Example- Find output of a 4-point Moving Average filter to the rectangular pulse $x[n] = 4p_5[n-6]$ using the FFT/IFFT algorithms.

Given a 4-point MA filter

$$h[n] = \frac{1}{4}\left[\delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3]\right]$$

For $h[n]$, the start index $Q=0$ + the length is $M=4$.

For $x[n] = 4p_5[n-6]$, the start index $P=4$ + the length is $N=5$.

$$N+M = 5+4 = 9 < L_{FFT} = 2^4 = 16$$

↳ Use 16-point FFT + IFFT

$y[n]$ will have values (non-zero) for

$$P+Q = 4+0 = 4 \le n \le P+Q+N+M-1-1$$
$$= 4+0+5+4-1-1$$

↓

$$4 \le n \le 11 \quad (8 \text{ points})$$

FFT/IFFT will yield 16 points (extras will just be zeros)

Using Table 4.1

$$H(\Omega) = 0.25\left[1 + e^{-j\Omega} + e^{-j2\Omega} + e^{-j3\Omega}\right] \quad -\infty < \Omega < \infty$$

&

$$X(\Omega) = 4\,\frac{\sin[(2+1/2)\Omega]}{\sin(\Omega/2)}\,e^{-j6\Omega} \quad -\infty < \Omega < \infty$$

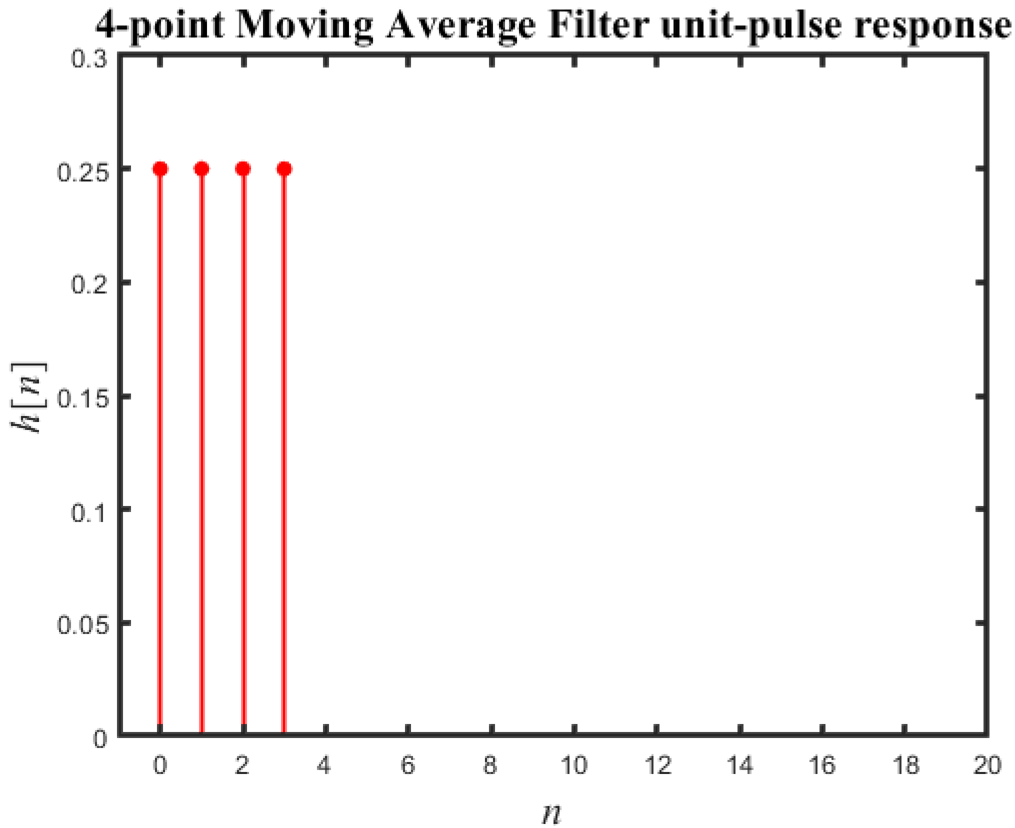
$$Y(\Omega) = H(\Omega)X(\Omega) \quad -\infty < \Omega < \infty$$

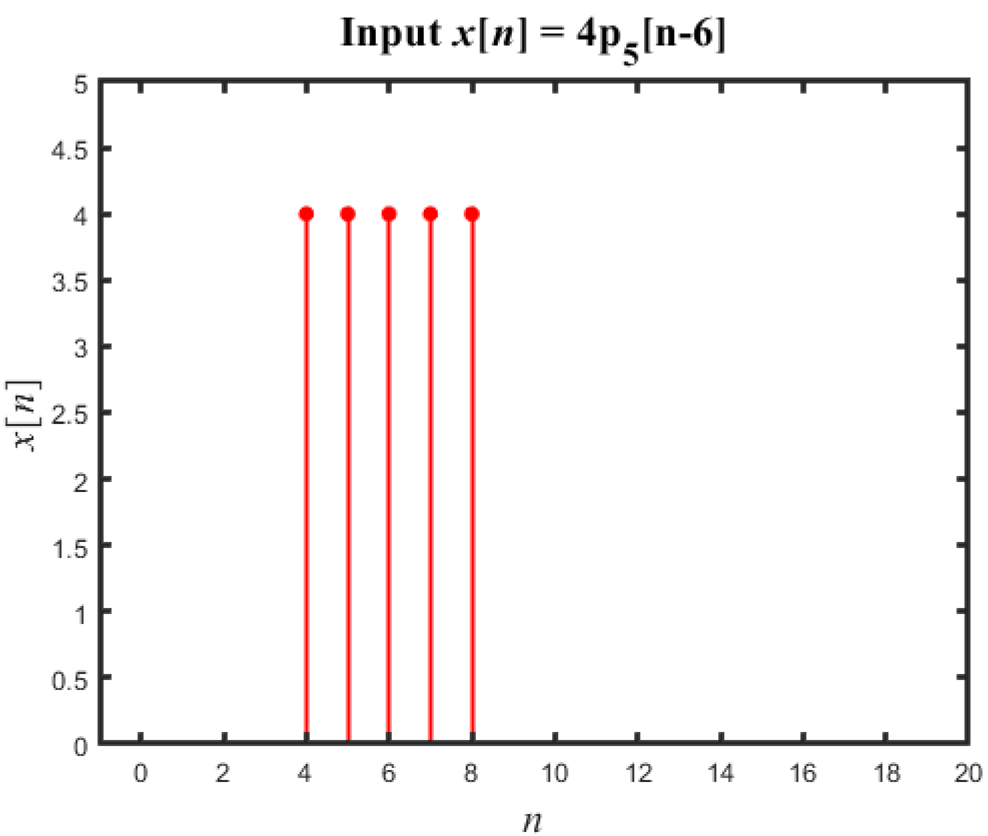
For FFT, $\Omega_k = \frac{2\pi k}{L_{FFT}}$

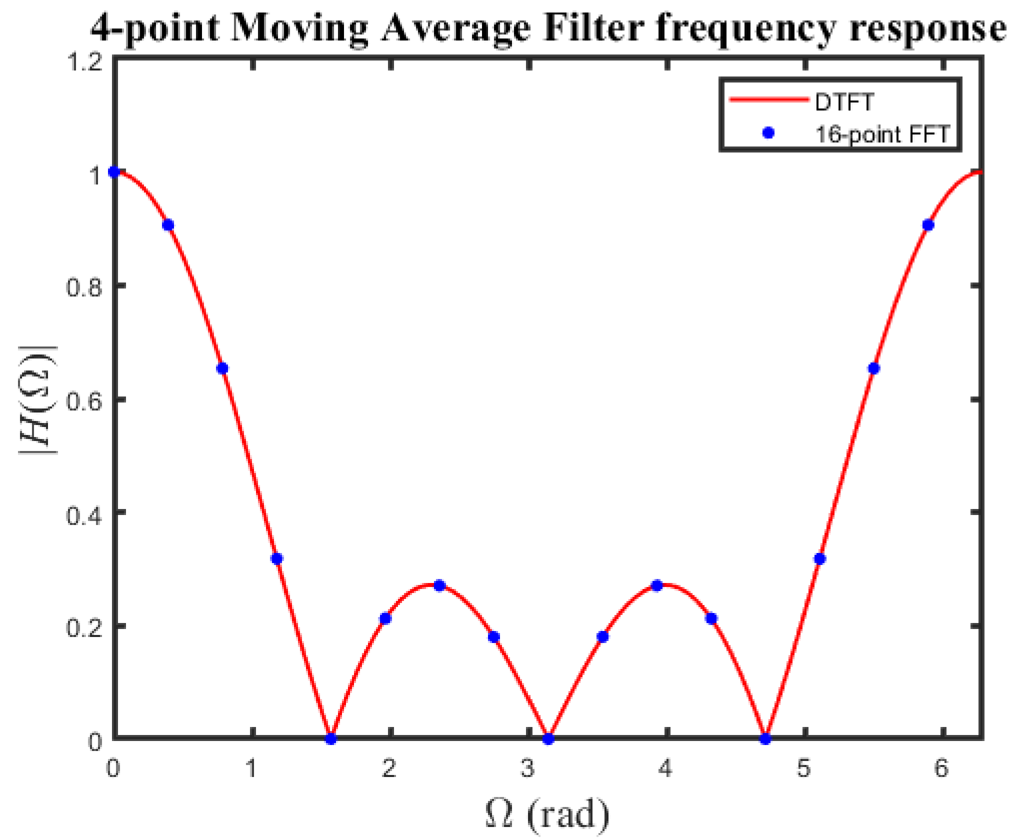
```
% chap5_DT_Fourier_analysis_FFT_example.m
% EE 313 Chap. 5 Implement DT Fourier analysis of a system
% using the FFT & IFFT algorithms.
% System- 4-point moving average filter (MA)
%          h[n]=0.25(delta[n]+delta[n-1]+delta[n-2]+delta[n-3])
%          H(Omega)=0.25(1+e^{-jOmega}+e^{-j2Omega}+e^{-j3Omega})
% Input- time-delayed DT rectangular pulse
%          x[n] = 4*p5[n-6]
%          X(Omega) = 4*e^{-j6Omega}*sin(2.5*Omega)/sin(0.5*Omega)
clear; close all; clc;
% Plot unit-pulse response h[n] and input x[n] signals
Q = 0; M = 4; nh = Q:1:Q+M-1;
h = [0.25,0.25,0.25,0.25]; % 4-point MA
P = 4; N = 5; nx = P:1:P+N-1;
x = [4,4,4,4,4]; % input
xzero = [0,0,0,0,x]; % input starting at n = 0
stem(nh,h,'r.','linewidth',1.5,'markersize',20), axis([-1 20 0 0.3]),
```

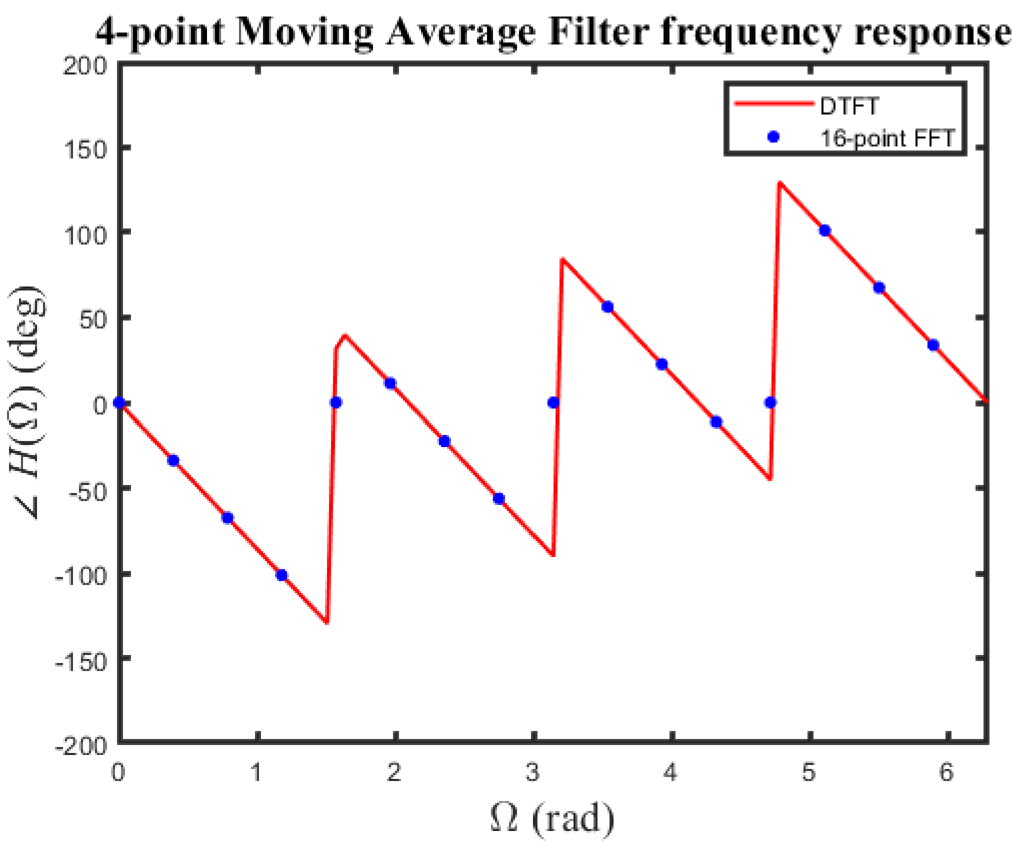
```
xlabel('{\itn}','fontsize',16,'fontname','times'),
ylabel('{\ith}[{\itn}]','fontsize',16,'fontname','times'),
title('4-point Moving Average Filter unit-pulse response',...
    'fontsize',16,'fontname','times'),
figure, stem(nx,x,'r.','linewidth',1.5,'markersize',20), axis([-1 20 0 5]),
xlabel('{\itn}','fontsize',16,'fontname','times'),
ylabel('{\itx}[{\itn}]','fontsize',16,'fontname','times'),
title('Input {\itx}[{\itn}] = 4p_5[n-
6]','fontsize',16,'fontname','times'),
% Choose L=16 > N+M=9 and calculate FFT of x[n] and h[n]
L = 16; k = 0:1:L-1; Omegak = 2*pi*k/L;
Hk = fft(h,L);
Xk = fft(x,L); % This Xk ignores phase shift due to start index P = 4
Xkzero = fft(xzero,L);
Yk = Xk.*Hk; Ykzero = Xkzero.*Hk;
% Calculate DTFT of h[n] and x[n]
Omega = eps:0.02*pi:2*pi; % DTFT freq. range
H = 0.25*(1+exp(-j*Omega) + exp(-j*2*Omega) + exp(-j*3*Omega));
X = 4*exp(-j*6*Omega).*sin(2.5*Omega)./sin(0.5*Omega);
Y = X.*H;
% Plot FFT and DTFT results for h[n]
figure, plot(Omega,abs(H),'r-',Omegak,abs(Hk),'b.'),
axis([0 2*pi 0 1.2]),
xlabel('\Omega (rad)','fontsize',16,'fontname','times'),
ylabel('|{\itH}(\Omega)|','fontsize',16,'fontname','times'),
title('4-point Moving Average Filter frequency response',...
    'fontsize',16,'fontname','times'),
legend('DTFT',[num2str(L),'-point FFT']);
figure, plot(Omega,angle(H)*180/pi,'r',Omegak,angle(Hk)*180/pi,'b.'),
axis([0 2*pi -200 200]),
ylabel('\angle {\itH}(\Omega)
(deg)','fontsize',16,'fontname','times'),
xlabel('\Omega (rad)','fontsize',16,'fontname','times'),
title('4-point Moving Average Filter frequency response',...
    'fontsize',16,'fontname','times'),
legend('DTFT',[num2str(L),'-point FFT']);
% Plot FFT and DTFT results for x[n]
figure, plot(Omega,abs(X),'r-',Omegak,abs(Xk),'b.',...
    Omegak,abs(Xkzero),'k+'),
axis([0 2*pi 0 25]),
xlabel('\Omega (rad)','fontsize',16,'fontname','times'),
ylabel('|{\itX}(\Omega)|','fontsize',16,'fontname','times'),
title('Input frequency response','fontsize',16,'fontname','times'),
```

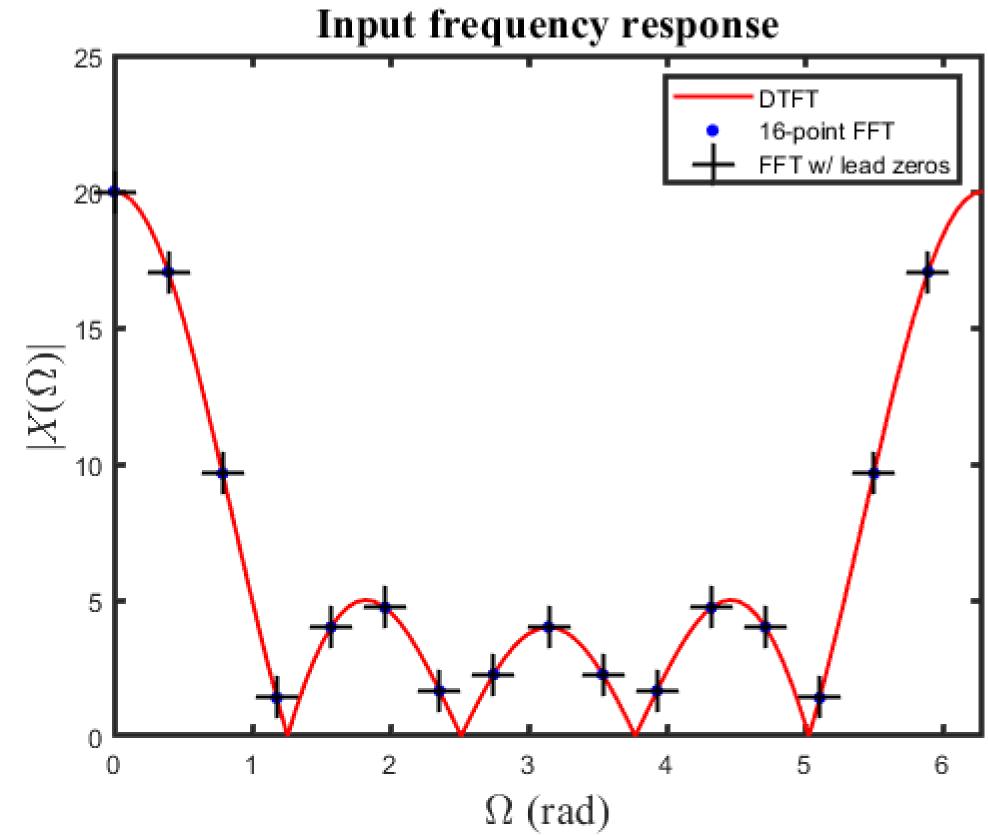
```
legend('DTFT',[num2str(L),'-point FFT'],'FFT w/ lead zeros');
figure,
plot(Omega,angle(X)*180/pi,'r',Omegak,angle(Xk)*180/pi,'b.',...
    Omegak,angle(Xkzero)*180/pi,'k+'),
axis([0 2*pi -200 275]),
ylabel('\angle {\itX}(\Omega)(deg)','fontsize',16,'fontname','times'),
xlabel('\Omega (rad)','fontsize',16,'fontname','times'),
title('Input frequency response','fontsize',16,'fontname','times'),
legend('DTFT',[num2str(L),'-point FFT'],'FFT w/ lead zeros');
% Plot FFT and DTFT results for y[n]
figure, plot(Omega,abs(Y),'r-',Omegak,abs(Yk),'b.',...
    Omegak,abs(Ykzero),'k+'),
axis([0 2*pi 0 25]),
xlabel('\Omega (rad)','fontsize',16,'fontname','times'),
ylabel('|{\itY}(\Omega)|','fontsize',16,'fontname','times'),
title('Output frequency response','fontsize',16,'fontname','times'),
legend('DTFT',[num2str(L),'-point FFT'],'FFT w/ lead zeros');
figure,
plot(Omega,angle(Y)*180/pi,'r',Omegak,angle(Yk)*180/pi,'b.',...
    Omegak,angle(Ykzero)*180/pi,'k+'),
axis([0 2*pi -200 275]),
ylabel('\angle {\itY}(\Omega)(deg)','fontsize',16,'fontname','times'),
xlabel('\Omega (rad)','fontsize',16,'fontname','times'),
title('Output frequency response','fontsize',16,'fontname','times'),
legend('DTFT',[num2str(L),'-point FFT'],'FFT w/ lead zeros');
% Find output by DT Fourier analysis
yfft = ifft(Yk); nyfft = (P+Q):1:(P+Q+L-1);
figure, stem(nyfft,yfft,'r.','linewidth',1.5,'markersize',20),
axis([-1 20 0 5]),
xlabel('{\itn}','fontsize',16,'fontname','times'),
ylabel('{\ity}_{FFT}[{\itn}]','fontsize',16,'fontname','times'),
title('System output computed using FFT & IFFT algorithms',...
    'fontsize',16,'fontname','times'),
% Find output by Convolving signals in the time-domain
yconv = conv(x,h); nyconv = (P+Q):1:(P+Q+N+M-2);
figure, stem(nyconv,yconv,'r.','linewidth',1.5,'markersize',20),
axis([-1 20 0 5]),
xlabel('{\itn}','fontsize',16,'fontname','times'),
ylabel('{\ity}_{conv}[{\itn}]','fontsize',16,'fontname','times'),
title('System output computed using DT convolution',...
    'fontsize',16,'fontname','times'),
set(findobj('type','line'),'linewidth',1.5,'markersize',16)
set(findobj('type','axes'),'linewidth',2)
```

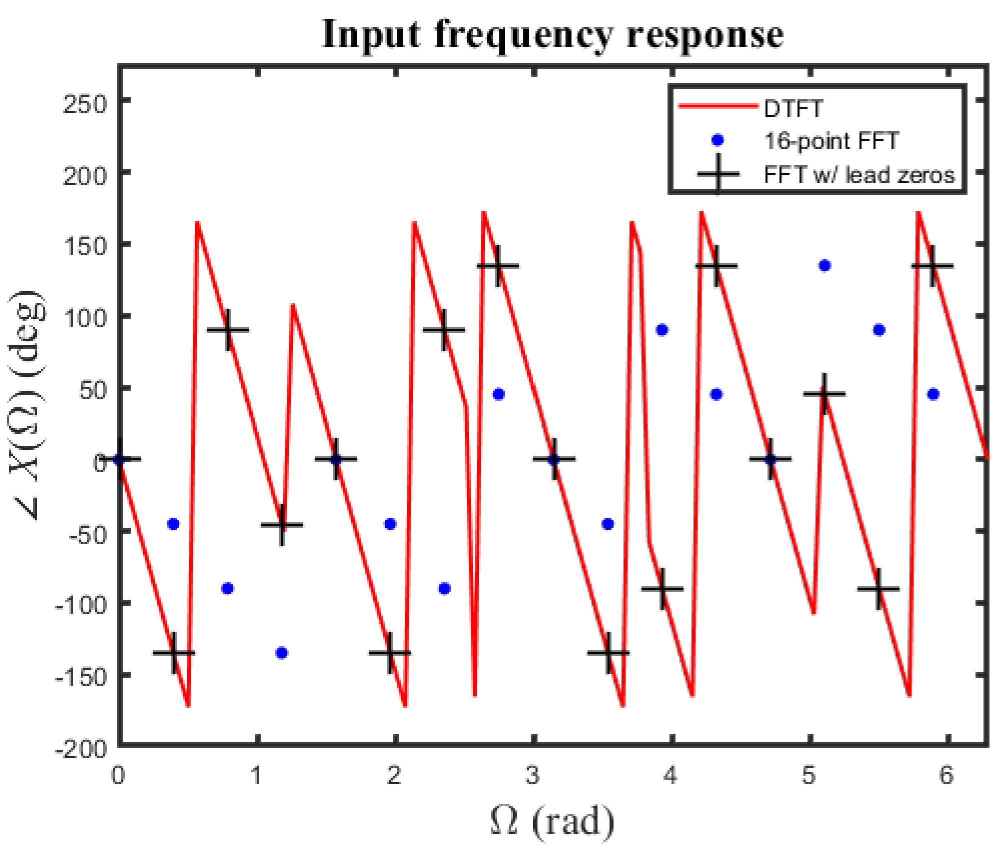

4-point Moving Average Filter unit-pulse response
h[n]
n
0.3
0.25
0.2
0.15
0.1
0.05
0
0
2
4
6
8
10
12
14
16
18
20


Input x[n] = 4p5[n-6]
x[n]
n
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
2
4
6
8
10
12
14
16
18
20

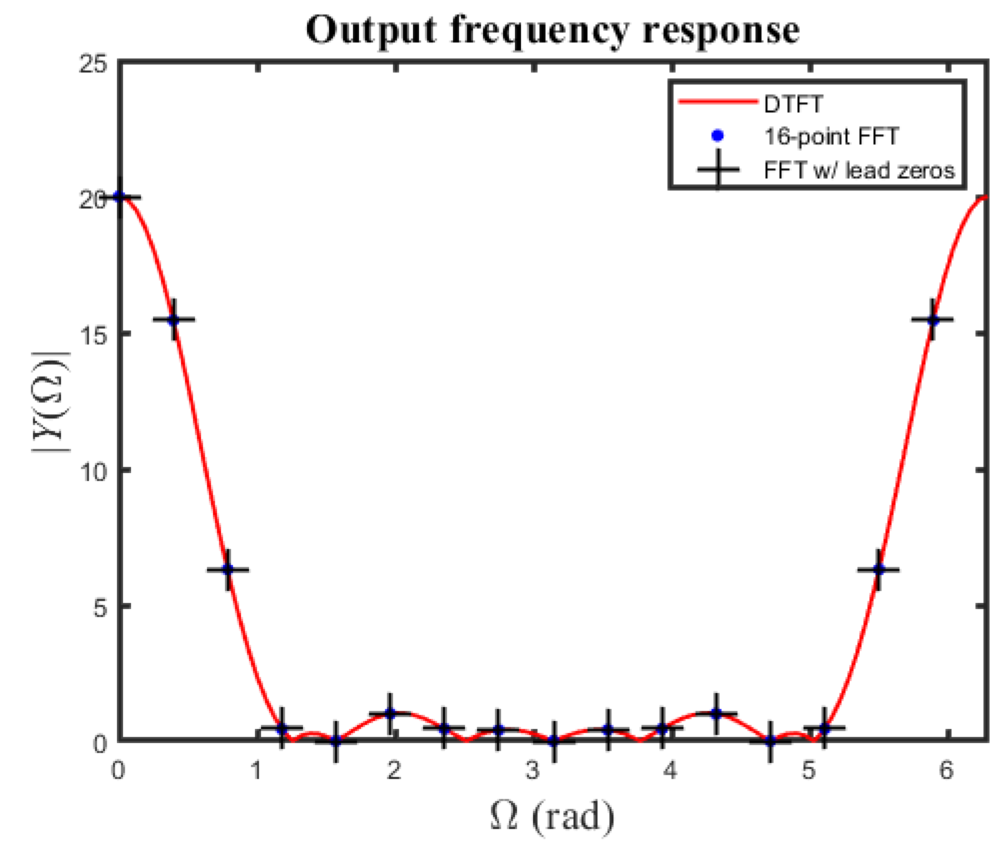

4-point Moving Average Filter frequency response
DTFT
16-point FFT
|H(Ω)|
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
Ω (rad)

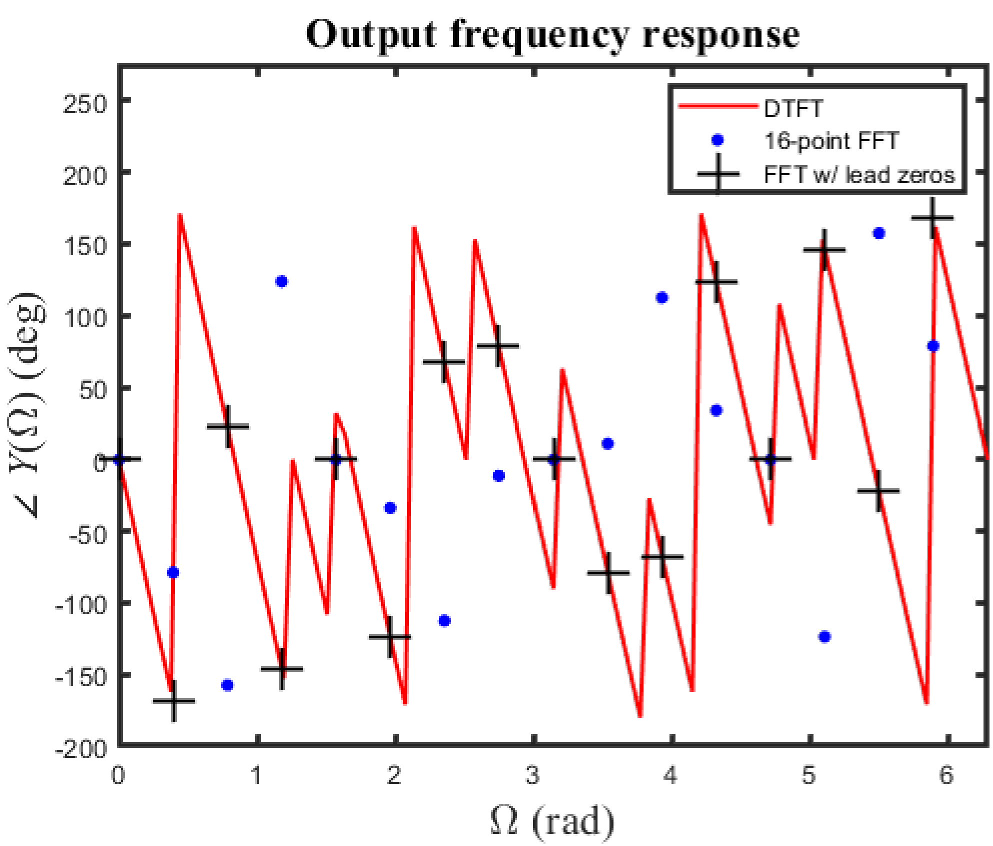

4-point Moving Average Filter frequency response
DTFT
16-point FFT
∠ H(Ω) (deg)
200
150
100
50
0
-50
-100
-150
-200
0
1
2
3
4
5
6
Ω (rad)



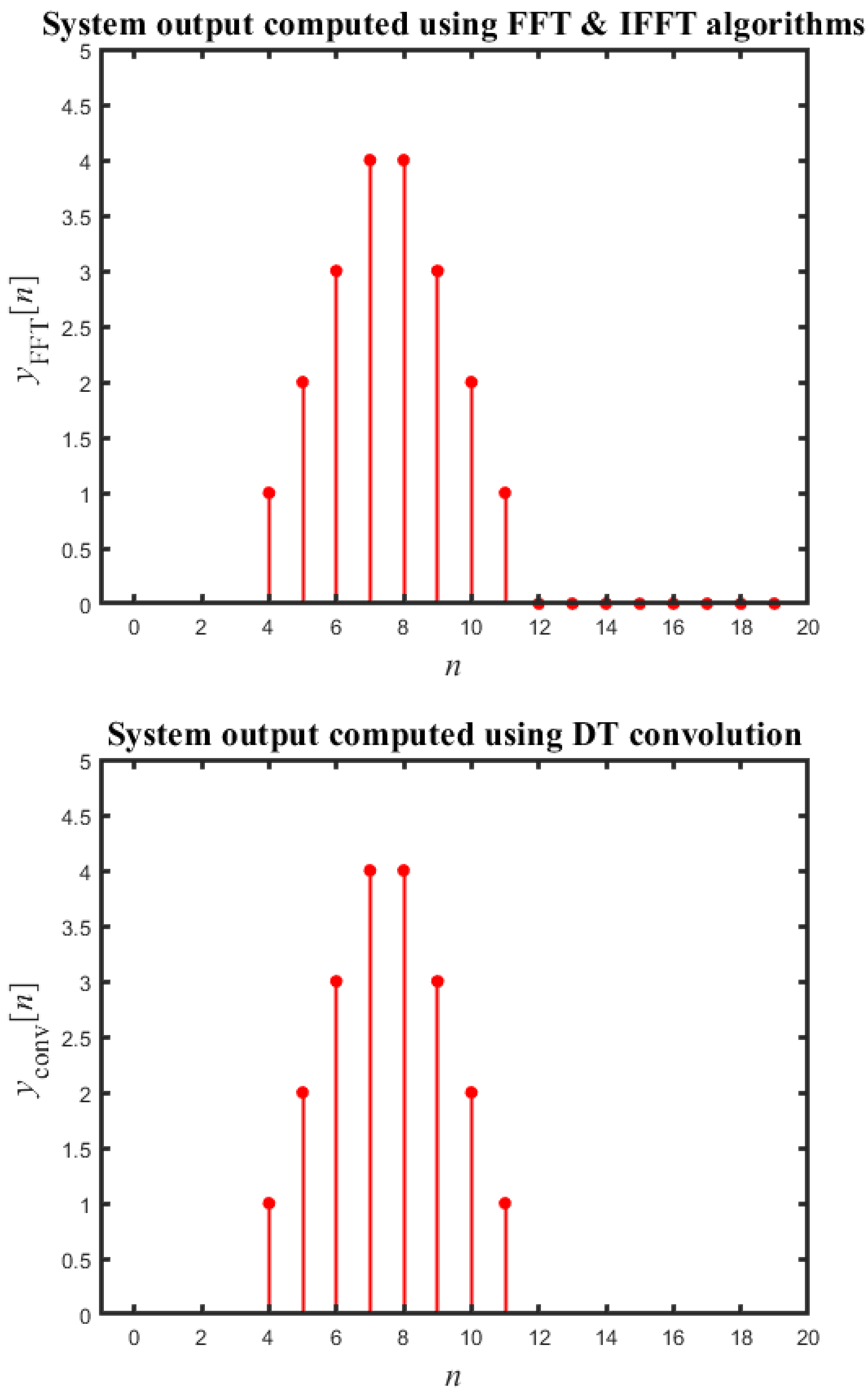


➢ **FFT (blue dot) doesn't account for the time-delay.**





- **FFT (blue dot) doesn't account for the time-delay.**


System output computed using FFT & IFFT algorithms
$y_{\mathrm{FFT}}[n]$
n
System output computed using DT convolution
$y_{\mathrm{conv}}[n]$
n