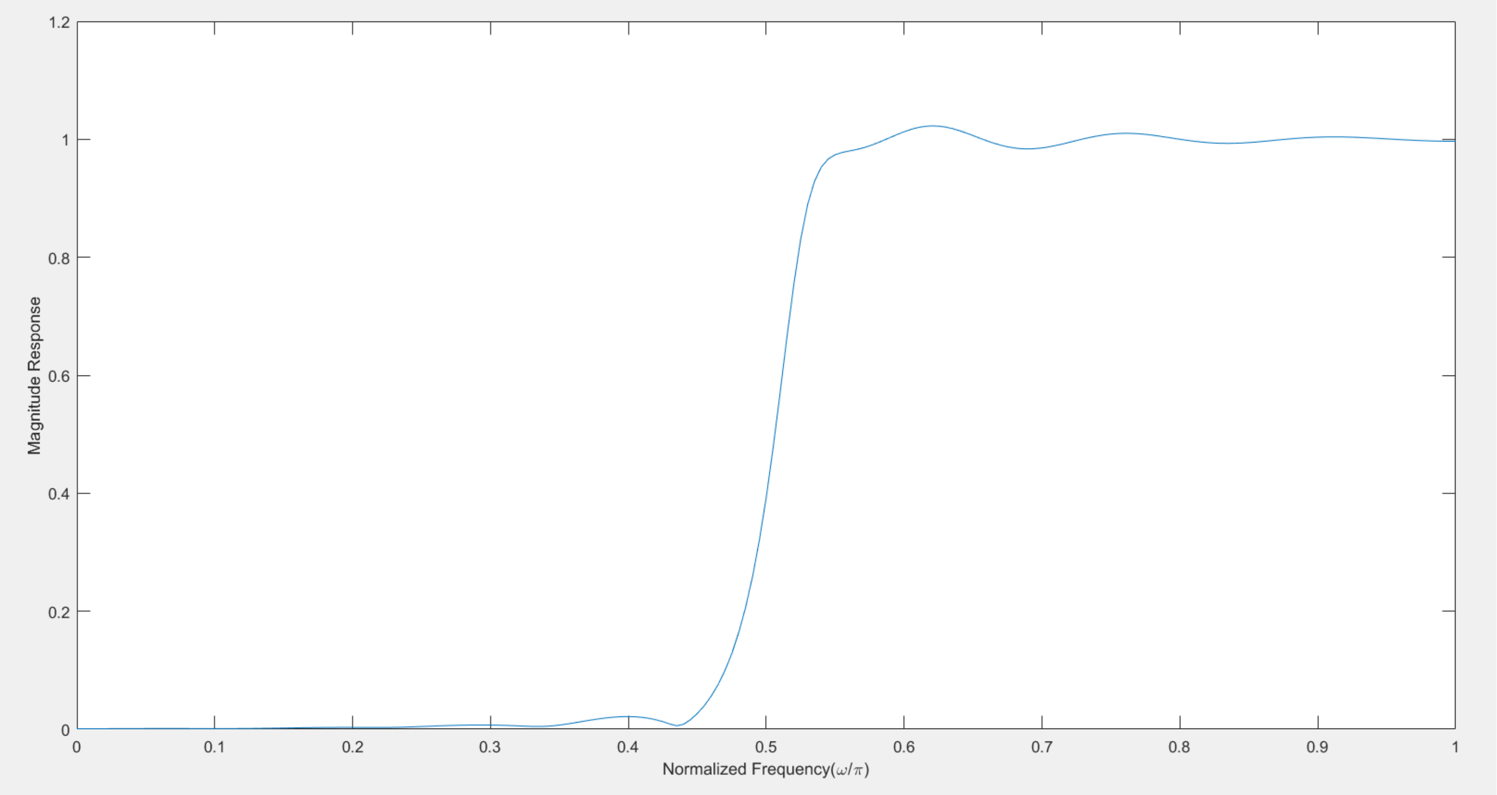
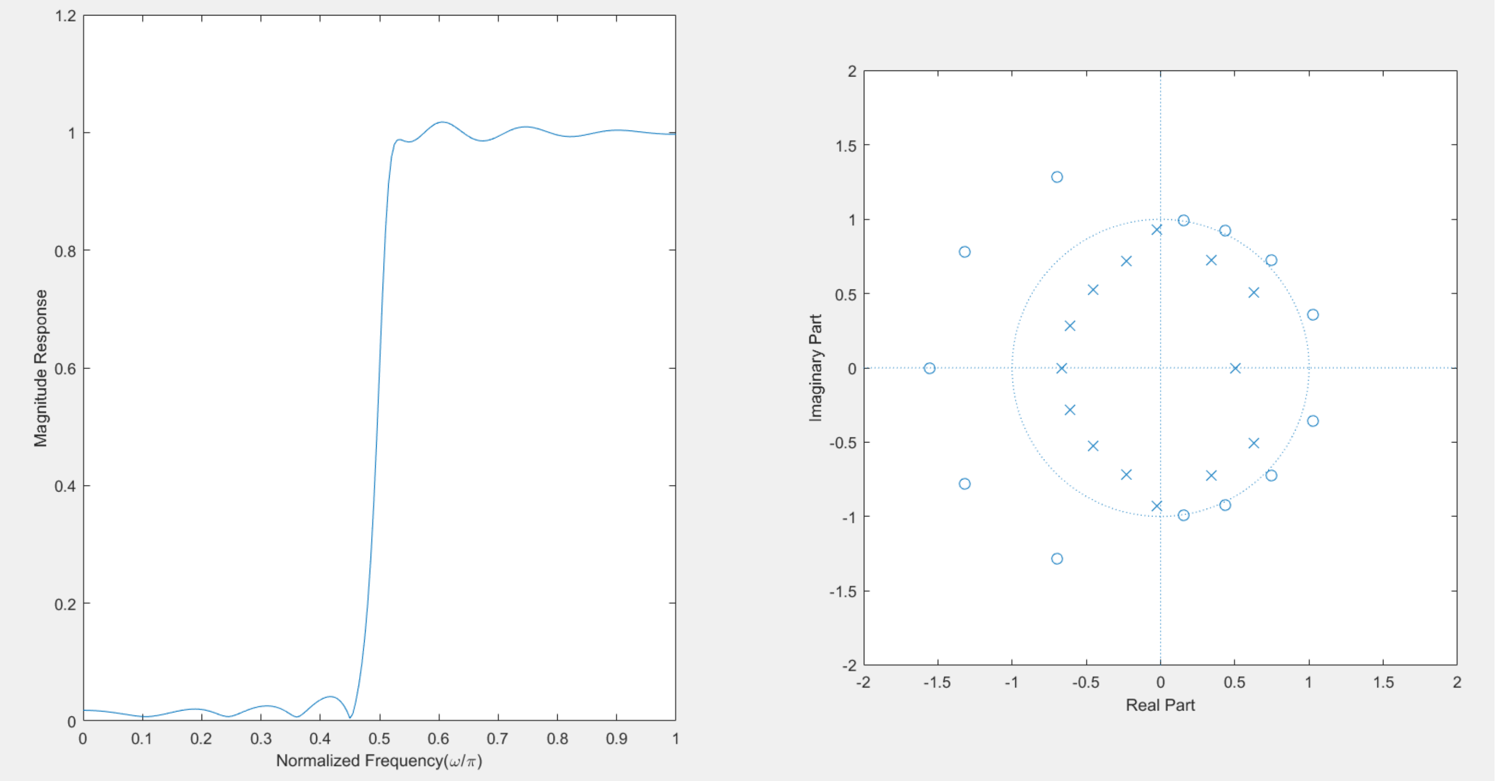
```
%                          DSP_Project_3
%
% Design of IIR_Digital_Filters
%
clear all;   % clear workspace
clc;        % clear command window
n = 14;
m = 14;
tau = 12;
ws = 0.475*pi;
wp = 0.525*pi;
delta = 0.00001;
epsilon = 0.00001;
pointw = 10000; % number of total sampling points ?
M = 100;
%
%
deltaw = pi/pointw; % smapling distance
deltaM = pi/M;     % smapling distance
nm = n + m + 1;
x = zeros(nm, 1);
j = 0 + i; % define i = j = (-1)^0.5
%
%
pointpass = 0; % number of sampling points
pointstop = 0;
PASS = zeros(0, 1); % store the answers in a vector
STOP = zeros(0, 1);
for iw = 0 : pointw
    w = iw * deltaw;
    if w >= wp
        pointpass = pointpass + 1;
        PASS(pointpass) = w;
    elseif w <= ws
        pointstop = pointstop + 1;
        STOP(pointstop) = w;
    end
end
Whp = ones(pointpass, 1); % W hat p
Whs = ones(pointstop, 1); % W hat s
%
%
delta_x = 10000;
iter = 0;
while delta_x > epsilon
    iter = iter + 1
    xk_1 = x;
```

```
r = zeros(nm, 1);
Qs = zeros(nm, nm);
Qp = zeros(nm, nm);
for ip = 1 : pointpass % r's integral
    w = PASS(ip);
    ew = zeros(nm, 1);
    for in = 1 : n
        ew(in) = exp(-j*tau*w) * exp(-j*in*w);
    end
    for im = 0 : m
        ew(im+in+1) = -exp(-j*im*w) ;
    end
    r = r + Whp(ip)*real(exp(j*tau*w)*ew);
    Qp = Qp + Whp(ip)*ew*ew';
end
r = (pi - wp) * r/pointpass;
Qp = real((pi - wp) * Qp/pointpass);
%
%
for is = 1 : pointstop
    w = STOP(is);
    ew = zeros(nm, 1);
    for im = 0 : m
        ew(im+in+1) = -exp(-j*im*w) ;
    end
    Qs = Qs + Whs(is)*ew*ew';
end
Qs = real(ws * Qs/pointstop);
Q = Qp + Qs;
%
%
% finding B & X
B = zeros(M+1, nm);
for im = 0 : M
    w = im * deltaM;
    for in = 1 : n
        B(im+1, in) = -cos(in*w);
    end
end
d = (1-delta) * ones(M+1, 1);
%
%
% MATLAB tools - quadratic programming
x = quadprog(Q, r, B, d);
a = x(1:n);
b = x(n+1:nm);
FR = abs(freqz(b, [1;a], 0:pi/200:pi));
```

```
    plot(0:1/200:1, FR);
    xlabel('Normalized Frequency(\omega/\pi)');
    ylabel('Magnitude Response');
    pause;
    %
    %
    % pass-band
    for ip = 1: pointpass
        w = PASS(ip);
        eaw = zeros(n, 1);
        for in = 1 : n
            eaw(in) = exp(-j*in*w);
        end
        Whp(ip) = 1 / (abs(1+a'*eaw))^2;
    end
    % stop-band
    for is = 1: pointstop
        w = STOP(is);
         eaw = zeros(n, 1);
        for in = 1 : n
            eaw(in) = exp(-j*in*w);
        end
        Whs(is) = 1 / (abs(1+a'*eaw))^2;
    end
    %
    %
    delta_x = norm(x - xk_1) / norm(x)
end
%
%
FR = abs(freqz(b, [1;a], 0:pi/200:pi));
subplot(1,2,1);
plot(0:1/200:1, FR);
xlabel('Normalized Frequency(\omega/\pi)');
ylabel('Magnitude Response');
%
subplot(1,2,2);
zplane(b',[1;a]');
axis([-2,2,-2,2]);
```

```
% IIR_Digital_Filters
```


Magnitude Response
Normalized Frequency(ω/π)


Magnitude Response
Normalized Frequency(ω/π)
Imaginary Part
Real Part