



BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH

Khoa: Công Nghệ Thông Tin



LAB REPORT 08

Student's ID :
Student's name : Hồ Phúc Lâm
Subject : PTHTDPT
Instructor : Nguyễn Thành Thái
Faculty : Công Nghệ Thông Tin
Completed Date : 23/10/2024

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

LAB 05 – Cài Đặt Matlab

CM3106: Multimedia Tutorial/Lab Class 1

1. Sóng sin và âm thanh:

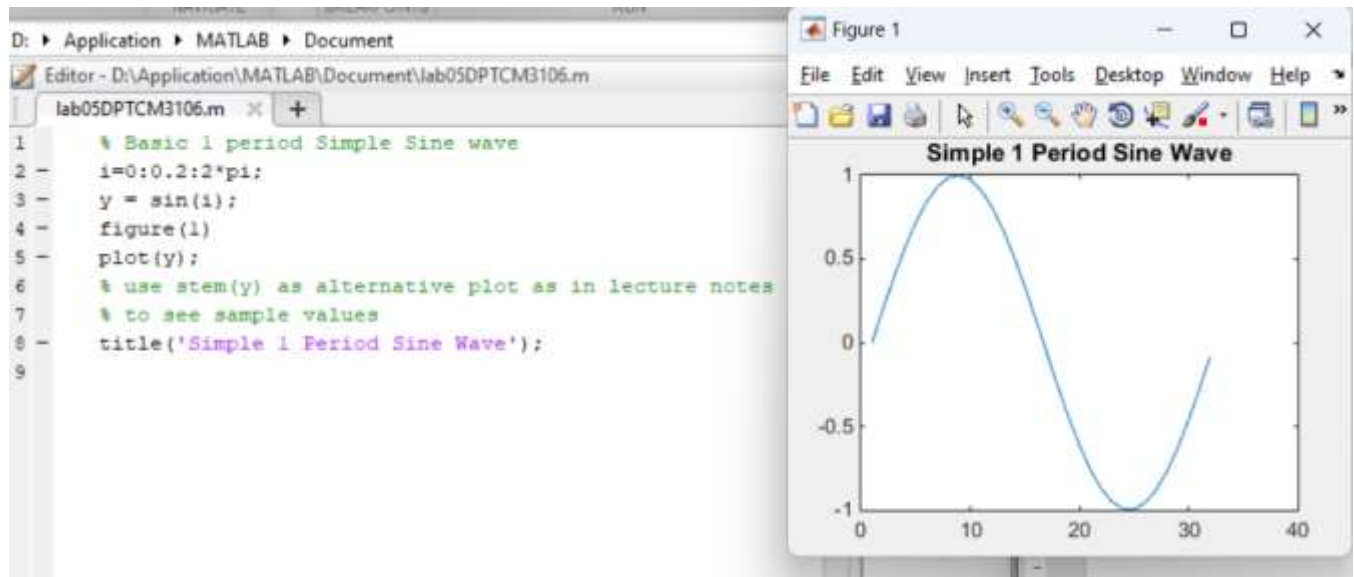
a. Phương trình tổng quát của sóng sin:

$$y = A \cdot \sin(2\pi \cdot n \cdot F_w / F_s)$$

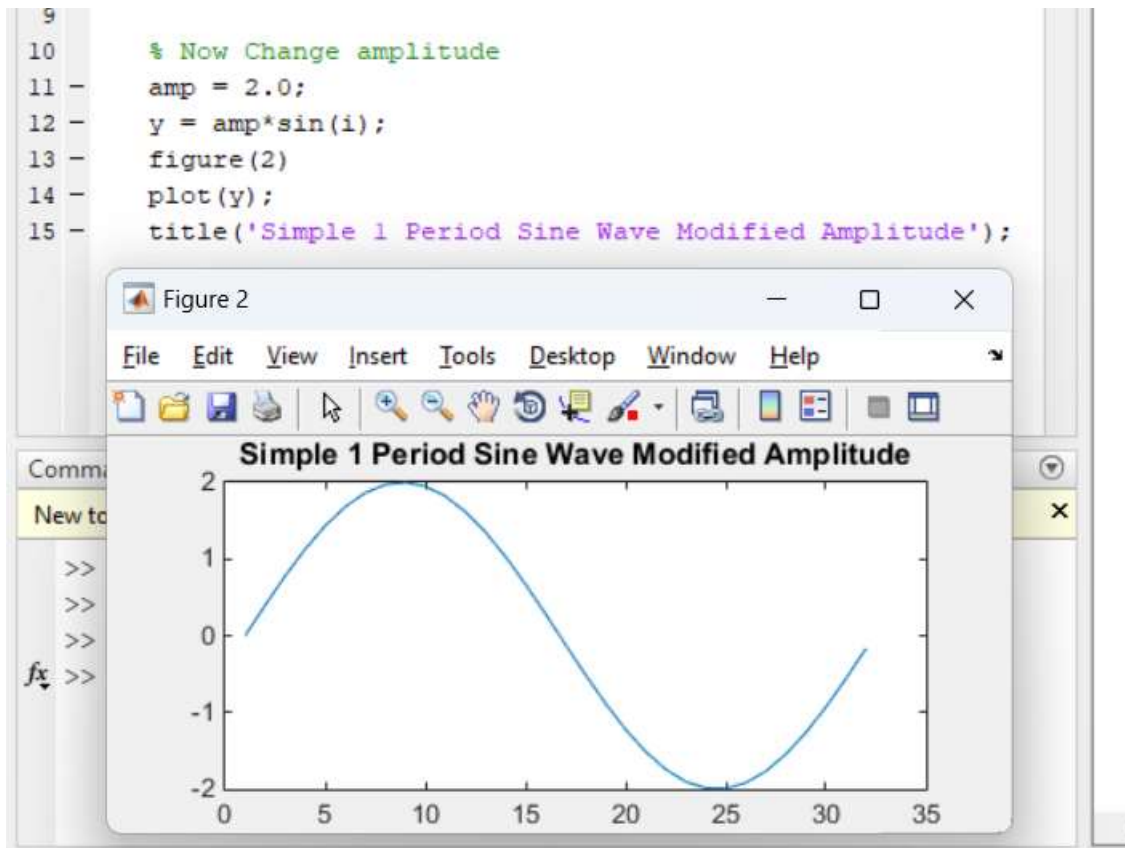
- A là biên độ.
 - F_w là tần số sóng.
 - F_s là tần số mẫu.
 - n là chỉ số mẫu.
- b. Sử dụng hàm sin() trong MATLAB để tạo sóng sin theo đơn vị radian.

2. Tạo sóng sin cơ bản trong MATLAB:

a. Vẽ đồ thị của một chu kỳ sóng sin đơn giản.

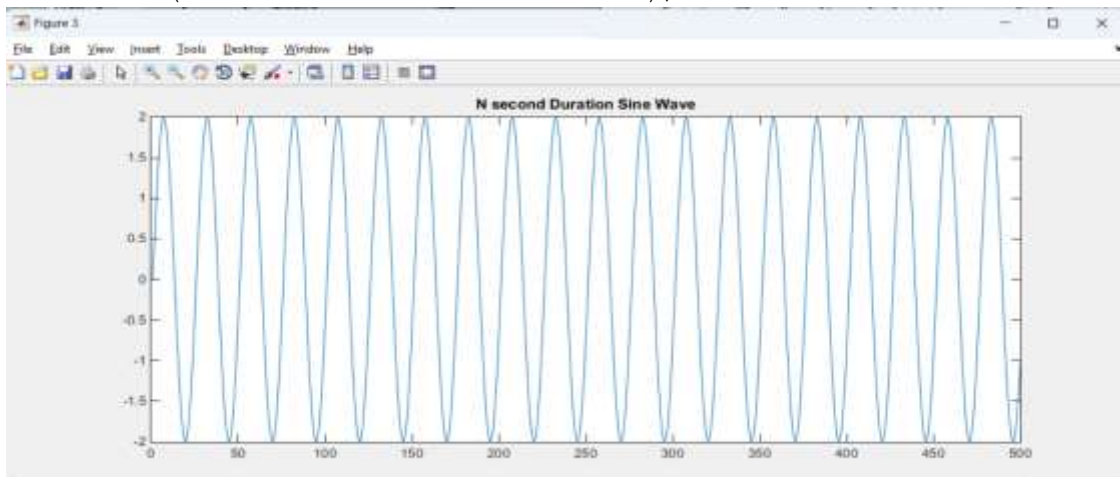


b. Thay đổi biên độ sóng sin bằng cách nhân với hệ số khuếch đại.

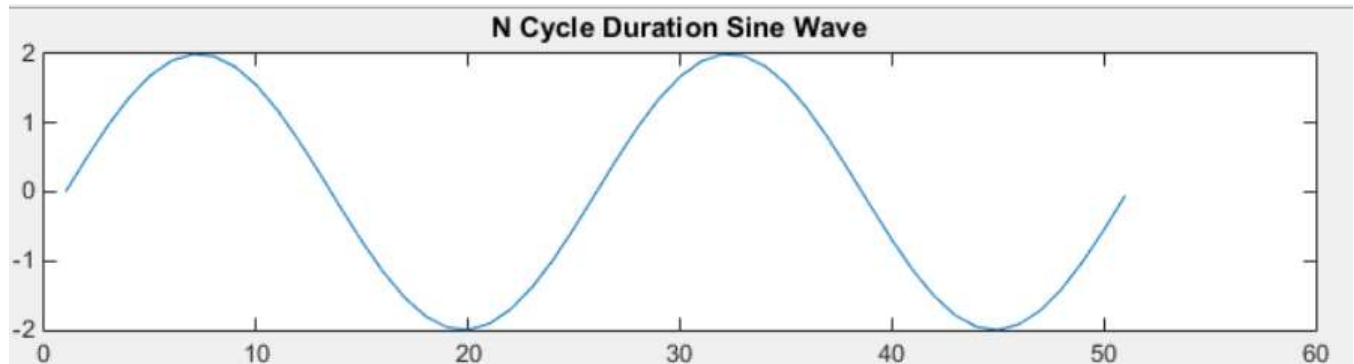


c. Thay đổi tần số sóng bằng cách điều chỉnh số lượng mẫu (số chu kỳ, thời gian).

```
F_s = 11025;
F_w = 440;
nsec = 2;
dur= nsec*F_s;
n = 0:dur;
y = amp*sin(2*pi*n*F_w/F_s);
figure(3)
plot(y(1:500));
title('N second Duration Sine Wave');
```



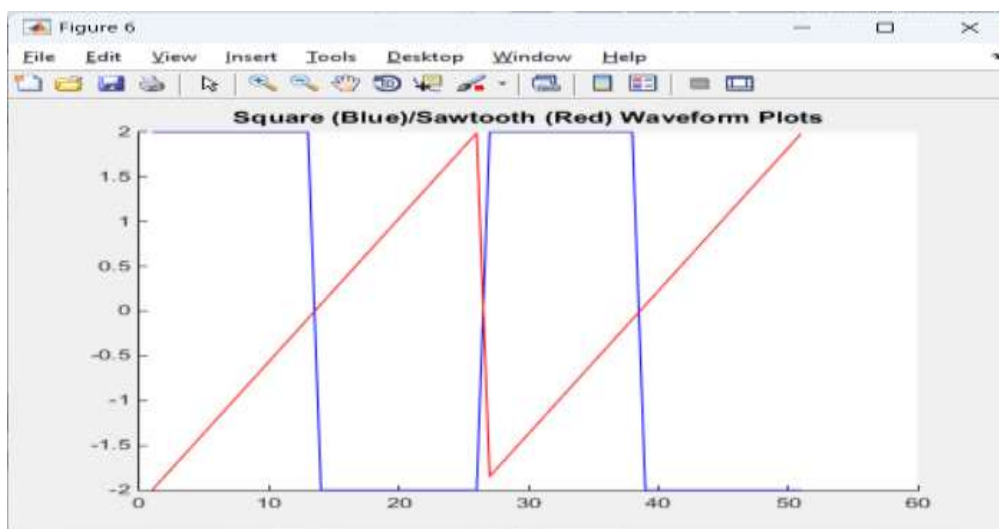
```
% To plot n cycles of a waveform
ncyc = 2;
n=0:floor(ncyc*F_s/F_w);
y = amp*sin(2*pi*n*F_w/F_s);
figure(4)
plot(y);
title('N Cycle Duration Sine Wave');
```



3. Tạo sóng vuông và sóng răng cưa:

- MATLAB hỗ trợ tạo sóng vuông và sóng răng cưa bằng cách sử dụng hàm `square()` và `sawtooth()`.

```
%3. Tạo sóng vuông và sóng răng cưa:
% Square and Sawtooth Waveforms created using Radians
ysq = amp*square(2*pi*n*F_w/F_s);
ysaw = amp*sawtooth(2*pi*n*F_w/F_s);
figure(6);
hold on
plot(ysq, 'b');
plot(ysaw, 'r');
title('Square (Blue)/Sawtooth (Red) Waveform Plots');
hold off;
```



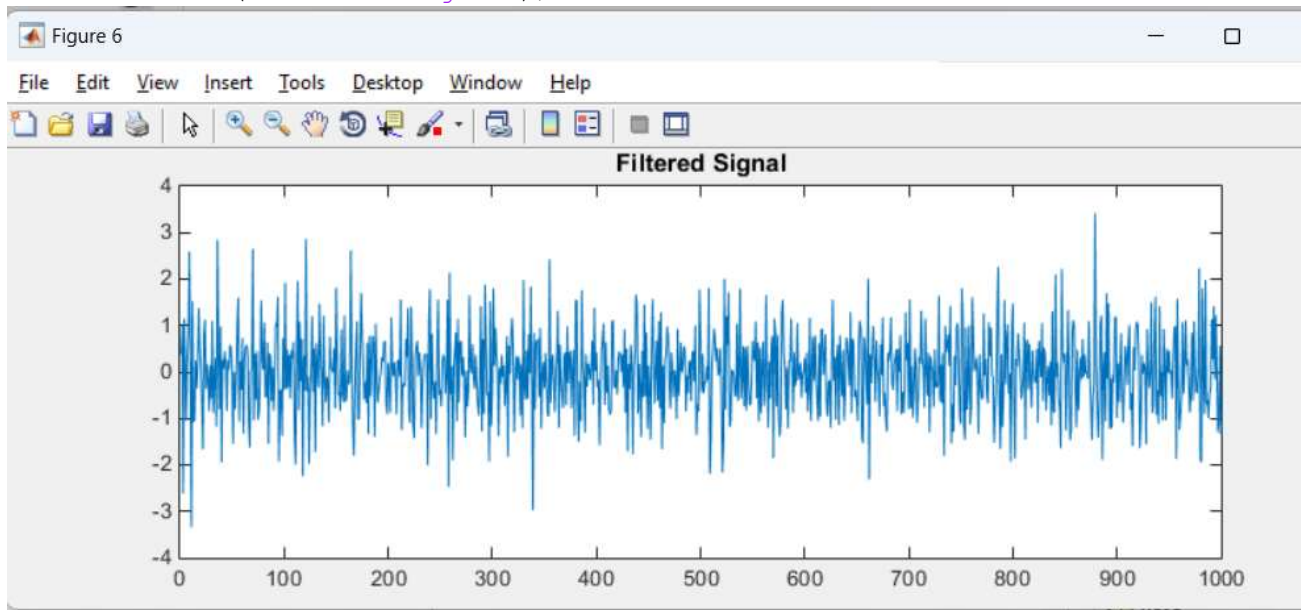
4. Lọc tín hiệu với bộ lọc IIR/FIR:

- a. MATLAB cung cấp hàm `filter()` để lọc dữ liệu dựa trên các bộ lọc số được mô tả bởi các vector A và B.

```
%4. Lọc tín hiệu với bộ lọc IIR/FIR:
fg=4000;
fa=48000;
k=tan(pi*fg/fa);
b(1)=1/(1+sqrt(2)*k+k^2);
b(2)=-2/(1+sqrt(2)*k+k^2);
b(3)=1/(1+sqrt(2)*k+k^2);
a(1)=1;
a(2)=2*(k^2-1)/(1+sqrt(2)*k+k^2);
a(3)=(1-sqrt(2)*k+k^2)/(1+sqrt(2)*k+k^2);
% Áp dụng bộ lọc IIR vào tín hiệu
input_signal = randn(1, 1000); % Ví dụ tín hiệu ngẫu nhiên

% Lọc tín hiệu
filtered_signal = filter(b, a, input_signal);

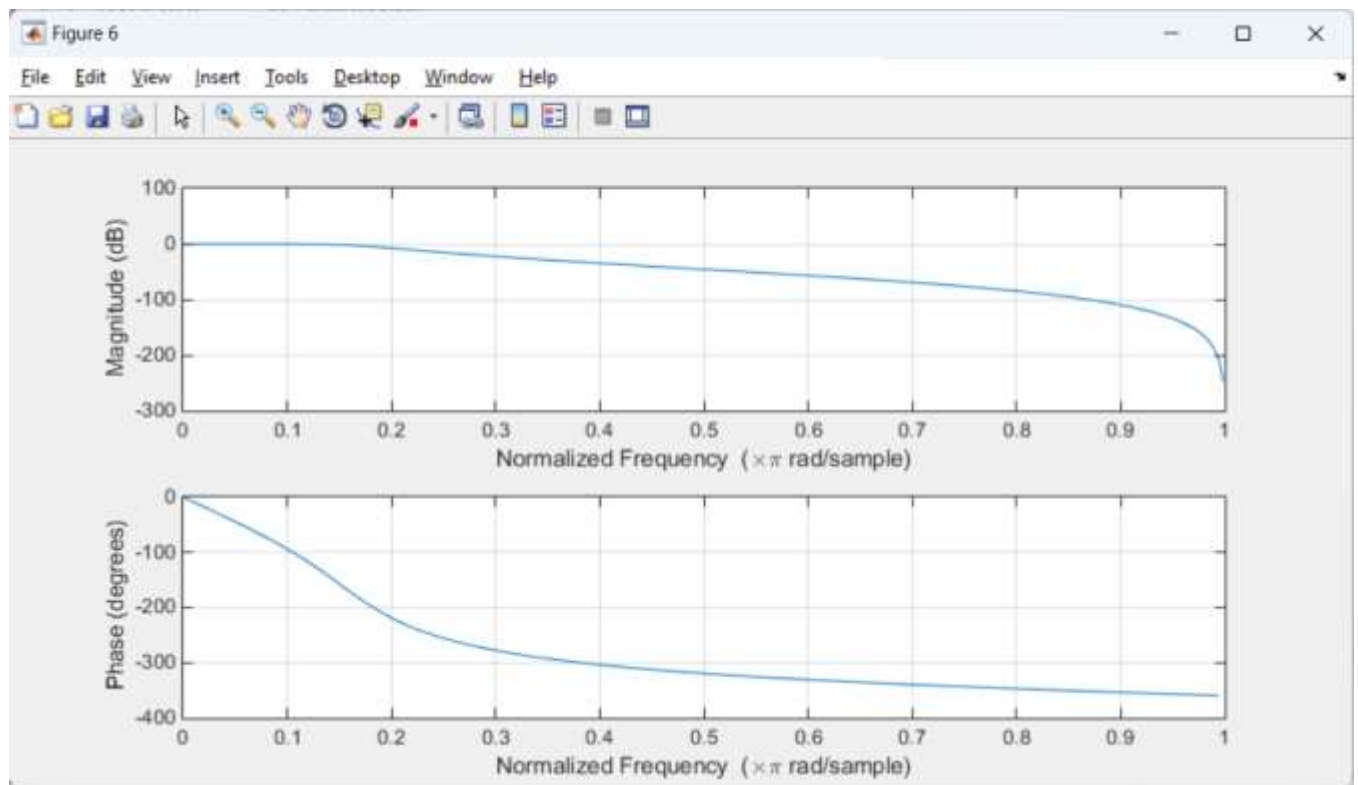
% Hiển thị tín hiệu đã lọc
plot(filtered_signal);
title('Filtered Signal');
```



- b. Có thể tạo các bộ lọc bằng tay hoặc sử dụng các bộ lọc định sẵn trong MATLAB như Butterworth.

```
%b.Có thể tạo các bộ lọc bằng tay hoặc sử dụng các bộ lọc định sẵn
sẵn trong MATLAB như Butterworth.
[b, a] = butter(n, Wn, 'type');

fs = 48000;      % Tần số lấy mẫu
fc = 4000;       % Tần số cắt
Wn = fc/(fs/2);  % Tần số cắt bình thường hóa (0 đến 1)
% Thiết kế bộ lọc Butterworth bậc 4
[b, a] = butter(4, Wn, 'low');
filtered_signal = filter(b, a, input_signal);
freqz(b, a);
```

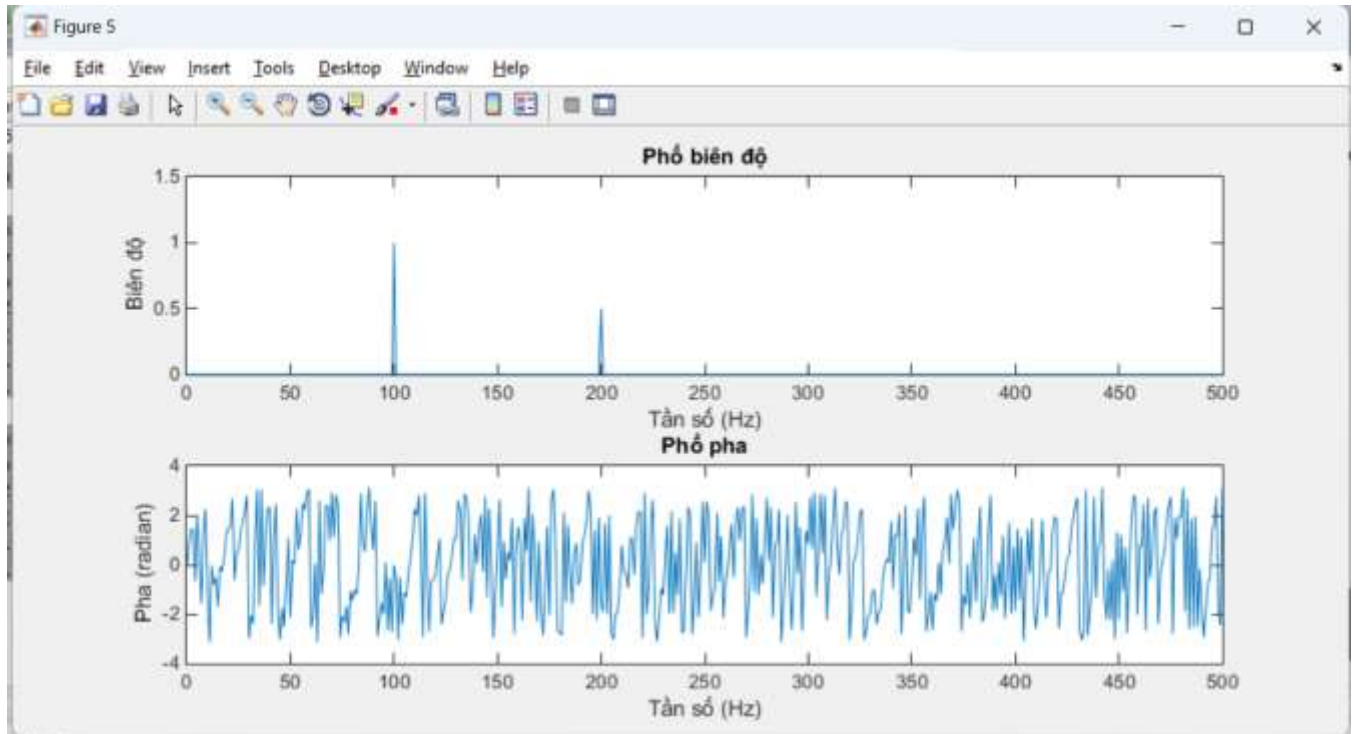


5. Biến đổi Fourier trong MATLAB:

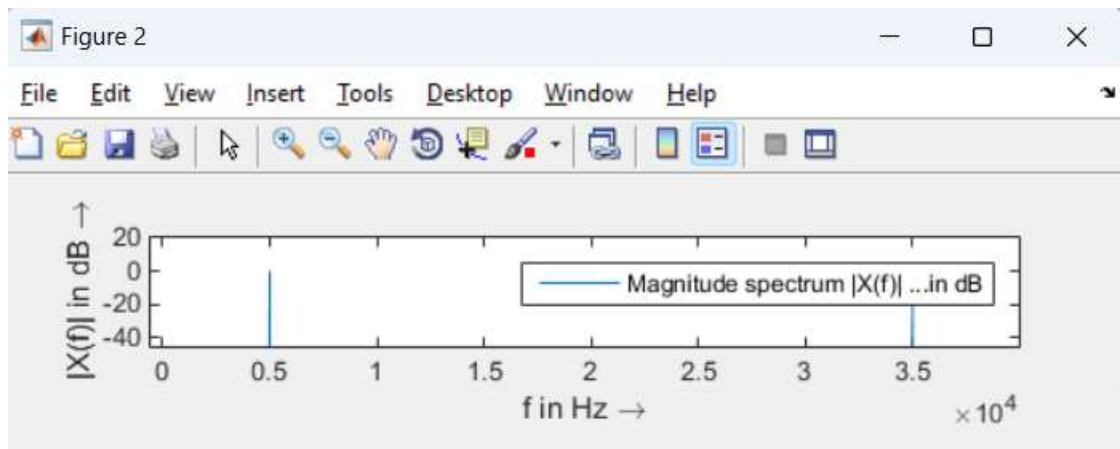
- Hàm `fft()` và `fft2()` được sử dụng để thực hiện phép biến đổi Fourier rời rạc 1D và 2D.
- Sau khi thực hiện phép biến đổi Fourier, có thể trực quan hóa phổ biên độ và phổ pha trong MATLAB.
- Công thức để chuyển đổi từ điểm mẫu sang tần số cho phép vẽ đồ thị phổ Fourier trên trục tần số Hz.

6. Ví dụ về đồ thị phổ Fourier trong MATLAB:

a. Các ví dụ về cách biểu diễn phổ biên độ và phổ pha sau biến đổi Fourier.

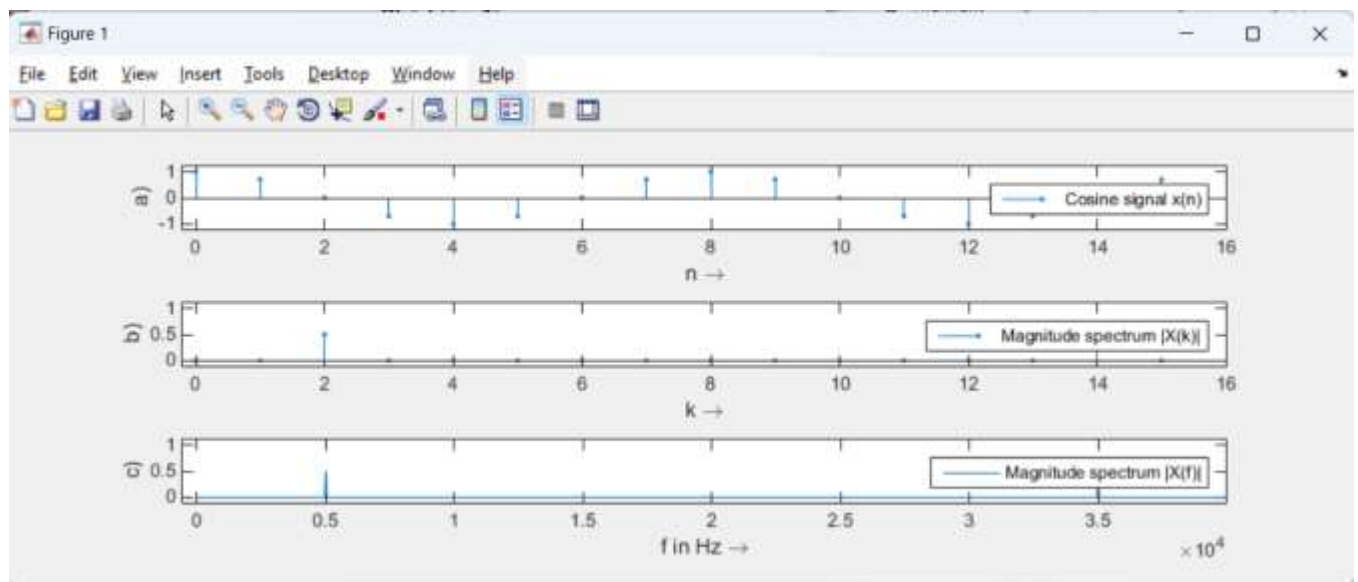


b. Cách sử dụng dải logarit để hiển thị phổ Fourier.



%bCá cách s? d?ng d?i logarit ?? hi?n th? ph? Fourier.

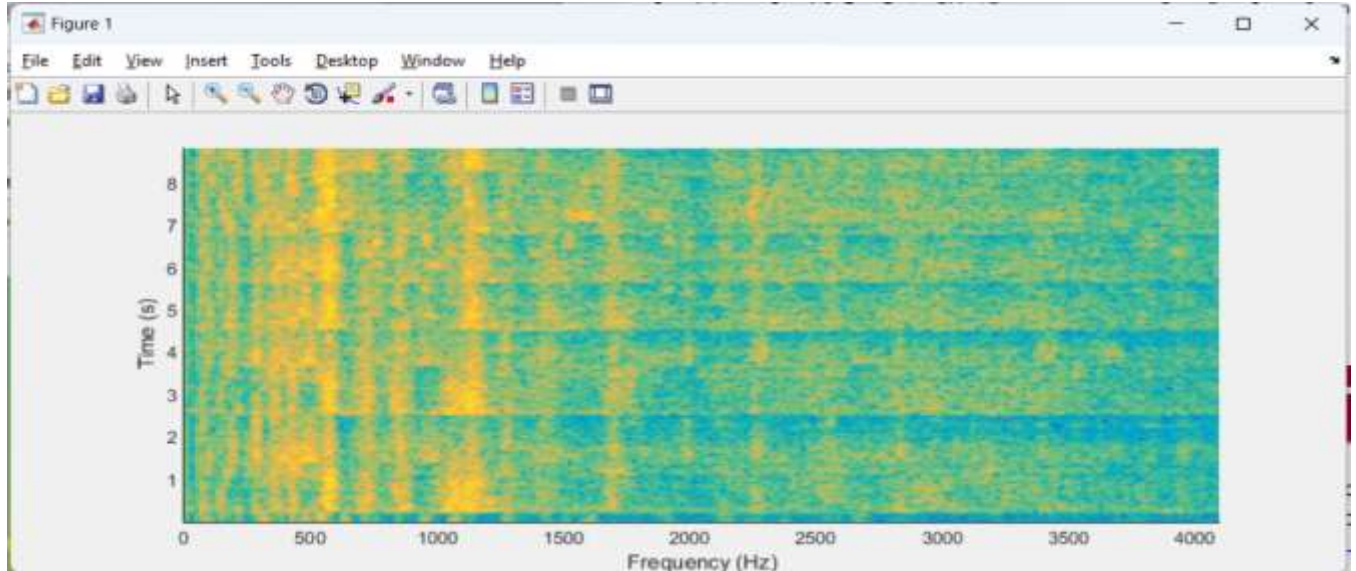
```
N=16;  
x=cos(2*pi*2*(0:1:N-1)/N)';  
figure(1)  
subplot(3,1,1);  
stem(0:N-1,x,'.');  
axis([-0.2 N -1.2 1.2]);  
legend('Cosine signal x(n)');  
ylabel('a)');  
xlabel('n \rightarrow');  
X=abs(fft(x,N))/N;  
subplot(3,1,2);stem(0:N-1,X,'.');  
axis([-0.2 N -0.1 1.1]);  
legend('Magnitude spectrum |X(k)|');  
ylabel('b)');  
xlabel('k \rightarrow');  
N=1024;  
x=cos(2*pi*(2*1024/16)*(0:1:N-1)/N)';  
  
FS=40000;  
f=(0:N-1)/N*FS;  
X=abs(fft(x,N))/N;  
subplot(3,1,3);plot(f,X);  
axis([-0.2*44100/16 max(f) -0.1 1.1]);  
legend('Magnitude spectrum |X(f)|');  
ylabel('c)');  
xlabel('f in Hz \rightarrow');  
figure(2)  
subplot(3,1,1);  
plot(f,20*log10(X./(0.5)));  
axis([-0.2*44100/16 max(f) ...  
-45 20]);  
legend('Magnitude spectrum |X(f)| ...in dB');  
ylabel('|X(f)| in dB \rightarrow');  
xlabel('f in Hz \rightarrow');
```



7. Spectrogram và hiển thị phổ theo thời gian:

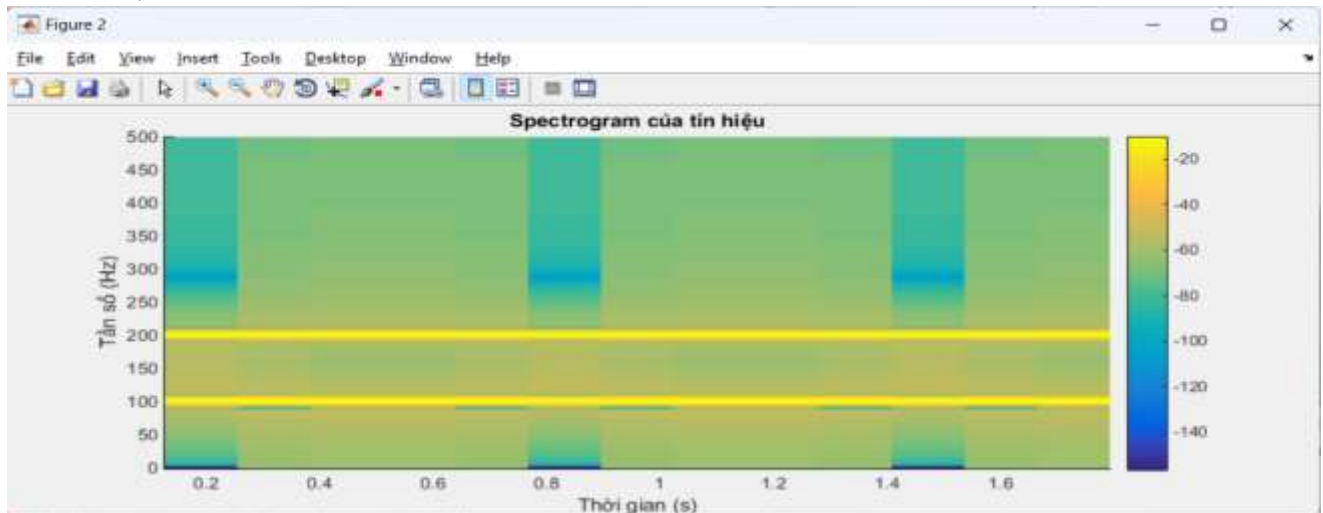
- a. Spectrogram được dùng để phân tích phân bố tần số theo thời gian ngắn (Short-Time Fourier Transform - STFT).

```
load('handel')
[N M] = size(y);
figure(1)
spectrogram(y, 512, 20, 1024, Fs);
```



- b. MATLAB cung cấp hàm spectrogram() để vẽ phổ âm thanh theo thời gian.

```
fs = 1000; % Tần số lấy mẫu (Hz)
t = 0:1/fs:2-1/fs; % Thời gian 2 giây
x = cos(2*pi*100*t) + sin(2*pi*200*t); % Tín hiệu tổng hợp với 2 tần số
% Vẽ spectrogram
figure;
spectrogram(x, 256, [], [], fs, 'yaxis');
title('Spectrogram của tín hiệu');
xlabel('Thời gian (s)');
ylabel('Tần số (Hz)');
colorbar;
```



8. Ví dụ về Spectrogram của bài hát:

- a. Trình bày cách Aphex Twin nhúng hình ảnh vào spectrogram của bản nhạc.
- b. Có thể sử dụng MATLAB để tạo và hiển thị hình ảnh nhúng trong âm thanh thông qua spectrogram.

9. Chuyển đổi từ hình ảnh sang âm thanh:

- a. MATLAB cung cấp một đoạn mã để chuyển đổi hình ảnh thành âm thanh bằng cách ánh xạ các dãy pixel trong hình ảnh với các tần số âm thanh.
- b. Âm thanh kết quả có thể được lưu dưới định dạng .wav và biểu diễn bằng cách sử dụng spectrogram.