z as shift (delay) operator

- We saw that in deriving the transfer function and frequency response of an averaging filter that it involved multiplying the input by $e^{-j\omega}$

$$Y(k) = .5e^{j\omega k} + .5e^{j\omega k} \cdot e^{-j\omega}$$

Means shift input by ω

- multiplying a phasor X by $e^{-j\omega}$ is equivalent to shifting that phasor later in time by one sample, which shifts its phase by -ω radians, which is the angle that the phasor rotates in one sample: $$e^{j\omega k} \cdot e^{-j\omega} = e^{j(\omega k - \omega)}$$

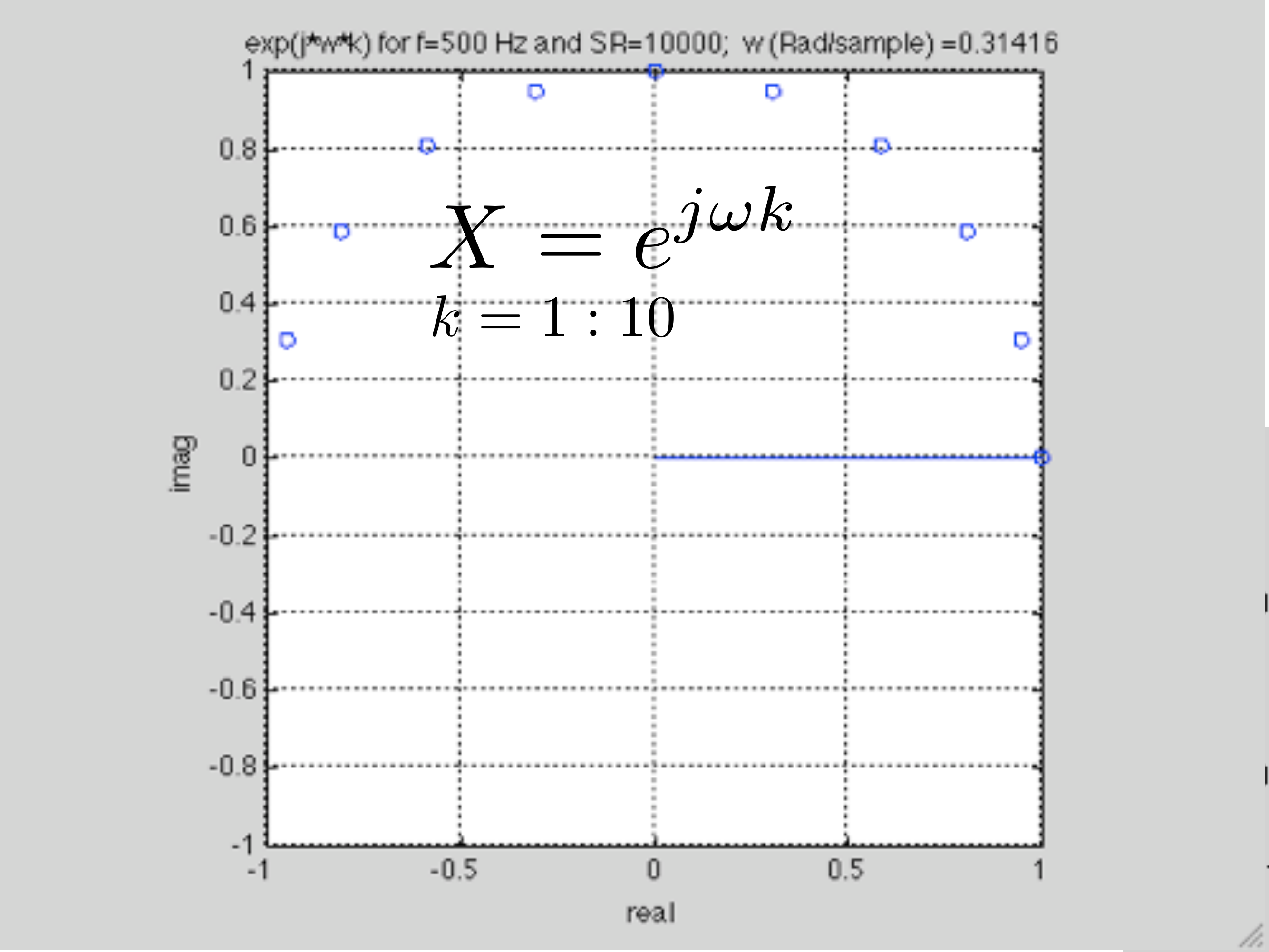
- Useful to define an shift operator z = $e^{j\omega}$

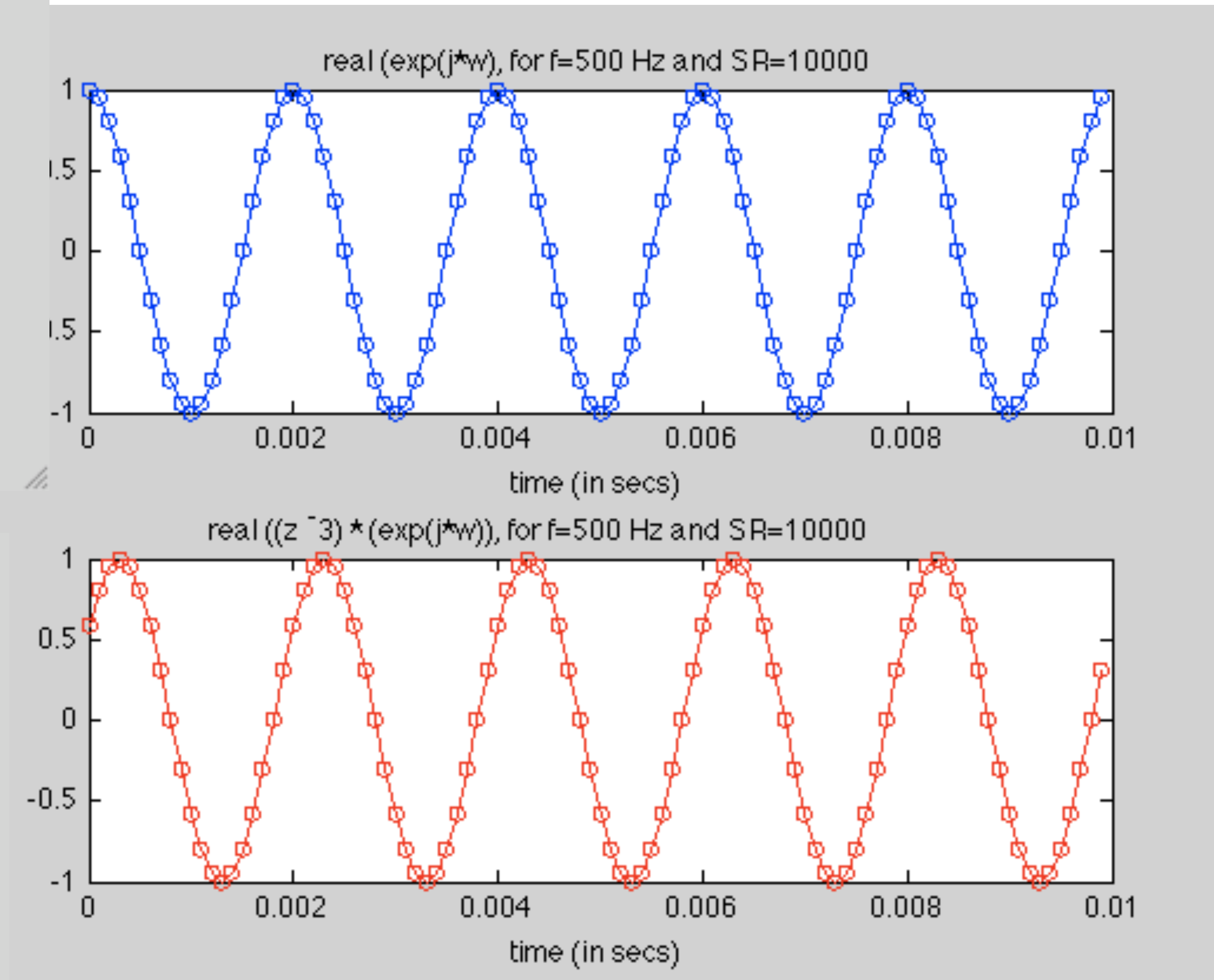
- If X is a phasor, of frequency ω radians/sample, then:
  - z*X shifts X one sample earlier, phase shift = ω radians
  - z^{-1}*X delays X by one sample, phase shift = -ω radians
  - z^{-3}*X delays X by 3 samples, phase shift = -3ω radians

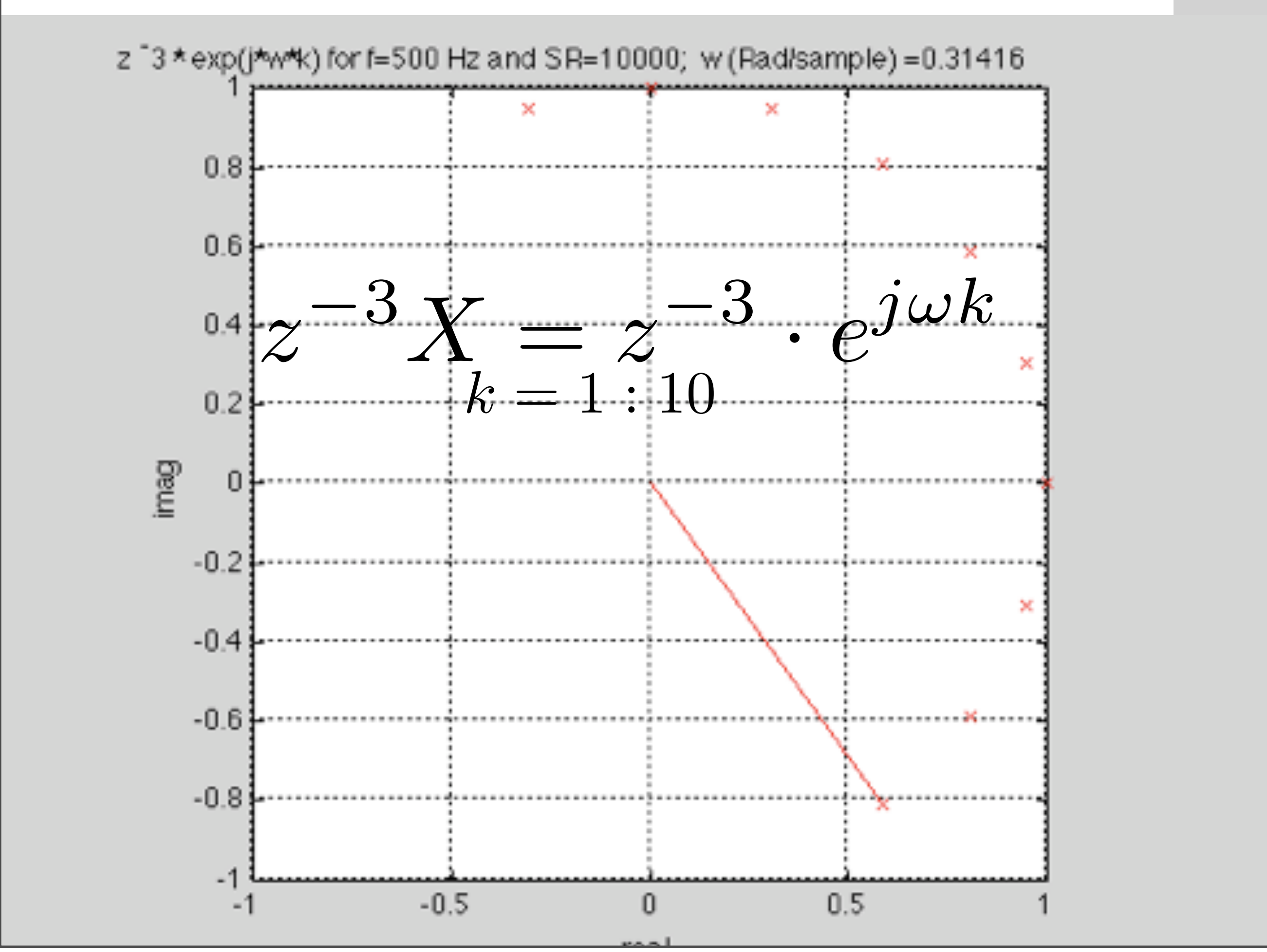
Example of shift

sr=10000 Hz; f=500 Hz

$\omega = \pi/10$ radians/sample







```matlab
% SHIFT_DEMO
% This demonstrates that a phasor is shifted in
% time by N samples when it is multiplied by z .^ N
% Also, the phase shift corresponds to w*n
% Where w is the frequency in radians per sample.




% Generate unshifted phasor:
srate = input ('Enter sampling rate in Hz: ')
f = input ('Enter sinusoid frequency in Hz: ')
N = input ('Enter power of z (shift operator): ')
dur = .5;
t = 0:1/srate:dur;
k = 0:dur*srate;            % k is sample number
w = 2*pi*f/srate;           % w is in radians per sample
unshifted = exp(j*w*k);

% Plot first ten samples in complex plane
% Begin by drawing a line from the origin to the first point.
figure(1)
clf;  %clears figure
first = [0 unshifted(1)];
plot (real(first), imag(first), 'b-');
axis ('square');
axis ([-1,1,-1,1]);
hold on
xlabel ('real')
ylabel ('imag')
grid
```

```
for i = 1:10
  plot (real(unshifted (i)), imag(unshifted(i)), 'o')
      pause(1)
end
title ([' exp(j*w*k) for f=',num2str(f), ' Hz and SR=', num2str(srate),         ';  w (Rad/sample) =',
num2str(w)])
pause

% Now shift signal by muliplying by z ^ N
z = exp(j*w) ^ N;
shifted = unshifted * z;

% Plot first ten shifted samples in complex plane
first = [0 shifted(1)];
plot (real(first), imag(first), 'r-');
for i = 1:10
  plot (real(shifted (i)), imag(shifted(i)), 'rx')
      pause(1)
end
title ([' z ^', num2str(N), ' * exp(j*w*k) for f=',num2str(f), ' Hz and SR=', num2str(srate),         ';
w (Rad/sample) =', num2str(w)])
pause

hold off
```

```
% Now plot unshifted and shifted signals as a function of time
y = real(unshifted);
figure(2)
clf;  %clears figure
axis ('normal')
subplot (211), plot (t(1:100), y(1:100), 'or-')
title (['real (exp(j*w), for f=',num2str(f),' Hz and SR=', num2str(srate)])
xlabel ('time (in secs)')

soundsc (y,srate)
pause (1)

y = real(shifted);
axis ('normal')
subplot (212), plot (t(1:100), y(1:100), 'or-')
title (['real ((z ^', num2str(N), ') * (exp(j*w)), for f=',num2str(f),' Hz and SR=',
num2str(srate)])
xlabel ('time (in secs)')

soundsc (y,srate)
```

Filtering Phasors using z

- Filter averaging 2 points:

$$Y = .5X + .5z^{-1}X$$

- This lets us see easily that

$$Y = (.5 + .5z^{-1})X$$

- So, $H(z) = (.5 + .5z^{-1})$

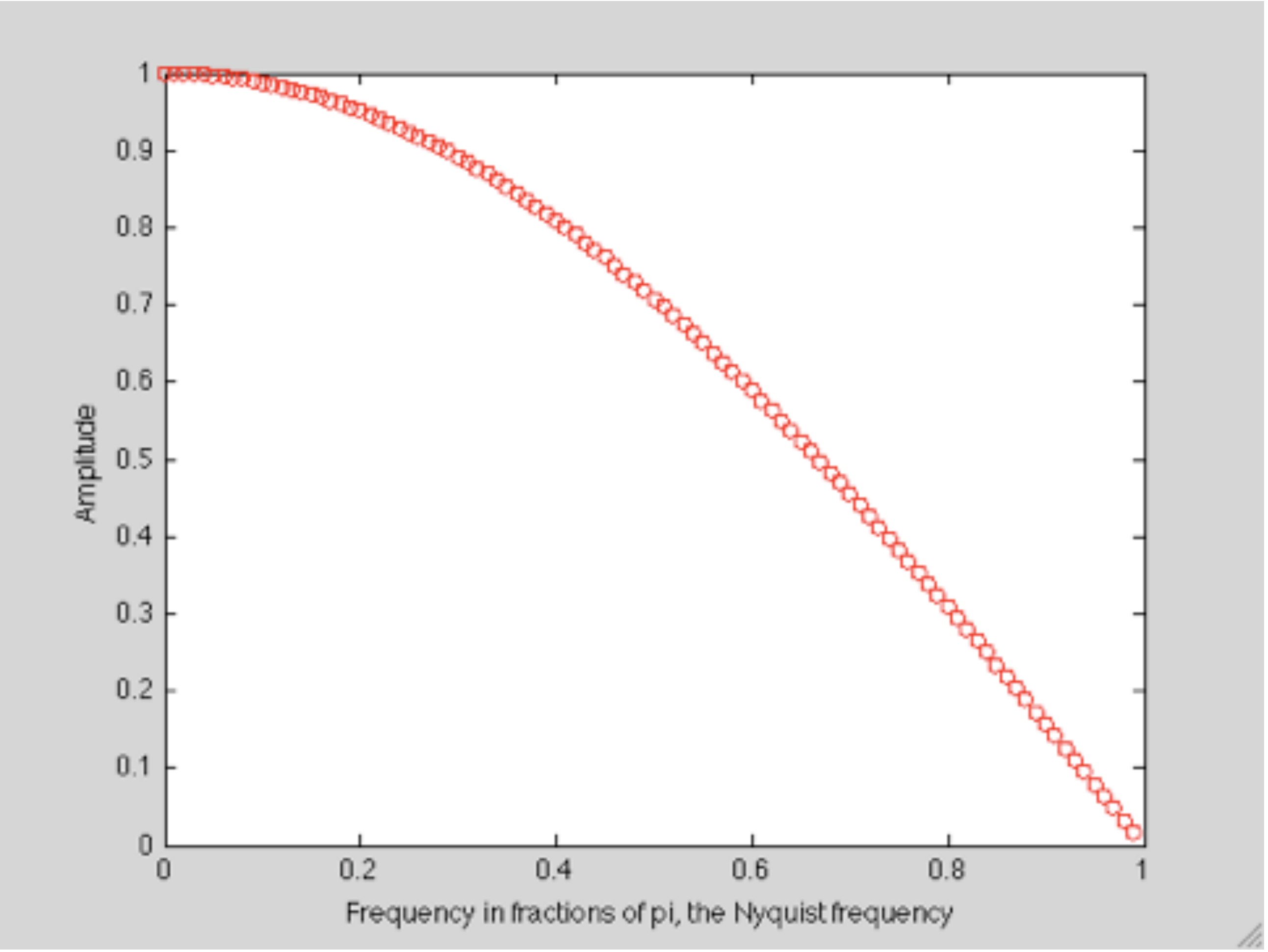
- And the frequency response *H(ω)* can be determined by evaluating for values of z on the unit circle *(R=1, ω=1:step:π)*

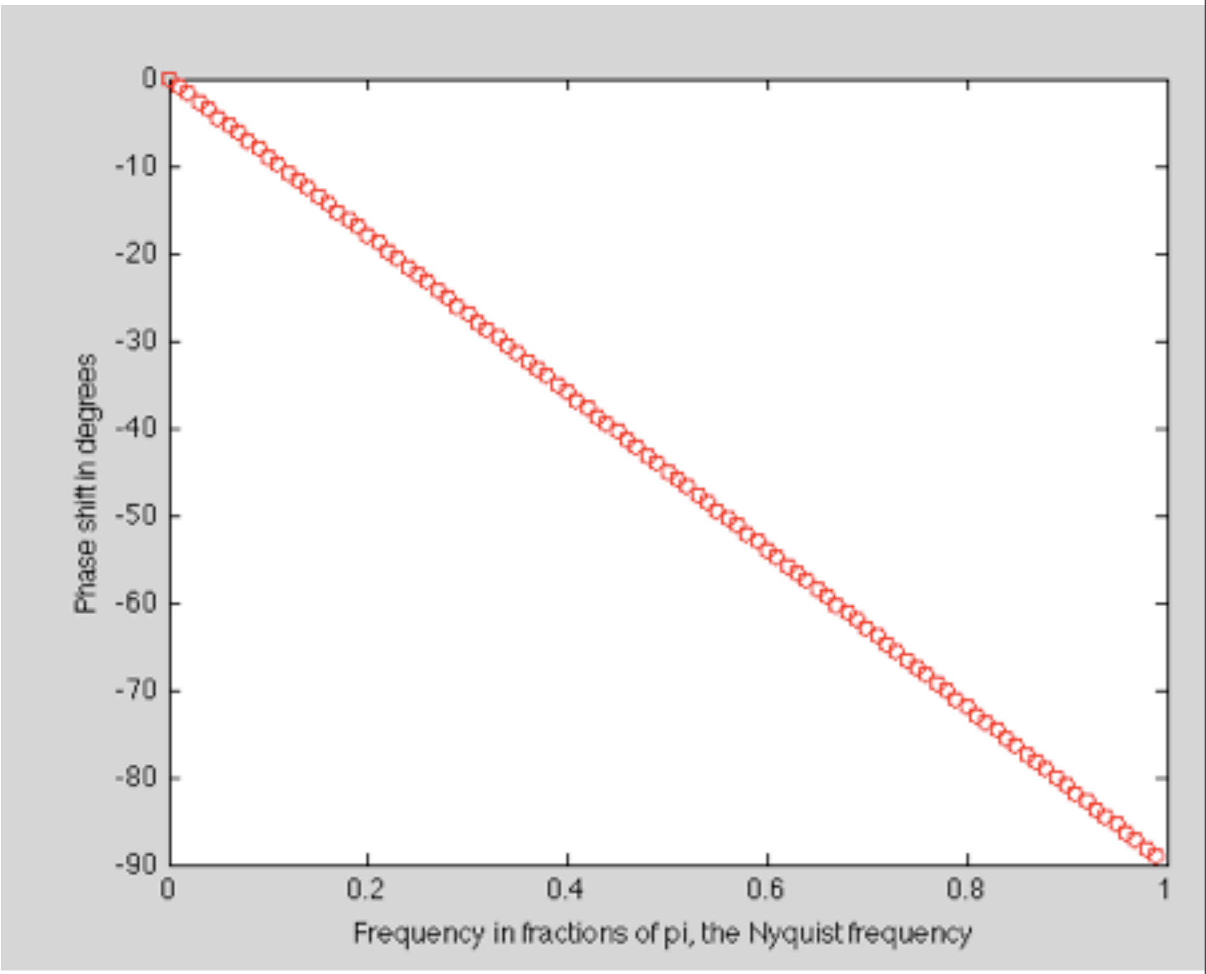
z-based Code for plotting amplitude, phase response

```
% Z_RESP
% Use MATLAB to sketch magnitude and phase of H(w)
% for the simple average2pts filter.


%
% Set up a vector of test frequencies to plot, from
% zero to the Nyquist frequency
% Frequencies are represented in radians per sample
nfreqs = 100;
w = 0:pi/nfreqs:pi-(pi/nfreqs);
z = exp(j*w);
%
% Give the expression for the value of the transfer function
% H as a function of frequency
% This will compute a vector of H values, one for each value of w.
H = .5*z.^0+ .5*z.^(-1);
%
% Plot the Amplitude of H as a function of frequency:
figure (1)
plot (w/pi,abs(H),'or')
xlabel ('Frequency in fractions of pi, the Nyquist frequency')
ylabel ('Amplitude')
%
% Plot the phase of H as a function of frequency
figure(2)
plot (w/pi,angle(H)*180/pi, 'or')
xlabel ('Frequency in fractions of pi, the Nyquist frequency')
ylabel ('Phase shift in degrees')
```

Amplitude Response

Phase Response





Transfer function for filter with M coefficients

$$Y = b_1 X + b_2 z^{-1} X + b_3 z^{-2} X + ... + b_M z^{-(M-1)} X$$

- Each term $b_i z^{-(i-1)} X$ represents X delayed by i-1 samples and weighted by b_i.

$$Y = (b_1 + b_2 z^{-1} + b_3 z^{-2} + ... + b_M z^{-(M-1)}) X$$

$$H(z) = b_1 + b_2 z^{-1} + b_3 z^{-2} + ... + b_M z^{-(M-1)}$$

$$Y = H(z) X$$

- Transfer function is polynomial in negative powers of z

$$X \longrightarrow H \longrightarrow Y$$

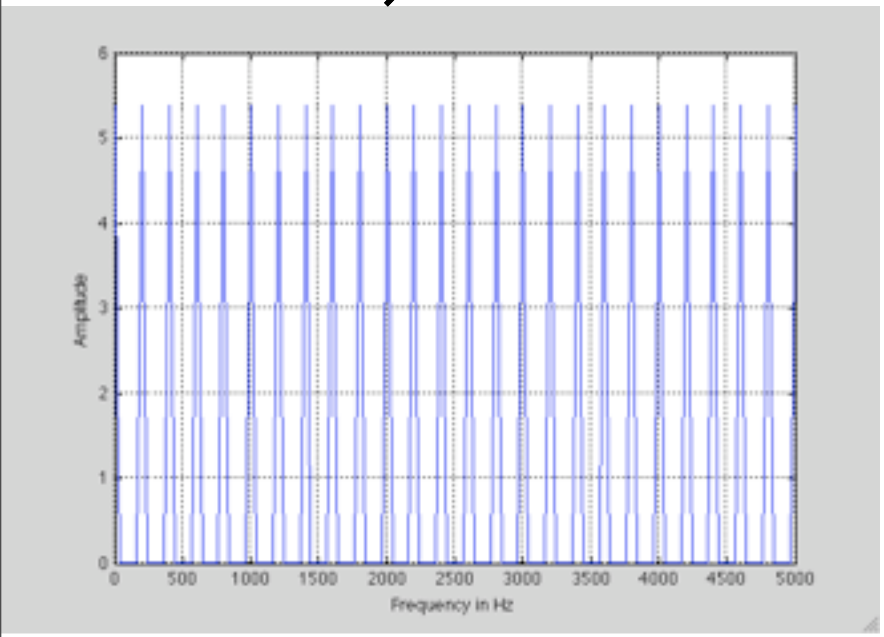
Example

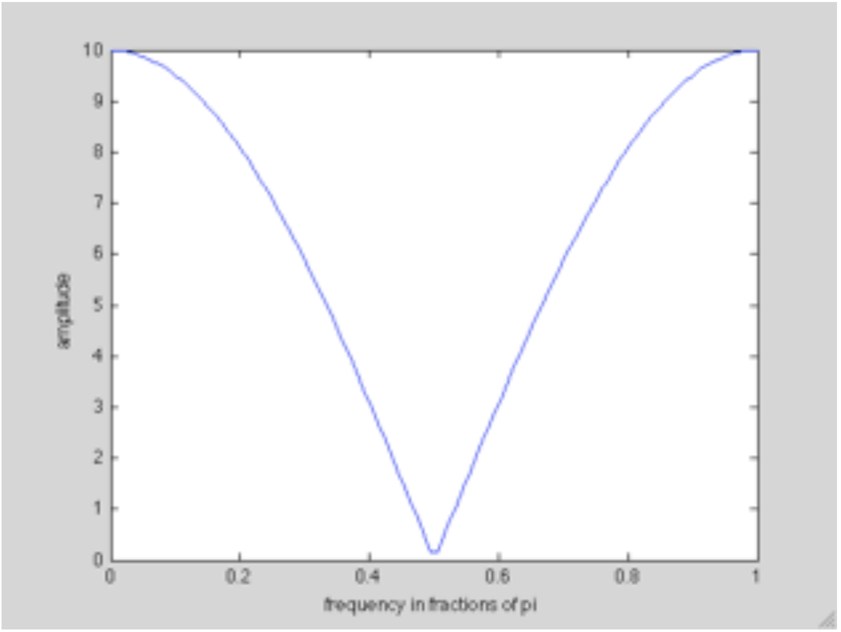
$$X = 200\text{ Hz buzz} \longrightarrow H = .5 + .5z^{-2} \longrightarrow Y$$

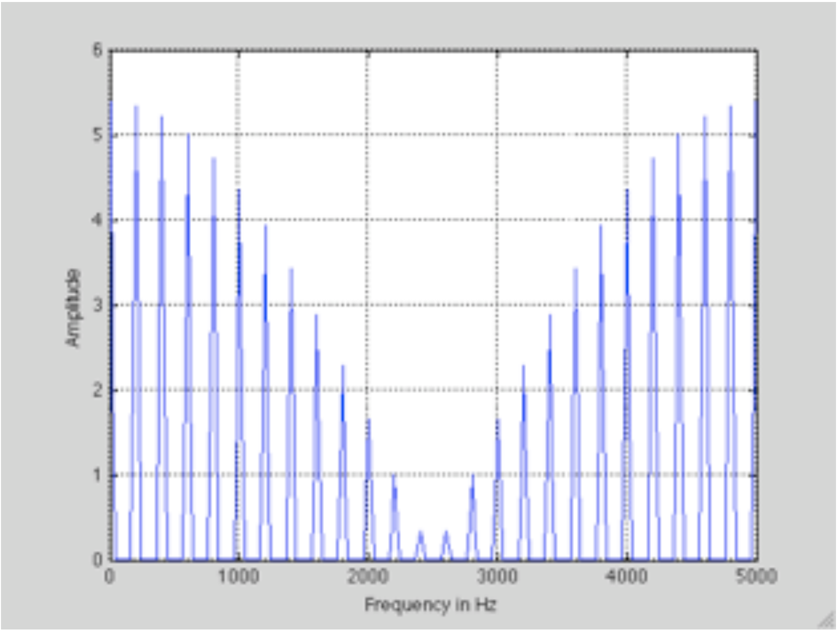
spectrum

$|H(\omega)|$

spectrum



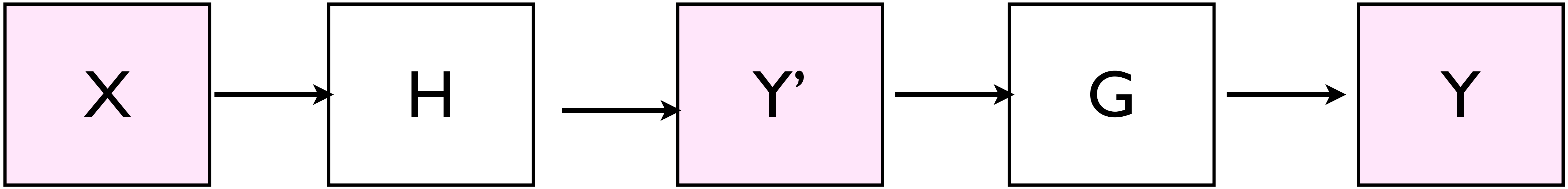




Frequency response for filter with M coefficients

```
function freqresp(b)

% Compute the frequency response of a moving average filter.
% b is a vector containing the coefficients
% outputs:
% mag = magnitude of transfer function at 100 test values of w
% phase = phase of transfer function at 100 test values of w

% first find the number of coefficients (filter order)
m = length(b);

% fill a vector of test frequencies to plot
nfreqs = 100;
w = linspace(0, pi, nfreqs);

% Compute a vector of z from w;
z = exp(j*w);

% Create a vector H that will eventually contain
% the transfer function.
% First set all the elements of this vector to 0
H = zeros(1,nfreqs);

% Compute H by summing the terms of the polynomial in z
for i=1:m
   H = H + b(i) .* z .^ (-(i-1));
end

% add your code here
% don't forget to add something to the function declaration command at the
% top to return mag and phase as outputs.
```

Cascading Filters

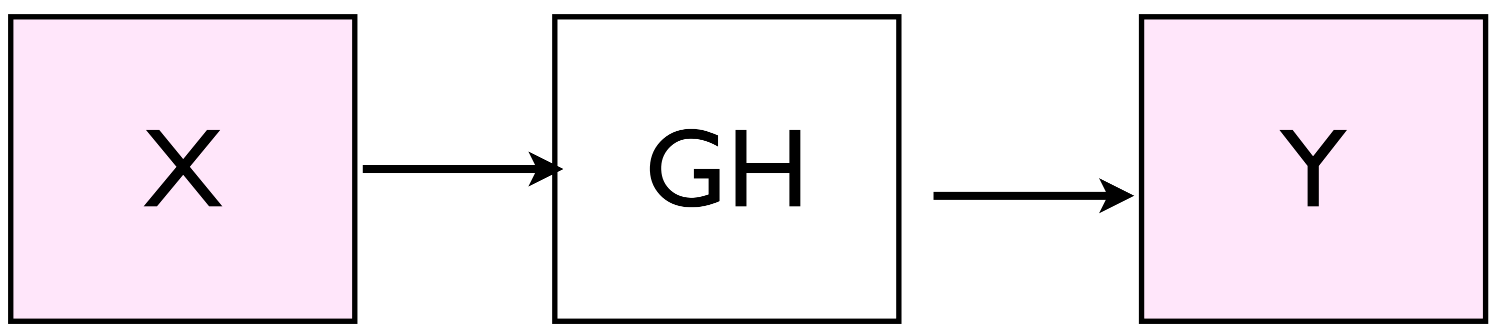


$$Y' = H(z)X \qquad\qquad Y = G(z)Y'$$

$$Y = G(z)\cdot(H(z)X) = (G(z)\cdot H(z))X$$

$$GH(z) = G(z)H(z)$$

$$Y = GH(z)X$$



If: $H(z) = h_1 + h_2 z^{-1} \quad G(z) = g_1 + g_2 z^{-1} + g_3 z^{-2}$

$$G(z)H(z) = h_1g_1 + h_1g_2z^{-1} + h_1g_3z^{-2} + \\ h_2g_1z^{-1} + h_2g_2z^{-2} + h_2g_3z^{-3}$$

$$= h_1g_1 + (h_1g_2 + h_2g_1)z^{-1} + (h_1g_2 + h_2g_2)z^{-2} + h_2g_3z^{-3} +$$

Amplitude Response of cascade

$$HG(\omega) = H(\omega)G(\omega)$$

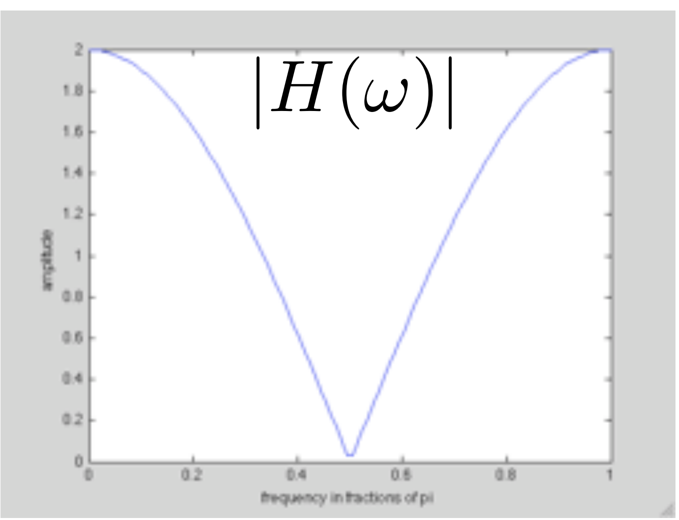
$$HG(\omega) = |H(\omega)|e^{ArgH(\omega)} \cdot |G(\omega)|e^{ArgG(\omega)}$$

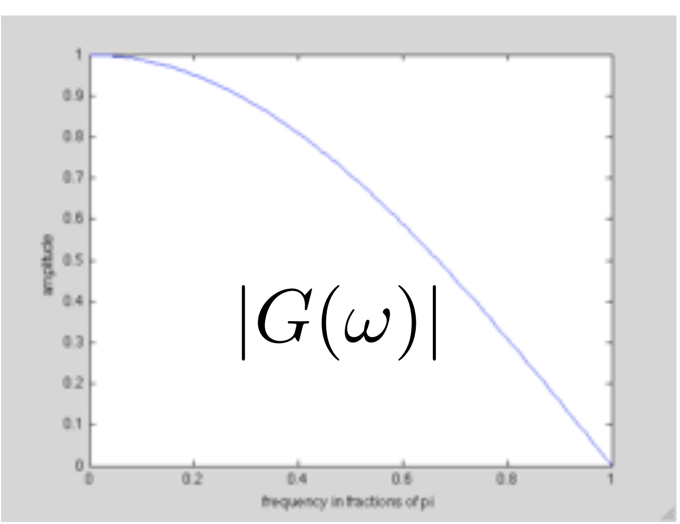
$$= |H(\omega)|G(\omega)|e^{ArgH(\omega)+ArgG(\omega)}$$

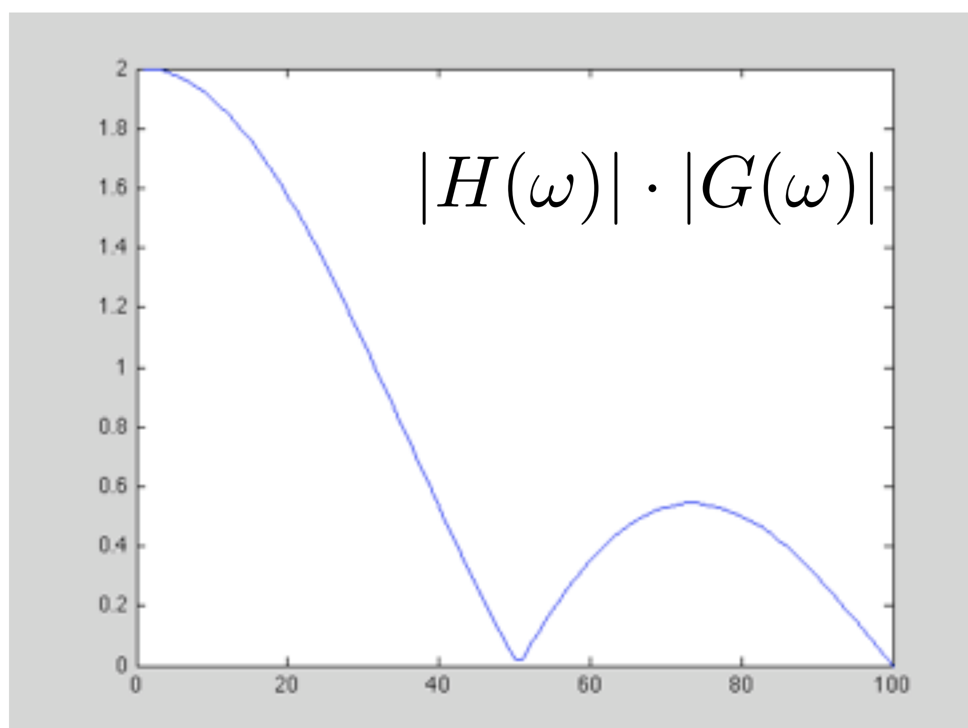
So $|HG(\omega) = |H(\omega)|G(\omega)|$

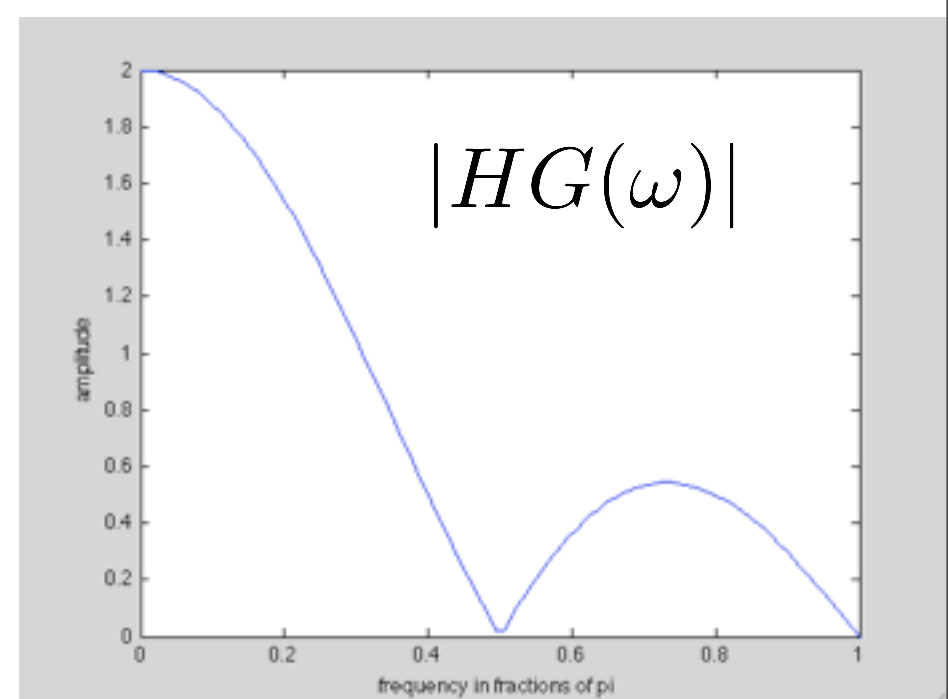
```
>> H = [1 0 1];
>> G =[.5 .5];
 >> HG = conv(H,G)   convolution
HG =
   0.5000    0.5000    0.5000    0.5000
```

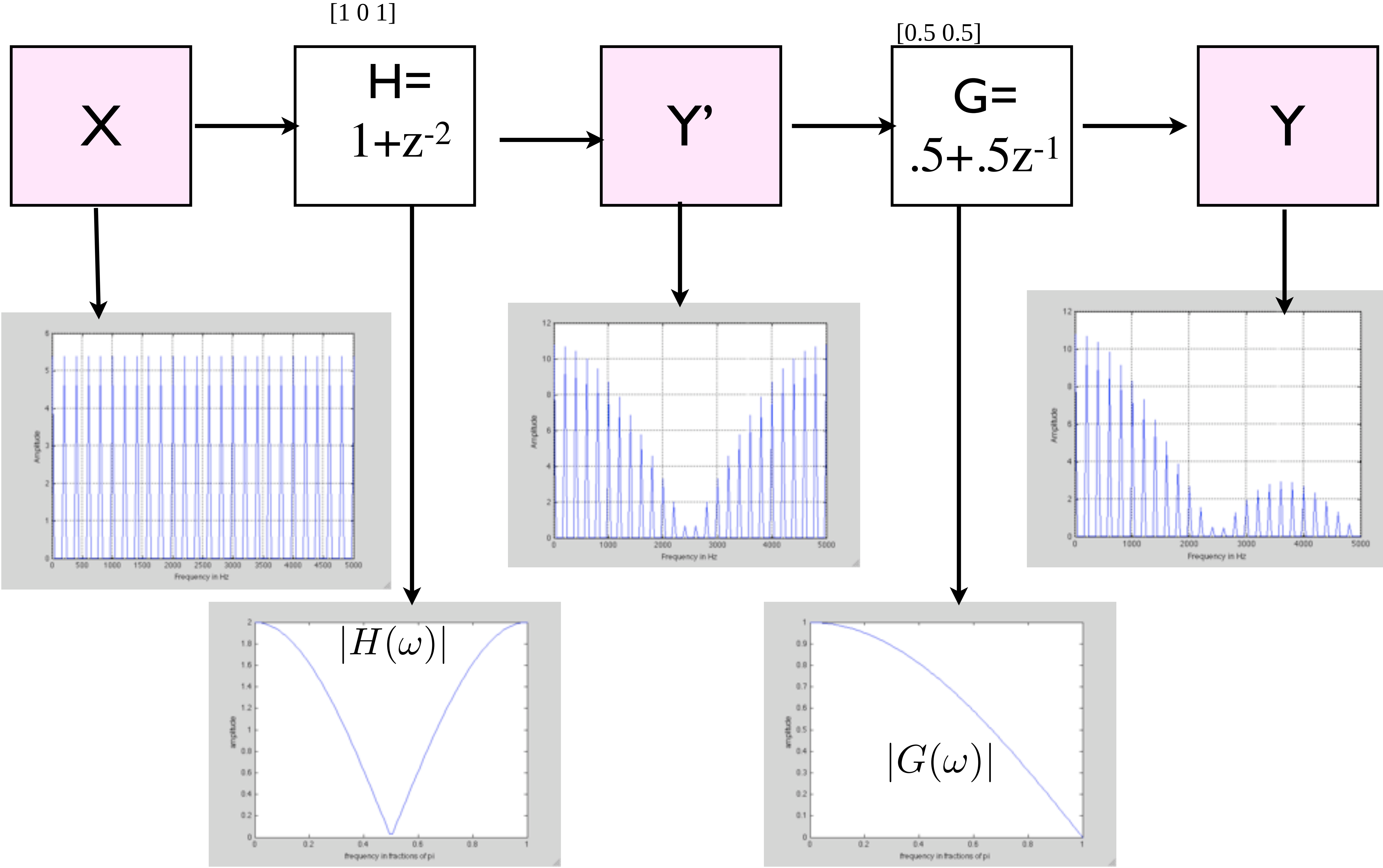
두 개의 filter
합친 coefficient 만드려면 conv!









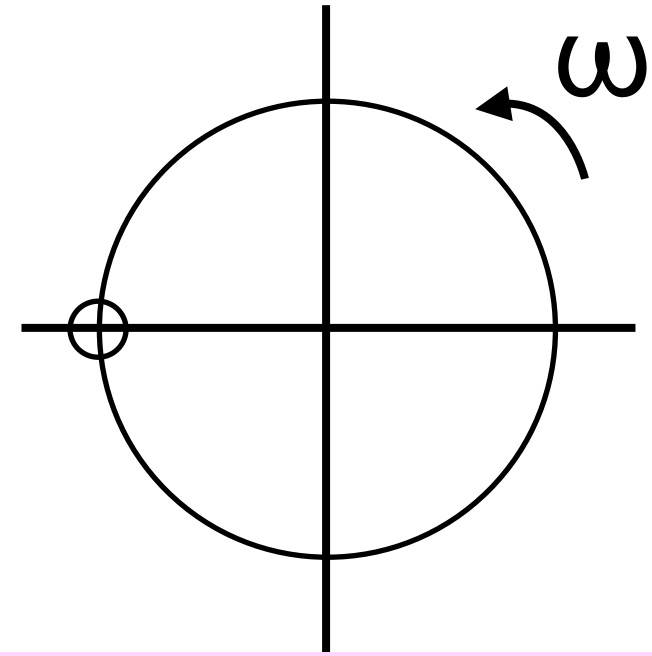

X
[1 0 1]
H=
$1+z^{-2}$
Y’
[0.5 0.5]
G=
$.5+.5z^{-1}$
Y
Amplitude
Frequency in Hz
$|H(\omega)|$
amplitude
frequency in fractions of pi
$|G(\omega)|$

Zeros in the magnitude response

$$H(z) = .5 + .5z^{-1} = .5 + \frac{.5}{z} = \frac{.5z}{z} + \frac{.5}{z}$$

$$H(z) = \frac{.5z + .5}{z}$$

$$|H(\omega)| = \frac{|.5z + .5|}{|z|}$$



Since $z=e^{j\omega}$, $|z| = 1$

$$|H(\omega)| = |.5z + .5|$$

= 0 at the roots of this polynomial

In general, the roots may not be on the unit circle, but the frequencies corresponding to the angles of the roots will always have minima of H(ω).

$$.5z + .5 = 0$$
$$.5z = -.5$$
$$z = -1$$

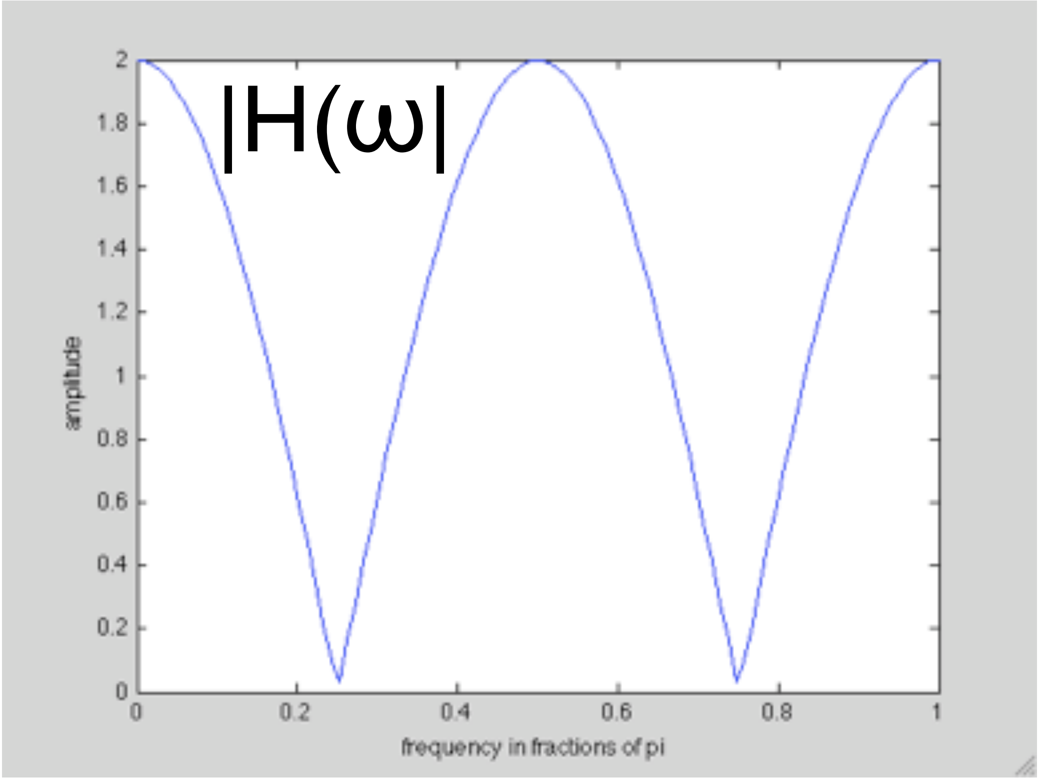
In this case, the angle of the root=π, which corresponds to this being a low -pass filter.

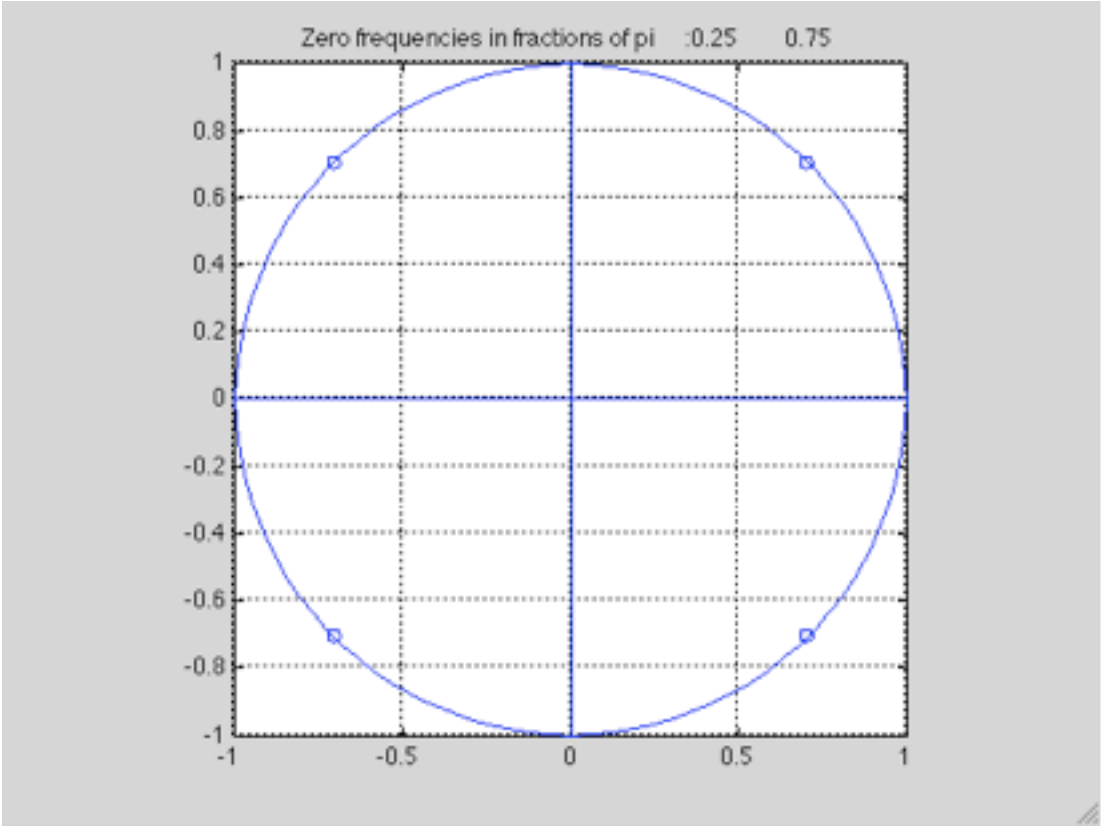
1+z^-2 = z^2 + 1 / z^2

이걸 0으로 만드는 지점!

Example

```
>> H = [1 0 0 0 1];
>> roots(H)
ans =
  -0.7071 + 0.7071i
  -0.7071 - 0.7071i
   0.7071 + 0.7071i
   0.7071 - 0.7071i
>> angle(roots(H))/pi
ans =
    0.7500
   -0.7500
    0.2500
   -0.2500
```





In general, the magnitude response is inversely proportional to the distance between that frequency (on the unit circle) and the location of the root.