```
%                           DSP_Project_10
%
% Design_of_3D_FIR_Cone_type_Filter
%
clear all;   % clear workspace
clc;        % clear command window
N = 20;     % -20 ~ 20
theta = 55;
wt = 0.2*pi;
pointw = 200;
%
%
theta_w = 55 * pi / 180;
if theta <= 45
   wup = pi;
else
    wup = pi / tan(theta_w);
end
deltaw = wup / pointw; % 1D
deltaw1 = pi / pointw; % 2D
%
%
r = zeros(3, 1);
Q = zeros(3, 3);
for iw = 0:pointw
   w1 = iw * deltaw;
   w2 = w1 * tan(theta_w);
   c = [1, -1-cos(w1)*cos(w2), -cos(w1)-cos(w2)]';
   r = r -2*cos(w1)*c;
   Q = Q + c*c';
end
r = r / (pointw + 1);
Q = Q / (pointw + 1);
T = -0.5 * inv(Q)  * r;
%
%
wc = acos(T(1));
t11 = T(2);
t01 = T(3);
t00 = t11;
t10 = 1 + t01;
%
% Design of 1D prototype lowpass filters
%
wp = wc;
ws = wp + wt;
P = zeros(N+1, 1);
```

```
Qs = zeros(N+1, N+1);
Qp = zeros(N+1, N+1);
pointp = 0;
points = 0;
for iw = 0:pointw
   w = iw * deltaw1;
   if w <= wp
      pointp = pointp + 1;
      c = zeros(N+1, 1);
      for in = 0:N
         c(in+1) = cos(in*w);
      end
      P = P - 2*c;
      Qp = Qp + c*c';
   elseif w >= ws
      points = points + 1;
      c = zeros(N+1, 1);
      for in = 0:N
         c(in+1) = cos(in*w);
      end
      Qs = Qs + c*c';
   end
end
P = wp * P / pointp;
Qp = wp * Qp / pointp;
Qs = (pi - ws) * Qs / points;
A = -0.5 * inv(Qp+Qs) * P;
%
h = zeros(2*N+1, 1);
h(N+1) = A(1);
h(1:N) = 0.5 * flipud(A(2:N+1)); % flip up down
h(N+2:2*N+1) = 0.5 * A(2:N+1);
% FR = abs(freqz(h, 1, 0:deltaw1:pi));
% subplot(1, 2, 1);
% plot(0:deltaw1/pi:1,FR);
% xlabel('Normalized Frequency');
% ylabel('Magnitude Response');
% title('1D Prototype Filter');
%
%
Tab = zeros(N+1, N+1);
Tab(1, 1) = 1;
Tab(2, 2) = 1;
for in = 2:N
   Tab(in+1, 2:N+1) = 2 * Tab(in, 1:N);
   Tab(in+1,:) = Tab(in+1,:) - Tab(in-1,:);
end
```

```
B = zeros(N+1, 1);
for in = 0:N
    B = B + A(in+1) * Tab(in+1,:)';
end
F = zeros(3, 3);
F(1, 1) = 0.25 * t11;
F(1, 2) = 0.5 * t10;
F(1, 3) = 0.25 * t11;
F(2, 1) = 0.5 * t01;
F(2, 2) = t00;
F(2, 3) = 0.5 * t01;
F(3, 1) = 0.25 * t11;
F(3, 2) = 0.5 * t10;
F(3, 3) = 0.25 * t11;
%
%
h2 = zeros(2*N+1, 2*N+1);
h2(N+1, N+1) = B(1);
h2(N:N+2, N:N+2) = h2(N:N+2, N:N+2) + B(2)*F;
FN = F;
for in = 2:N
    FN = conv2(FN, F);
    h2(N+1-in:N+1+in, N+1-in:N+1+in) = h2(N+1-in:N+1+in, N+1-
in:N+1+in) + B(in+1) * FN;
end
%
%
FR2 = abs(freqz2(h2, -1:2/100:1, -1:2/100:1));
XX = zeros(101, 101);
for i = 0:100
   XX(:, i+1) = (-1:2/100:1)';
end
YY = XX';
subplot(2, 2, 1);
plot3(XX, YY, FR2);
axis([-1, 1, -1, 1, 0, 1.1]);
xlabel('\omega_1');
ylabel('\omega_2');
zlabel('Magnitude Response');
title('2D FIR Fan Type Filter');
%
wc_fan = wc;
h_fan = h;
B_fan = B;
%
% Design of 2D  Circular Low-pass Filter
%
```

```
wr = wup;
%
%
deltaw = 0.5 * pi / pointw; % 1D
deltaw1 = pi / pointw; % 2D
%
%
r = zeros(2, 1);
Q = zeros(2, 2);
for iw = 0:pointw
   w1 = wr * cos(iw * deltaw);
   w2 = wr * sin(iw * deltaw);
   c = [1, 1-cos(w1)*cos(w2)]';
   r = r -2*(0.5*cos(w1) + 0.5*cos(w2))*c;
   Q = Q + c*c';
end
r = r / (pointw + 1);
Q = Q / (pointw + 1);
T = -0.5 * inv(Q) * r;
%
%
wc = acos(T(1));
r11 = T(2);
r00 = -r11;
r10 = 0.5;
r01 = 0.5;
%
% Design of 1D prototype lowpass filters
%
wp = wc;
ws = wp + wt;
P = zeros(N+1, 1);
Qs = zeros(N+1, N+1);
Qp = zeros(N+1, N+1);
pointp = 0;
points = 0;
for iw = 0:pointw
   w = iw * deltaw1;
   if w <= wp
      pointp = pointp + 1;
      c = zeros(N+1, 1);
      for in = 0:N
         c(in+1) = cos(in*w);
      end
      P = P - 2*c;
      Qp = Qp + c*c';
   elseif w >= ws
```

```
        points = points + 1;
        c = zeros(N+1, 1);
        for in = 0:N
            c(in+1) = cos(in*w);
        end
        Qs = Qs + c*c';
    end
end
P = wp * P / pointp;
Qp = wp * Qp / pointp;
Qs = (pi - ws) * Qs / points;
A = -0.5 * inv(Qp+Qs) * P;
%
h = zeros(2*N+1, 1);
h(N+1) = A(1);
h(1:N) = 0.5 * flipud(A(2:N+1)); % flip up down
h(N+2:2*N+1) = 0.5 * A(2:N+1);
% FR = abs(freqz(h, 1, 0:deltaw1:pi));
% subplot(1, 2, 1);
% plot(0:deltaw1/pi:1,FR);
% xlabel('Normalized Frequency');
% ylabel('Magnitude Response');
% title('1D Prototype Filter');
%
%
Tab = zeros(N+1, N+1);
Tab(1, 1) = 1;
Tab(2, 2) = 1;
for in = 2:N
    Tab(in+1, 2:N+1) = 2 * Tab(in, 1:N);
    Tab(in+1,:) = Tab(in+1,:) - Tab(in-1,:);
end
B = zeros(N+1, 1);
for in = 0:N
    B = B + A(in+1) * Tab(in+1,:)';
end
F = zeros(3, 3);
F(1, 1) = 0.25 * r11;
F(1, 2) = 0.5 * r10;
F(1, 3) = 0.25 * r11;
F(2, 1) = 0.5 * r01;
F(2, 2) = r00;
F(2, 3) = 0.5 * r01;
F(3, 1) = 0.25 * r11;
F(3, 2) = 0.5 * r10;
F(3, 3) = 0.25 * r11;
%
```

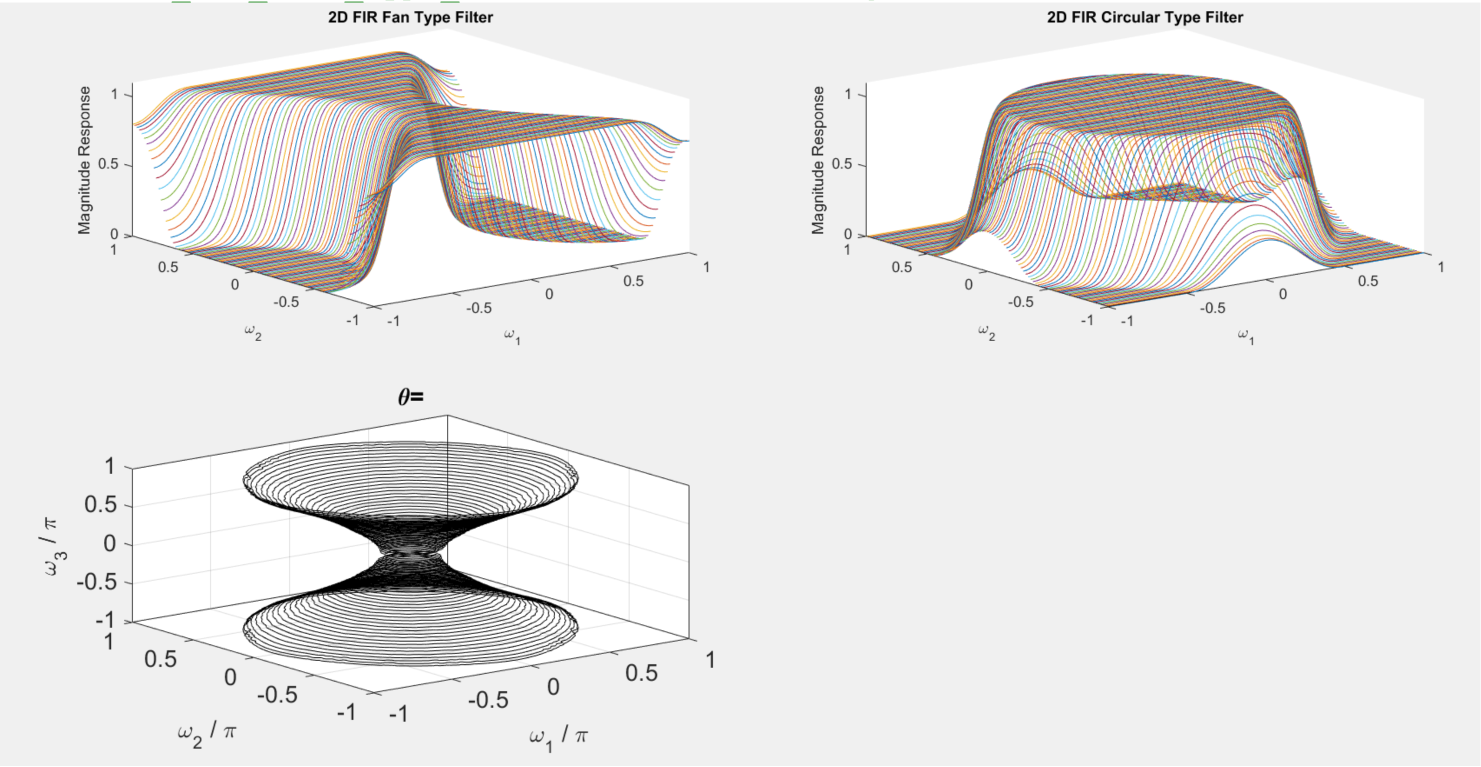
```
%
h2 = zeros(2*N+1, 2*N+1);
h2(N+1, N+1) = B(1);
h2(N:N+2, N:N+2) = h2(N:N+2, N:N+2) + B(2)*F;
FN = F;
for in = 2:N
    FN = conv2(FN, F);
    h2(N+1-in:N+1+in, N+1-in:N+1+in) = h2(N+1-in:N+1+in, N+1-
in:N+1+in) + B(in+1) * FN;
end
%
%
FR2 = abs(freqz2(h2, -1:2/100:1, -1:2/100:1));
XX = zeros(101, 101);
for i = 0:100
   XX(:, i+1) = (-1:2/100:1)';
end
YY = XX';
subplot(2, 2, 2);
plot3(XX, YY, FR2);
axis([-1, 1, -1, 1, 0, 1.1]);
xlabel('\omega_1');
ylabel('\omega_2');
zlabel('Magnitude Response');
title('2D FIR Circular Type Filter');
%
%
wc_cir = wc;
h_cir = h;
%
%
%
%
t000 = t00 + t10 * r00;
t100 = t10 * r10;
t010 = t10 * r01;
t001 = t01 + t11 * r00;
t110 = t10 * r11;
t101 = t11 * r10;
t011 = t11 * r01;
t111 = t11 * r11;
%
%
F = zeros(3, 3, 3);
F(1, 1, 1) = 0.125 * t111;
F(1, 1, 2) = 0.25 * t110;
F(1, 1, 3) = 0.125 * t111;
```

```
F(1, 2, 1) = 0.25 * t101;
F(1, 2, 2) = 0.5 * t100;
F(1, 2, 3) = 0.25 * t101;
F(1, 3, 1) = 0.125 * t111;
F(1, 3, 2) = 0.25 * t110;
F(1, 3, 3) = 0.125 * t111;
%
F(2, 1, 1) = 0.25 * t011;
F(2, 1, 2) = 0.5 * t010;
F(2, 1, 3) = 0.25 * t011;
F(2, 2, 1) = 0.5 * t001;
F(2, 2, 2) = t000;
F(2, 2, 3) = 0.5 * t001;
F(2, 3, 1) = 0.25 * t011;
F(2, 3, 2) = 0.5 * t010;
F(2, 3, 3) = 0.25 * t011;
%
F(3, 1, 1) = 0.125 * t111;
F(3, 1, 2) = 0.25 * t110;
F(3, 1, 3) = 0.125 * t111;
F(3, 2, 1) = 0.25 * t101;
F(3, 2, 2) = 0.5 * t100;
F(3, 2, 3) = 0.25 * t101;
F(3, 3, 1) = 0.125 * t111;
F(3, 3, 2) = 0.25 * t110;
F(3, 3, 3) = 0.125 * t111;
%
%
h3 = zeros(2*N+1, 2*N+1, 2*N+1);
h3(N+1, N+1, N+1) = B_fan(1);
h3(N:N+2, N:N+2, N:N+2) = h3(N:N+2, N:N+2, N:N+2) + B_fan(2)*F;
FN = F;
for in = 2:N
    FN = convn(FN, F); %
    h3(N+1-in:N+1+in, N+1-in:N+1+in, N+1-in:N+1+in) = ...
        h3(N+1-in:N+1+in, N+1-in:N+1+in, N+1-in:N+1+in) +
B_fan(in+1) * FN;
end
%
%
A3=zeros(N+1,N+1,N+1);
A3(1,1,1)=h3(N+1,N+1,N+1);
A3(2:N+1,1,1)=2*h3(N+2:2*N+1,N+1,N+1);
A3(1,2:N+1,1)=2*h3(N+1,N+2:2*N+1,N+1);
A3(1,1,2:N+1)=2*h3(N+1,N+1,N+2:2*N+1);
A3(2:N+1,2:N+1,1)=4*h3(N+2:2*N+1,N+2:2*N+1,N+1);
A3(2:N+1,1,2:N+1)=4*h3(N+2:2*N+1,N+1,N+2:2*N+1);
```

```
A3(1,2:N+1,2:N+1)=4*h3(N+1,N+2:2*N+1,N+2:2*N+1);
A3(2:N+1,2:N+1,2:N+1)=8*h3(N+2:2*N+1,N+2:2*N+1,N+2:2*N+1);
%
% plot iso_surface for cone filter design
%
subplot(2, 2, 3);
point=32;
deltaw=2*pi/32;
HF3=zeros(point+1,point+1,point+1);
XX=zeros(point+1,point+1,point+1);
YY=zeros(point+1,point+1,point+1);
ZZ=zeros(point+1,point+1,point+1);
for i1=0:point
    w1=-pi+i1*deltaw;
    for i2=0:point
        w2=-pi+i2*deltaw;
        for i3=0:point
            w3=-pi+i3*deltaw;
            XX(i1+1,i2+1,i3+1)=w1/pi;
            YY(i1+1,i2+1,i3+1)=w2/pi;
            ZZ(i1+1,i2+1,i3+1)=w3/pi;
            for n1=0:N/2
                for n2=0:N/2
                    for n3=0:N/2

HF3(i1+1,i2+1,i3+1)=HF3(i1+1,i2+1,i3+1)+A3(n1+1,n2+1,n3+1)*cos(n1*
w1)*cos(n2*w2)*cos(n3*w3);
                    end
                end
            end
        end
    end
end
HF3=abs(HF3);
[F,V]=isosurface(XX,YY,ZZ,HF3,0.99);
[rowV, colV]=size(V);
%
[Z1 ind]=sort(V(:,3));
X1=V(ind,1);
Y1=V(ind,2);
%
Zu=Z1(rowV/2+1:rowV,1);
Xu=X1(rowV/2+1:rowV,1);
Yu=Y1(rowV/2+1:rowV,1);
Zd=Z1(1:rowV/2,1);
Xd=X1(1:rowV/2,1);
Yd=Y1(1:rowV/2,1);
```

```
%
ti = -1:.0025:1;
[XI,YI] = meshgrid(ti,ti);
ZU = griddata(Xu,Yu,Zu,XI,YI);
ZD = griddata(Xd,Yd,Zd,XI,YI);
%
contour3(XI,YI,ZU,35,'k'); hold % with black plot
contour3(XI,YI,ZD,35,'k');
%
view(3);
title('\theta=','FontSize',16);
set( gca,'FontSize', 16);
xlabel('\omega_1 / \pi','FontSize',16);
set(gca,'xtick',linspace(-1,1,5));
ylabel('\omega_2 / \pi','FontSize',16);
set(gca,'ytick',linspace(-1,1,5));
zlabel('\omega_3 / \pi','FontSize',16);
set(gca,'ztick',linspace(-1,1,5));
%
grid on;
hidden off;
%
%
% 3D_FIR_Cone_type_Filter (theta = 55 degrees)
```


2D FIR Fan Type Filter
Magnitude Response
$\omega_2$
$\omega_1$
2D FIR Circular Type Filter
Magnitude Response
$\omega_2$
$\omega_1$
$\theta$=
$\omega_3$ / $\pi$
$\omega_2$ / $\pi$
$\omega_1$ / $\pi$