

# Project #4 - Integrals and Intervals

LI YICHENG\*

USCID:7827077047

email: l.y.c.liyicheng@gmail.com

USC Viterbi of Engineering

2(a)

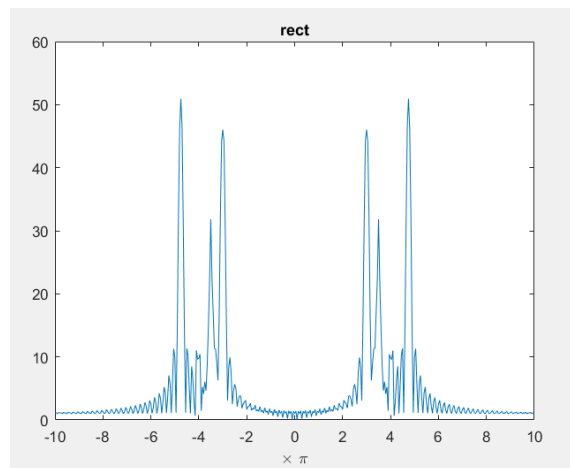


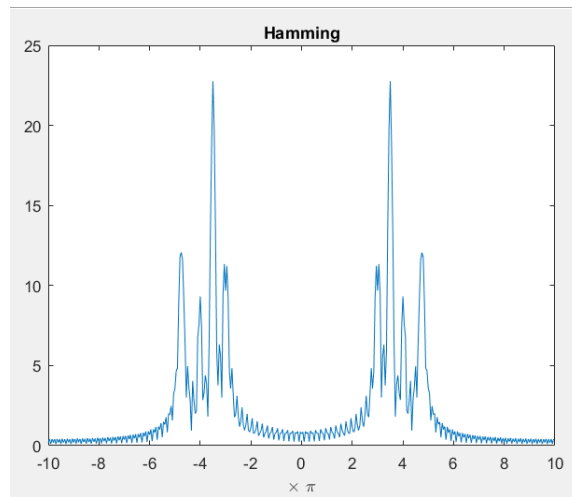
Figure 1: DFT spectral estimation

The spectral peaks at 3, 3.5, 4.75 can be seen however we may not tell if at  $4\pi$ , it is a peak or a sidelobe.

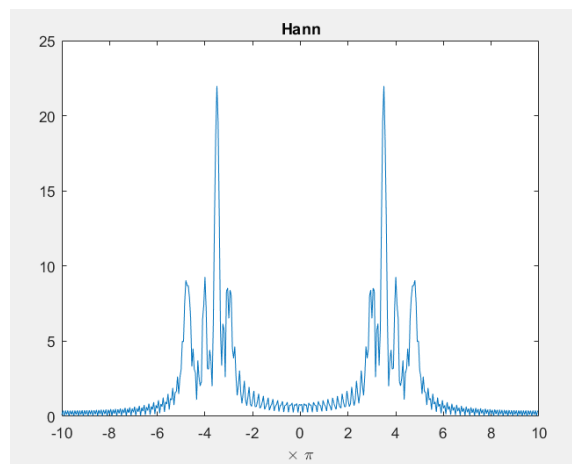
2(b)

---

\*github link: <https://github.com/IAMLYCHEE/EE483>



**Figure 2:** *Hamming*



**Figure 3:** *Hann*

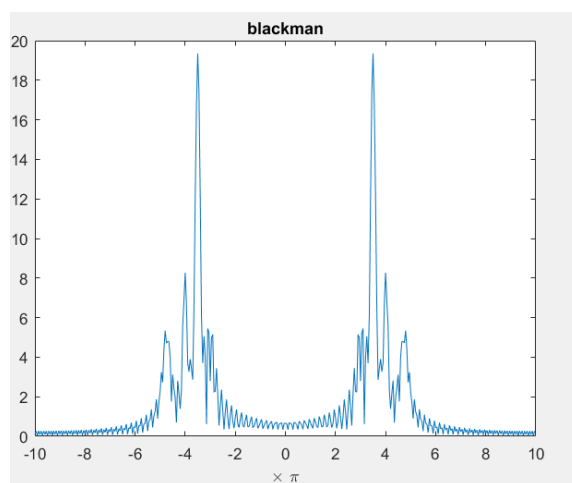


Figure 4: Blackman

Hamming window performs best to distinguish 4 peaks however, after applying these windows, the amplitude characteristic of the original signal may be eliminated.

**2(c) 100 Hz results:**(the x axis is scaled from frequency from  $-10\pi$  to  $10\pi$ ):

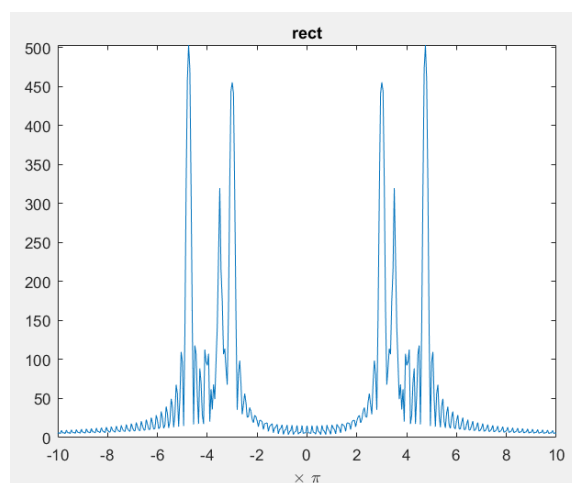
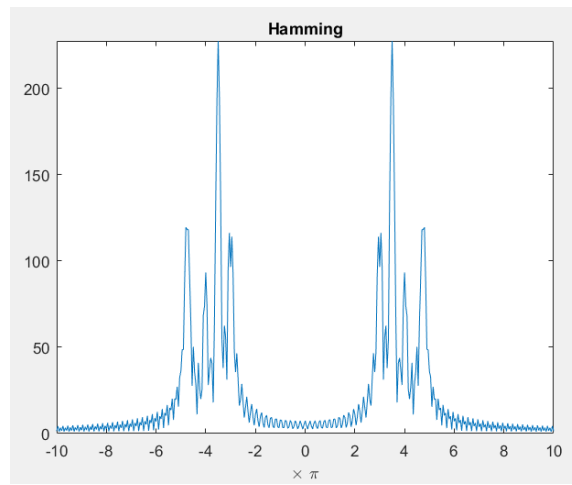
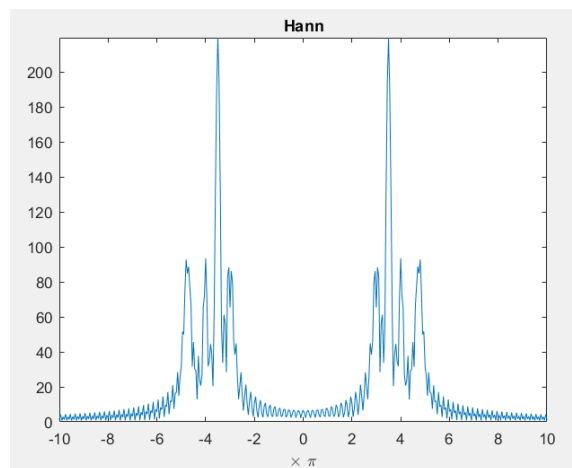


Figure 5



**Figure 6:** *Hamming*



**Figure 7:** *Hann*

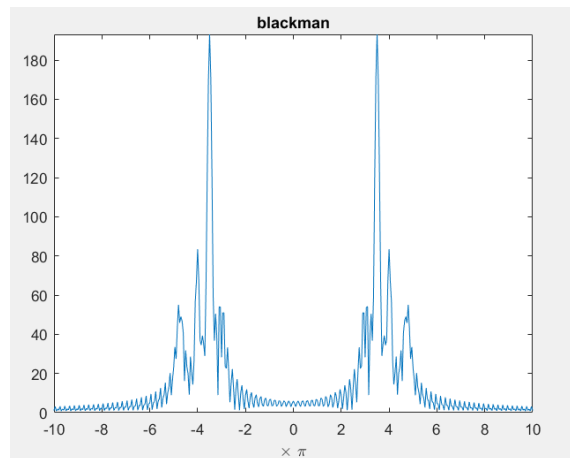


Figure 8: Blackman

If compared to the figures got with 10 Hz sampling, the 100 Hz sampling ones actually did not lead to improvements in the spectral estimation. Although it has effect on the amplitude however such effects also applied on sidelobe.

2(d)

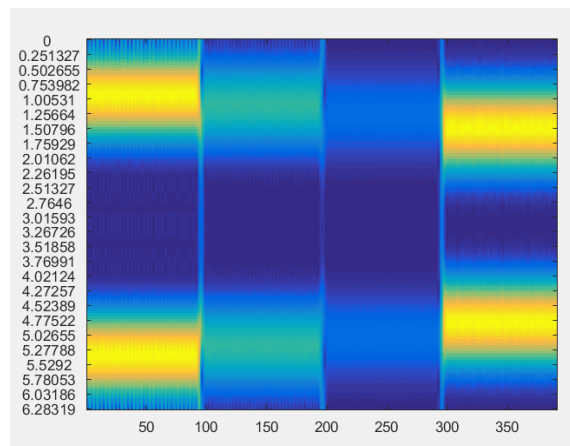


Figure 9: STFT

Yes, the images shows the feature of the giving signal, that its frequency is getting higher and in every 10 ten second the frequency remains the same.

appendix: code 2(a)(b)

```
1 clear
2 %a
```

```

3 fs = 100;
4 t = 0 : 1/fs : 40-1/fs;
5 N = length(t);
6 index = t * fs;
7 signal = cos(3 * pi * t) .* (t < 10) +...
8     1/2 * sin(3.5 * pi * t) .* (t >= 10 & t < 20)+...
9     1/6 * cos(4 * pi * t) .* (t >= 20 & t < 30) +...
10    sin(4.75 * pi * t) .* (t >= 30 & t < 40);
11 X = fft(signal);
12 t = (index - N/2)/ N * fs * 2;
13 plot(t,abs(fftshift(X)))
14 xlabel('\times \pi')
15 axis([-10 10 0 max(abs(X))])
16 title('rect')
17 %b
18 %hamming
19 window = hamming(N);
20 x = signal .* window';
21 X = fft(x);
22 t = (index - N/2)/ N * fs * 2;
23 figure
24 plot(t,abs(fftshift(X)));
25 xlabel('\times \pi')
26 axis([-10 10 0 max(abs(X))])
27 title('Hamming')
28
29 %hann
30 window = hann(N);
31 x = signal .* window';
32 X = fft(x);
33 t = (index - N/2)/ N * fs * 2;
34 figure
35 plot(t,abs(fftshift(X)))
36 xlabel('\times \pi')
37 axis([-10 10 0 max(abs(X))])
38 title('Hann')
39
40 %blackman
41 window = blackman(N);
42 x = signal .* window';
43 X = fft(x);
44 t = (index - N/2)/ N * fs * 2;
45 figure
46 plot(t,abs(fftshift(X)))
47 xlabel('\times \pi')
48 axis([-10 10 0 max(abs(X))])
49 title('blackman')

```

2(c)

```

1 clear
2 %a
3 fs = 100;
4 t = 0 : 1/fs : 40-1/fs;
5 N = length(t);
6 index = t * fs;
7 signal = cos(3 * pi * t) .* (t < 10) +...
8     1/2 * sin(3.5 * pi * t) .* (t >= 10 & t < 20) +...
9     1/6 * cos(4 * pi * t) .* (t >= 20 & t < 30) +...
10    sin(4.75 * pi * t) .* (t >= 30 & t < 40);
11 X = fft(signal);
12 t = (index - N/2)/ N * fs * 2;
13 plot(t, log(abs(fftshift(X))))
14 xlabel(' \times \pi ')
15 title('rect')
16 %b
17 %hamming
18 window = hamming(N);
19 x = signal .* window';
20 X = fft(x);
21 t = (index - N/2)/ N * fs * 2;
22 figure
23 plot(t, log(abs(fftshift(X))))
24 xlabel(' \times \pi ')
25 title('Hamming')
26
27 %hann
28 window = hann(N);
29 x = signal .* window';
30 X = fft(x);
31 t = (index - N/2)/ N * fs * 2;
32 figure
33 plot(t, log(abs(fftshift(X))))
34 xlabel(' \times \pi ')
35 title('Hann')
36
37 %blackman
38 window = blackman(N);
39 x = signal .* window';
40 X = fft(x);
41 t = (index - N/2)/ N * fs * 2;
42 figure
43 plot(t, log(abs(fftshift(X))))
44 xlabel(' \times \pi ')
45 title('blackman')

```

**2(d)**

```

1 clear

```

```

2 fs = 10;
3 t = 0 : 1/fs : 40-1/fs;
4 N = length(t);
5 index = t * fs;
6 signal = cos(3 * pi * t) .* (t < 10) +...
7     1/2 * sin(3.5 * pi * t) .* (t >= 10 & t < 20)+...
8     1/6 * cos(4 * pi * t) .* (t >= 20 & t < 30) +...
9     sin(4.75 * pi * t) .* (t >= 30 & t < 40);
10 tWindow = 1;
11 windowLength = fs * tWindow;
12 window = hamming(windowLength)';
13 yAmount = length(0:0.05:2*pi);
14 result = zeros(yAmount, N-windowLength + 1);
15 for m = 0 : N - windowLength
16     k = 1;
17     for omega = 0 : 0.05 : 2 * pi
18         index = m + 1 : m + windowLength;
19         result(k,m+1)=sum(signal(index) .* window .* exp(-1i*omega*index));
20         k = k + 1; %put less frequency down the axis
21     end
22 end
23 imagesc(abs(result));
24 ytickLabels = linspace(0,2*pi,length(1:5:size(result,1)));
25 set(gca, 'YTick',1:5:126 , 'YTickLabel', ytickLabels)

```