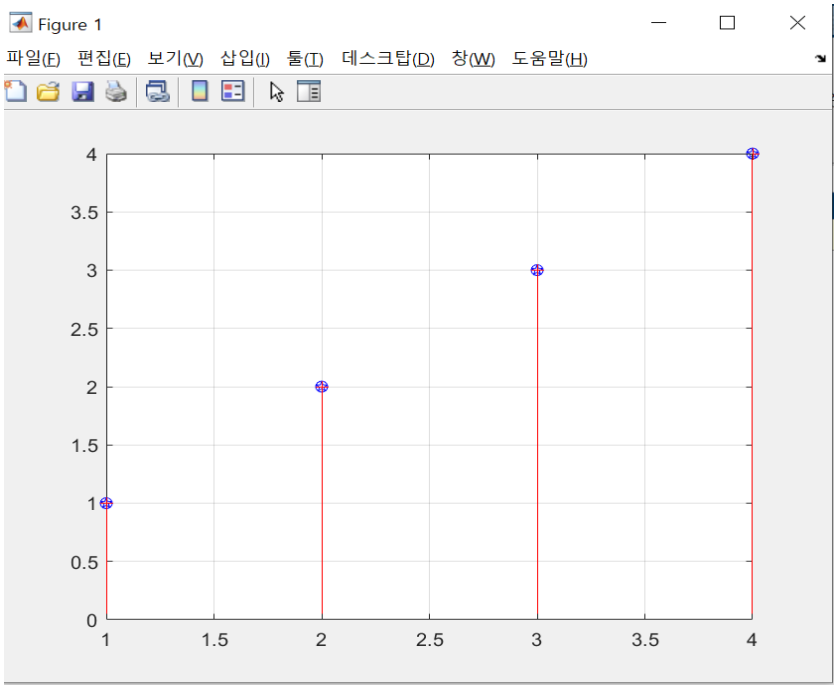
Assignment #10

융합공학부 20153865 김민석

[Practice]



알게 된 것)

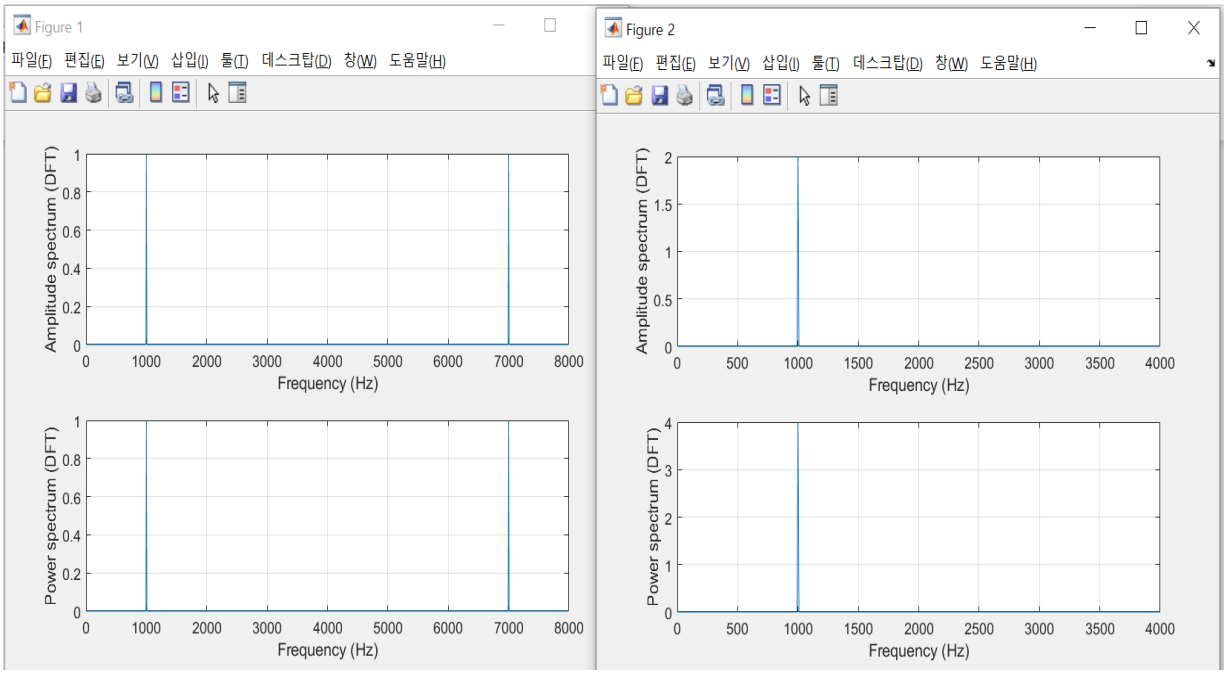
Stem function에 대해 알게 되었다. 이산 시퀀스 데이터를 플로팅해주는 함수이며, x축을 지정할 수도 있고, 안하고 사용할 수도 있다. 그리고, fft 와 ifft가 역관계여서, 고속푸리에 변환을 한 후에, 역고속푸리에 변환을 하게 되면, 그대로이게 된다.

Code)

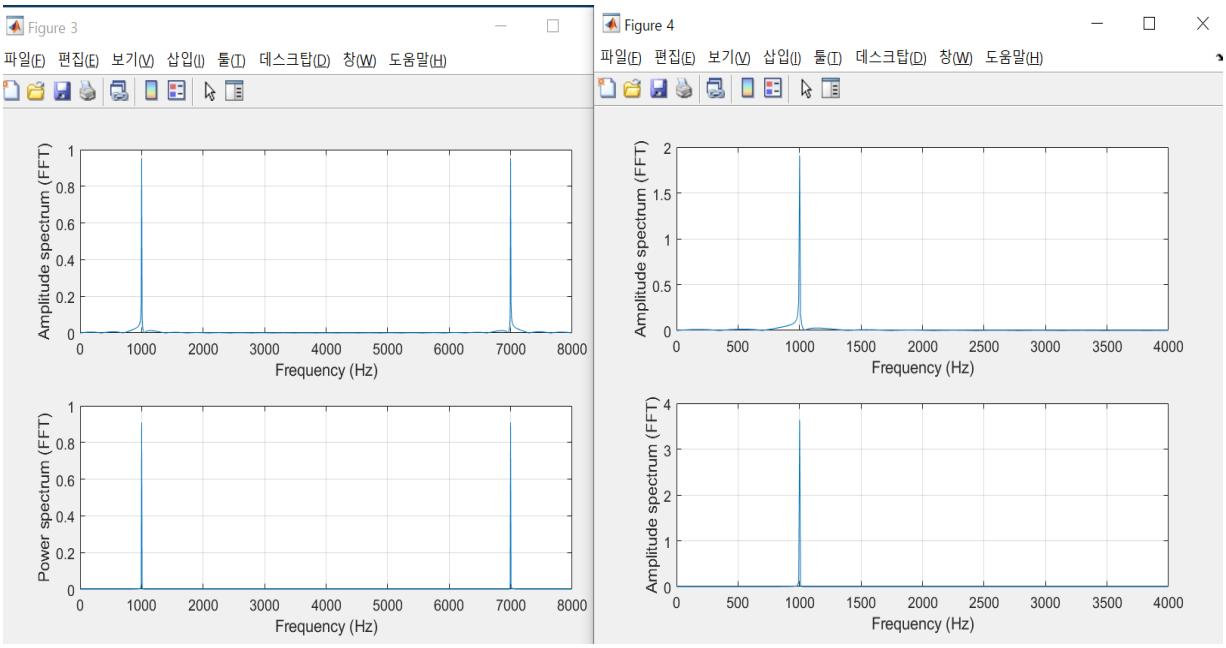
```
>> x = [1 2 3 4];

>> N = length(x);

>> X1 = fft(x,N);

>> x1 = ifft(X1,N);

>> stem(real(x1),'bo'); hold on; grid on

>> stem(x, 'r+'); hold off;
```

[EX 4.8]

a. DFT



b. FFT



알게된 것)

실제 Matlab 코드로 DFT와 FFT의 결과 그래프는 같게 나타나며, FFT에서 zero padding 추가 방법을 알게 되었습니다. 연산이 빠른 FFT를 대규모 프로젝트를 할 때 이용해야 겠다고 더욱 느끼게 되었습니다.

Code)

```
>> %Generate the sine wave sequence
>> fs=8000; %sampling rate
>> N=1000; %Number of data points
>> x=2*sin(2000*pi*[0:1:N-1]/fs);
>>
>> % Apply the DFT algorithm
>> figure(1)
>> xf = abs(fft(x))/N; % Compute the amplitude spectrum
>> P=xf.*xf; %Compute power spectrum
>> f=[0:1:N-1]*fs/N; % Map the frequency bin to frequency (Hz)
>> subplot(2,1,1); plot(f,xf); grid
>> xlabel('Frequency (Hz)'); ylabel('Amplitude spectrum (DFT)');
>> subplot(2,1,2);plot(f,P); grid
>> xlabel('Frequency (Hz)'); ylabel('Power spectrum (DFT)');

>> figure(2)
>> % Convert it to one side spectrum
>> xf(2:N)=2*xf(2:N); % Get the single-side spectrum
>> P=xf.*xf; %Calculate the power spectrum
>> f = [0:1:N/2]*fs/N %Frequencies up to the folding frequency
>> subplot(2,1,1); plot(f,xf(1:N/2+1)); grid
>> xlabel('Frequency (Hz)'); ylabel('Amplitude spectrum (DFT)');
>> subplot(2,1,2); plot(f,P(1:N/2+1)); grid
```

```
>> xlabel('Frequency (Hz)'); ylabel('Power spectrum (DFT)');


>> figure(3)
>> % Zero padding to the length of 1024
>> x=[x,zeros(1,23)];
>> N=length(x);
>> xf=abs(fft(x))/N; % Compute amplitude spectrum with zero padding
>> P=xf.*xf;
>> f=[0:1:N-1]*fs/N; % Map frequncy bin to frequency (Hz)
>> subplot(2,1,1); plot(f,xf);grid
>> xlabel('Frequency (Hz)'); ylabel('Amplitude spectrum (FFT)');
>> subplot(2,1,2);plot(f,P);grid
>> xlabel('Frequency (Hz)'); ylabel('Power spectrum (FFT)');


>> figure(4)
>> % Convert it to one side spectrum
>> xf(2:N) = 2*xf(2:N);
>> P=xf.*xf;
>> f=[0:1:N/2]*fs/N;
>> subplot(2,1,1); plot(f,xf(1:N/2+1));grid
>> xlabel('Frequency (Hz)'); ylabel('Amplitude spectrum (FFT)');
>> subplot(2,1,2); plot(f,P(1:N/2+1)); grid
>> xlabel('Frequency (Hz)'); ylabel('Amplitude spectrum (FFT)');
```

[EX 4.11]

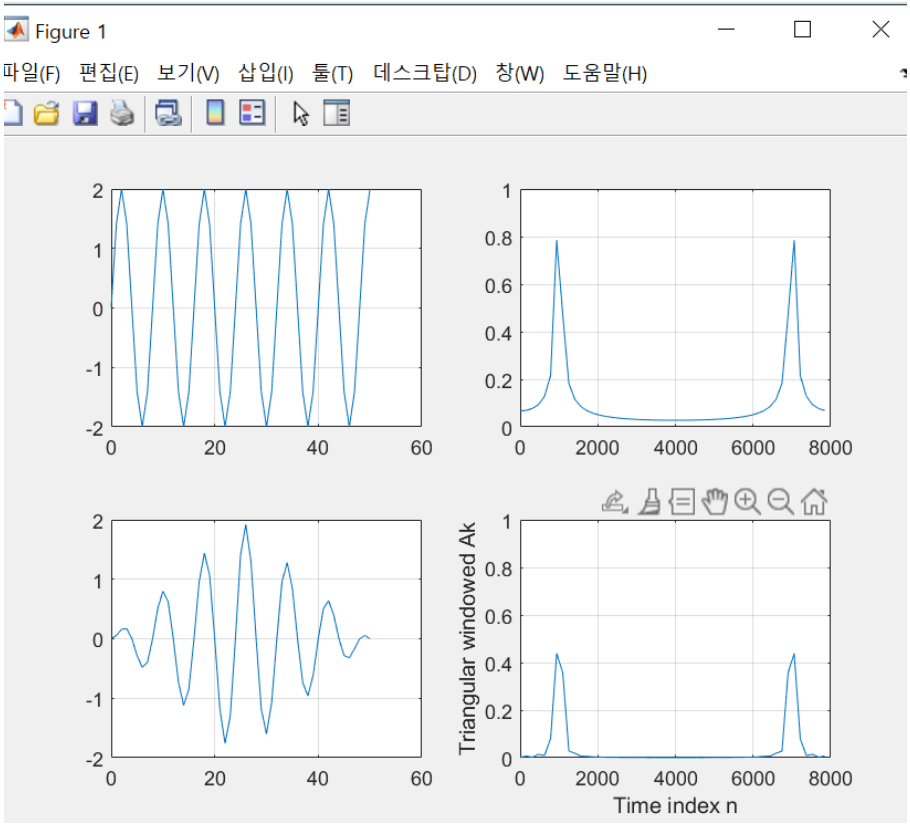


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Triangular windowed Ak
Time index n

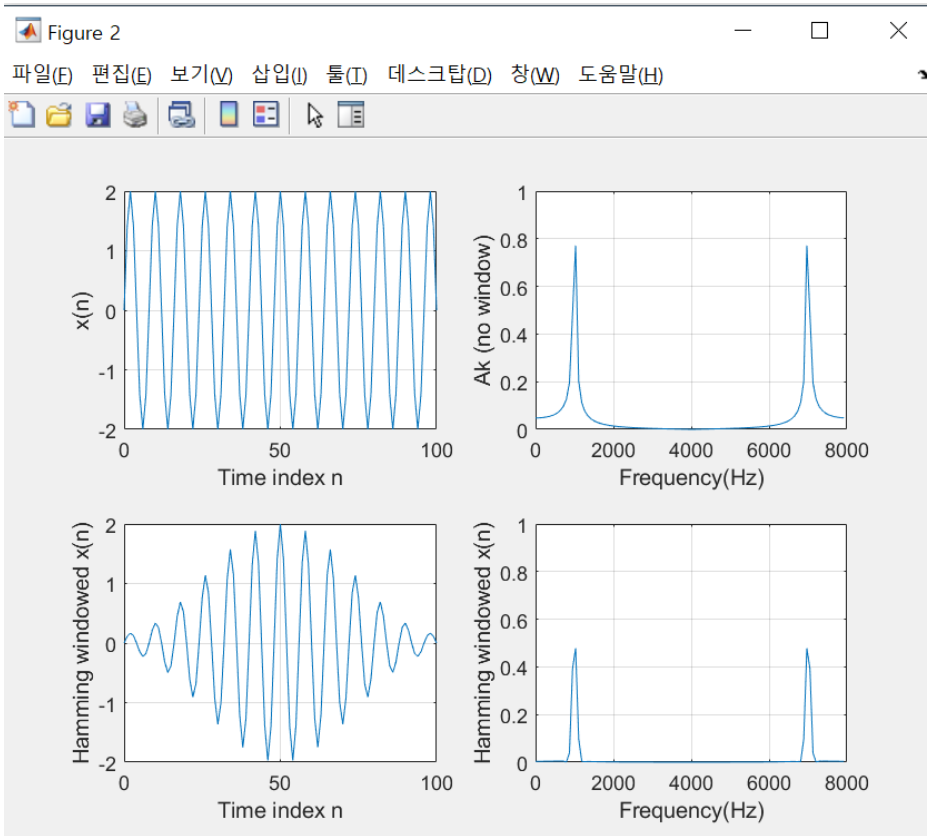
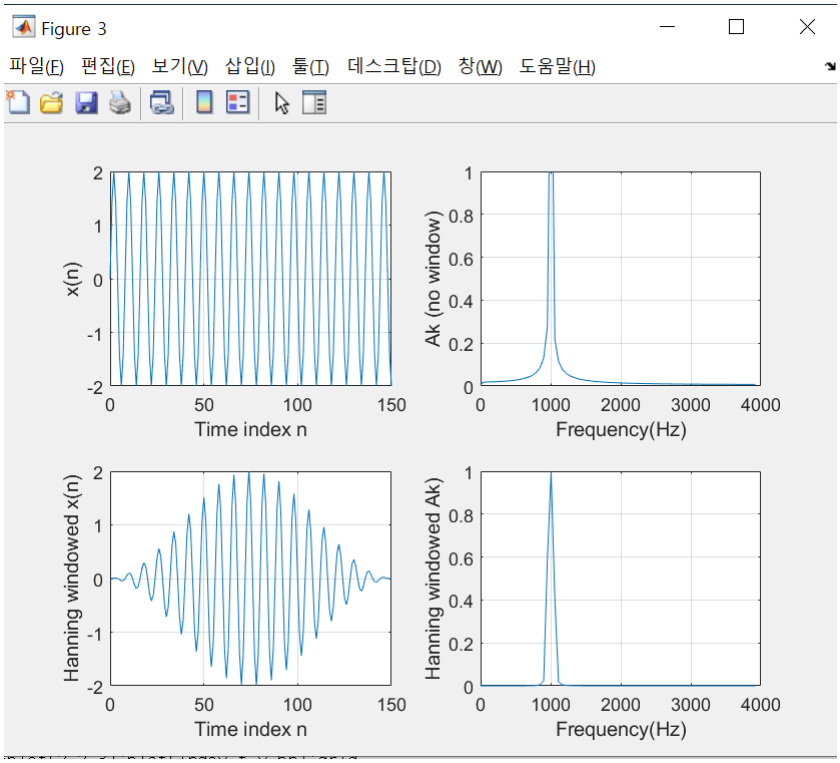


Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
x(n)
Time index n
Ak (no window)
Frequency(Hz)
Hamming windowed x(n)
Time index n
Hamming windowed x(n)
Frequency(Hz)



알게된 것)

Triangular, hamming, hanning window function 에 따라서 스펙트럼이 바뀌는 것을 보게 되었고, 각각의 window function 들을 적용할 때, matlab 코드에서 객체 방식으로 구현하게 되는 것을 알게 되었습니다.

Code)

```
>> fs=8000; T=1/fs;
>> x=2*sin(2000*pi*[0:1:50]*T);
>> N=length(x);
>> index_t=[0:1:N-1];
>> f=[0:1:N-1]*8000/N;
>> xf=abs(fft(x))/N;
>> figure(1)
```

```
>> x_b=x.*bartlett(N)';
>> xf_b=abs(fft(x_b))/N;
>> subplot(2,2,1); plot(index_t, x_b);grid
>> xlabel('Time index n'); ylabel('x(n)');
>> subplot(2,2,3); plot(index_t,x_b); grid
>> xlabel('Time index n'); ylabel('Triangular windowed x(n)');
>> subplot(2,2,2); plot(f,xf_b);grid; axis([0 8000 0 1]);
>> xlabel('Time index n'); ylabel('Ak (no window)');
>> subplot(2,2,4); plot(f,xf_b);grid; axis([0 8000 0 1]);
>> xlabel('Time index n'); ylabel('Triangular windowed Ak');
>> subplot(2,2,1); plot(index_t, x);grid
>> subplot(2,2,3); plot(index_t,x); grid
>> subplot(2,2,3); plot(index_t,x_b); grid
>> subplot(2,2,2); plot(f,xf);grid; axis([0 8000 0 1]);
>>
>> figure(2)
>> x=2*sin(2000*pi*[0:1:100]*T);
>> N=length(x);
>> index_t=[0:1:N-1];
>> f=[0:1:N-1]*fs/N;
>> xf=abs(fft(x))/N;
>> x_hm=x.*hamming(N)';
>> xf_hm=abs(fft(x_hm))/N;
>> subplot(2,2,1);plot(index_t,x);grid
```

```
>> xlabel('Time index n'); ylabel('x(n)');
>> subplot(2,2,3); plot(index_t, x_hm); grid
>> xlabel('Time index n'); ylabel('Hamming windowed x(n)');
>> subplot(2,2,2); plot(f,xf); grid; axis([0 fs 0 1]);
>> xlabel('Time index n'); ylabel('Ak (no window)');
>> subplot(2,2,4); plot(f,xf_hm); grid; axis([0 fs 0 1]);
>> xlabel('Frequency(Hz)'); ylabel('Hamming windowed x(n)');
>> subplot(2,2,3); plot(index_t, x_hm); grid
>> xlabel('Time index n'); ylabel('Hamming windowed x(n)');
>> subplot(2,2,2); plot(f,xf); grid; axis([0 fs 0 1]);
>> xlabel('Frequency(Hz)'); ylabel('Ak (no window)');
>>
>> figure(3)
>> x=2*sin(2000*pi*[0:1:150]*T);
>> N=length(x);
>> index_t=[0:1:N-1];
>> f=[0:1:N-1]*fs/N;
>> xf=2*abs(fft(x))/N;xf(1)=xf(1)/2;
>> x_hn=x.*hanning(N)';
>> xf_hn=2*abs(fft(x_hn))/N;xf_hn(1)=xf_hn(1)/2;
>> subplot(2,2,1);plot(index_t,x);grid
>> xlabel('Time index n'); ylabel('x(n)');
>> subplot(2,2,3);plot(index_t,x_hn);grid
>> xlabel('Time index n'); ylabel('Hanning windowed x(n)');
```

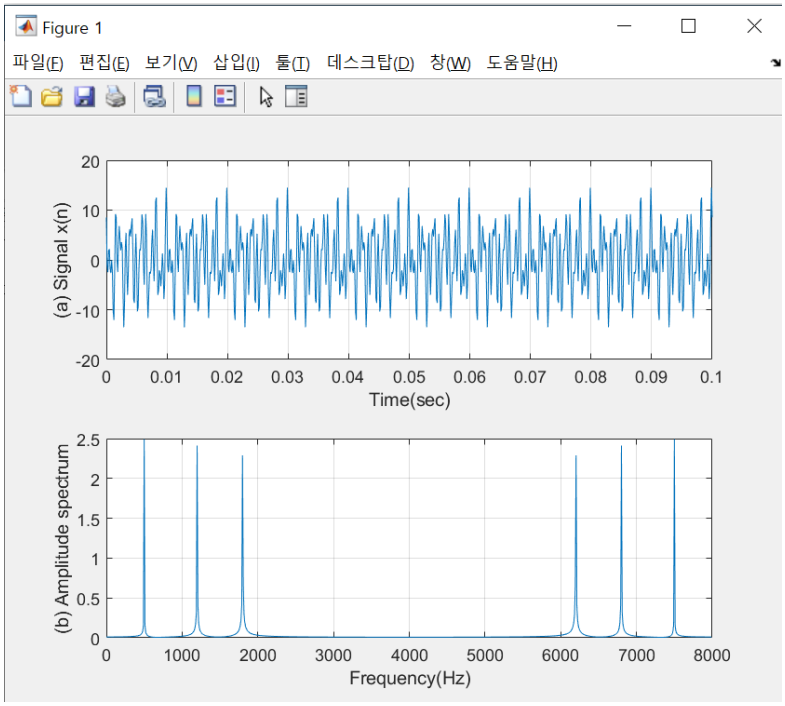
>> subplot(2,2,2);plot(f(1:(N-1)/2),xf(1:(N-1)/2));grid;axis([0 fs/2 0 1]);

>> xlabel('Frequency(Hz)'); ylabel('Ak (no window)');

>> subplot(2,2,4);plot(f(1:(N-1)/2),xf_hn(1:(N-1)/2));grid;axis([0 fs/2 0 1]);

>> xlabel('Frequency(Hz)'); ylabel('Hanning windowed Ak)');

[Prob 4.29]



알게된 것)

세개의 함수를 합치게 되면, 본래 함수의 주파수들이, fast-fourier-transform에 의해 나타나게 된다. 이것을 이용하면, 실생활에서 어떠한 wave가 무엇을 기반으로 구성되어 있는지, 분석할 수 있을 것 같습니다.

Code)

>> fs = 8000;

>> T = 1/fs;

>> t=0:T:0.1;

```
>> x1 = 5*cos(2*pi*500*t);

>> x2 = 5*cos(2*pi*1200*t+0.25*pi);

>> x3 = 5*cos(2*pi*1800*t+0.5*pi);

>> x=x1+x2+x3;

>> N=length(x);

>> index_t=[0:1:N-1];

>> f=[0:1:N-1]*fs/N;

>> Ak=abs(fft(x))/N;

>> subplot(2,1,1);plot(t,x);grid

>> xlabel('Time(sec)');ylabel('(a) Signal x(n)');

>> subplot(2,1,2);plot(f,Ak);grid

>> xlabel('Frequency(Hz)');ylabel('(b) Amplitude spectrum');
```