Homework 1

Multipath Propagation

Name: Jinhan Zhang

Class: ECE 251A Digital Signal Processing I

Date: 01/16/2017

Multipath Propagation

• Objective

Build an equalizer to restore the original signal in multipath propagation, comparing the effect of using different values of N in the process.

• Background

In underwater communication channel we get three impulses at the receiver for every impulse sent from the source. This phenomenon is mainly due to the reflection from the surface of water and the seabed. In this problem, we use the model $\mathrm{h[n]} = \delta[\mathrm{n}] - \frac{1}{2}\delta[\mathrm{n} - n_0]$ to represent what we get at the receiver which considers only 2 impulse received, and we want to build an equalizer to get the undistorted signal.

• Approach

Firstly, to deal with the case in which N=16, we do 16-point FFT on the received signal $\mathrm{h}[n]$ to get $\mathrm{H}[k]$. Secondly we get $H_1[k]$ by the inverse of $\mathrm{H}[k]$ and do IFFT on $H_1[k]$ to get $h_1[n]$. And thirdly we do convolution between $\mathrm{h}[n]$ and $h_1[n]$ by 32-point FFT (convolution in time domain equals to multiply in frequency domain) and IFFT to get the restored signal $h_2[n]$. And finally we do all above again in the case that N=64 and compare $h_1[n]$ and $h_2[n]$ in both time domain and frequency domain to see the difference with different N value.

- Results

Plot 1 just shows the series of our model, plot 2 shows that there are 4 poles and 4 zeros of H(z) respectively. Plot 3 and plot 4 show the fluctuation of amplitude and phase caused by multipath Propagation. We see from the plots that the multipath makes the constant amplitude and phase to be sine-like.

Plot 5 and 6 shows the 16-point FFT amplitude and phase of model series, the curves are unsmooth because that the number of FFT point is small. Plot 7 and 8 shows the amplitude and phase of $H_1(k)$ have opposite tendency compared with those of H(k), and that is due to $H_1(k)$ is got by the inverse of H(k). Plot 9 and 10 just show the series and zeros/poles of $h_1(n)$, while plot 11 and 12 show its properties in frequency domain. Plot 13 shows the series we restore by all above, and we see that there is a remaining distortion at n=16. The remaining distortion is mainly because that we only have $\mathrm{H(k)} * H_1(k) = 1$ at N distinct frequencies and we don't have $\mathrm{H}(e^{j\omega}) * H_1(e^{j\omega}) = 1$ for all ω. Plot 14 is the zeros and poles of its corresponding $H_2(k)$ after we truncate the trailing zeros of $h_2(n)$. Plot 15 and 16 show the result $h_1(n)$ do to equalize the channel that amplitude and phase fluctuate around the desired signal in a very small range.

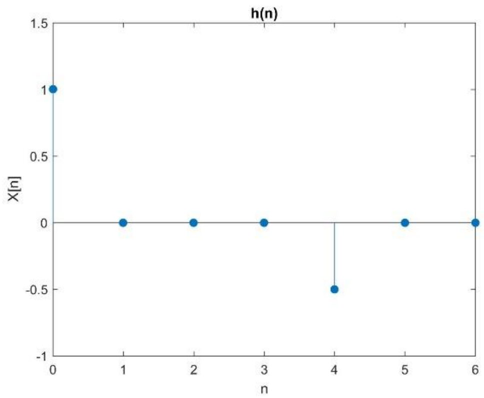
Plot 17-28 are obtained by the same process to Plot 5-16 by substitute N from 16 to 64. Compared with Plot 5-8, Plot 17-20 are smoother since we have more points. Again, Plot 21 and 22 just show the series and zeros/poles of $h_1(n)$ when N=64, while plot 23 and 24 show its properties in frequency domain which are smoother by the increment of points too. Plot 25 shows the series we restore by the new N value, which looks much better than that of N=16 and we can see any obvious remaining distortion in the plot. And Plot 26 shows the zeros and poles of its corresponding $H_2(k)$ after we truncate the trailing zeros. Plot 27 and 28 show the result $h_1(n)$ do to equalize the channel. And at this time we can't see any fluctuation when we use the same y-axis range with N=16. But when we blow up the regional part, we can still see fluctuation but within a much smaller range compared to N=16 case.

- Summary

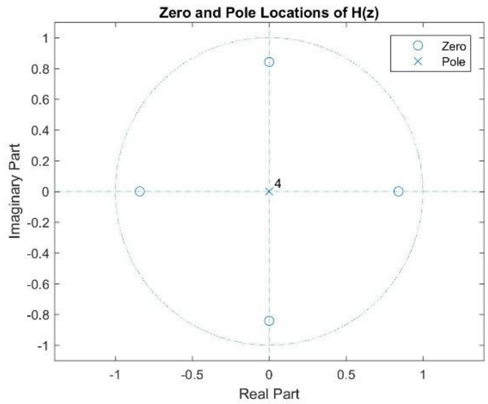
However, from the two cases above, we can see that with a bigger N, we will have a better equalizer to restore signals distorted by multipath. And the remaining distortion usually exists even if it can be in a tiny orders of magnitude.

- Plots

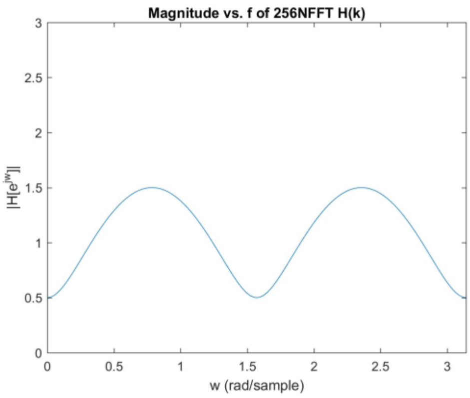
Part A:



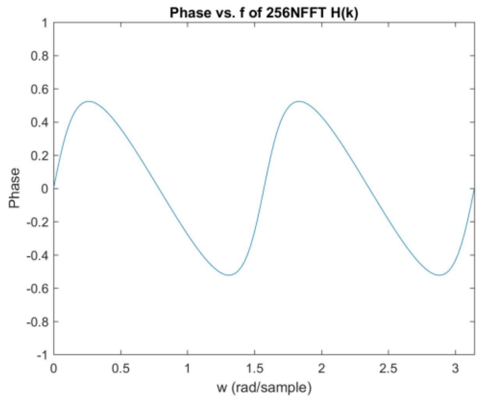
Plot 1



Plot 2

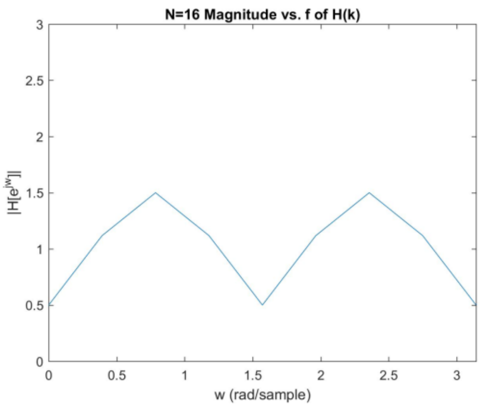


Plot 3

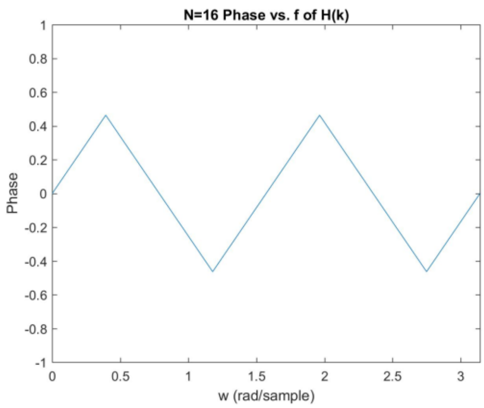


Plot 4

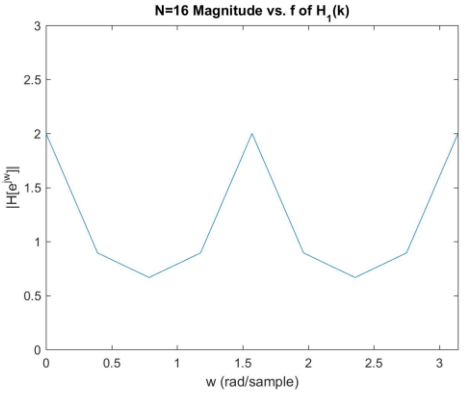
Part B:



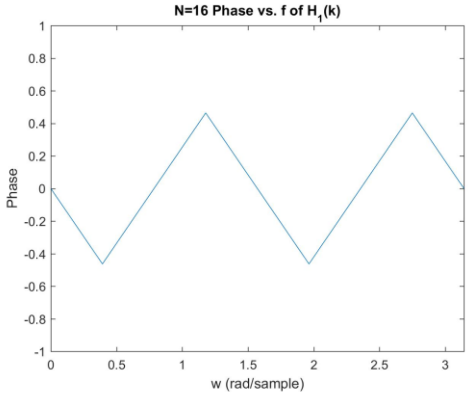
Plot 5



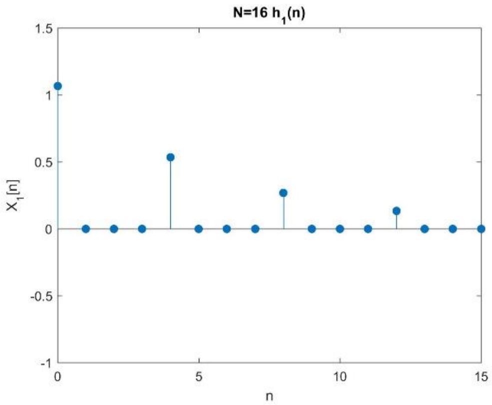
Plot 6



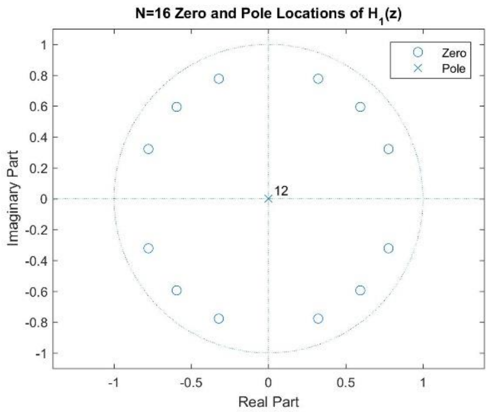
Plot 7



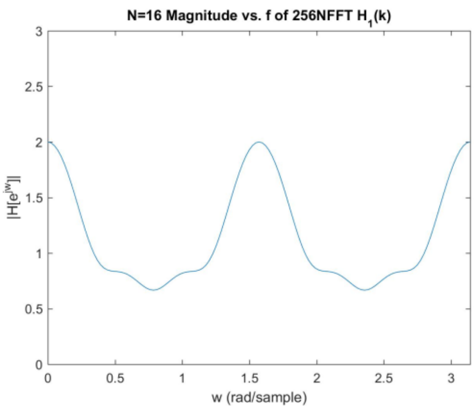
Plot 8



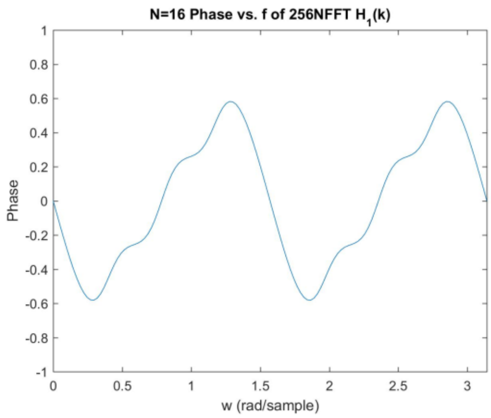
Plot 9



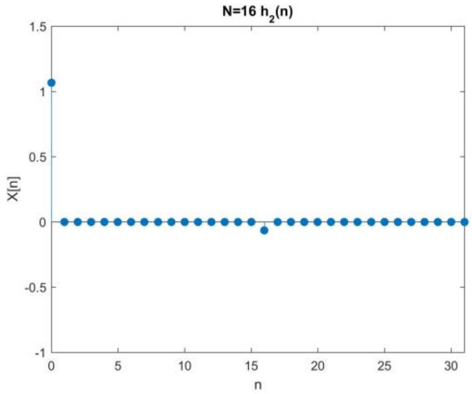
Plot 10



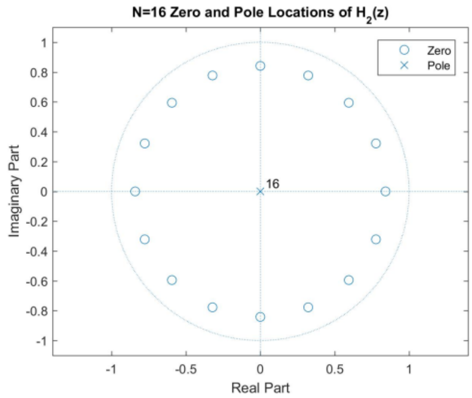
Plot 11



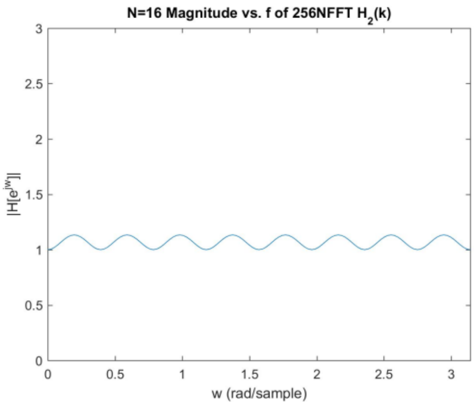
Plot 12



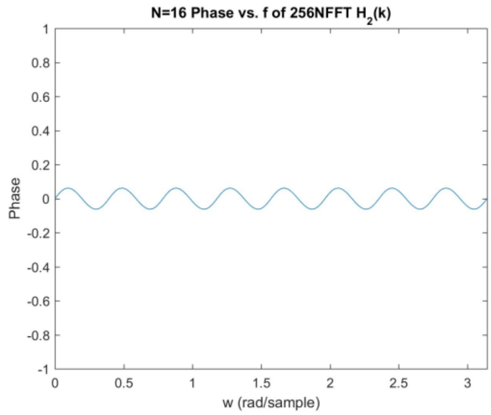
Plot 13



Plot 14

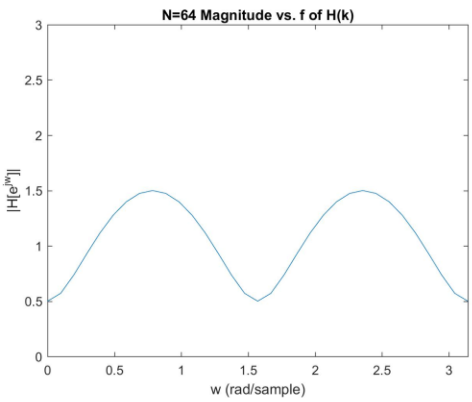


Plot 15

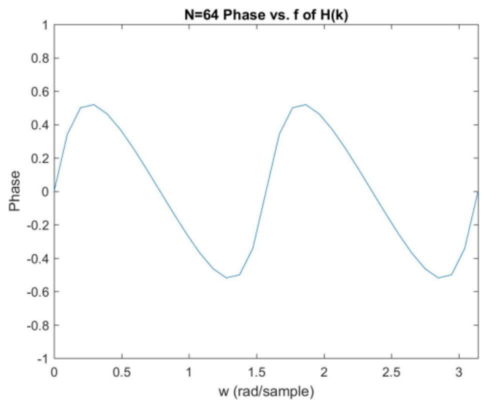


Plot 16

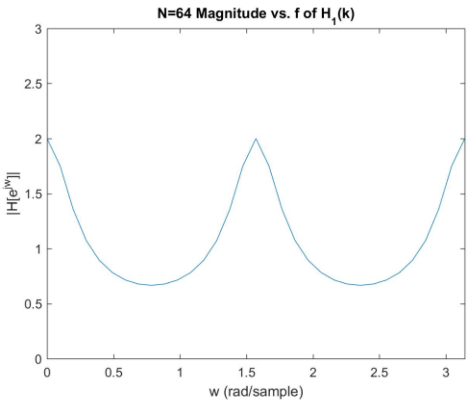
Part C:



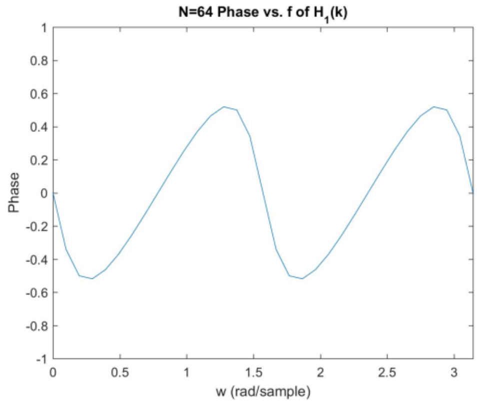
Plot 17



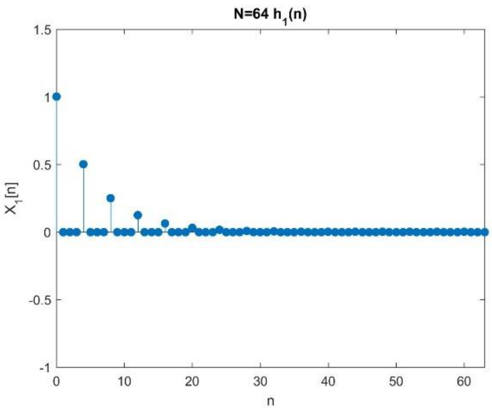
Plot 18



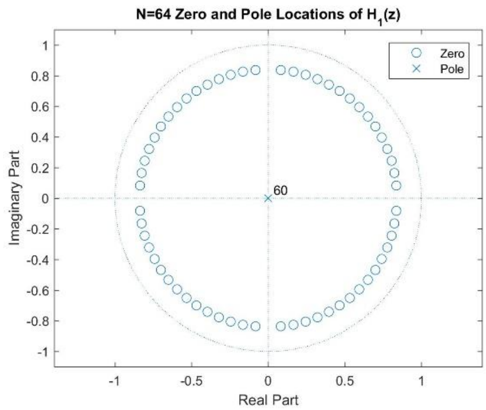
Plot 19



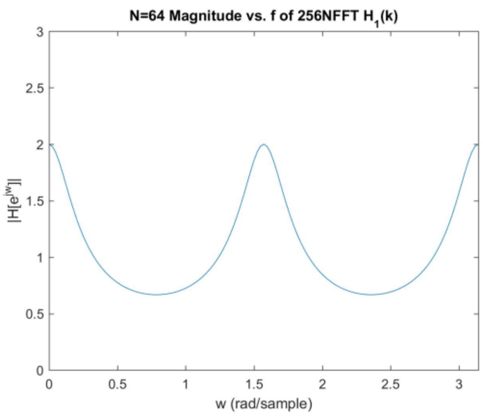
Plot 20



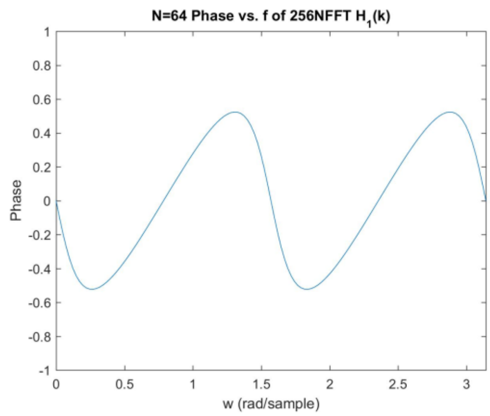
Plot 21



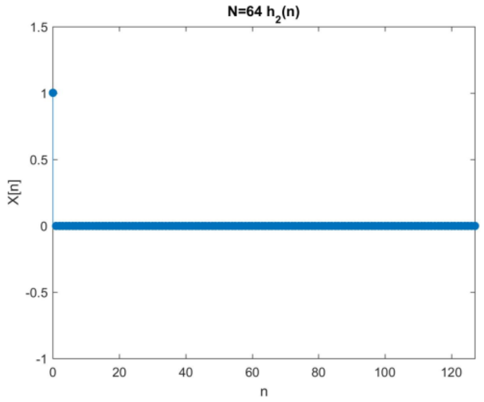
Plot 22



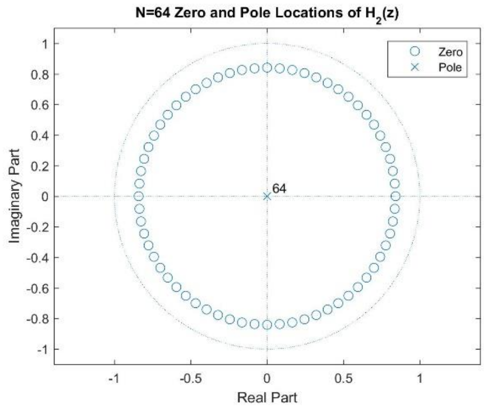
Plot 23



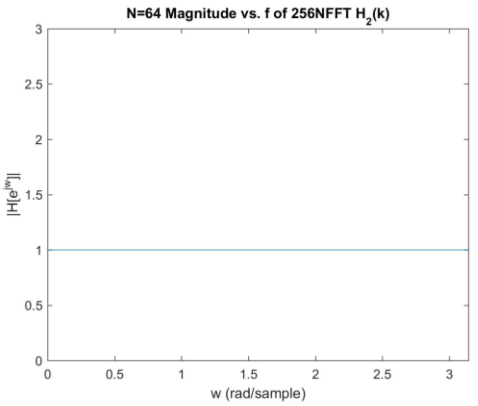
Plot 24



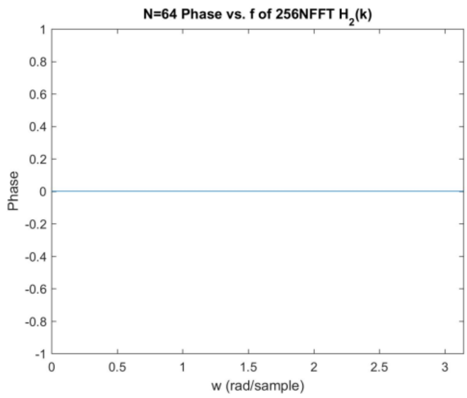
Plot 25



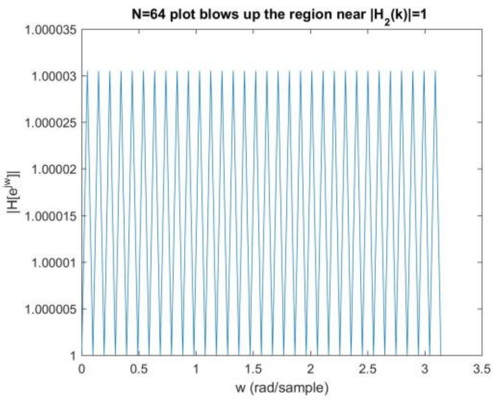
Plot 26



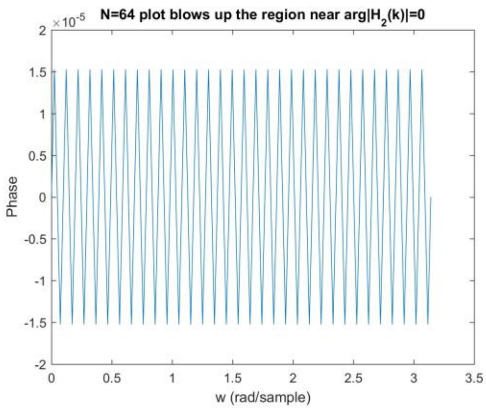
Plot 27



Plot 28



Plot 29



Plot 30

- Appendix

Function plot_series:

```
function [ ] = plot_series( h )

figure;
x = 0: length(h)-1;
stem(x,h,'filled');axis([0 length(h)-1 -1 1.5]);
xlabel('n');ylabel('X[n]');title('h(n)');

figure;
zplane(h);legend('Zero','Pole');
title('Zero and Pole Locations');

end
```

Function NFFT_plot:

```
function [ ] = NFFT_plot( f , N )

figure;
t=0:N/2;
plot(t/N*2*pi,abs(f(1:N/2+1)));
axis([0 pi 0 3]);
xlabel('w');ylabel('H[e^{jw}]');title('Magnitude vs.
f');

figure;
angle=phase(f);
plot(t/N*2*pi,angle(1:N/2+1));
axis([0 pi -1 1]);
xlabel('w');ylabel('Phase');title('Phase vs. f');

end
```

Call script:(just substitute N=16 to N=64 for Part C):

```
%%%%%%%%%%%%%%Part A%%%%%%%%%%%%%%%%%%%%%%%%
h=[1,0,0,0,-0.5,0,0];
plot_series( h );
F=fft(h,256);
NFFT_plot( F , 256 );
%%%%%%%%%%%%%%Part B%%%%%%%%%%%%%%%%%%%%%%%%
%%%1
N=16;
```

```
H=fft(h,N);
NFFT_plot( H , N );

H1=1./H;
NFFT_plot( H1 , N );
%%%2
h1=ifft(H1,N);
plot_series( h1 );
%%%3
F=fft(h1,256);
NFFT_plot( F , 256 );
%%%4
h2=ifft(fft(h,2*N).*fft(h1,2*N));
%%%5
plot_series(h2(1:N+1));
%%%6
F=fft(h2,256);
NFFT_plot( F , 256 );
```