Digital Signal Processing

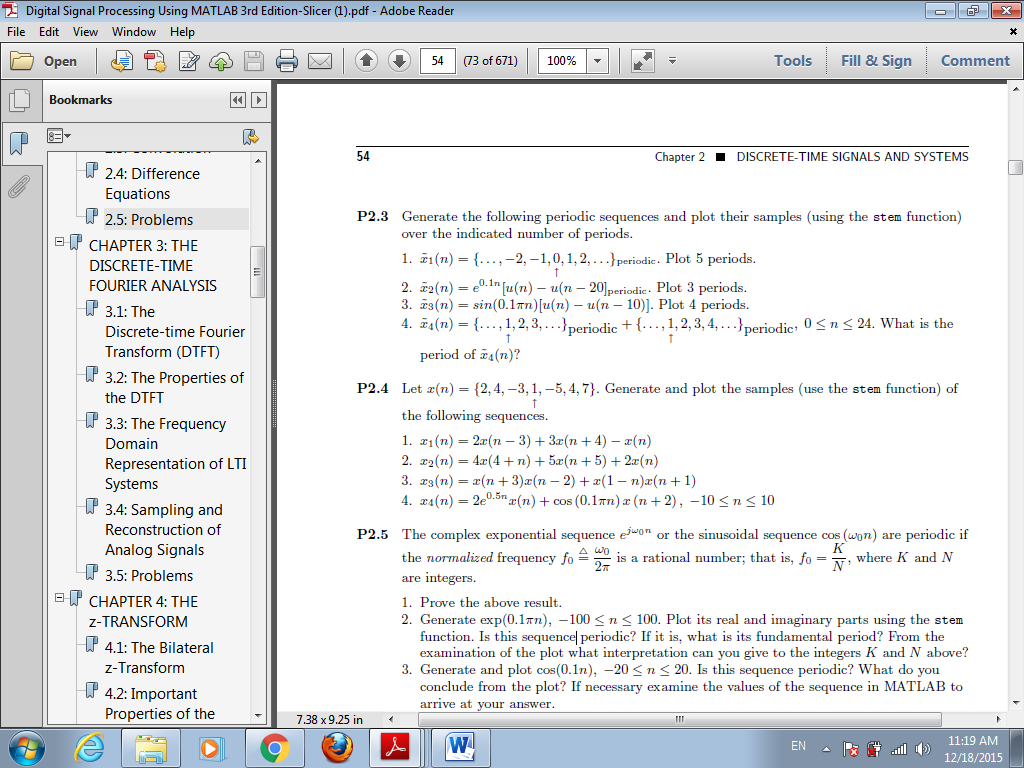
Final Project

Matlab Assignment

Aliaa Mohamed Abbas 3747

Eyad Sherif Nawar 4292

Problem Statement:



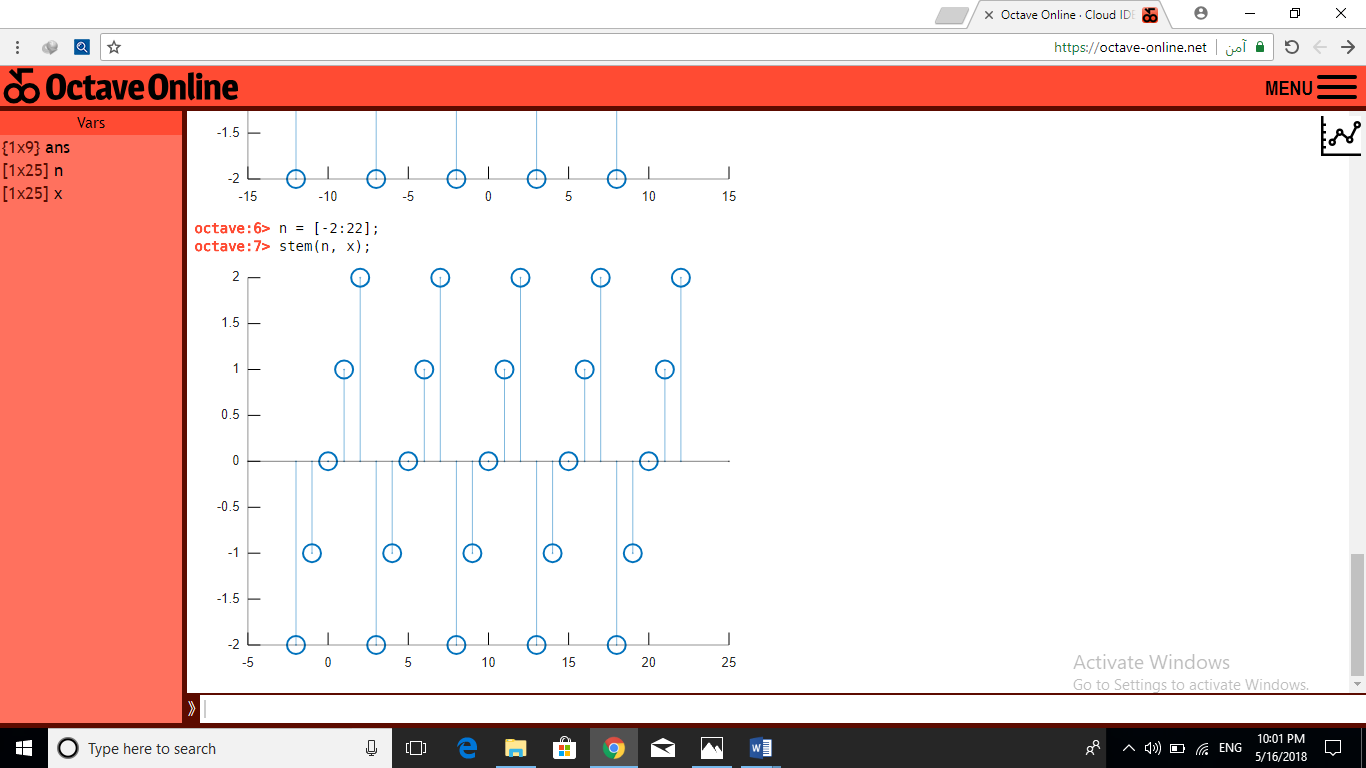
Matlab Code & Results:

**octave:2>** n = [-2:22]; %% Range of 5 periods

**octave:3>** x = [-2, -1, 0, 1, 2]; %% One period

**octave:4>** x = [x, x, x, x, x]; %% Repeat 5 times

**octave:5>** stem(n, x); %% Stem();



**octave:6>** n = [0:20]; %% range of step sequence

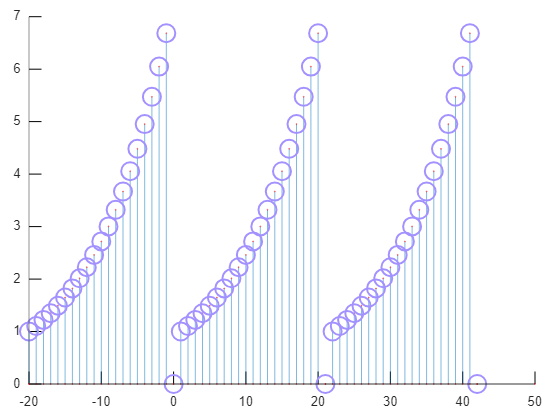
**octave:7>** xstep = stepseq(0,0,20)-stepseq(20,0,20);

**octave:8>** x = exp(0.1\*n).\*(xstep);

**octave:9>** x = [x, x, x]; %% Three Periods

**octave:10>** n = [-20:42]; %% Range for three Periods with padding

**octave:11>** stem(n,x);



**octave:13>** n = [0:10];

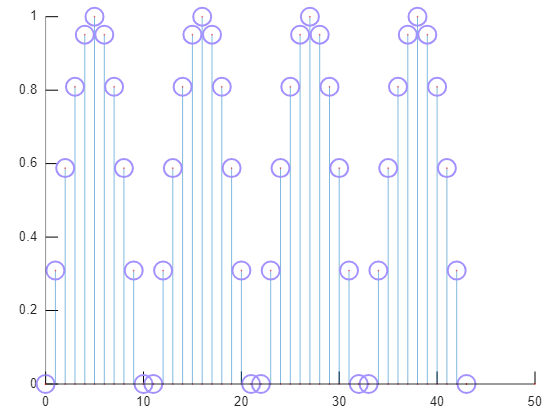
**octave:14>** xstep **=** stepseq(0, 0, 10)-stepseq(10, 0, 10)

**octave:15>** x = sin(0.1\*pi\*n).\*(xstep);

**octave:16>** x = [x, x, x, x]; %% Four Periods

**octave:17>** n = [0:43]; %% Range of Four Periods w/ padding

**octave:18>** stem(n,x);



**octave:19>** n = [0:24];

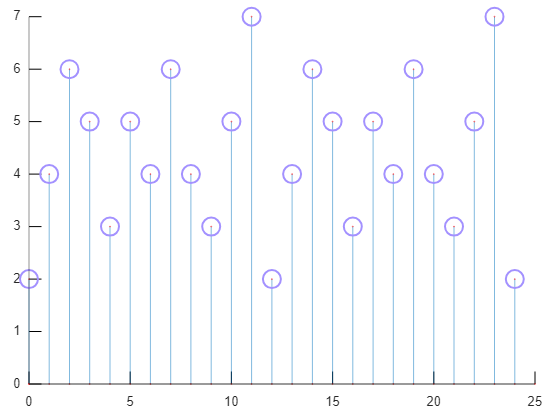
**octave:20>** x1 = [1, 2, 3];

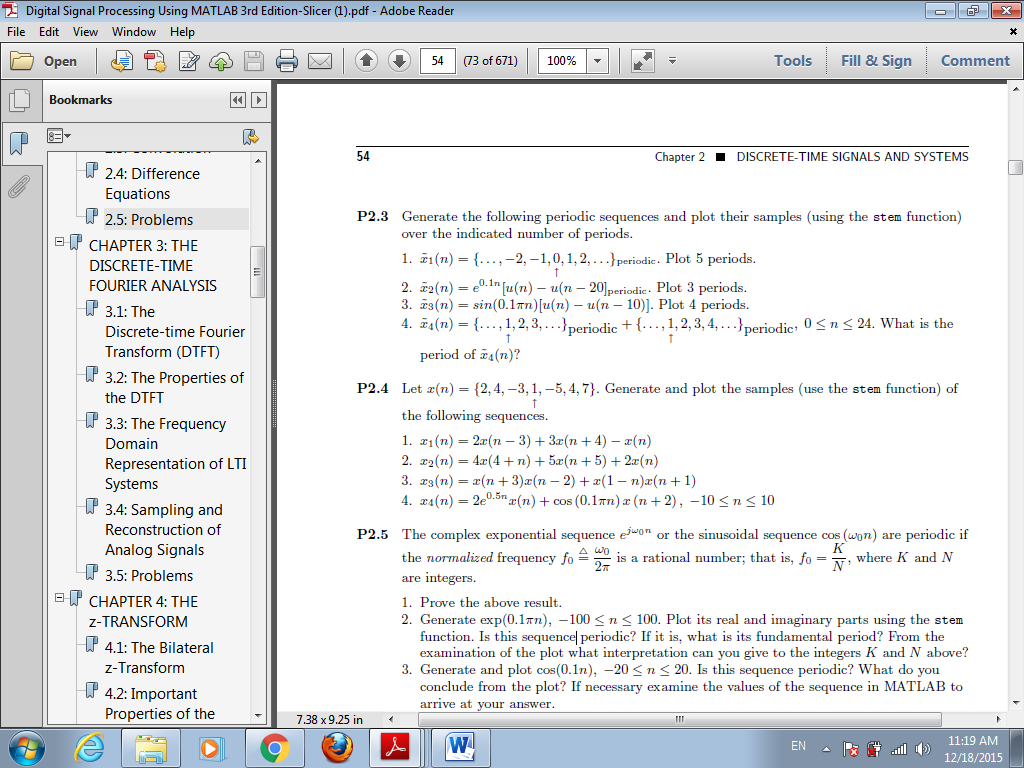
**octave:21>** x1 = [x1, x1, x1, x1, x1, x1, x1, x1, x1(1)];

**octave:22>** x2 = [1, 2, 3, 4];

**octave:23>** x2 = [x2, x2, x2, x2, x2, x2, x2(1)];

**octave:24>** xtot = x1+x2;

**octave:25>** stem(n, xtot); 



**octave:1>** x = [2, 4, -3, 1, -5, 4, 7];

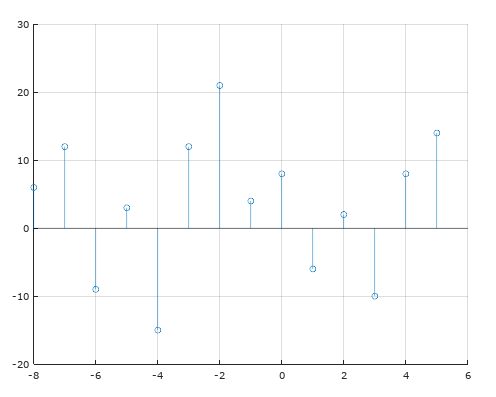
**octave:2>** n = [-3:3]

**octave:3>** [x1, n1] = sigshift(x, n, 3);

**octave:4>** [x2, n2] = sigshift(x, n, -4);

**octave:5>** [xtot, ntot] = sigadd(2\*x1, n1, 3\*x2, n2, -1\*x, n);

**octave:6>** stem(ntot, xtot);

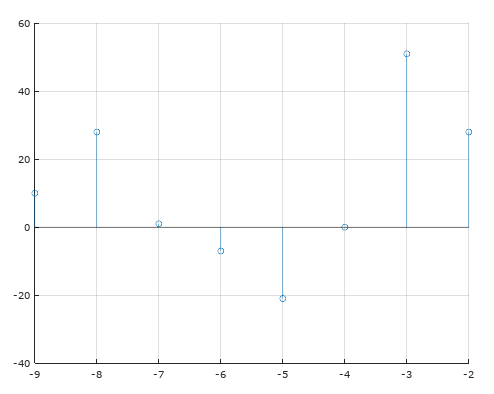


**octave:7>** [x1, n1] = sigshift(x, n, -4);

**octave:8>** [x2, n2] = sigshift(x, n, -5);

**octave:9>** [xtot, ntot] = sigadd(4\*x1, n1, 5\*x2, n2, 2\*x, n);

**octave:10>** stem(ntot, xtot);



**octave:16>** [term1, nt1] = sigshift(x, n, -3);

**octave:17>** [term2, nt2] = sigshift(x, n, 2);

**octave:18>** [x1, n1] = sigmult(term1, nt1, term2, nt2);

**octave:19>** [term1, nt1] = sigshift(x, n, -1);

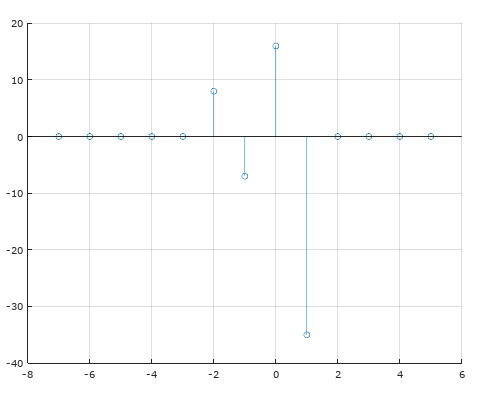
**octave:20>** [term2, nt2] = sigfold(x, n);

**octave:21>** [term2, nt2] = sigshift(term2, nt2, 1);

**octave:22>** [x2, n2] = sigmult(term1, nt1, term2, nt2);

**octave:23>** [xtot, ntot] = sigadd(x1, n1, x2, n2);

**octave:24>** stem(ntot, xtot);



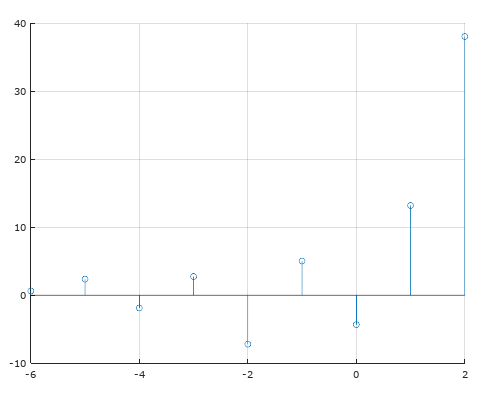
**octave:25>** x1 = 2\*exp(0.5\*n).\*x;

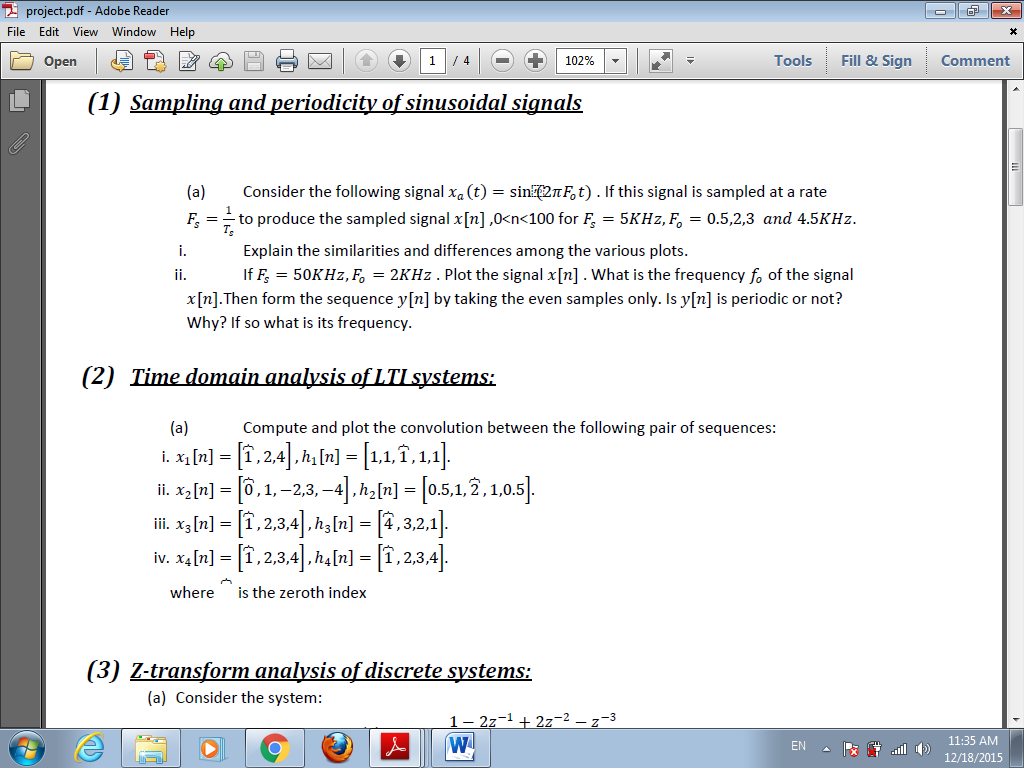
**octave:26>** [x2, n2] = sigshift(x, n, -2);

**octave:27>** x2 = cos(0.1\*pi\*n).\*x;

**octave:28>** [xtot, ntot] = sigadd(x1, n, x2, n2);

**octave:29>** stem(ntot, xtot);





**octave:1>** nx = [0:2];

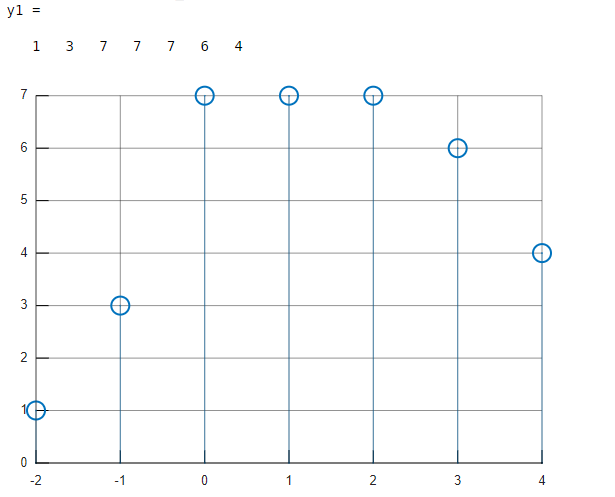
**octave:2>** x = [1, 2, 4];

**octave:3>** nh = [-2:2];

**octave:4>** h = [1, 1, 1, 1, 1];

**octave:5>** [y, ny] = conv\_m(x, nx, h, nh);

**octave:6>** stem(ny, y);



**octave:7>** nx = [0:4];

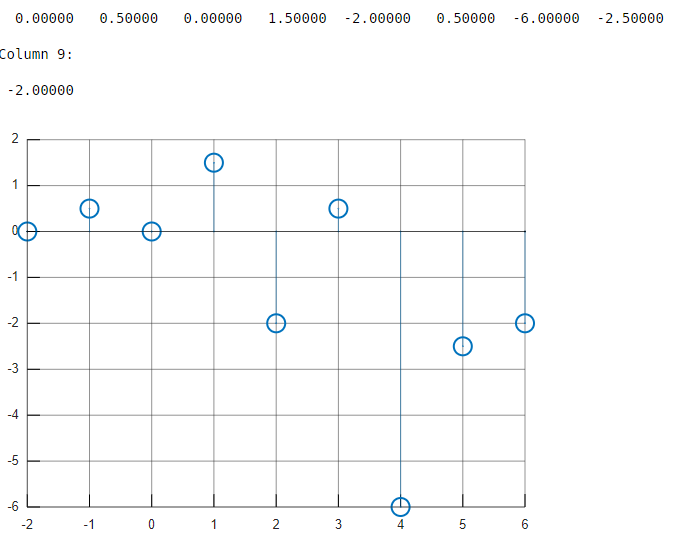
**octave:8>** x = [0, 1, -2, 3, -4];

**octave:9>** nh = [-2:2];

**octave:10>** h = [0.5, 1, 2, 1, 0.5];

**octave:11>** [y, ny] = conv\_m(x, nx, h, nh);

**octave:12>** stem(ny, y);



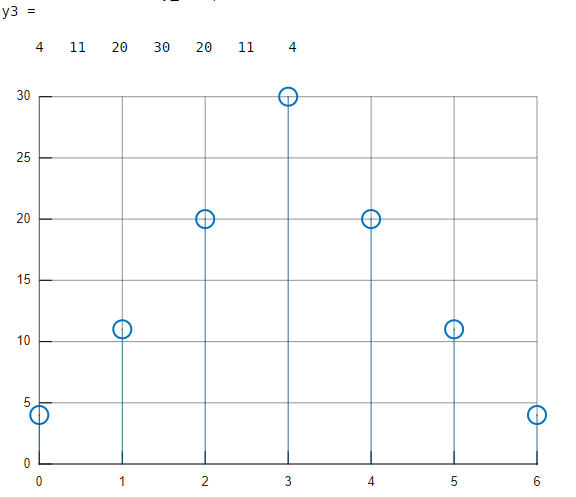
**octave:13>** n = [0:6];

**octave:14>** x = [1, 2, 3, 4];

**octave:15>** h = [4, 3, 2, 1];

**octave:16>** y = conv(x, h);

**octave:17>** stem(n, y);

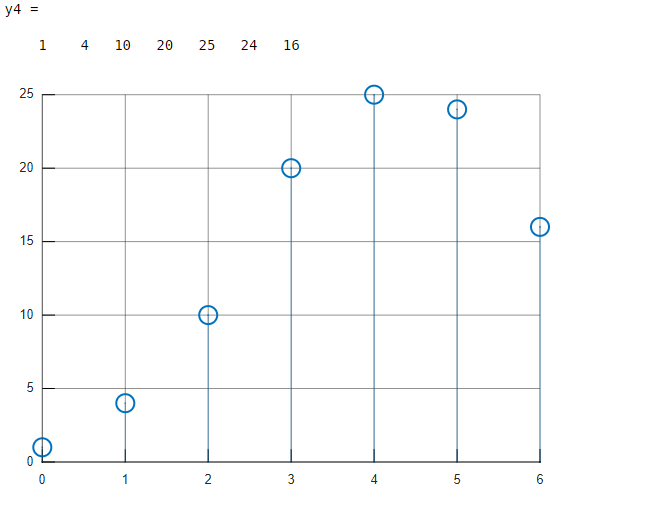


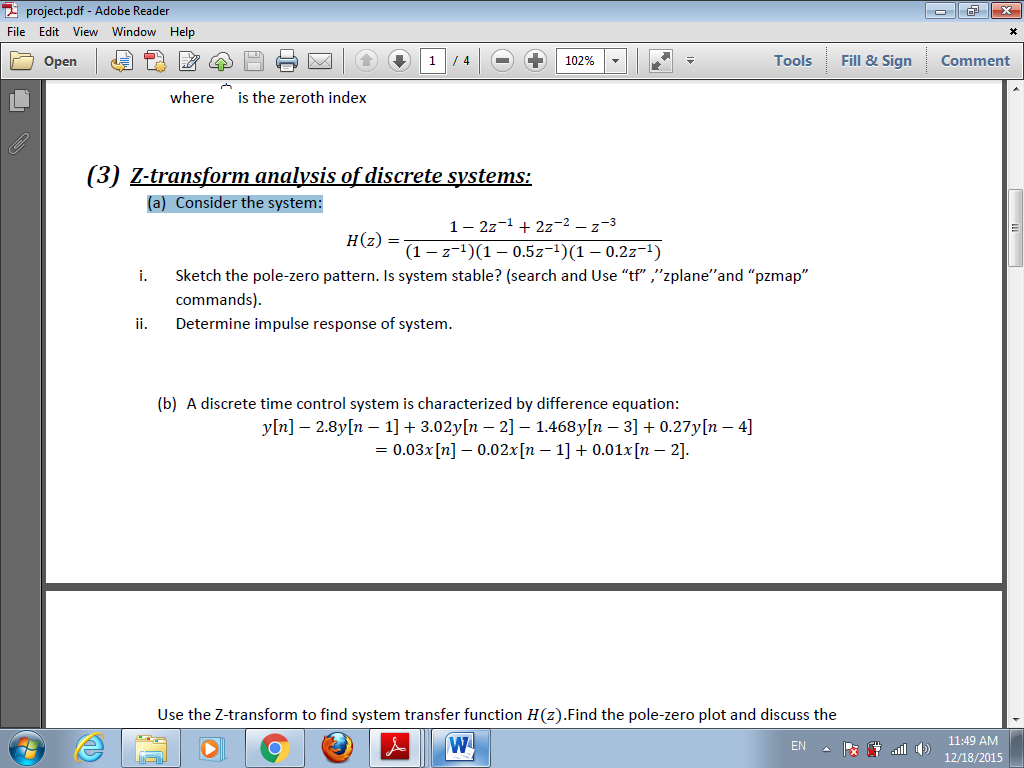
**octave:18>** x = [1, 2, 3, 4];

**octave:19>** h = [1, 2, 3, 4];

**octave:20>** y = conv(x, h);

**octave:21>** stem(n, y);





**octave:1>** num = [1 -2 2 -1];

**octave:2>** den1 = [1 -1];

**octave:3>** den2 = [1 -0.5];

**octave:4>** den3 = [1 -0.2];

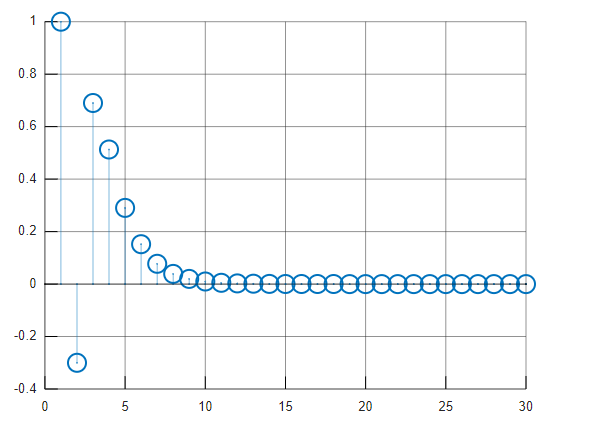
**octave:5>** den12 = conv(den1, den2);

**octave:6>** den = conv(den12, den3);

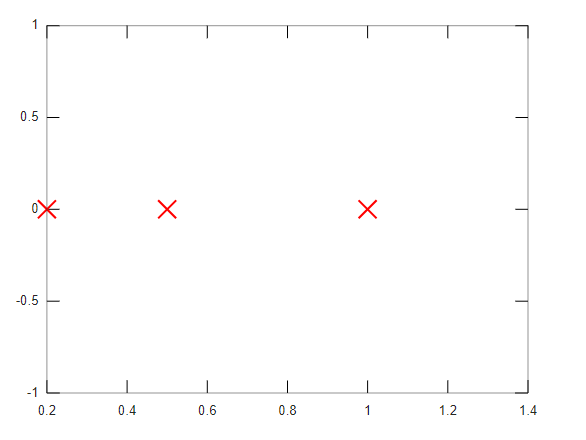
**octave:7>** H = tf(num, den,-1, ‘Variable’, ‘z^-1’);

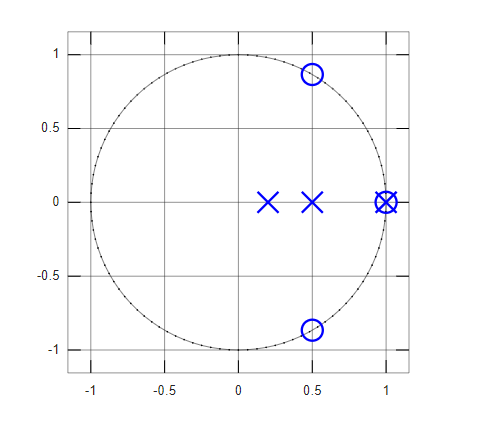
**octave:8>** h = filter(num,den, [1 zeros(1, 29)]);

**octave:9>** stem(h);

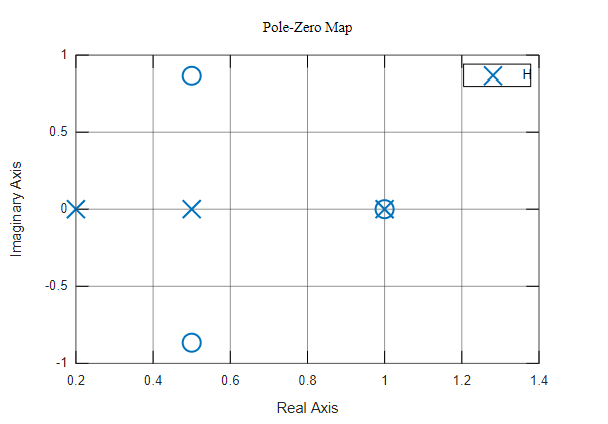


**octave:10>** zplane([num], [den]);





**octave:11>** pzmap(H);



𝑦 (𝑛) −2.8𝑦 (𝑛−1) +3.02𝑦 (𝑛−2) −1.468y( 𝑛−3) +0.27𝑦 (𝑛−4 )=

0.03x( n)−0.02𝑥 (𝑛−1 )+0.01𝑥 (𝑛−2) .

Use the Z-transform to find system transfer function H (𝑧).Find the pole-zero plot and discuss the stability. Then determine and plot the system output when

𝑥 (𝑛) = 5\*u(𝑛).

**octave:12>** num = [0.03 -0.02 0.01 0 0];

**octave:13>** den = [1 -2.8 3.02 -1.468 0.27];

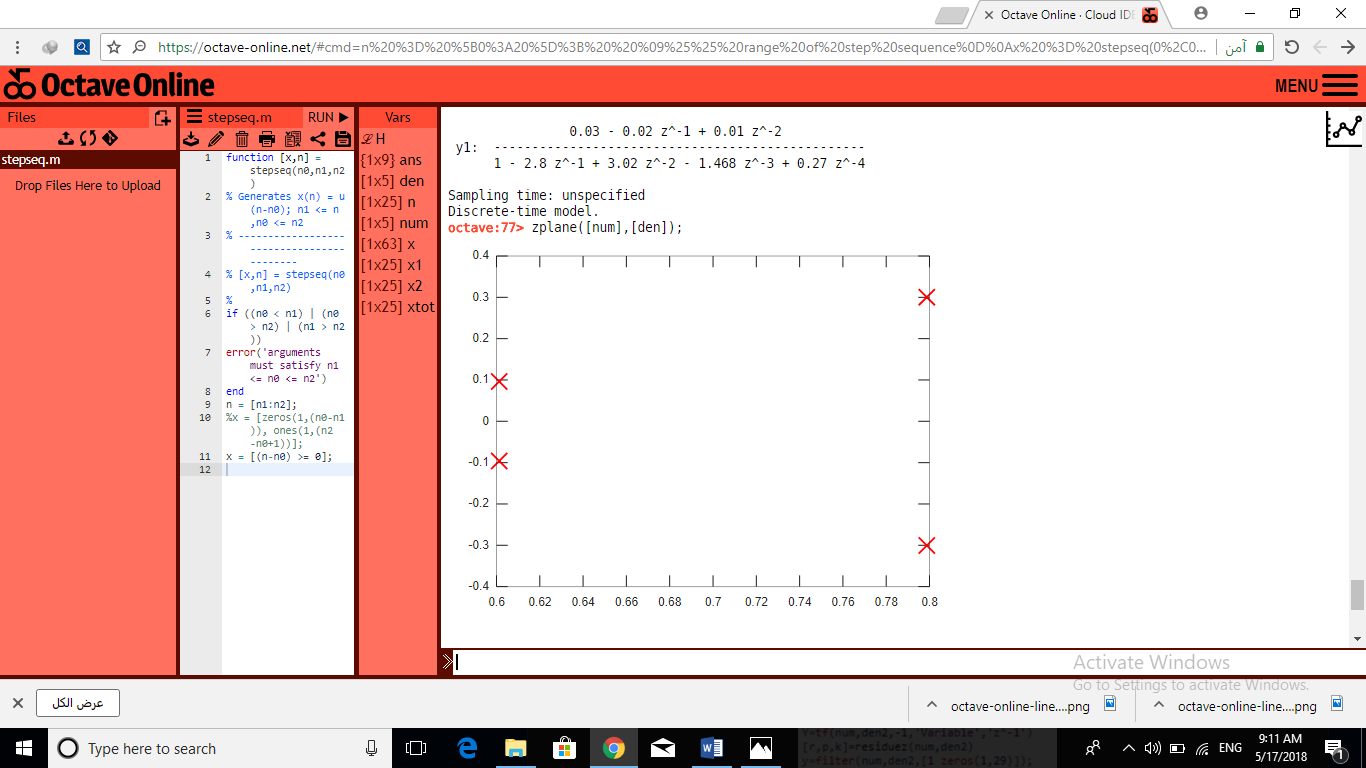
**octave:14>** H = tf(num, den, -1,'Variable', 'z^-1')

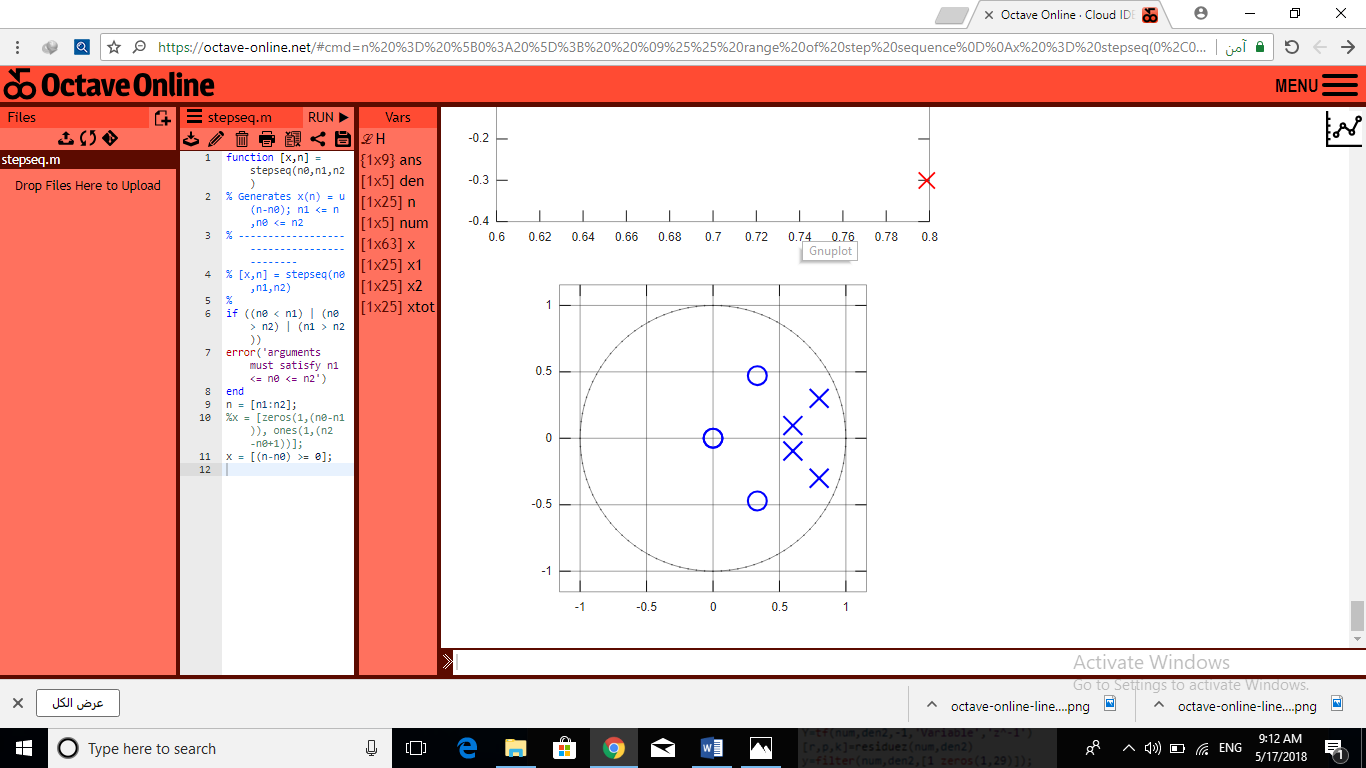
0.03 - 0.02 z^-1 + 0.01 z^-2

y1: -------------------------------------------------

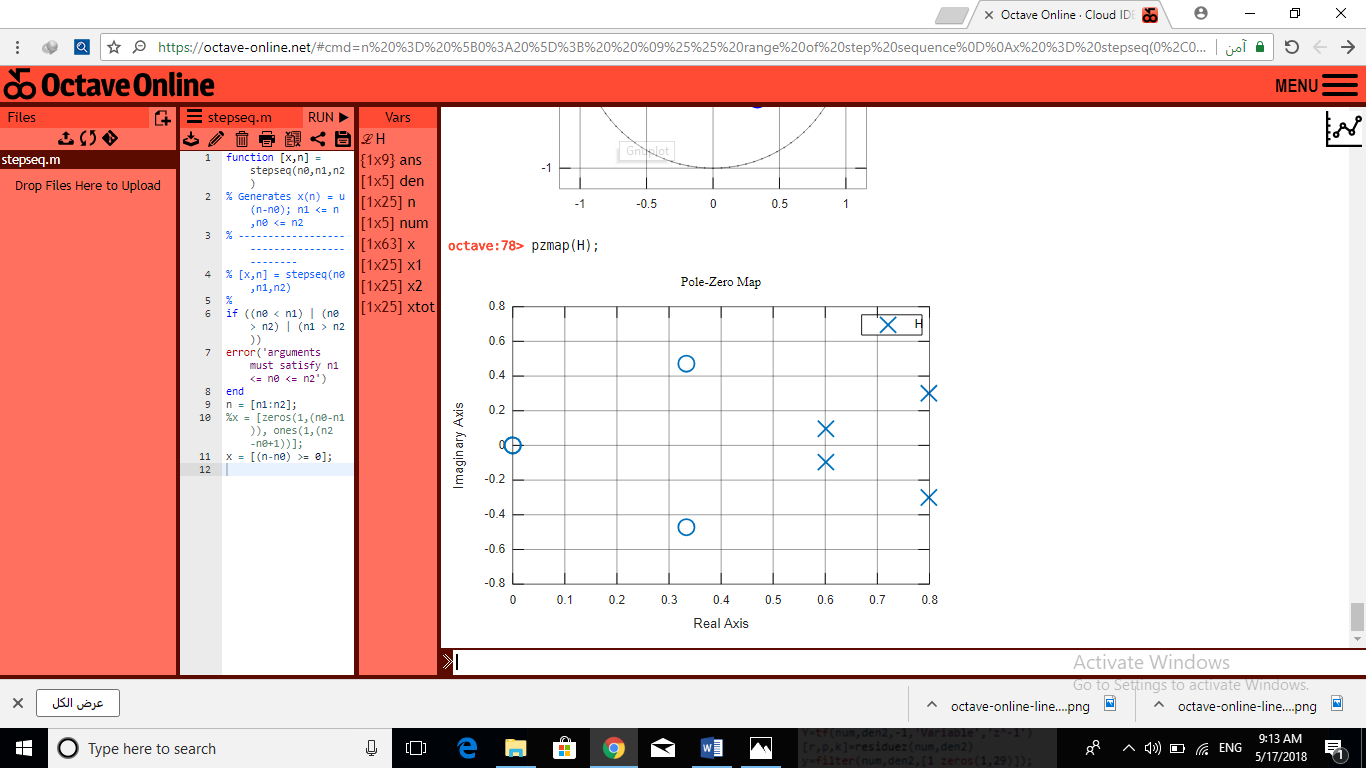
1 - 2.8 z^-1 + 3.02 z^-2 - 1.468 z^-3 + 0.27 z^-4

**octave:15>** zplane([num],[den]);





**octave:16>** pzmap(H);



**octave:17>** x = [1 -1];

**octave:18>** num = 5\*num;

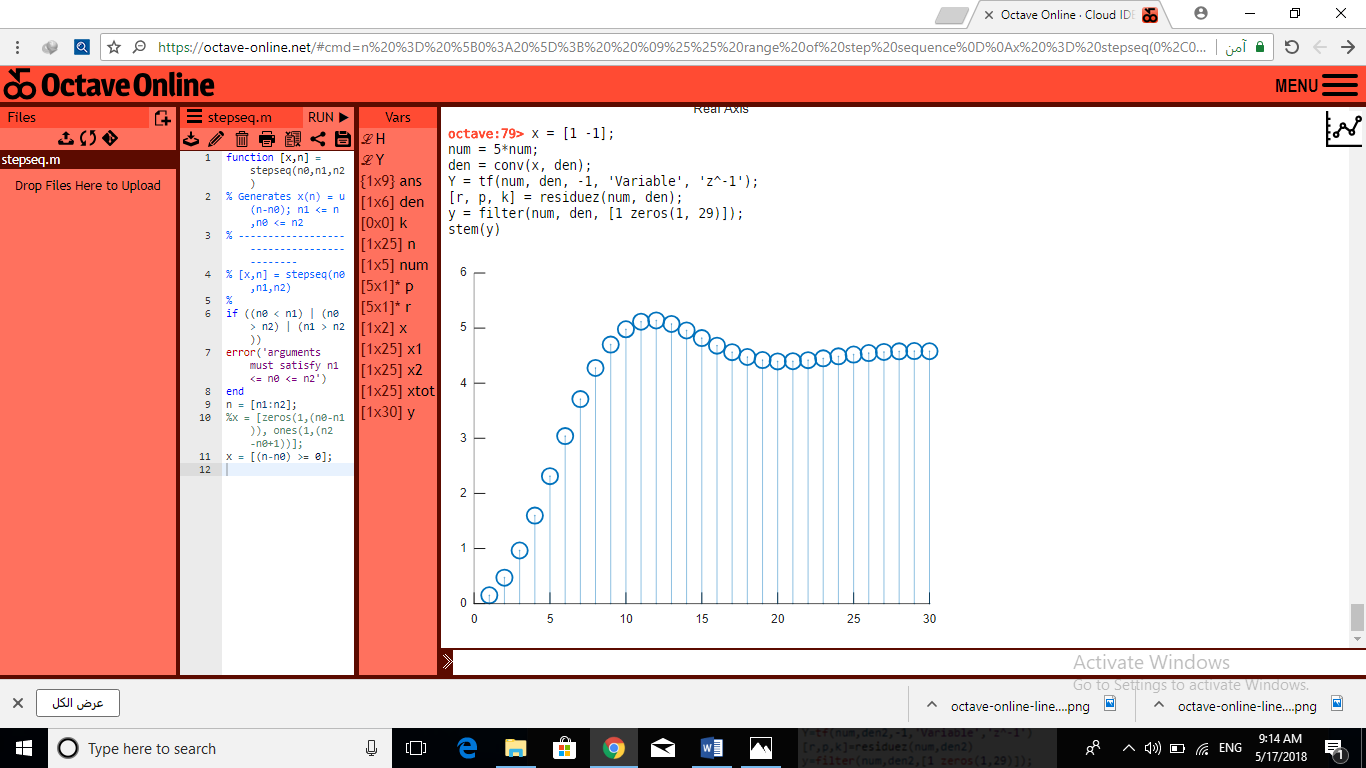
**octave:19>** den = conv(x, den);

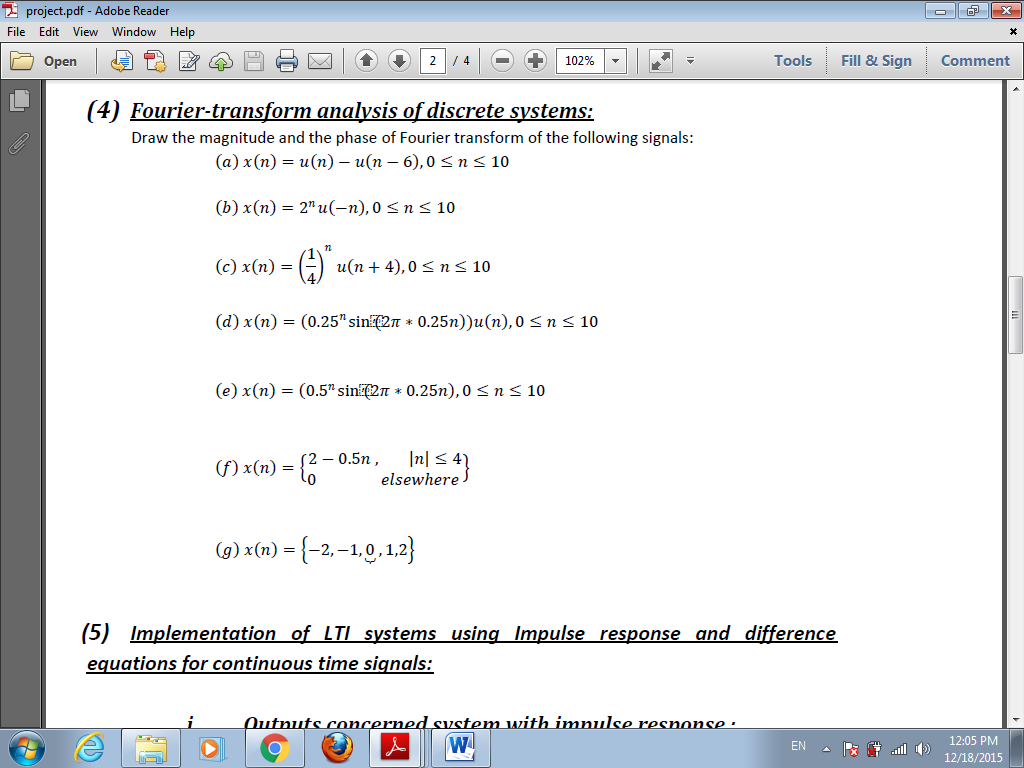
**octave:20>** Y = tf(num, den, -1, 'Variable', 'z^-1');

**octave:21>** [r, p, k] = residuez(num, den);

**octave:22>** y = filter(num, den, [1 zeros(1, 29)]);

**octave:23>** stem(y)





**octave:5>** n = [0:10];

**octave:6>** xa = stepseq(0, 0, 10)-stepseq(6, 0, 10);

**octave:7>** xstep = stepseq(0, 0, 10);

**octave:8>** xb = (2.^n).\*(fliplr(xstep));

**octave:9>** xc = (0.25.^n).\*(xstep);

**octave:10>** xd = ((0.25).^n).\*(sin(2\*pi\*(0.25).\*n)).\*(xstep);

**octave:11>** xe = ((0.5).^n).\*(sin(2\*pi\*(0.25).\*n));

**octave:12>** n = [-4:4];

**octave:13>** xf = 2-((0.5).\*n);

**octave:14>** n = [-2:2];

**octave:15>** xg = n;

%% For Every Signal we repeat the following commands:

**octave:16>** ftA = fft(xa);

**octave:17>** magA = abs(ftA);

**octave:18>** phaseA = unwrap(angle(ftA));

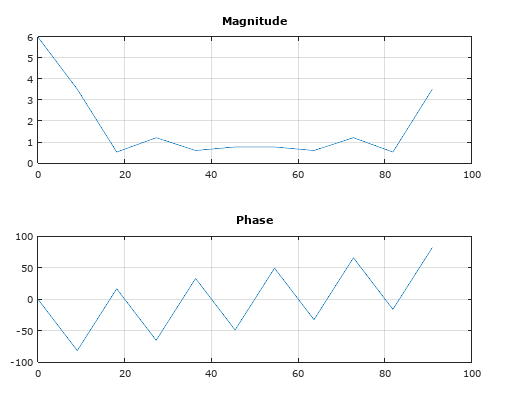
**octave:19>** frequencyA = (0:length(ftA)-1)\*100/length(ftA);

**octave:20>** plot(frequencyA, magA);

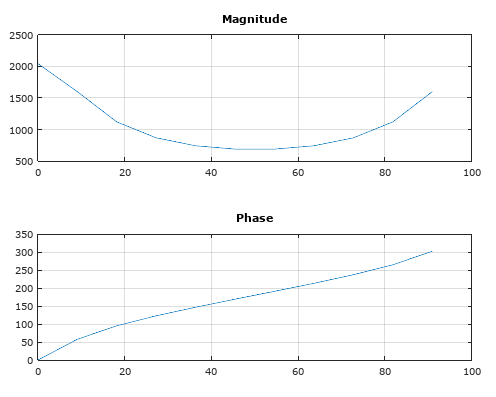
**octave:21>** plot(frequencyA, phaseA\*180/pi);

%%

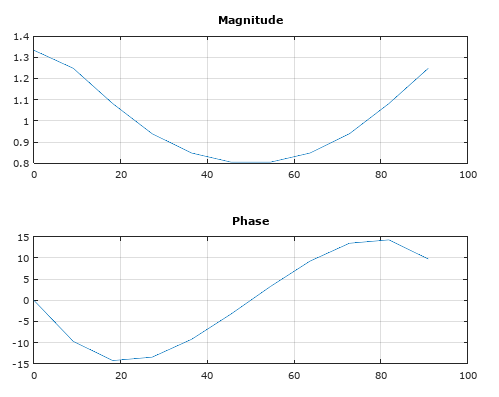
A)



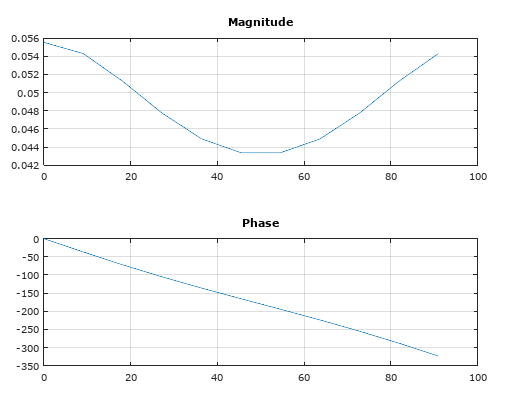
B)

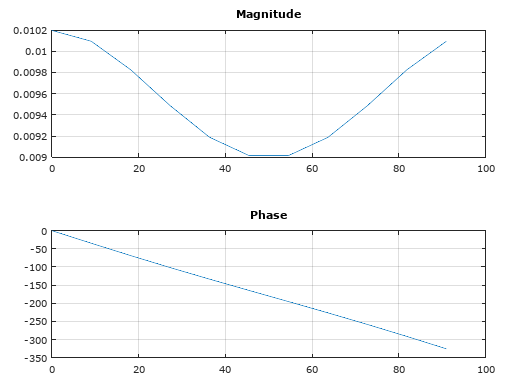


C)



D)



E)

f

