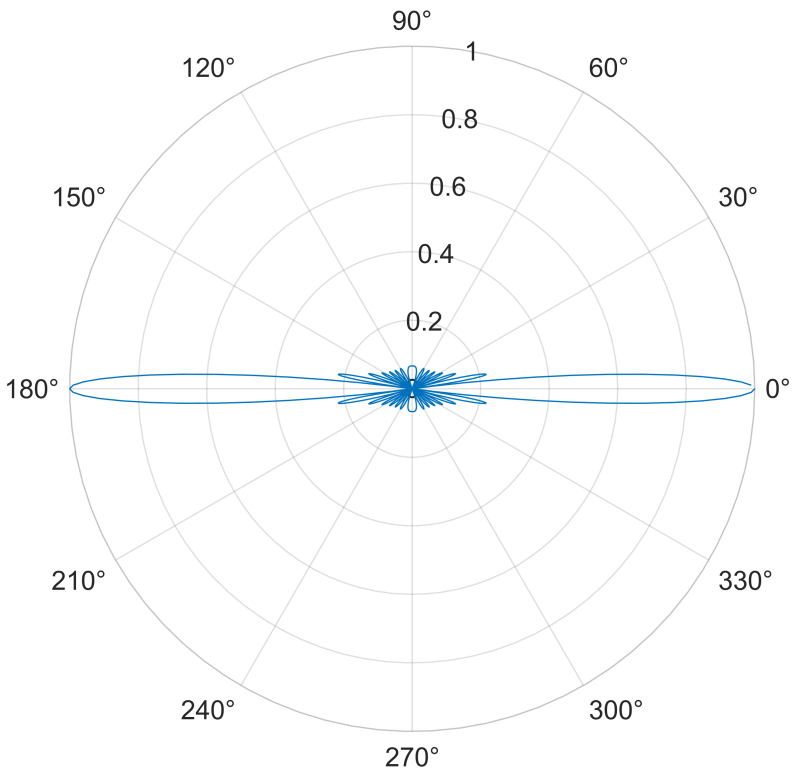
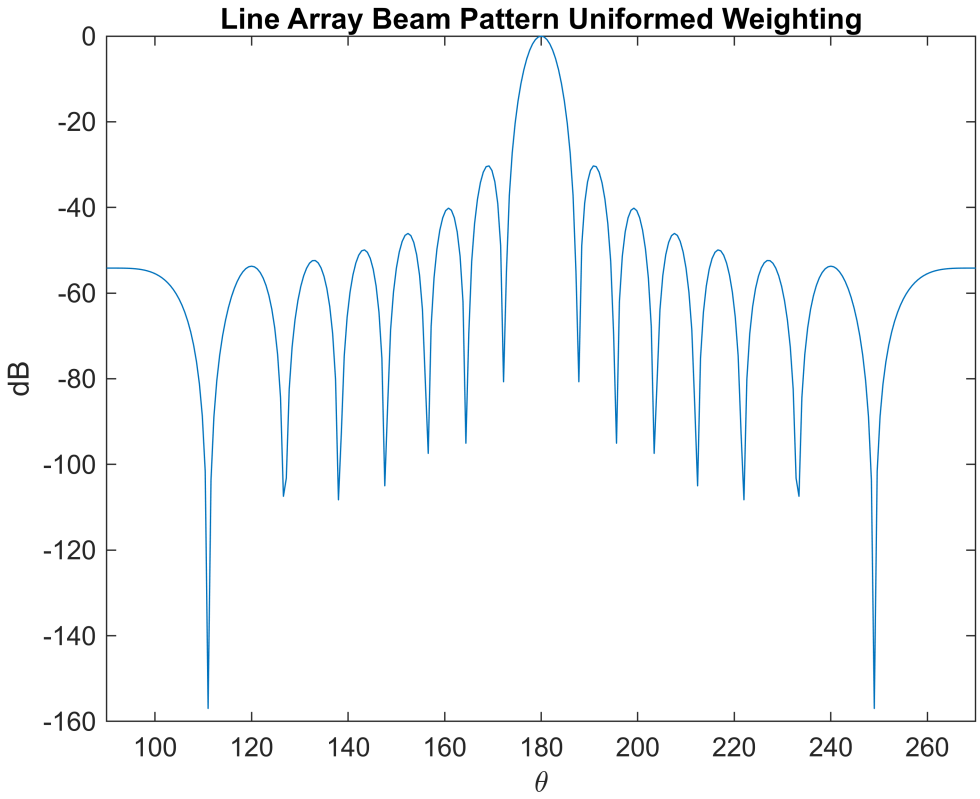
```
%CLS Continuous Line Source Arrays
%intial
% Parameters
c=1500;
fc=1000;
lambda=c/fc;
k=2*pi/lambda;
d=lambda/2; %element spacing
kd=k*d;
N=15; %number of elements
theta_s=pi/300; %theta spacing
t=[0 1/6 1/5 1/4 1/3 1/2 7/6 6/5 5/4 4/3 3/2 1]*pi; %phase
theta=0:theta_s:2*pi; %index for theta
N_t=length(theta); %the length of theta
v=.5*kd*sin(theta-t(1)); %null
H=(1/N)*sin(N*v)./(sin(v)); %

%plots
figure;
polarplot(theta,H)
```



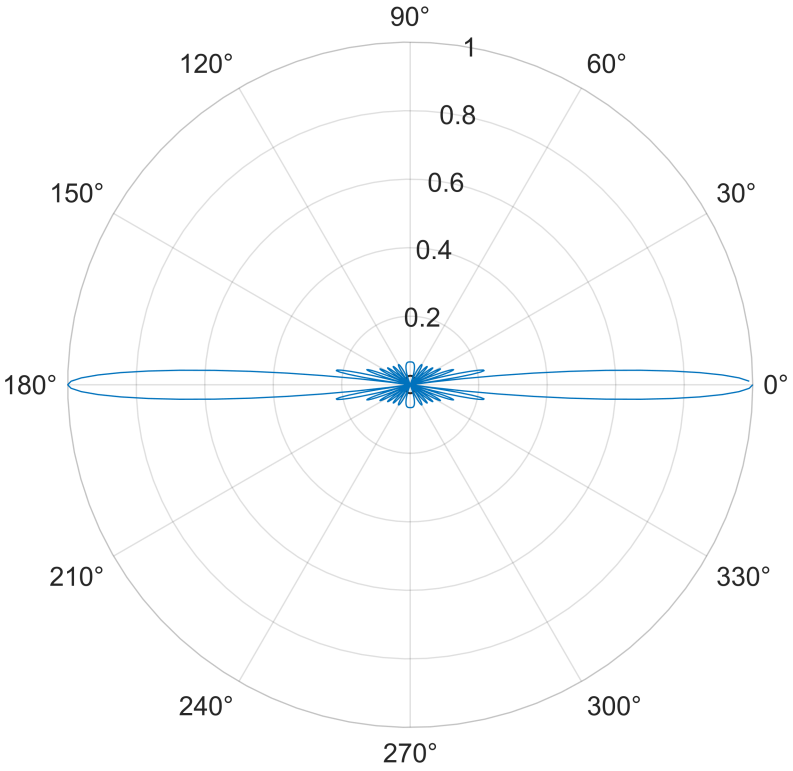

```
figure;
rect=rectwin(length(N_t));
H_rect= H.*rect;
plot(theta*(180/pi),20*log(H_rect))
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```

```
xlim([90 270])
title('Line Array Beam Pattern Uniformed Weighting')
xlabel('\theta')
ylabel('dB')
```



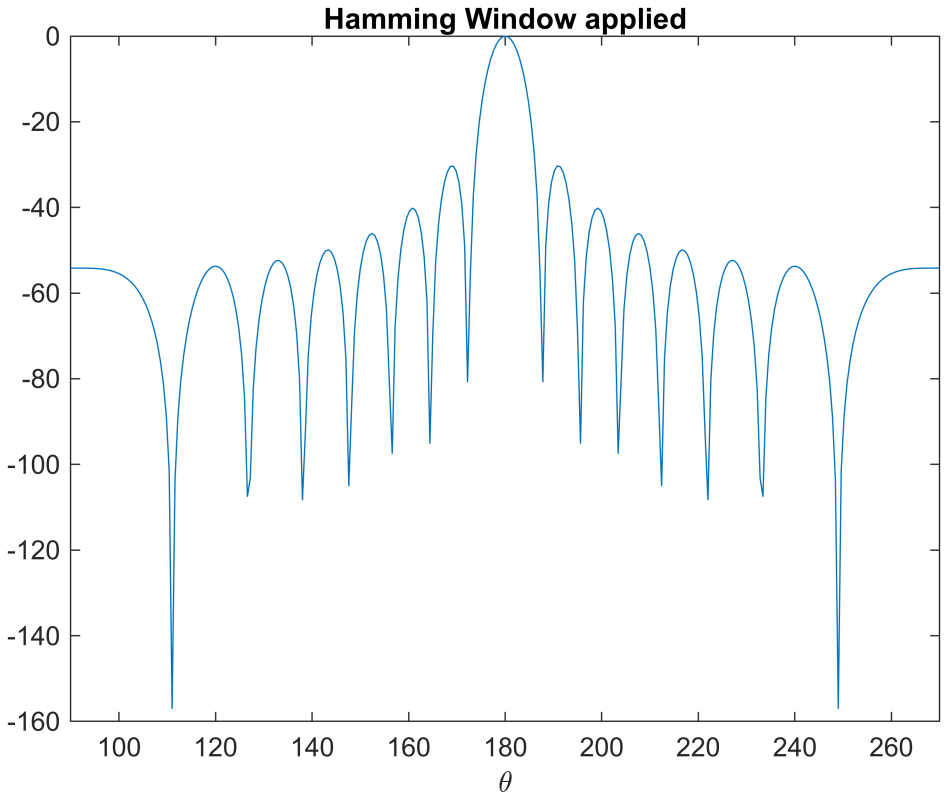

```
% Apply Hamming window to the radiation pattern
window_theta = hamming(length(N_t))';

H_Hamming = H .* window_theta;
figure;
polarplot(theta,H_Hamming)
hold on
polarplot(theta,window_theta)
hold off
```



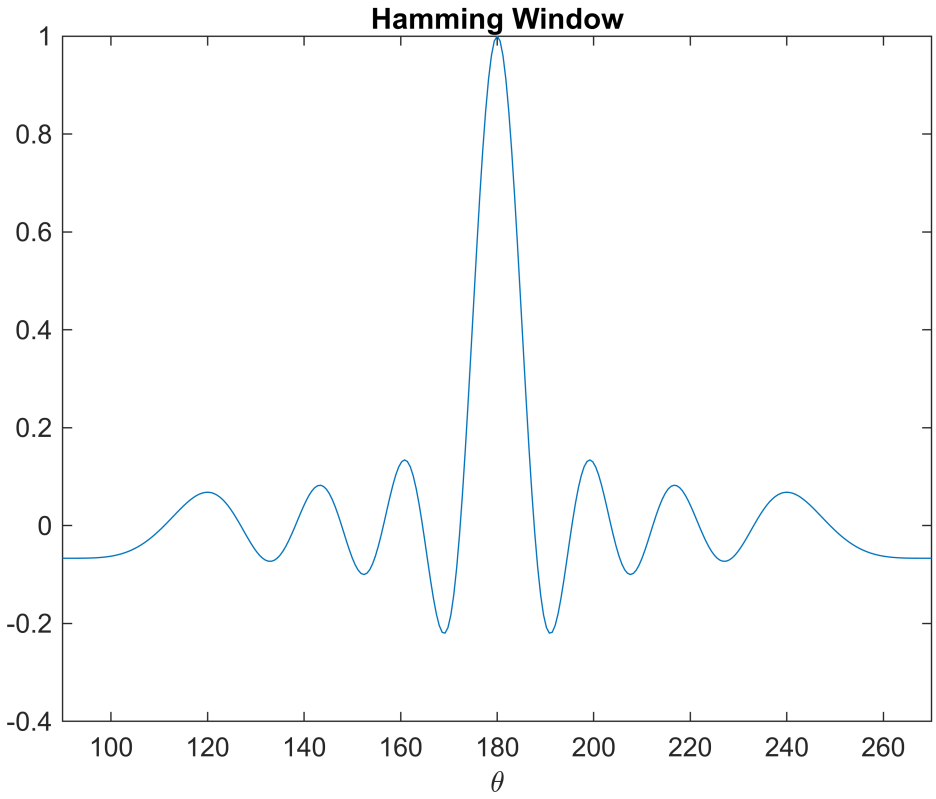

```
figure;
plot(theta*(180/pi),20*log(H_Hamming))
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```

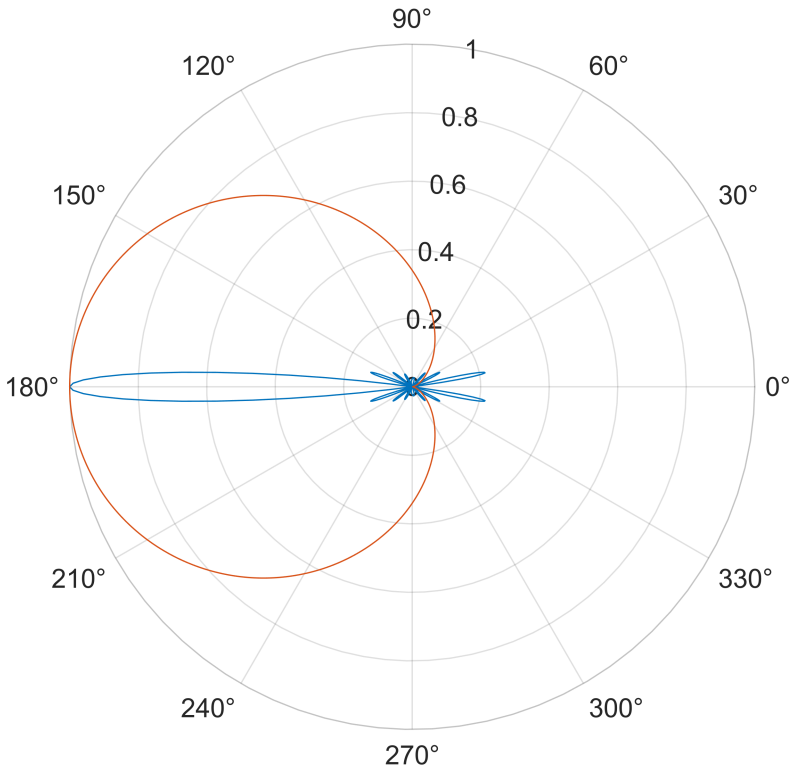
```
title('Hamming Window applied')
xlim([90 270])
xlabel('\theta')
```




```
figure;
plot(theta*(180/pi),H_Hamming)
title('Hamming Window')
xlim([90 270])
xlabel('\theta')
```



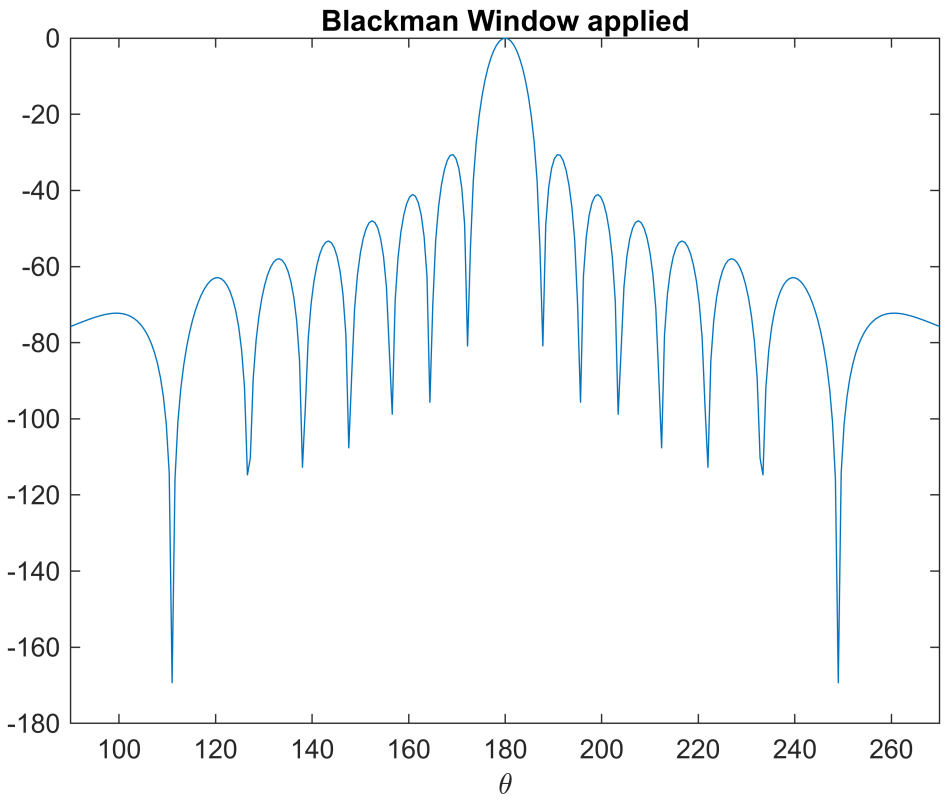

```
% Apply Blackman window to the radiation pattern
window_theta = blackman(length(theta))';

H_Blackman = H .* window_theta;
figure;
polarplot(theta,H_Blackman)
hold on
polarplot(theta,window_theta)
hold off
```



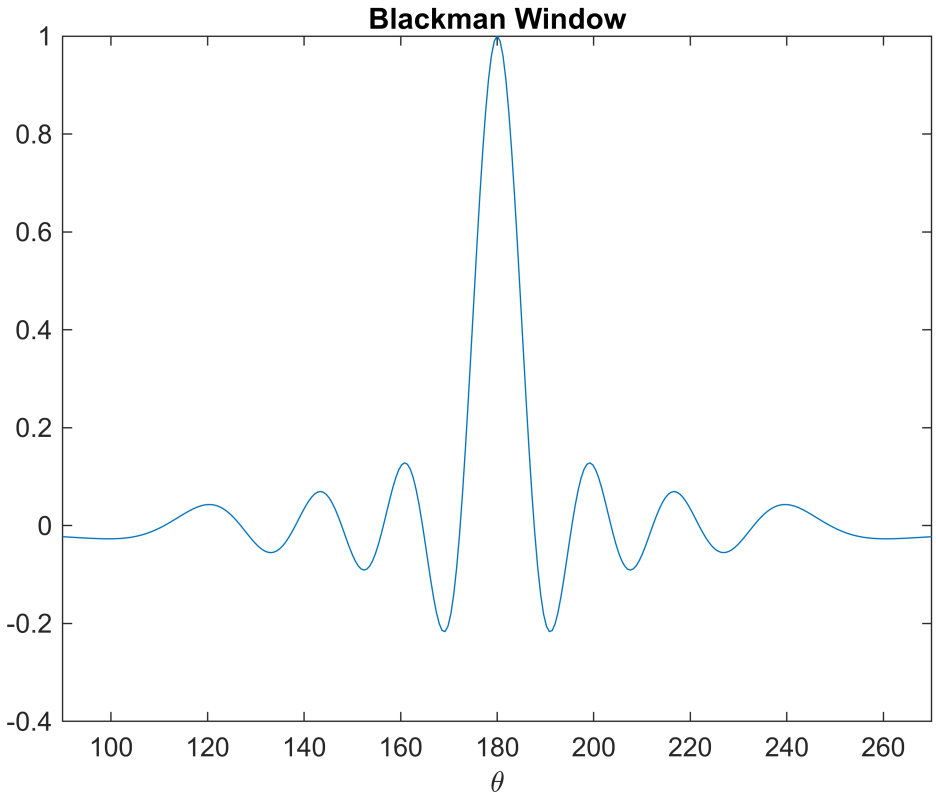

```
figure;
plot(theta*(180/pi),20*log(H_Blackman))
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```

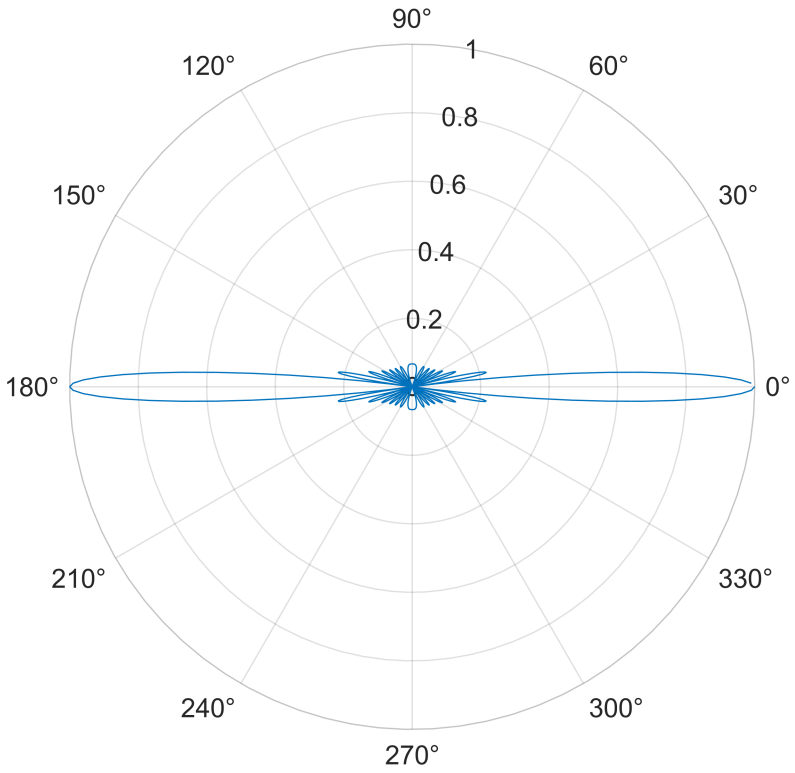
```
title('Blackman Window applied')
xlim([90 270])
xlabel('\theta')
```




```
figure;
plot(theta*(180/pi),H_Blackman)
title('Blackman Window')
xlim([90 270])
xlabel('\theta')
```



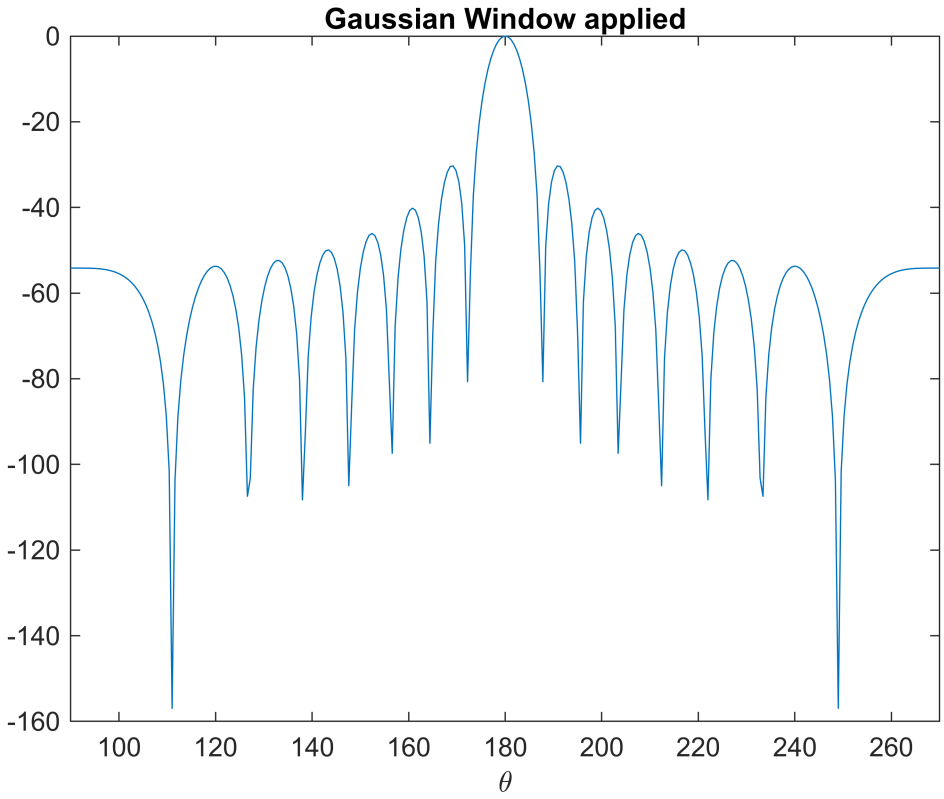

```
% Apply Gaussian window to the radiation pattern
window_theta = gausswin(length(N_t))';

H_Gausswin = H .* window_theta;
figure;
polarplot(theta,H_Gausswin)
hold on
polarplot(theta,window_theta)
hold off
```



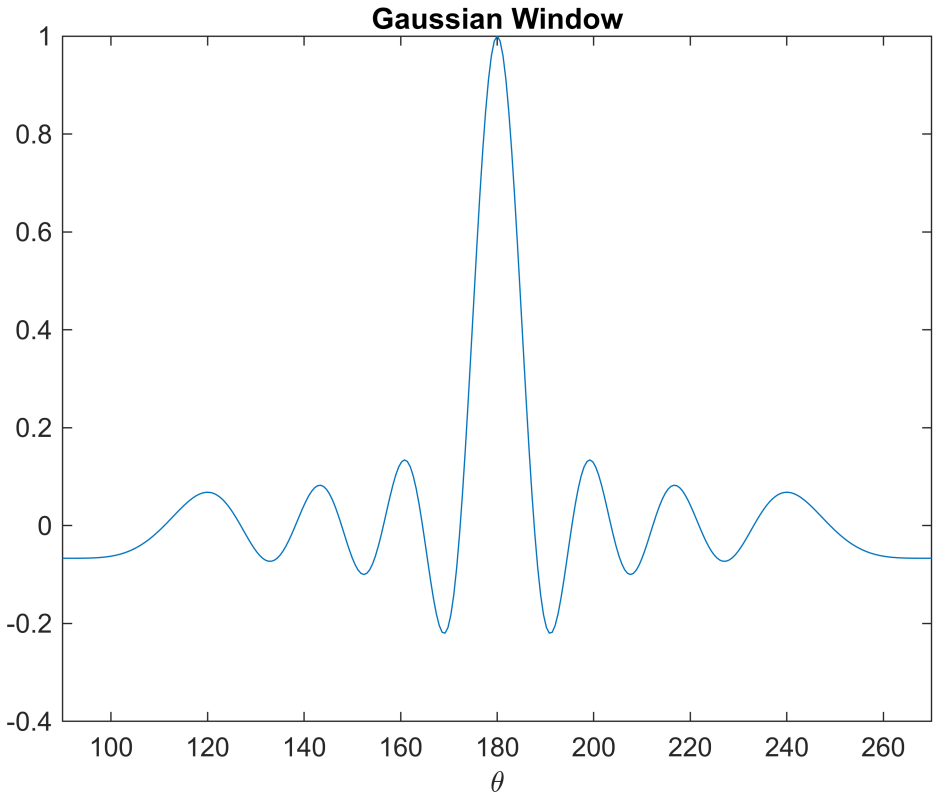

```
figure;
plot(theta*(180/pi),20*log(H_Gausswin))
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```

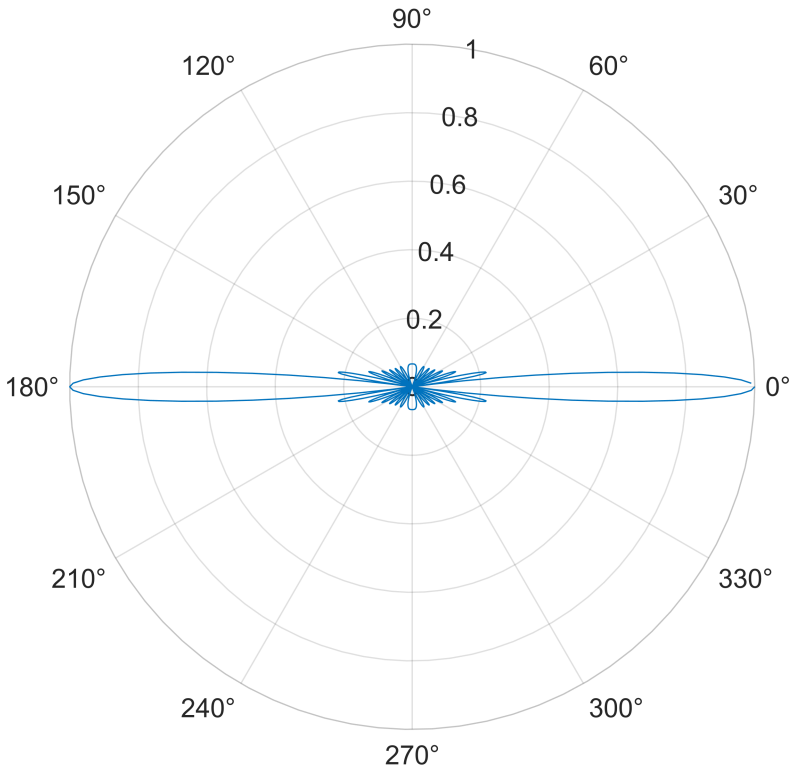
```
title('Gaussian Window applied')
xlim([90 270])
xlabel('\theta')
```




```
figure;
plot(theta*(180/pi),H_Gausswin)
title('Gaussian Window')
xlim([90 270])
xlabel('\theta')
```



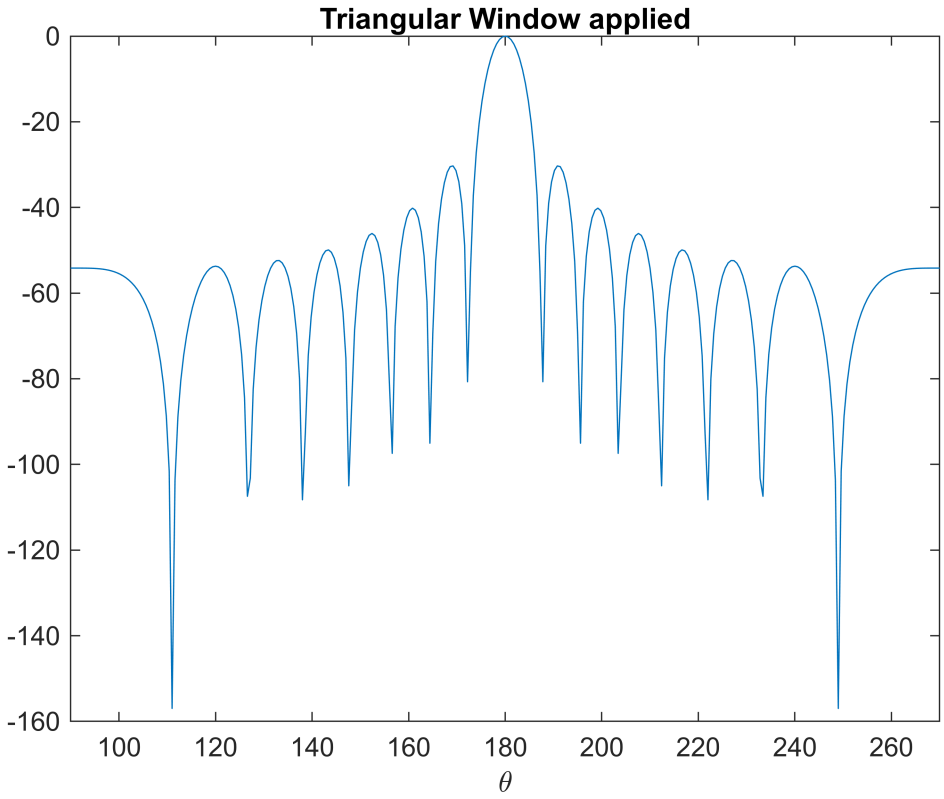

```
% Apply Triangle window to the radiation pattern
window_theta = triang(length(N_t))';

H_Triang = H .* window_theta;
figure;
polarplot(theta,H_Triang)
hold on
polarplot(theta,window_theta)
hold off
```



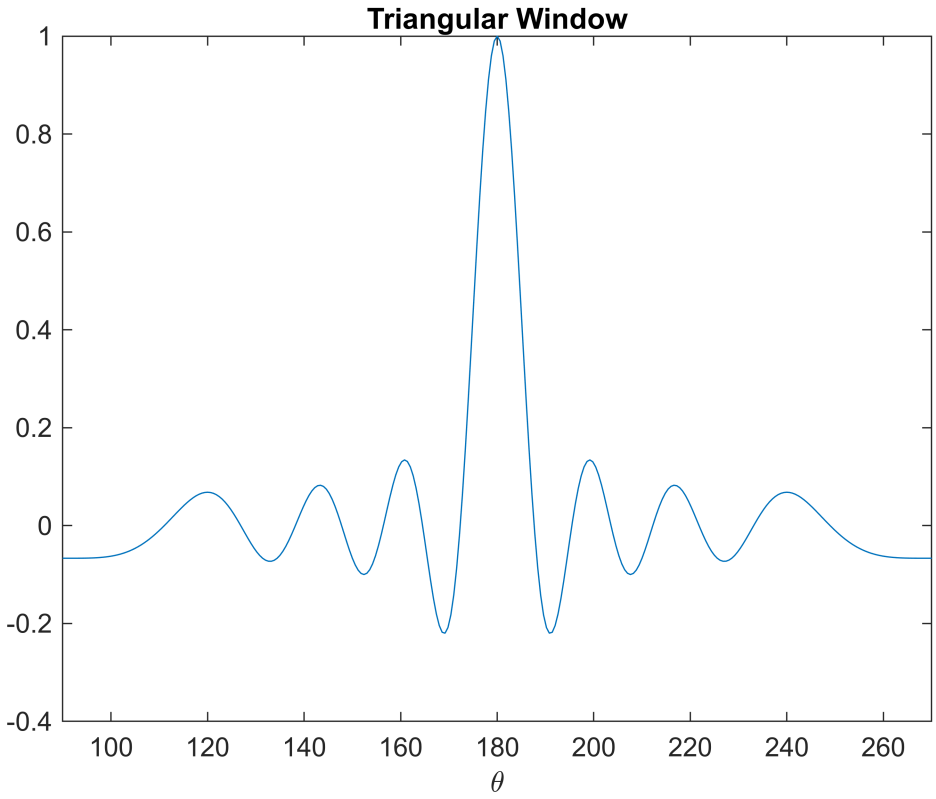

```
figure;
plot(theta*(180/pi),20*log(H_Triang))
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```

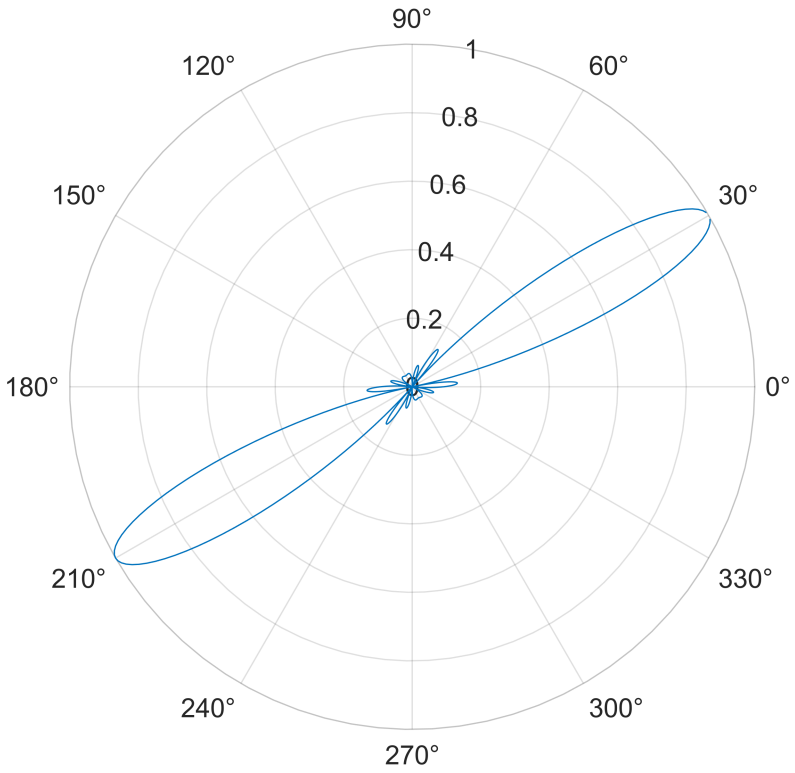
```
title('Triangular Window applied')
xlim([90 270])
xlabel('\theta')
```



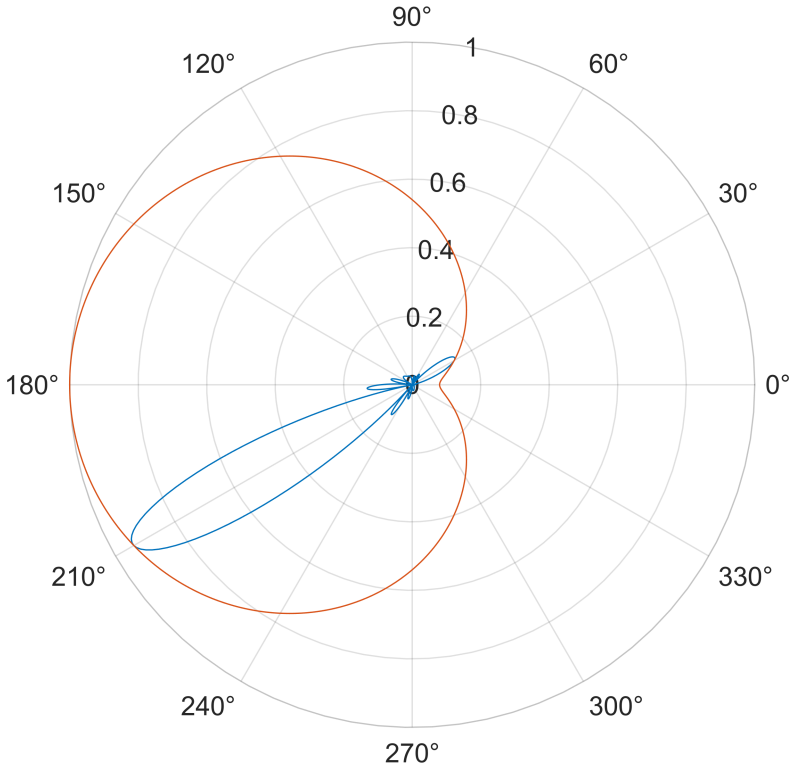

```
figure;
plot(theta*(180/pi),H_Triang)
title('Triangular Window')
xlim([90 270])
xlabel('\theta')
```



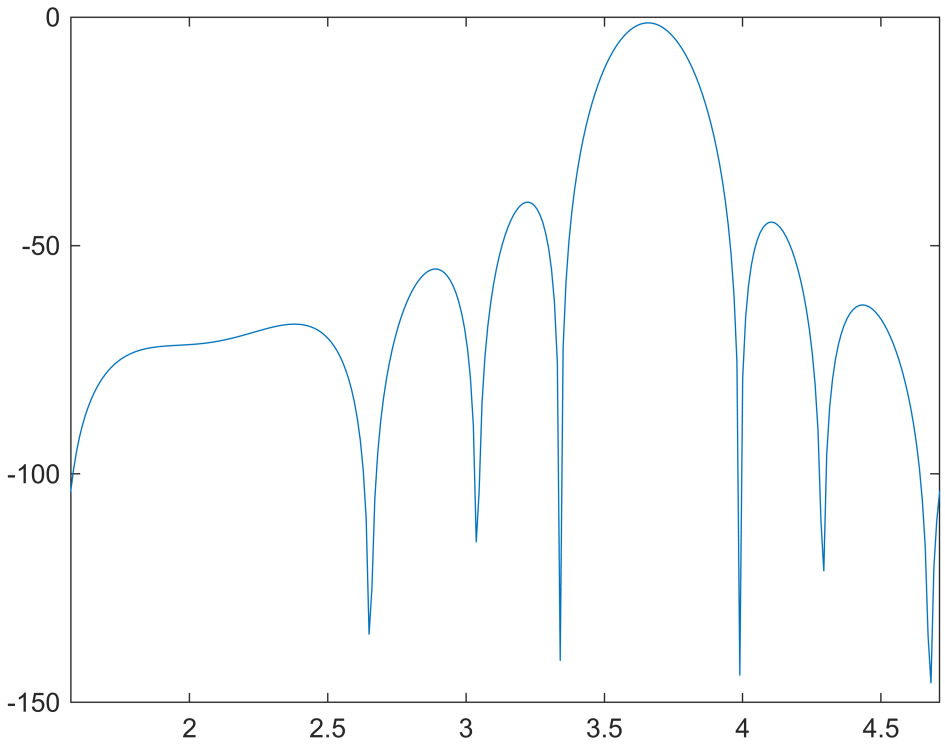

```
%question 3
ka=12;
X=ka.*sin(theta-t(2));
Hcp=abs(2*besselj(1,X)./X);
figure;
polarplot(theta,Hcp)
```



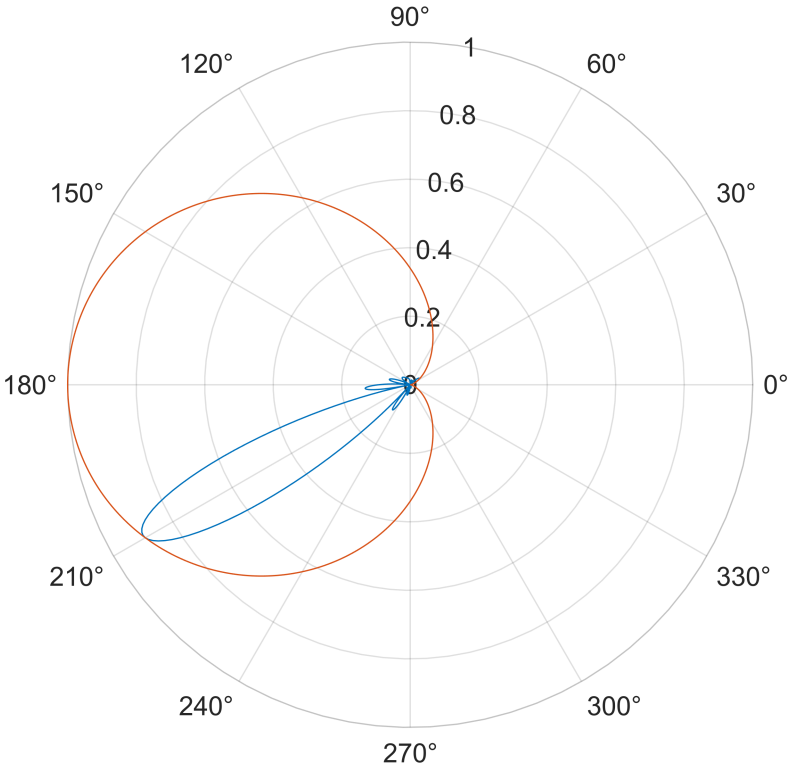

```
% Apply Hamming window to the radiation pattern
window_theta = hamming(length(theta))';
Hcp_Hamming = Hcp.* window_theta;
figure;
polarplot(theta,Hcp_Hamming)
hold on
polarplot(theta,window_theta)
hold off
```



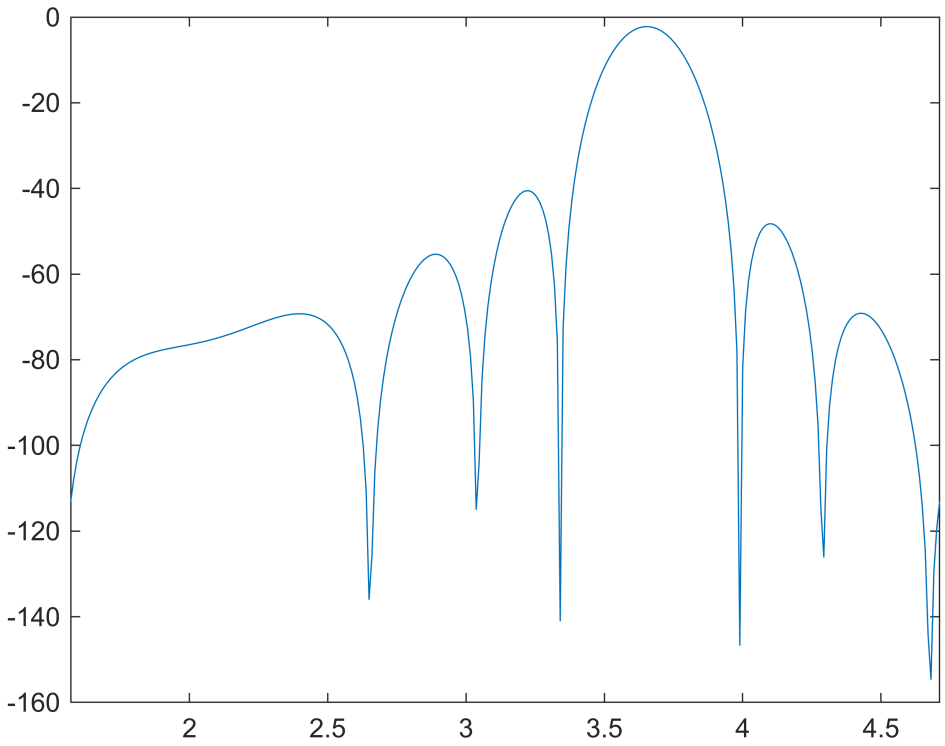

```
figure;
plot(theta,20*log(Hcp_Hamming))
xlim([pi/2 (3/2)*pi])
```



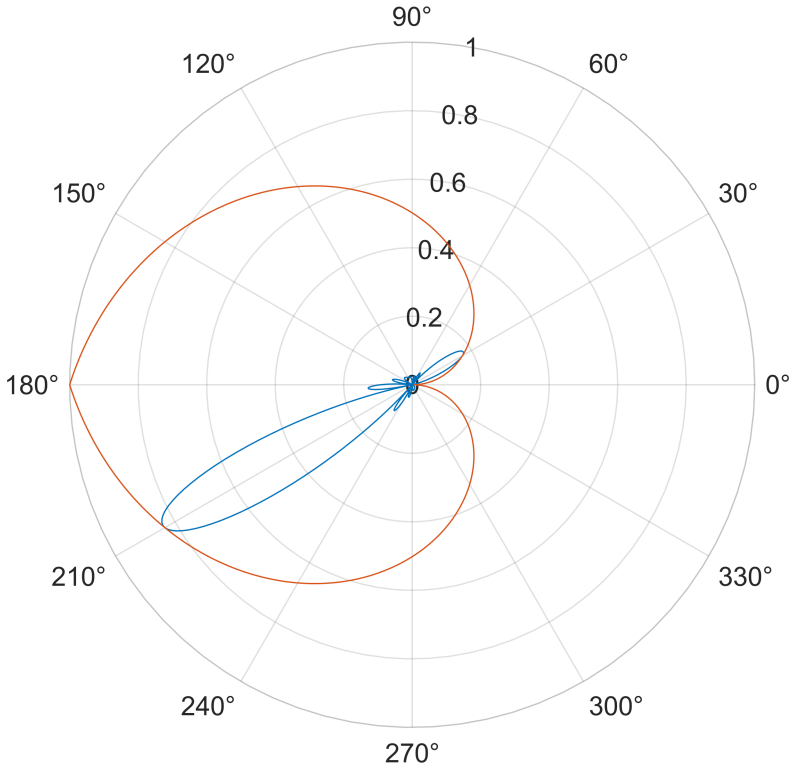

```
% Apply Blackman window to the radiation pattern
window_theta = blackman(length(theta))';
X_Blackman = Hcp .* window_theta;
figure;
polarplot(theta,X_Blackman)
hold on
polarplot(theta,window_theta)
hold off
```



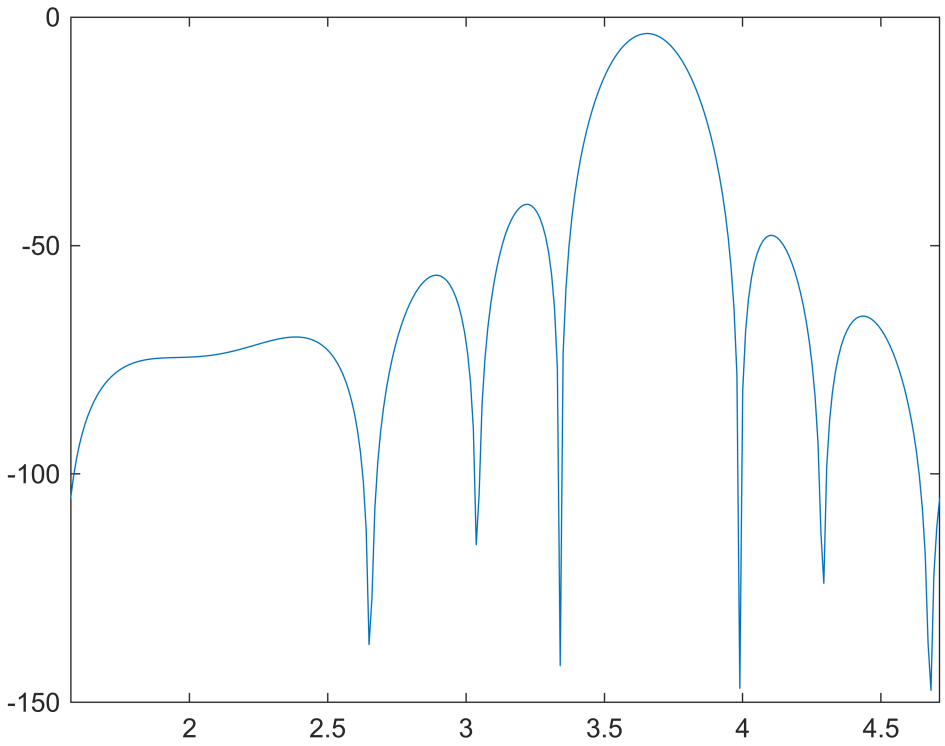

```
figure;
plot(theta,20*log(X_Blackman))
xlim([pi/2 (3/2)*pi])
```




```
% Apply Triangle window to the radiation pattern
window_theta = triang(length(theta))';
X_triang = Hcp .* window_theta;
figure;
polarplot(theta,X_triang)
hold on
polarplot(theta,window_theta)
hold off
```

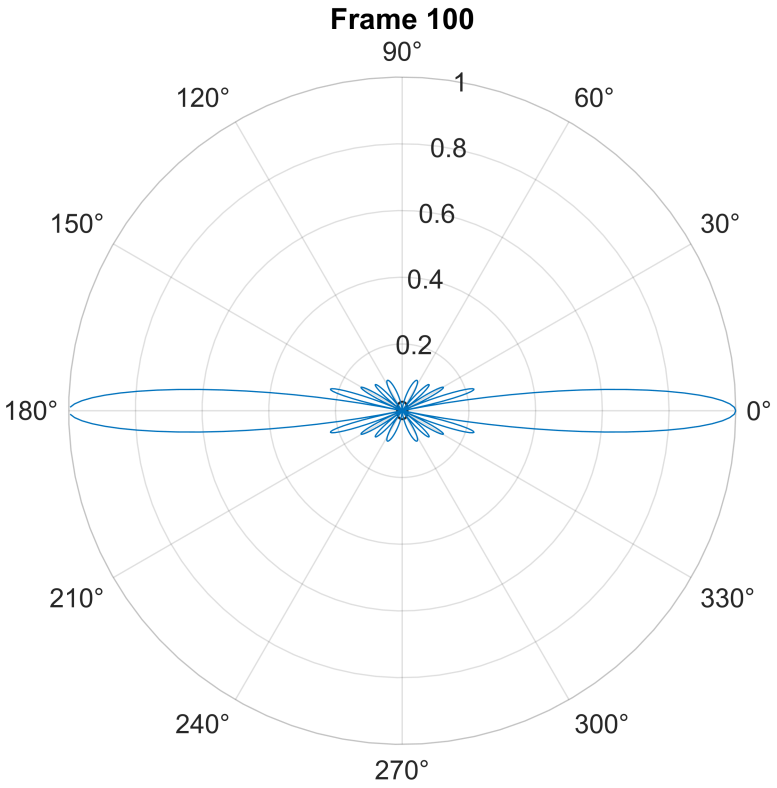



```
figure;
plot(theta,20*log(X_triang))
xlim([pi/2 (3/2)*pi])
```

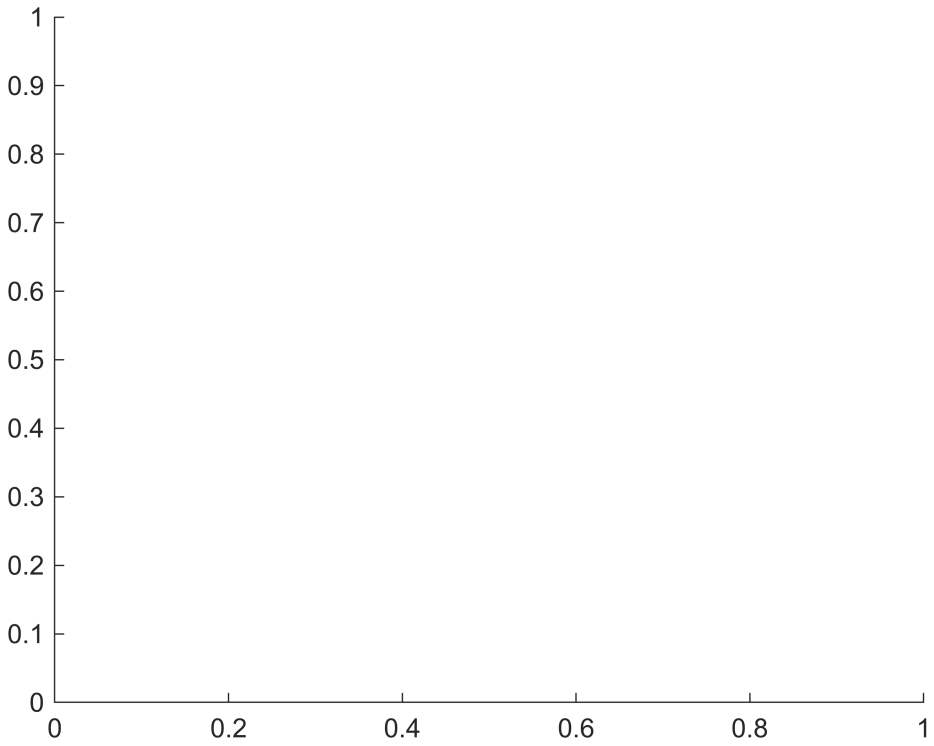



```
%matlab movie
clear
% Example: Creating a simple MATLAB movie
    c=1500;
    fc=1000;
    lambda=c/fc;
    k=2*pi/lambda;
    d=lambda/2;
    kd=k*d;
    N=10;


% Set up a figure
figure;

% Define the number of frames
num_frames = 100;
theta_s=pi/300; %theta spacing
t=linspace(0,pi,num_frames); %steering angle
theta=0:theta_s:2*pi; %index for theta
N_t=length(theta); %the length of theta
% Create a loop to generate frames
for frame = 1:num_frames
```

```
    % Your code to generate each frame goes here
    % For demonstration purposes, let's plot a sine wave that moves across the screen
    v=.5*kd*sin(theta-t(frame)); %null
    H=(1/N)*sin(N*v)./(sin(v)); %
%     x = linspace(0, 2*pi, 100);
%     y = sin(x - 0.1 * frame);

%     plot(x, y, 'LineWidth', 2);
    polarplot(theta,H)
    % Set axis limits
%     axis([0 2*pi -1 1]);

    % Title with frame number
    title(['Frame ' num2str(frame)]);

    % Capture the current frame
    frames(frame) = getframe(gcf);
end
```

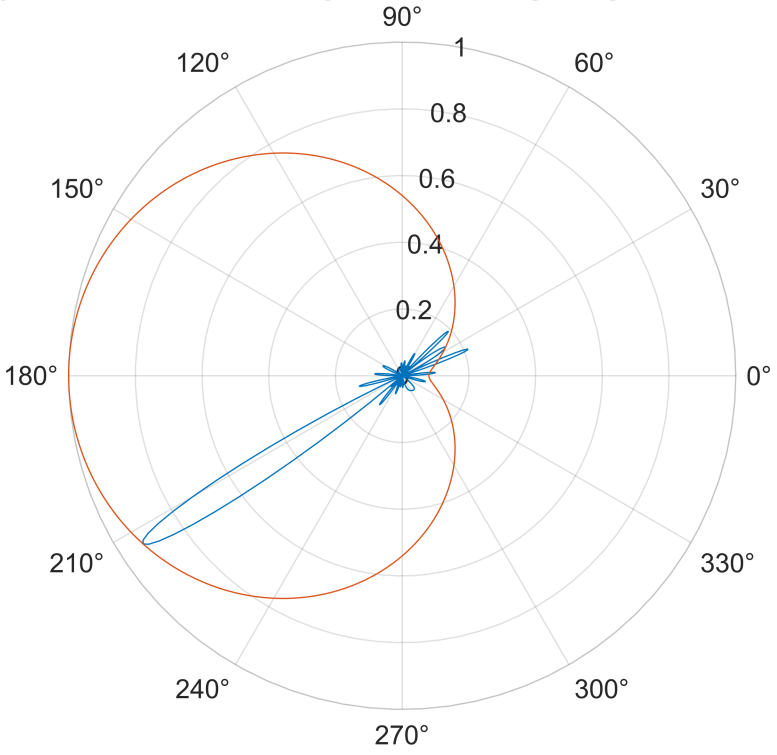



```
% Create the movie
figure;
movie(frames, 2);  % Play the movie at 2 frames per second
```

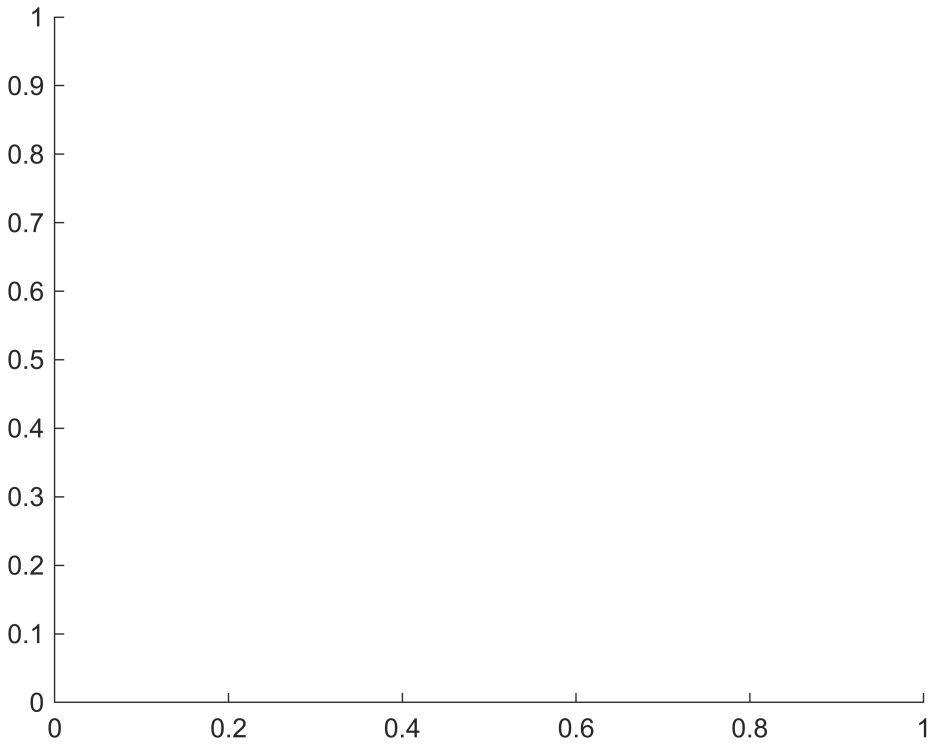



```
%matlab movie steering Beam pattern
clear
c=1500;
fc=1000;
lambda=c/fc;
k=2*pi/lambda;
d=lambda/2;
kd=k*d;
N=15;

% Set up a figure
figure;

% Define the number of frames
num_frames = 100;
theta_s=pi/300; %theta spacing
t=linspace(-.005,.005,num_frames); %steering time delay
ctd=c*t/d;
theta=0:theta_s:2*pi; %index for theta
N_t=length(theta); %the length of theta

% Apply Hamming window to the radiation pattern
window_theta = hamming(length(theta))';

% Create a loop to generate frames
```

```
for frame = 1:num_frames
    v=.5*kd*sin(theta-ctd(frame));
    H=(1/N)*sin(N*v)./(sin(v));
    H_Hamming = H .* window_theta;


    polarplot(theta,H_Hamming)
    hold on
    polarplot(theta,window_theta)
    hold off

    % Title with frame number
    title(['Line Array Beam Pattern Steering with Hamming Wieghted Elements overlay']);

    % Capture the current frame
    frames(frame) = getframe(gcf);
end
```




```
% Create the movie
figure;
movie(frames);  % Play the movie at 2 frames per second
```


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1