**ECE3051 – Analog and Digital Signal Processing, Fall Semester 2022-2023**

**ELA DA – 2, Slot: L25-L26**

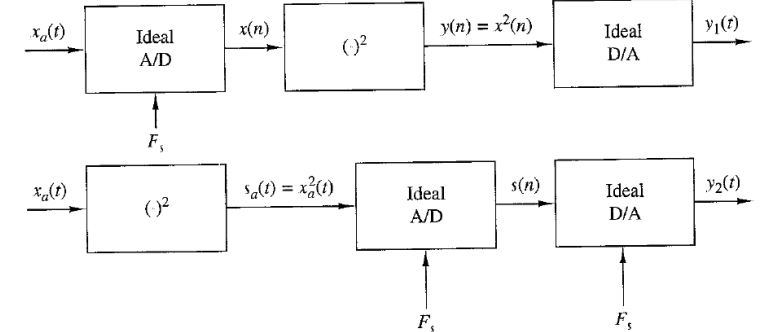
**By: Jonathan Rufus Samuel (20BCT0332) Date: 11.09.2022**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ELA DA 2 – DOS: 11.09.2022**

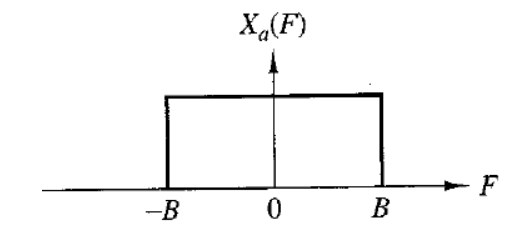
**Task - 2: SAMPLING AND RECONSTRUCTION OF SIGNALS**

**Q1) Consider the two systems shown in the below fig:**



**a. Sketch the spectra of the various signals if 𝑥𝑎(𝑡) has the Fourier transform shown in the below fig. and 𝐹𝑠 = 2𝐵. How are 𝑦1(𝑡) and 𝑦2(𝑡) related to 𝑥𝑎(𝑡)?**

**b. Determine 𝑦1(𝑡) and 𝑦2(𝑡) if 𝑥𝑎 (𝑡) = cos 2π𝐹0𝑡, 𝐹0 = 20Hz, and 𝐹𝑠 = 50𝐻𝑧 and 𝐹𝑠 = 30Hz.**

****

**CODE:**

%Task - 2: SAMPLING AND RECONSTRUCTION OF SIGNALS

%Name: Jonathan Rufus Samuel (20BCT0332)

%Course: ECE3051 - ELA

%SubTask 1 - Consider the two systems shown in the below fig.

%a. Sketch the spectra of the various signals if 𝑥𝑎(𝑡) has the Fourier transform shown

% in the below fig. and 𝐹𝑠 = 2𝐵. How are 𝑦1(𝑡) and 𝑦2(𝑡) related to 𝑥𝑎(𝑡)?

<SHOWN IN NOTES>

%b. Determine 𝑦1(𝑡) and 𝑦2(𝑡) if 𝑥𝑎(𝑡) = cos 2π𝐹0𝑡, 𝐹0 = 20Hz, and 𝐹𝑠 = 50𝐻𝑧 and 𝐹𝑠 = 30Hz.

%Case a) Fs = 50Hz

n0 = -20:20;

syms x(n);

x(n) = cos(4\*pi\*n/5);

subplot(321),plot(n0,x(n0));

title('Sequence #1.1 - x(n)');

xlabel('x(n)');

ylabel('n0');

syms y(n);

y(n) = 1/2 + (1/2)\*cos(8\*pi\*n/5);

subplot(322),plot(n0,y(n0));

title('Sequence #1.2 - y(n)');

xlabel('y(n)');

ylabel('n0');

syms y1(t);

y1(t) = 1/2 + cos(20\*pi\*t)\*(1/2);

subplot(323),plot(n0,y1(n0));

title('Sequence #1.3 - y1(t) and y2(t)');

xlabel('y1(t)');

ylabel('t');

%Case b) Fs = 30Hz

n0 = -20:20;

syms x(n);

x(n) = cos(2\*pi\*n/3);

subplot(324),plot(n0,x(n0));

title('Sequence #2.1 - x(n)');

xlabel('x(n)');

ylabel('n0');

syms y(n);

y(n) = x(n)^2;

subplot(325),plot(n0,y(n0));

title('Sequence #2.2 - y(n)');

xlabel('y(n)');

ylabel('n0');

syms y1(t);

y1(t) = 1/2 + cos(20\*pi\*t)\*(1/2);

subplot(326),plot(n0,y1(n0));

title('Sequence #2.3 - y1(t) and y2(t)');

xlabel('y1(t)');

ylabel('t');

**OUTPUT:**

>> %b. Determine 𝑦1(𝑡) and 𝑦2(𝑡) if 𝑥𝑎(𝑡) = cos 2π𝐹0𝑡, 𝐹0 = 20Hz, and 𝐹𝑠 = 50𝐻𝑧 and 𝐹𝑠 = 30Hz.

%Case a) Fs = 50Hz

n0 = -20:20;

syms x(n);

x(n) = cos(4\*pi\*n/5);

subplot(321),plot(n0,x(n0));

title('Sequence #1.1 - x(n)');

xlabel('x(n)');

ylabel('n0');

syms y(n);

y(n) = 1/2 + (1/2)\*cos(8\*pi\*n/5);

subplot(322),plot(n0,y(n0));

title('Sequence #1.2 - y(n)');

xlabel('y(n)');

ylabel('n0');

syms y1(t);

y1(t) = 1/2 + cos(20\*pi\*t)\*(1/2);

subplot(323),plot(n0,y1(n0));

title('Sequence #1.3 - y1(t) and y2(t)');

xlabel('y1(t)');

ylabel('t');

%Case b) Fs = 30Hz

n0 = -20:20;

syms x(n);

x(n) = cos(2\*pi\*n/3);

subplot(324),plot(n0,x(n0));

title('Sequence #2.1 - x(n)');

xlabel('x(n)');

ylabel('n0');

syms y(n);

y(n) = x(n)^2;

subplot(325),plot(n0,y(n0));

title('Sequence #2.2 - y(n)');

xlabel('y(n)');

ylabel('n0');

syms y1(t);

y1(t) = 1/2 + cos(20\*pi\*t)\*(1/2);

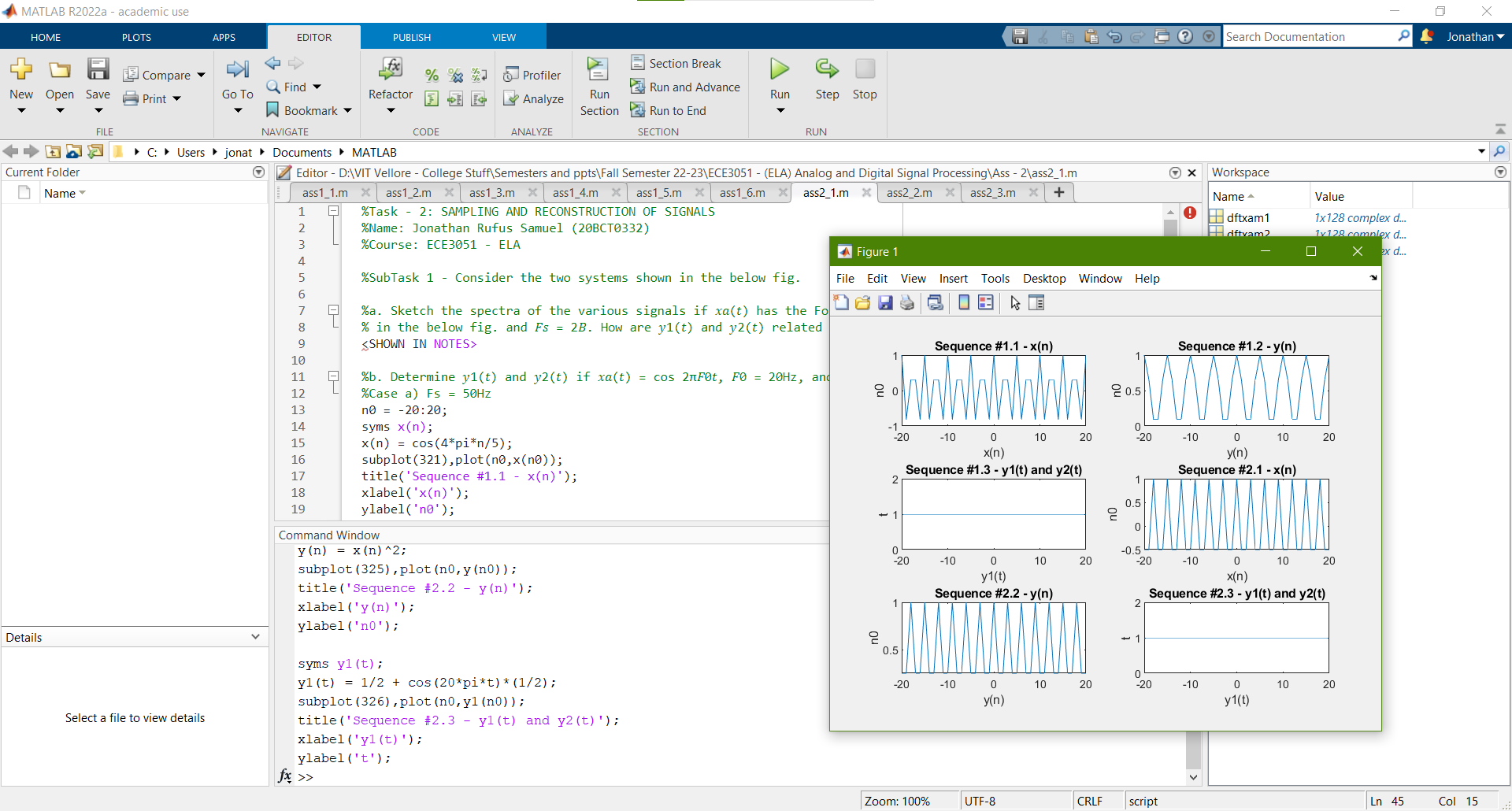
subplot(326),plot(n0,y1(n0));

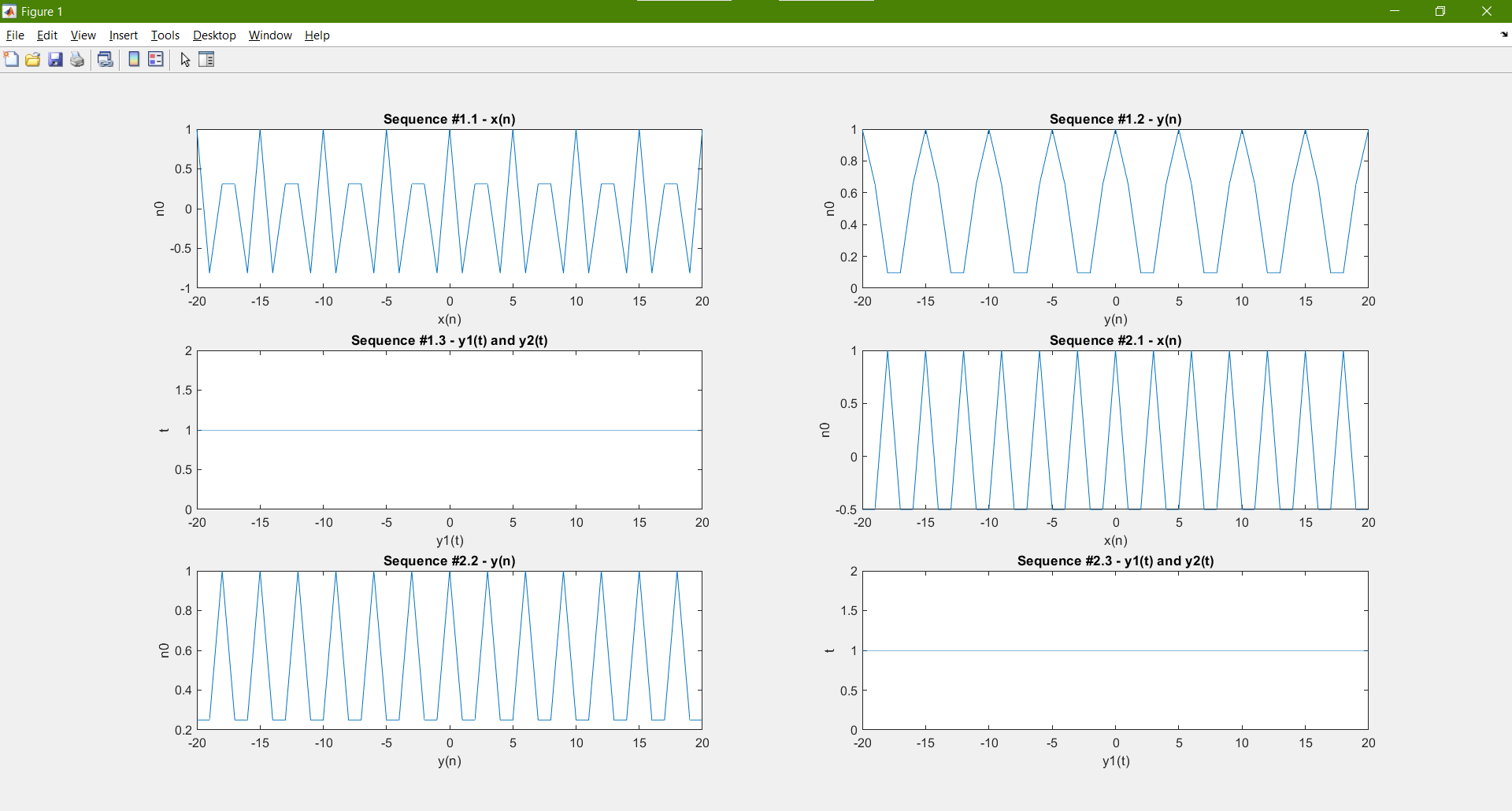
title('Sequence #2.3 - y1(t) and y2(t)');

xlabel('y1(t)');

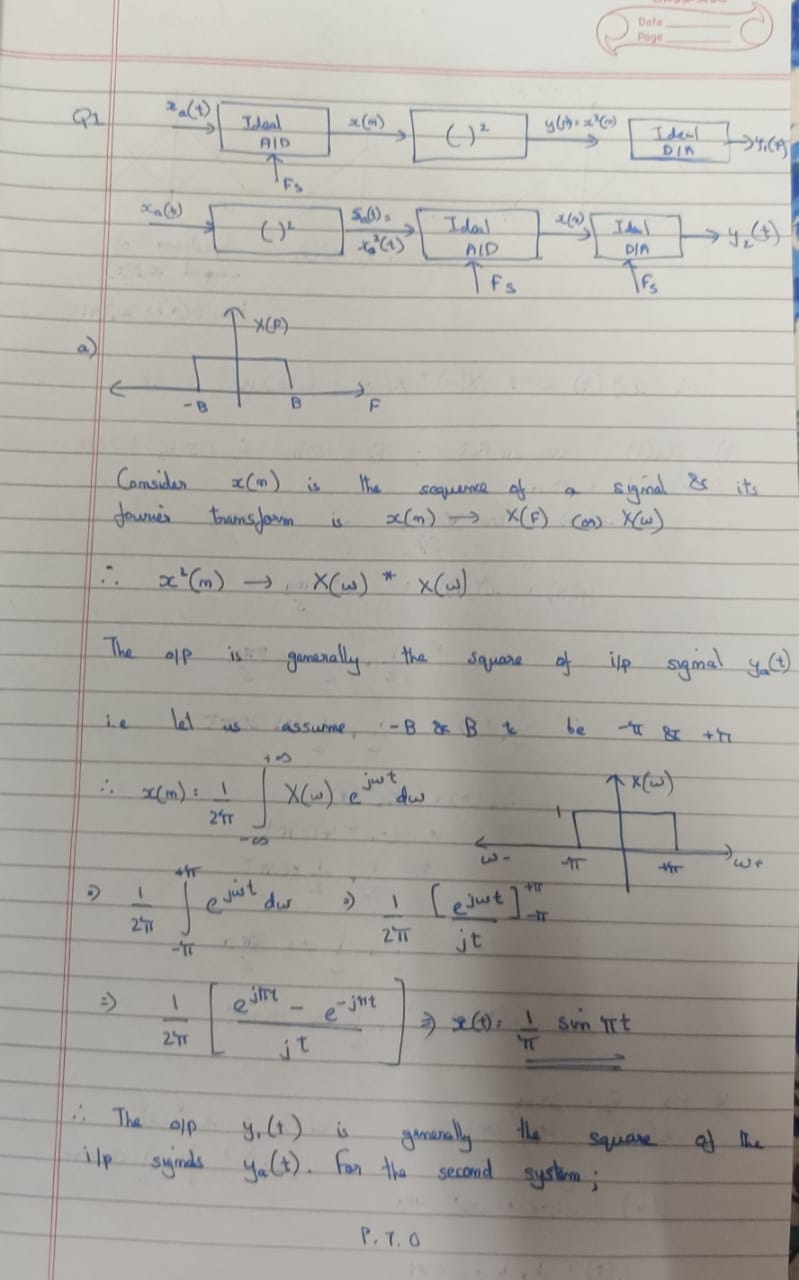
ylabel('t');

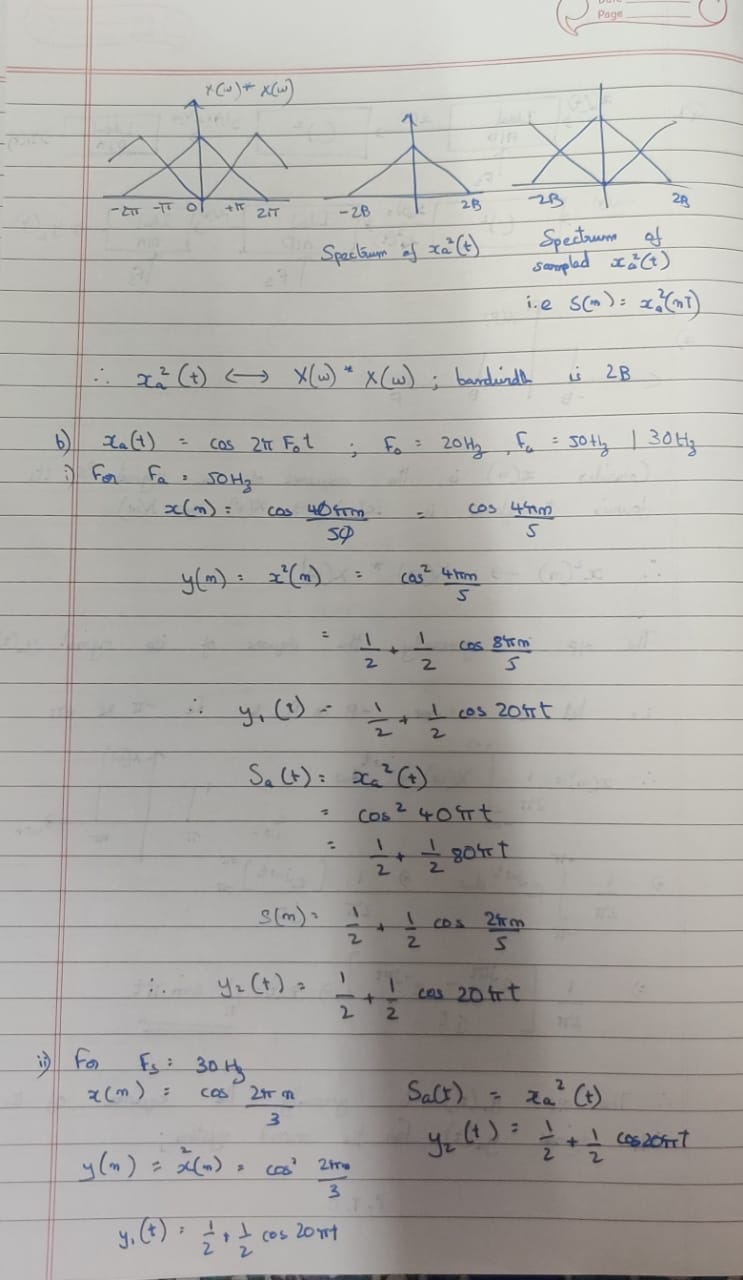
>>





**WORKING OUT:**





**Q2) Frequency analysis of amplitude-modulated discrete-time signal-The discrete-time 𝑥(𝑛) = cos 2𝜋𝑓1𝑛 + cos 2𝜋𝑓2𝑛, 𝑓1 = 1/18, 𝑓2 = 5/128, modulates the amplitude of the carrier 𝑥𝑐(𝑛) = cos 2𝜋𝑓𝑐𝑛 with 𝑓𝑐 = 50/128. The resulting amplitude-modulated signal is 𝑥am(𝑛) = 𝑥(𝑛)𝑥𝑐 (𝑛) = 𝑥(𝑛)cos 2𝜋𝑓𝑐?**

**Sketch the signals 𝑥(𝑛), 𝑥𝑐(𝑛), and 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 255. • Compute and sketch the 128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 127. N=128 • Compute and sketch the 128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 99. • Compute and sketch the 256-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 179. • Explain the results obtained in parts (b) through (d), by deriving the spectrum of the amplitude-modulated signal and comparing it with the experimental results.**

**CODE:**

%SubTask 2 - Frequency analysis of amplitude-modulated discrete-time

%signal-The discrete-time 𝑥(𝑛) = cos 2𝜋𝑓1𝑛 + cos 2𝜋𝑓2𝑛, 𝑓1 = 1/18, 𝑓2 =

% 5/128, modulates the amplitude of the carrier 𝑥𝑐(𝑛) = cos 2𝜋𝑓𝑐𝑛 with

% 𝑓𝑐 = 50/128. The resulting amplitude-modulated signal is 𝑥am(𝑛) =

% 𝑥(𝑛)𝑥𝑐 (𝑛) = 𝑥(𝑛)cos 2𝜋𝑓𝑐𝑛

%a) Sketch the signals 𝑥(𝑛), 𝑥𝑐(𝑛), and 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 255.

n = 0:255;

f1 = 1/18; f2 = 5/128;

fc = 50/128;

xn = cos(2\*pi\*f1\*n)+cos(2\*pi\*f2\*n);%x[n]

xcn = cos(2\*pi\*fc\*n);%xc[n]

xam = xn.\*xcn;%xam[n]

subplot(3,1,1);

stem(n,xn);

title('Signal x(n)');

xlabel('x(n)');

ylabel('n');

subplot(3,1,2);

stem(n,xcn);

title('Signal xc(n)');

xlabel('xc(n)');

ylabel('n');

subplot(3,1,3);

stem(n,xam);

title('Signal xam(n)');

xlabel('xam(n)');

ylabel('n');

%b) Compute and sketch the 128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 127. N=128

dftxam1 = fft(xam(1:128),128);%calculating 128 point dft for 0<=n<=127

plot(abs(dftxam1));

title('128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 127');

xlabel('x(n)');

ylabel('n');

%c) Compute and sketch the 128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 99.

dftxam2 = fft(xam(1:100),128);%calculating 128 point dft for 0<=n<=99

plot(abs(dftxam2));

title('128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 99');

xlabel('x(n)');

ylabel('n');

%d) Compute and sketch the 256-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 179.

dftxam3 = fft(xam(1:180),256);%calculating 256 point dft for 0<=n<=179

plot(abs(dftxam3));

title('256-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 179');

xlabel('x(n)');

ylabel('n');

**OUTPUT:**

>> %SubTask 2 - Frequency analysis of amplitude-modulated discrete-time

%signal-The discrete-time 𝑥(𝑛) = cos 2𝜋𝑓1𝑛 + cos 2𝜋𝑓2𝑛, 𝑓1 = 1/18, 𝑓2 =

% 5/128, modulates the amplitude of the carrier 𝑥𝑐(𝑛) = cos 2𝜋𝑓𝑐𝑛 with

% 𝑓𝑐 = 50/128. The resulting amplitude-modulated signal is 𝑥am(𝑛) =

% 𝑥(𝑛)𝑥𝑐 (𝑛) = 𝑥(𝑛)cos 2𝜋𝑓𝑐𝑛

%a) Sketch the signals 𝑥(𝑛), 𝑥𝑐(𝑛), and 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 255.

n = 0:255;

f1 = 1/18; f2 = 5/128;

fc = 50/128;

xn = cos(2\*pi\*f1\*n)+cos(2\*pi\*f2\*n);%x[n]

xcn = cos(2\*pi\*fc\*n);%xc[n]

xam = xn.\*xcn;%xam[n]

subplot(3,1,1);

stem(n,xn);

title('Signal x(n)');

xlabel('x(n)');

ylabel('n');

subplot(3,1,2);

stem(n,xcn);

title('Signal xc(n)');

xlabel('xc(n)');

ylabel('n');

subplot(3,1,3);

stem(n,xam);

title('Signal xam(n)');

xlabel('xam(n)');

ylabel('n');

%b) Compute and sketch the 128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 127. N=128

dftxam1 = fft(xam(1:128),128);%calculating 128 point dft for 0<=n<=127

plot(abs(dftxam1));

title('128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 127');

xlabel('x(n)');

ylabel('n');

%c) Compute and sketch the 128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 99.

dftxam2 = fft(xam(1:100),128);%calculating 128 point dft for 0<=n<=99

plot(abs(dftxam2));

title('128-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 99');

xlabel('x(n)');

ylabel('n');

%d) Compute and sketch the 256-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 179.

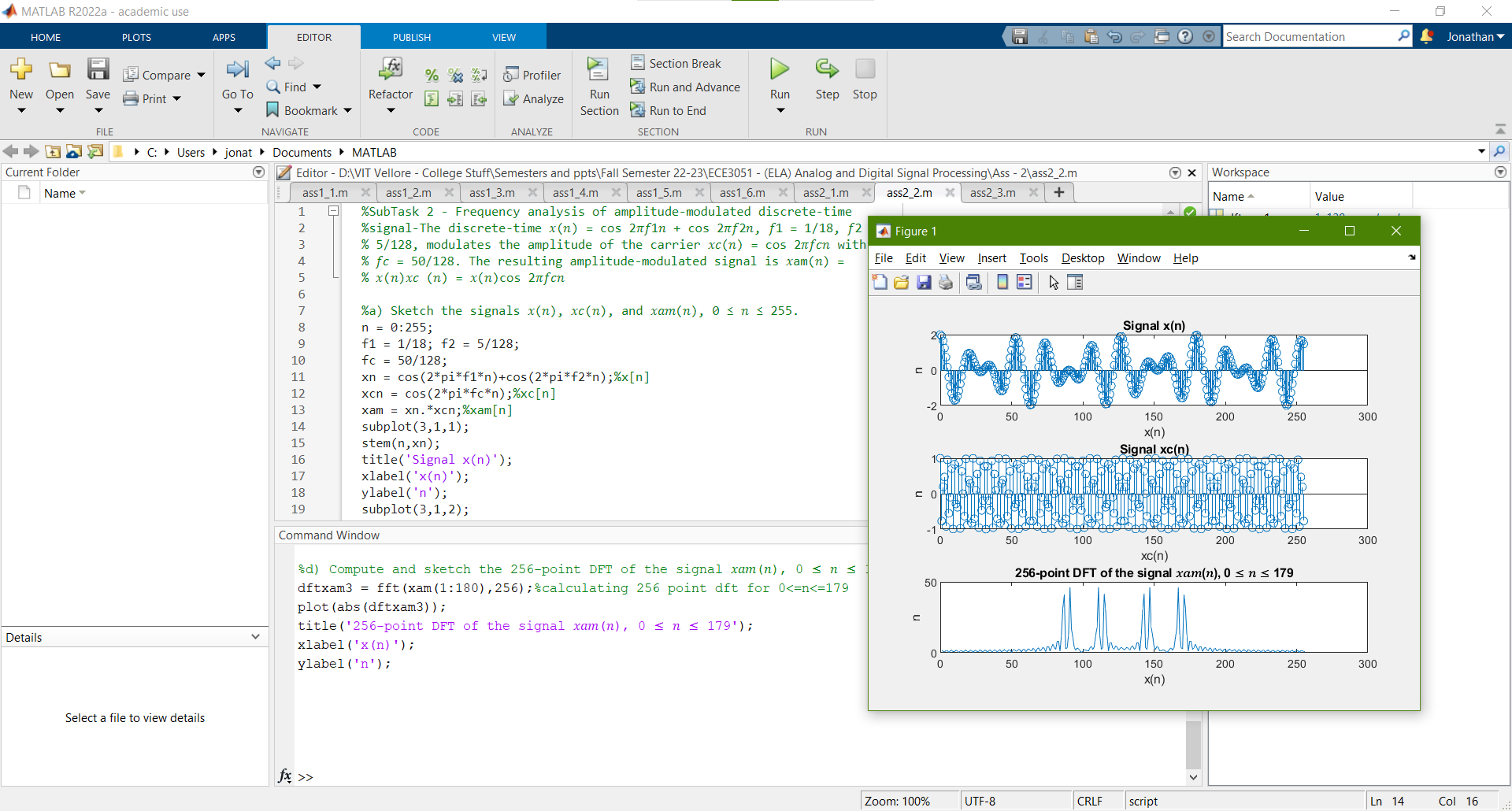
dftxam3 = fft(xam(1:180),256);%calculating 256 point dft for 0<=n<=179

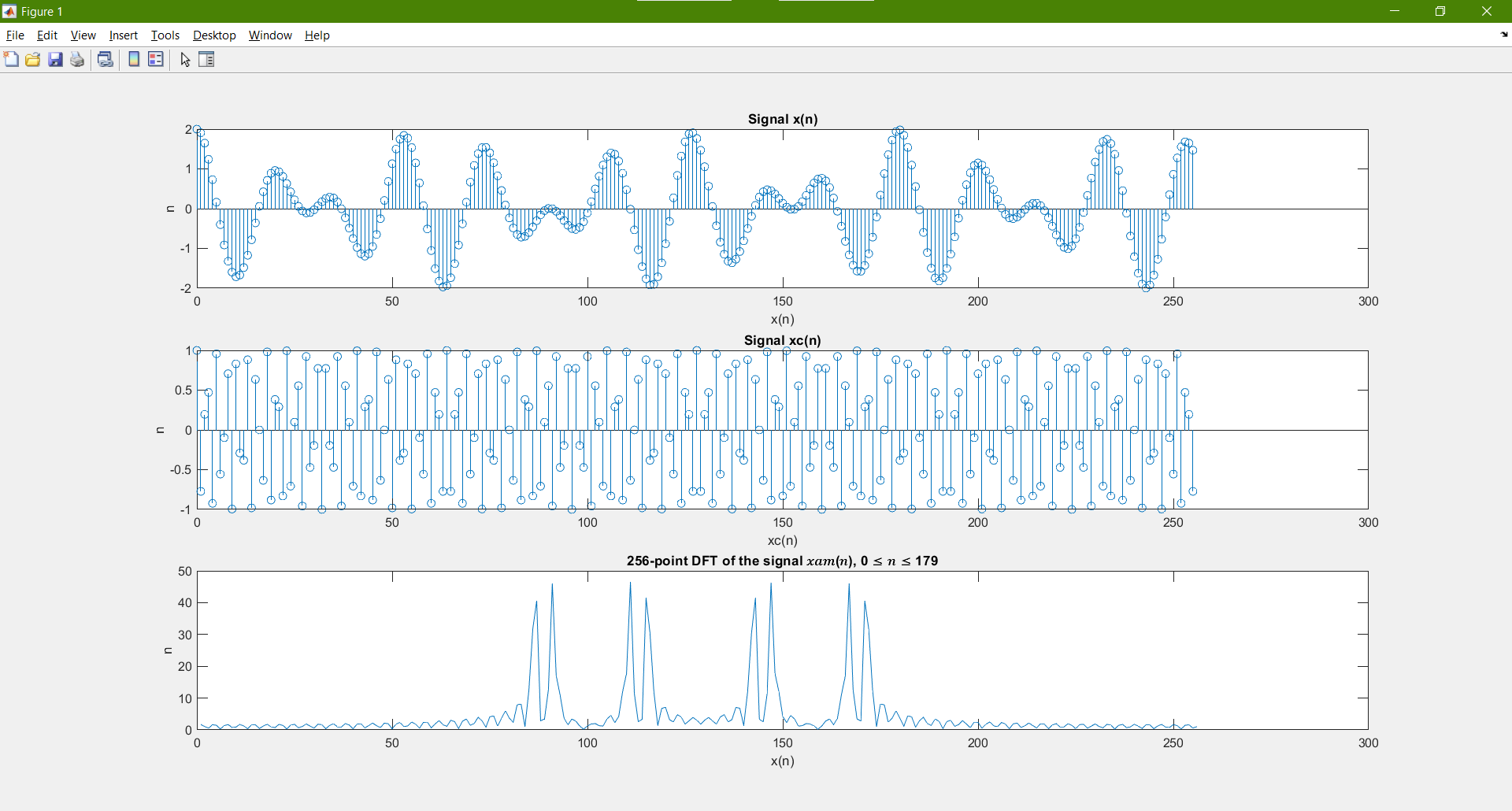
plot(abs(dftxam3));

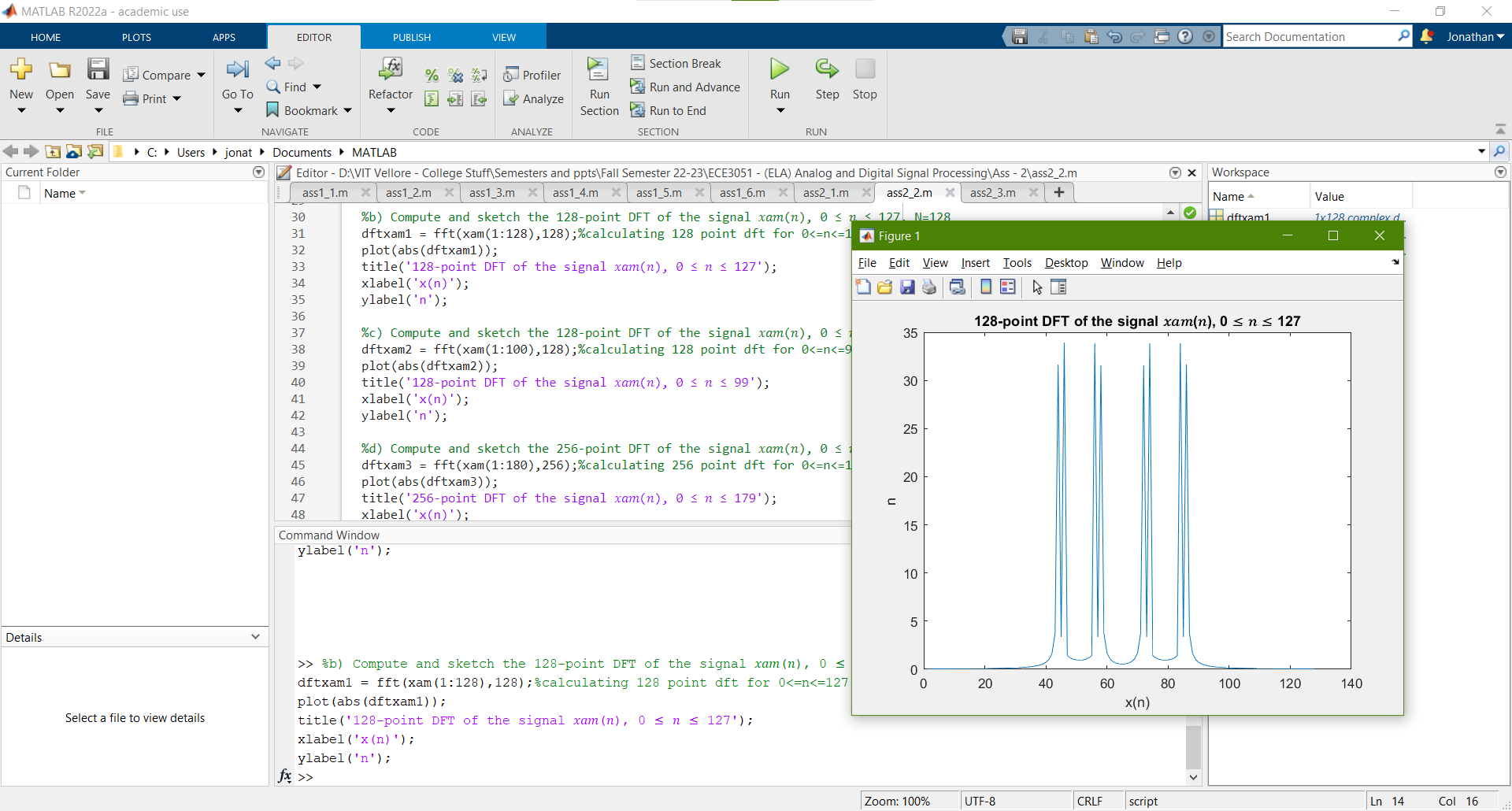
title('256-point DFT of the signal 𝑥𝑎𝑚(𝑛), 0 ≤ 𝑛 ≤ 179');

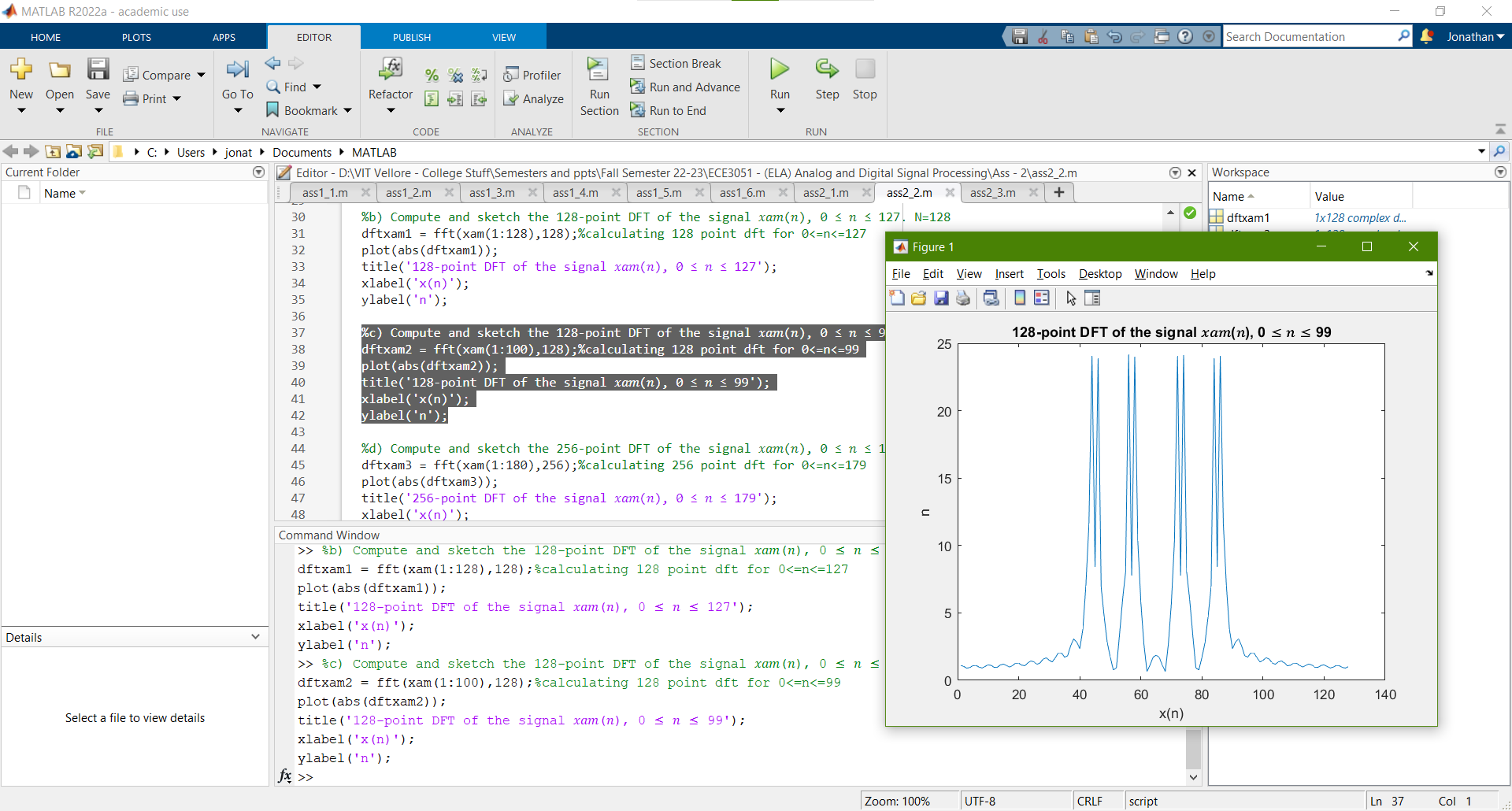
xlabel('x(n)');

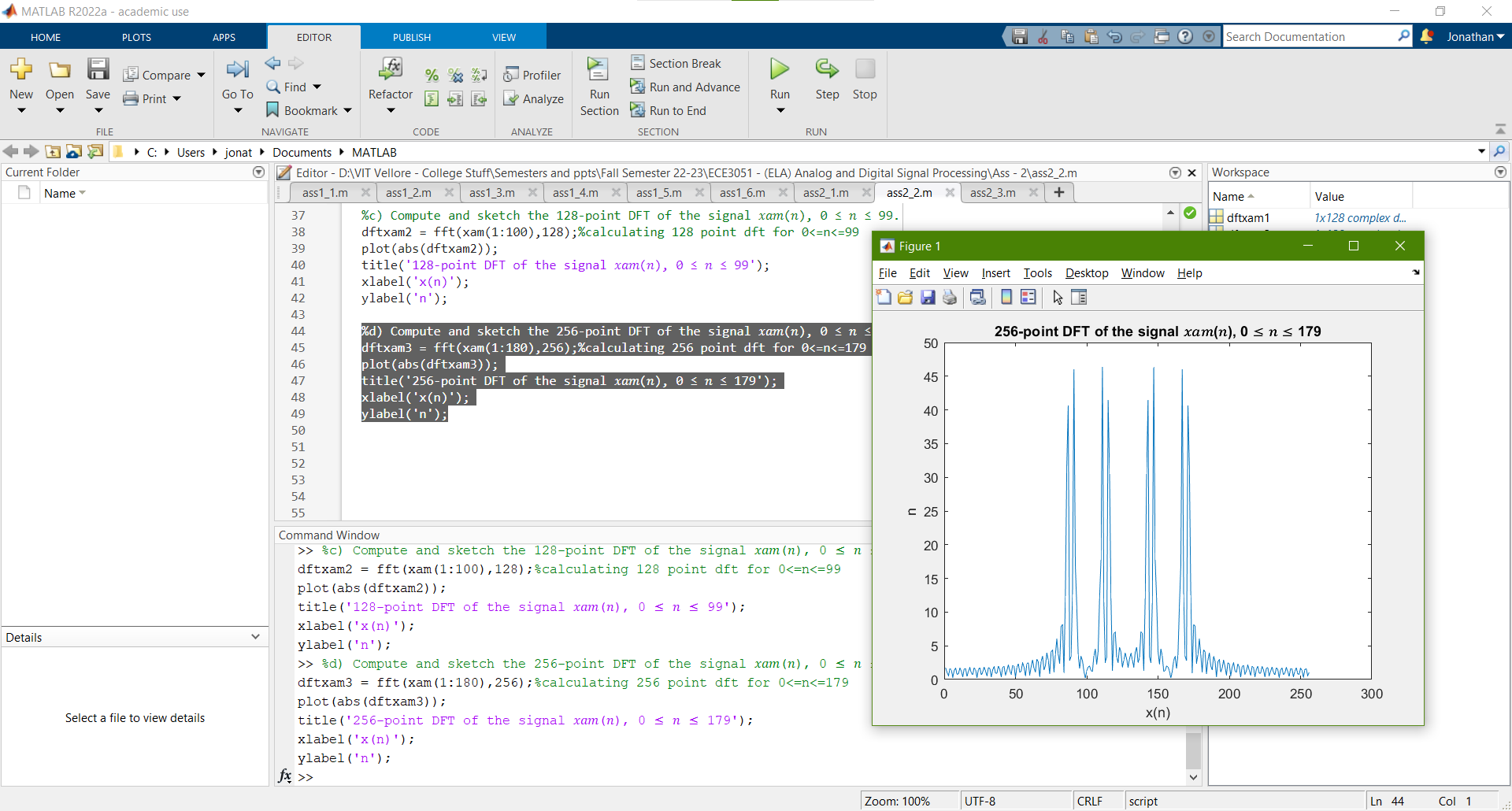
ylabel('n');



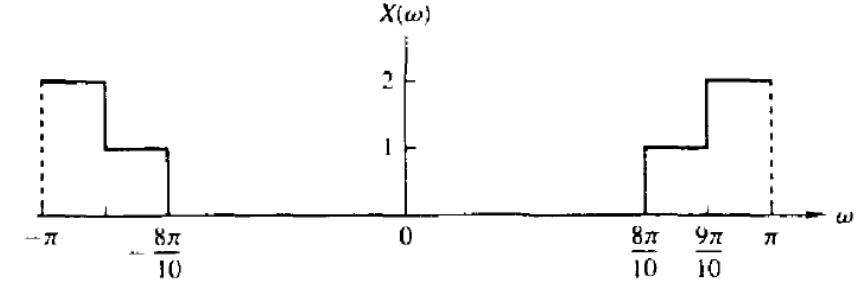








**Q3) Determine the signal 𝑥(𝑛) if its Fourier transform is as given in the figure below:**

****

**CODE:**

%SubTask 3 - Determine the signal 𝑥(𝑛) if its Fourier transform is as

%given in the below figure (Figure shown in final document)

%(Problem Working shown in final Document)

%Inverse Fourier Transform for the given figure:

% x(n) = 1/pi[2sin(9pi/10) - sin(8pi/10)]; x(0) = 3/10

syms x(n);

x(n) = 1/pi \* (2\*sin(9\*pi\*n/10) - sin(8\*pi\*n/10));

t = -50:50; %t = -20:20;

subplot(211),plot(t,x(t));

title('Signal x(n) - Derived by Inverse Fourier Transform');

xlabel('time (t)');

ylabel('Magnitude (x(n))');

grid;

%Verification of x(0)

disp(x(0));

**OUTPUT:**

>> %SubTask 3 - Determine the signal 𝑥(𝑛) if its Fourier transform is as

%given in the below figure (Figure shown in final document)

%(Problem Working shown in final Document)

%Inverse Fourier Transform for the given figure:

% x(n) = 1/pi[2sin(9pi/10) - sin(8pi/10)]; x(0) = 3/10

syms x(n);

x(n) = 1/pi \* (2\*sin(9\*pi\*n/10) - sin(8\*pi\*n/10));

t = -50:50; %t = -20:20;

subplot(211),plot(t,x(t));

title('Signal x(n) - Derived by Inverse Fourier Transform');

xlabel('time (t)');

ylabel('Magnitude (x(n))');

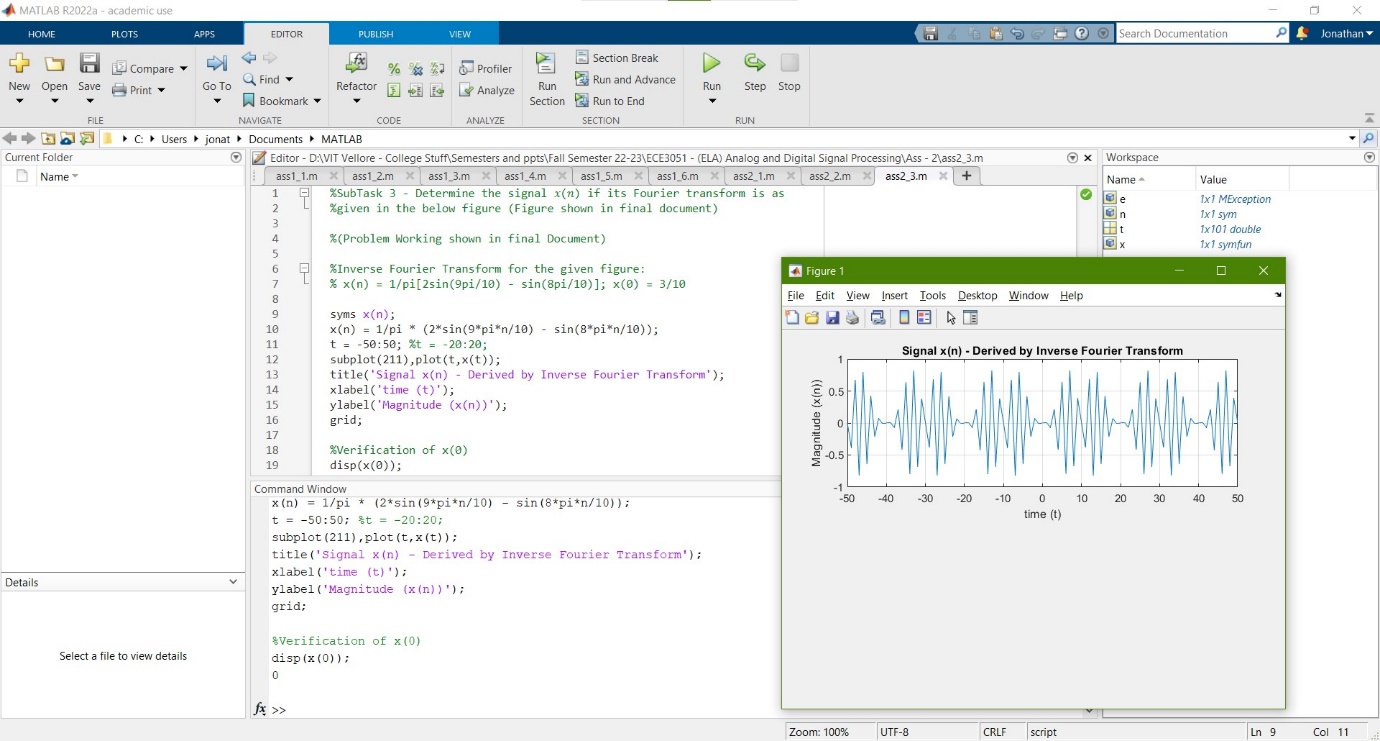
grid;

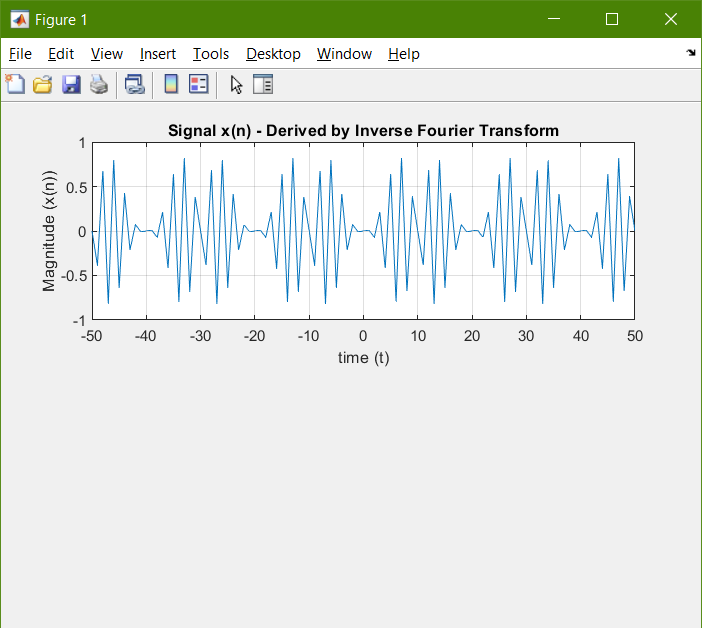
%Verification of x(0)

disp(x(0));

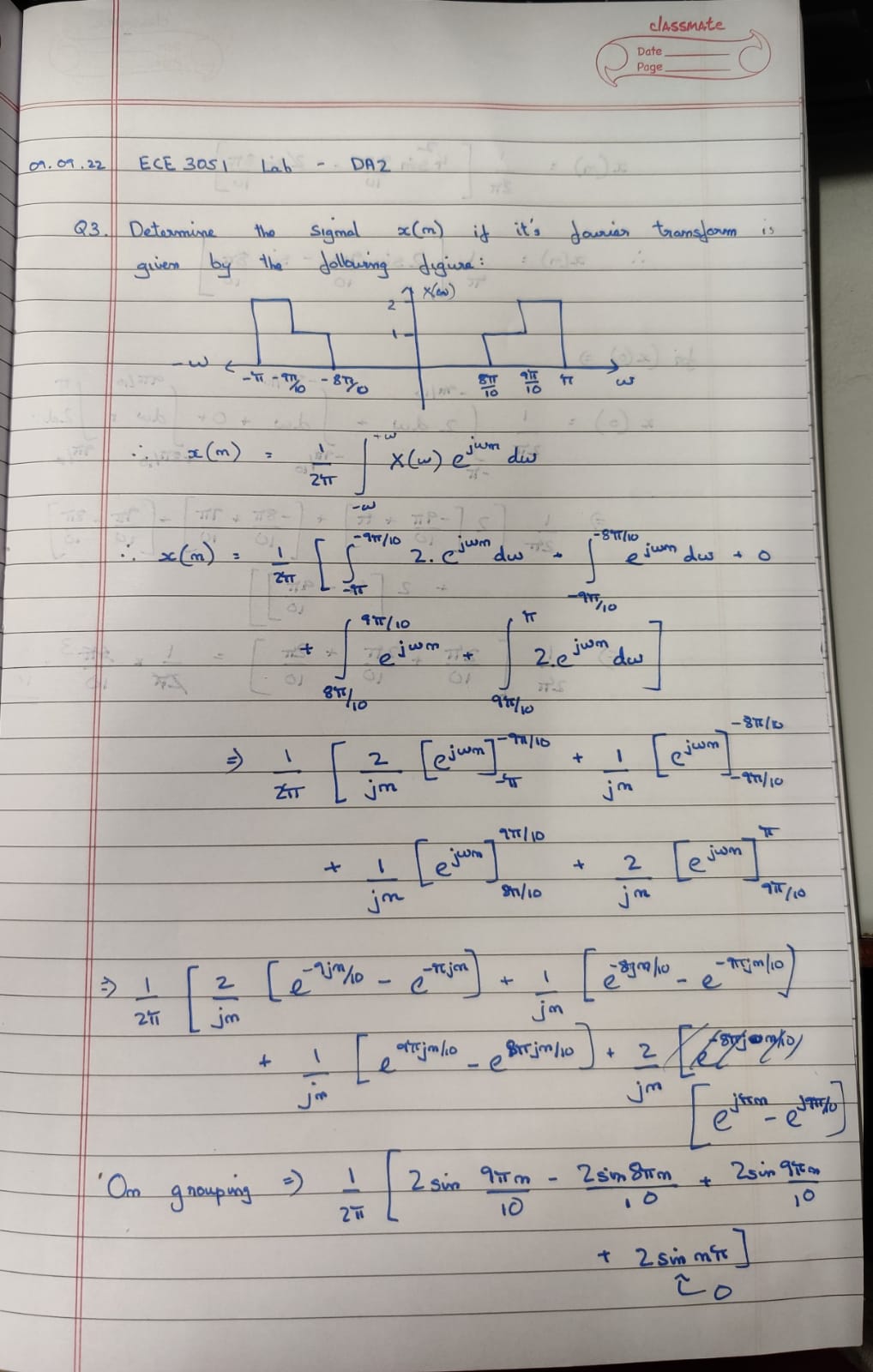
0

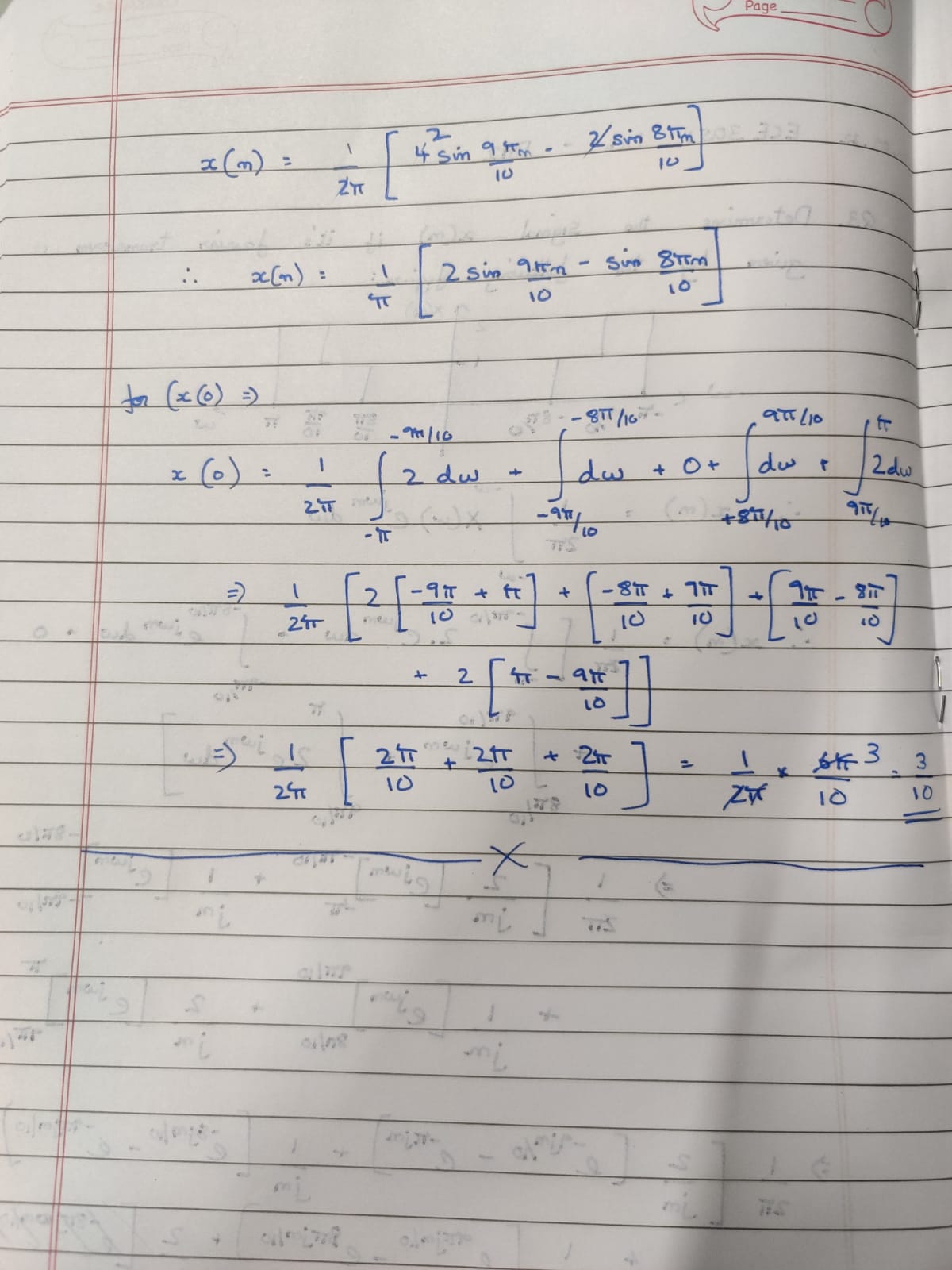
>>





**WORKING OUT:**

****

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_