odd, $e^{-j\frac{2\pi km}{N}} - e^{-j\frac{2\pi k(\frac{N}{2}+m)}{N}} = 2e^{-j\frac{2\pi km}{N}}$

so $\sum_{m=0}^{\frac{N}{2}-1} y(n) e^{-j\frac{4\pi mn}{N}} = \sum_{m=0}^{\frac{N}{2}-1} 2x(m) e^{-j\frac{2\pi mk}{N}} \leftarrow k = 2m+1.$

Therefore $y(n) e^{-j\frac{4\pi mn}{N}} = 2X(m) e^{-j\frac{4\pi m^2 + 2\pi m}{N}}$

$n=m$, $0 \le n \le \frac{N}{2}-1$, $0 \le m \le \frac{N}{2}-1$ $\therefore$ so $y(n) = 2X(m) e^{-j\frac{2\pi m}{N}} = 2X(m)\left[\cos(\frac{2\pi m}{N}) - j\sin(\frac{2\pi m}{N})\right]$

Hence: $y(n) = 2X(n)\left[\cos(\frac{2\pi n}{N}) - j\sin(\frac{2\pi n}{N})\right] \; 0 \le n \le \frac{N}{2}-1$

Problem 2:

Solve:

(a) $x(n) = \{\underset{\uparrow}{3}, 0, -1, 2\}$

$y(n) = \{\underset{\uparrow}{1}, 5, 4, -2\}$

$z(m) = \sum_{n=0}^{3} x(n) y((m-n))_4 \quad m = 0, 1, 2, 3$

when $m=0$, $z(0) = \sum_{n=0}^{3} x(n) y((-n))_4$

$x(n) = \{\underset{\uparrow}{3}, 0, -1, 2\}$

$y(-n)_4 = \{y(0), y(3), y(2), y(1)\} = \{\underset{\uparrow}{1}, -2, 4, 5\}$

So $z(0) = 3 - 4 + 10 = 9$

when $m=1$, $z(1) = \sum_{n=0}^{3} x(n) y((1-n))_4$

$x(n) = \{\underset{\uparrow}{3}, 0, -1, 2\}$

$y((1-n))_4 = \{y(1), y(0), y(3), y(2)\} = \{5, \underset{\uparrow}{1}, -2, 4\}$

So $z(1) = 15 + 2 + 8 = 25$.

when $m=2$, $z(2) = \sum_{n=0}^{3} x(n) y((2-n))_4$

$x(n) = \{\underset{\uparrow}{3}, 0, -1, 2\}$

$y((2-n))_4 = \{y(2), y(1), y(0), y(3)\} = \{\underset{\uparrow}{4}, 5, 1, -2\}$

So $z(2) = 12 - 1 - 4 = 7$

when $m=3$, $z(3) = \sum_{n=0}^{3} x(n) y((3-n))_4$

$x(n) = \{\underset{\uparrow}{3}, 0, -1, 2\}$

$y((3-n))_4 = \{y(3), y(2), y(1), y(0)\} = \{\underset{\uparrow}{-2}, 4, 5, 1\}$

So $z(3) = \boxed{-6 - 5 + 2} = -9$

Therefore $\sum_{n=0}^{3} x(n) y((m-n))_4 = \{9, 25, 7, -9\}$

(b)

The Matlab code is shown as follows:

```
x=[3,0,-1,2];
y=[1,5,4,-2];
Z1=fft(x,4).*fft(y,4); % multiply the spectrum of x by y with
                         the use of fft
z1=ifft(Z1,4)          % obtain the result of circular
                         convolution with the use of ifft
```

The result is shown as follows:

```
>> hmw4

z1 =

     9    25     7    -9
```

Therefore, the result in (b) is the same as the result in (a).

(c)

The Matlab code is shown as follows:

```
z2=conv(x,y)
```

The result is shown as follows:

```
z2 =

     3    15    11    -9     6    10    -4
```

(d)

The Matlab code is shown as follows:

```
Z3=fft(x,7).*fft(y,7);
z3=ifft(Z3,7)
```

The result is shown as follows:

```
z3 =

    3.0000   15.0000   11.0000   -9.0000    6.0000   10.0000   -4.0000
```

Therefore, the result in (d) is the same as the result in (c).

3. Solve:

(a)

The Matlab code is shown as follows:

```
y=sampdata;
figure(1)
stem(0:length(y)-1,y)
title('x(n) The input sequence')
xlabel('n')
ylabel('Amplitude')
```

The figure of the input sequence x(n) is shown in Fig. 1:

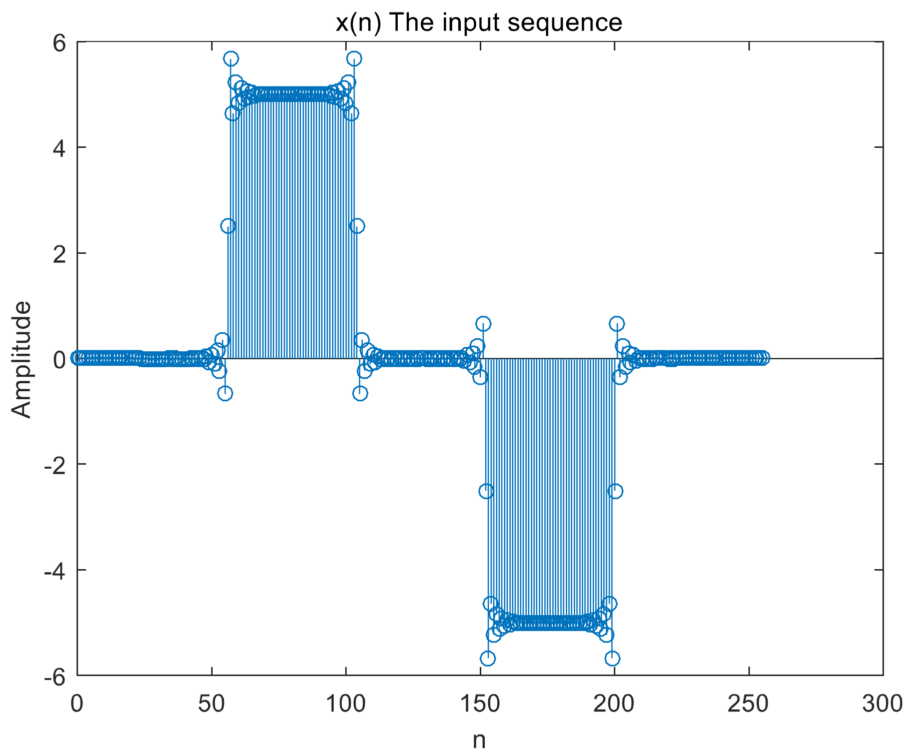

Fig. 1. The input sequence x(n)

(b)

Use the following Matlab code to design a 32 coefficient low pass FIR filter.

```
order=32;
ws=0.749;
wc=0.85*ws;
F=[0.0 wc ws 1.0];
A=[1.0 0.95 0.01 0.0];
b=firpm(order, F, A);
```

(c)

Use the Matlab routine **freqz** to compute the magnitude and frequency response of the resulting filter. The Matlab code is shown as follows:

```
w=pi*(0:0.005:1.0);
h=freqz(b,A,w);
hmag=abs(h);
hphase=angle(h);
figure(2)   % Plot the magnitude response of the resulting
filter
plot(w,hmag)
title('Frequency Response Plot')
xlabel('Normalized Frequency (radians)')
ylabel('Magnitude')
figure(3)   % Plot the phase response of the resulting filter
plot(w,hphase)
title('Phase Response Plot')
xlabel('Normalized Frequency (radians)')
ylabel('Magnitude')
```

The figures of the magnitude and phase response of the resulting filter are shown in Fig. 2 and Fig. 3 respectively.

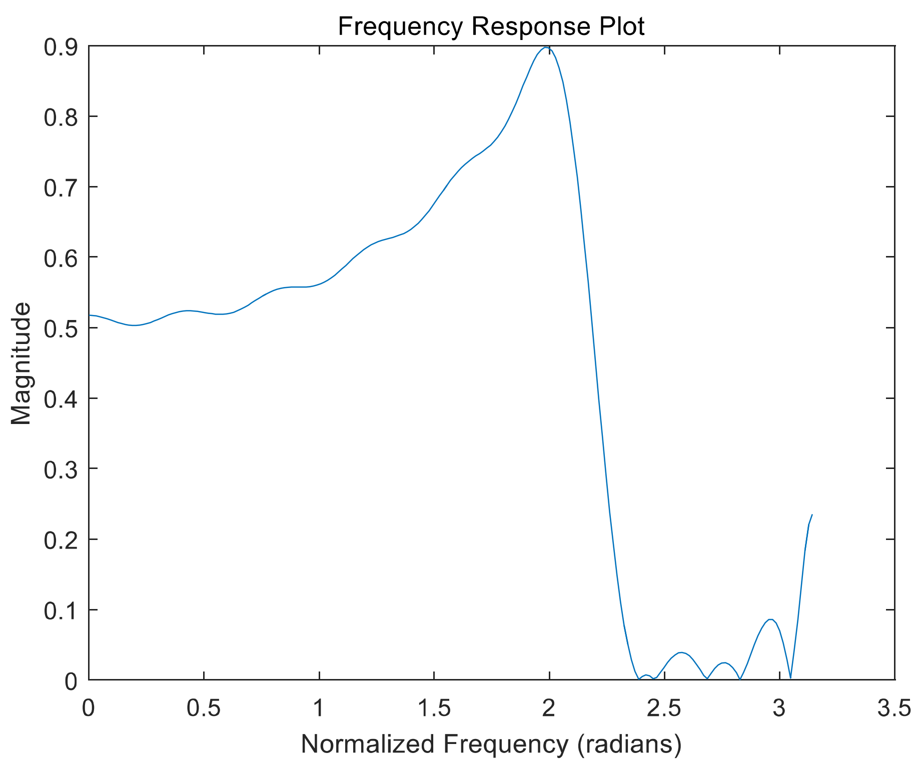

Fig. 2. The magnitude response of the resulting filter

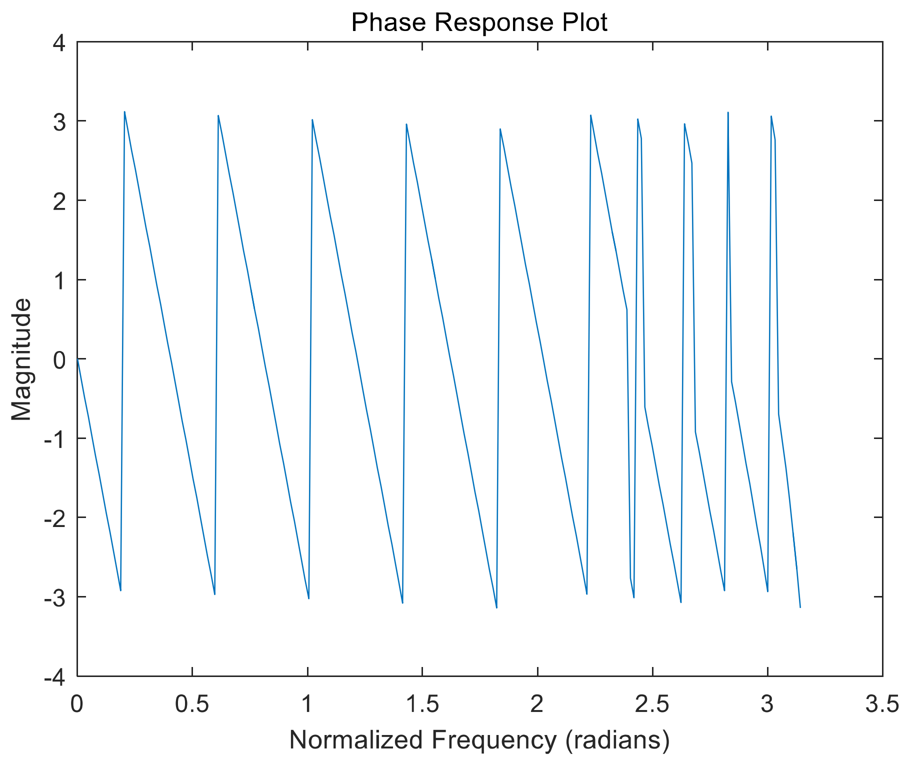

Fig. 3. The phase response of the resulting filter

(d)

Use the Matlab routine **conv** to compute the output of this filter, y1(n), if x(n) is applied as the input. The Matlab code is shown as follows:

```
[h,t]=impz(b,A); % Calculate impulse response of the filter
y1n=conv(y,h');   % use conv to get y1(n)
figure(4)         % Plot the output sequence y1(n)of the filter
length(y1n);
stem(0:length(y1n)-1,y1n)
title('y1(n) The output sequence')
xlabel('n')
ylabel('Amplitude')
```

The figure of the output sequence y1(n) is shown in Fig. 4:

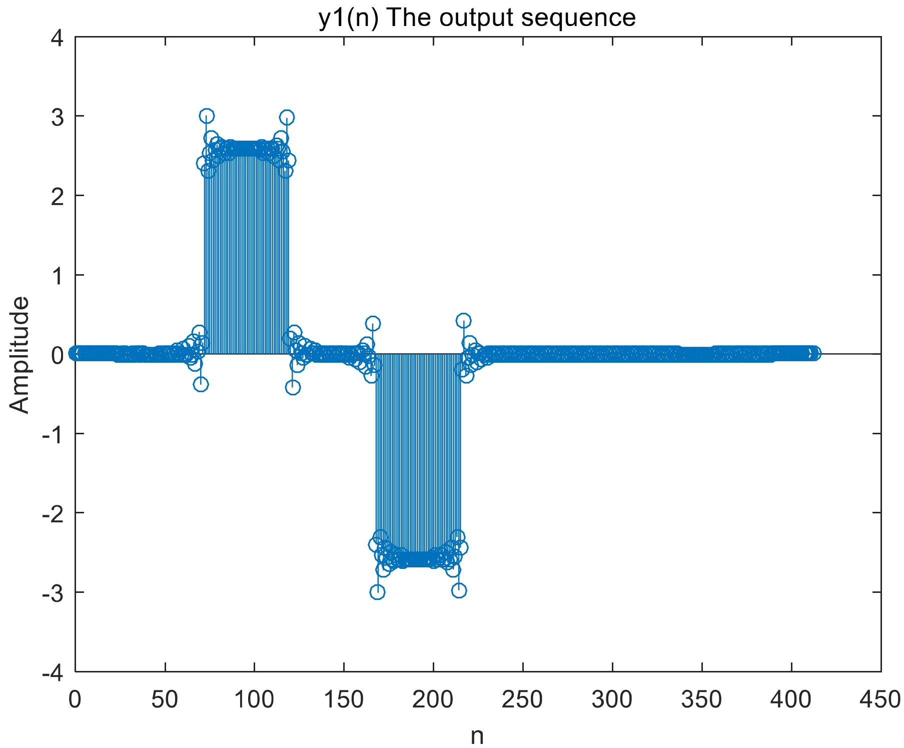

Fig. 4. The output sequence y1(n) of the resulting filter

(e)

Use the Matlab routines **fft** and **ifft** to simulate a DFT/IDFT direct computation of the output of this filter if the sequence x(n) is applied as the input. Name this output y2(n). The Matlab code is shown as follows:

```
Y2=fft(y,length(y1n)).*fft(h',length(y1n));
y2n=ifft(Y2,length(y1n));  % use fft and ifft to get y1(n)
figure(5)                  % Plot the output sequence y2(n)
                             of the filter
length(y2n);
stem(0:length(y2n)-1,y2n)
title('y2(n) The output sequence')
xlabel('n')
ylabel('Amplitude')
```

The figure of the output sequence y2(n) is shown in Fig. 5:

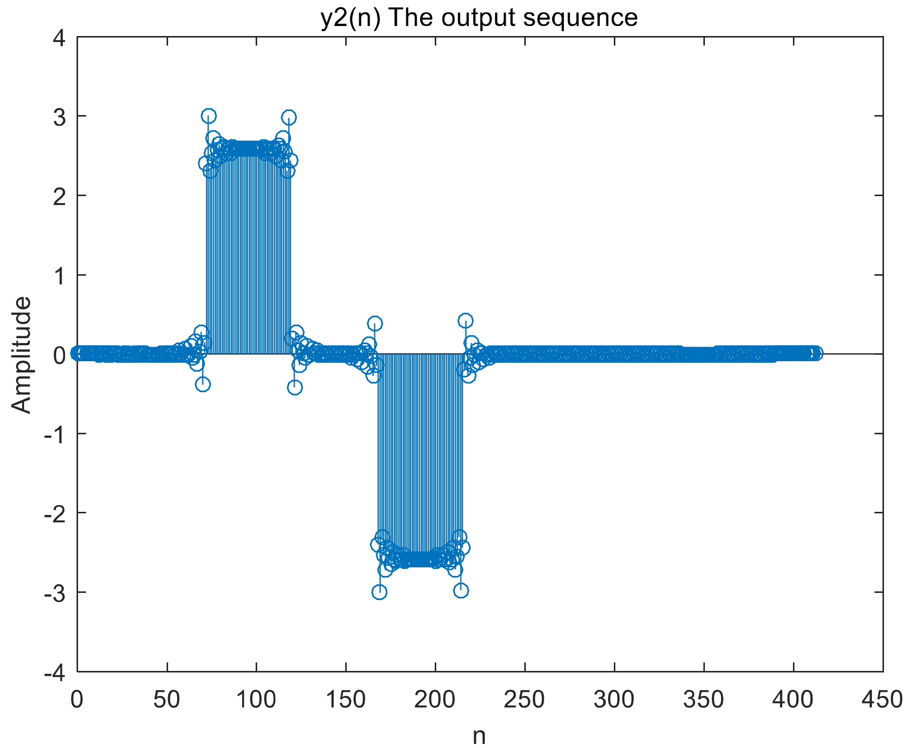

Fig. 5. The output sequence y2(n) of the resulting filter

(f)

Compute the absolute error between the computed signals y1(n) and y2(n). The Matlab code is shown as follows:

```
figure(6)              % Plot the absolute error between y1(n) and
                         y2(n)
error=abs(y1n-y2n);
stem(0:length(error)-1,error)
title('The absolute error between y1(n) and y2(n)')
xlabel('n')
ylabel('Absolute error')
```

The figure of the absolute error between y1(n) and y2(n) is shown in Fig. 6:

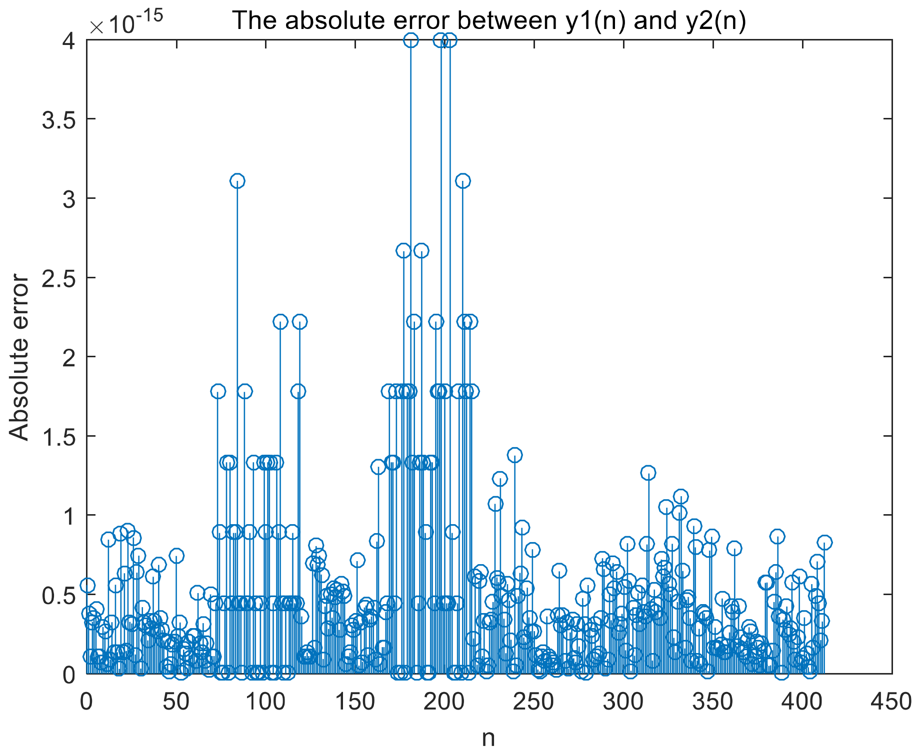

Fig. 6. Absolute Error between $y1(n)$ and $y2(n)$

As is clearly demonstrated, the absolute error is definitely small so that people can actually ignore it while conducting research. It's okay for us to use FFT algorithm to resolve engineering problems concerning signal processing without worrying about losing too much accuracy.

Problem 4:

Solve:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}} \quad k=0,1,\cdots,N-1$$

$$X_c(k) = \sum_{n=0}^{N-1} x(n)\cdot \cos\frac{4\pi k_0 n}{N} e^{-j\frac{2\pi nk}{N}}, \quad 0\le n\le N-1$$

due to $\cos\frac{4\pi k_0 n}{N} = \frac{e^{j\frac{4\pi k_0 n}{N}} + e^{-j\frac{4\pi k_0 n}{N}}}{2}$

$$\text{So } X_c(k) = \frac{1}{2}\sum_{n=0}^{N-1}\left[x(n) e^{-j\frac{2\pi nk}{N}} e^{j\frac{2\pi n\cdot 2k_0}{N}} + x(n) e^{-j\frac{2\pi nk}{N}} e^{-j\frac{2\pi n\cdot 2k_0}{N}}\right]$$

$$= \frac{1}{2}\sum_{n=0}^{N-1}\left[x(n) e^{-j\frac{2\pi n(k-2k_0)}{N}} + x(n) e^{-j\frac{2\pi n(k+2k_0)}{N}}\right]$$

$$= \frac{1}{2}\left[X((k-2k_0))_N + X((k+2k_0))_N\right]$$

$$X_s(k) = \sum_{n=0}^{N-1} x(n) \sin\frac{4\pi k_0 n}{N} e^{-j\frac{2\pi nk}{N}}, \quad 0\le n\le N-1$$

due to $\sin\frac{4\pi k_0 n}{N} = \frac{e^{j\frac{4\pi k_0 n}{N}} - e^{-j\frac{4\pi k_0 n}{N}}}{2j}$

$$X_s(k) = \frac{1}{2j}\sum_{n=0}^{N-1}\left[x(n) e^{-j\frac{2\pi nk}{N}} e^{j\frac{2\pi n\cdot 2k_0}{N}} - x(n) e^{-j\frac{2\pi nk}{N}} e^{-j\frac{2\pi n\cdot 2k_0}{N}}\right]$$

$$= \frac{1}{2j}\sum_{n=0}^{N-1}\left[x(n) e^{-j\frac{2\pi n(k-2k_0)}{N}} - x(n) e^{-j\frac{2\pi n(k+2k_0)}{N}}\right]$$

$$= \frac{1}{2j}\left[X((k-2k_0))_N - X((k+2k_0))_N\right]$$

Problem 5:

Solve:

(a) The total number of samples in the sequence: $L_1 = 31912$

The minimum number of samples to use for the FFT in order to obtain a linear convolution result:

$$N_1 = 31912 + 130 - 1 = 32041$$

The next number that is a power of 2 is: $N_1 = 2^{15} = 32768$

The total number of complex multiplications is $k_1 = (2+1) \times 32768 \times \log_2 32768 + 32768 = 1507368$

(b) If we use a block size of $N_2 = 512\ (2^9)$ samples for the overlap method, then each time we use $L_2 = 512 - 130 + 1 = 383$ new samples in each input block of data.

We would have to use $B_2 = \left\lceil \frac{32041}{383} \right\rceil = \lceil 83.66 \rceil = 84$ blocks

The total number of complex multiplications required is

$$k_2 = 84 \times [2 \times 512 \times \log_2 512 + 512] + 512 \times \log_2 512 = 821760$$

(c) If we use a block size of $N_3 = 1024\ (2^{10})$ samples for the overlap method, then each time we use $L_3 = 1024 - 130 + 1 = 895$ new samples in each input block of data.

We would have to use $B_3 = \left\lceil \frac{32041}{895} \right\rceil = \lceil 35.8 \rceil = 36$ blocks.

The total number of complex multiplications required is

$$k_3 = 36 \times [2 \times 1024 \times \log_2 1024 + 1024] + 1024 \times \log_2 1024 = 784384$$

(d) If we use a block size of $N_4 = 2048\ (2^{11})$ samples for the overlap method, then each time we use $L_4 = 2048 - 130 + 1 = 1919$ new samples in each input block of data.

We would have to use $B_4 = \left\lceil \frac{32041}{1919} \right\rceil = \lceil 16.7 \rceil = 17$ blocks

The total number of complex multiplications required is

$$k_4 = 17 \times [2 \times 2048 \times \log_2 2048 + 2048] + 2048 \times \log_2 2048 = 823296$$

(e) The number of each block getting bigger doesn't necessarily mean fewer complex multiplications. When considering the block size of new samples, we should think about factors such as memory, cost, trade-off between sizes and numbers of complex multiplications.