Zero Order Hold Filtering and Recovery

lets assume a signal x(t) is given by:

$$x(t) = \frac{4}{\omega_m \pi t^2} \cdot \sin^2(\omega_m t)\cos(\omega_m t)\sin(2\omega_m t)$$

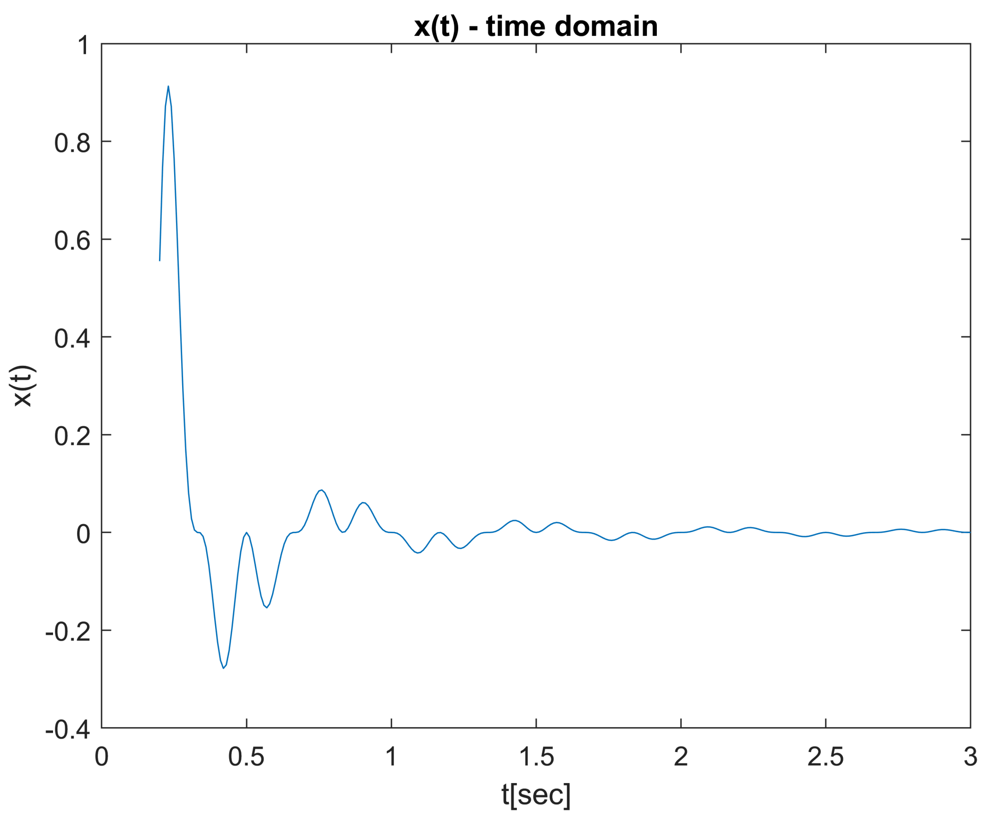
where:

$$\omega_m = 3\pi$$

```
wm = 3*pi;
t = 0.2:1/100:3;
x = @(t) 4./(wm*pi*t.^2).*(sin(wm*t)).^2.*(cos(wm*t)).*(sin(2*wm*t));
ylim([-0.8 1.6])
```

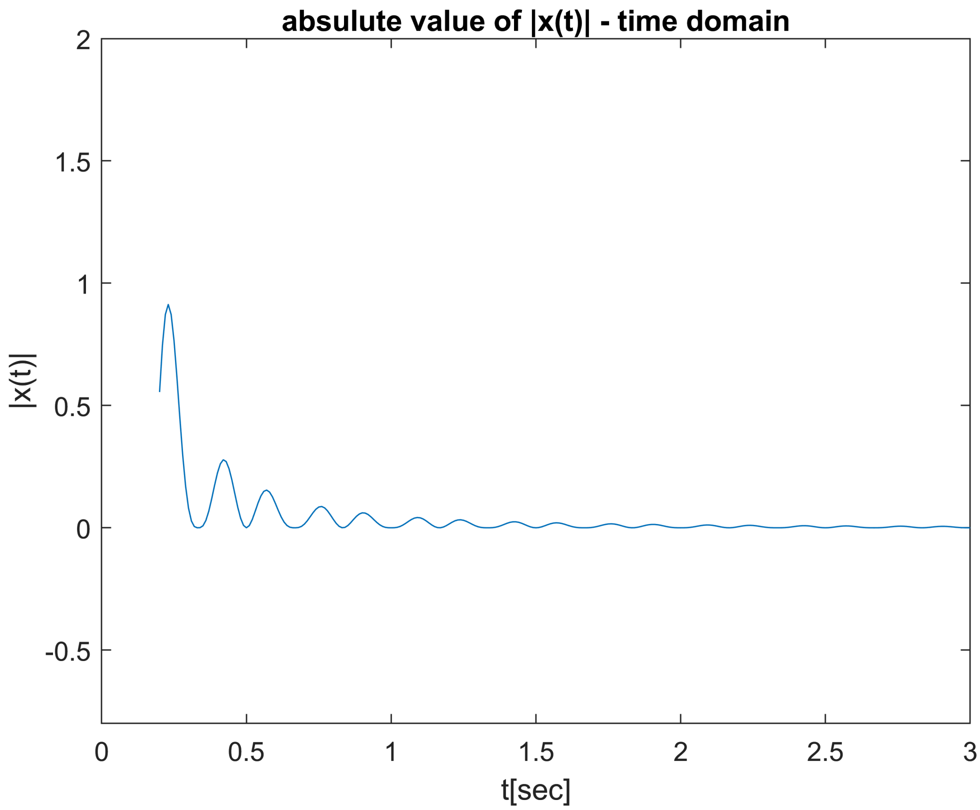
let'stake a look on $x(t)$ in time:

```
plot(t,x(t));title ('x(t) - time domain'); ylabel('x(t)'); xlabel('t[sec]')
```



we are asked to display the absulute value $|x(t)|$:

```
plot(t,abs(x(t)));title ('absulute value of |x(t)| - time domain'); ylabel('|x(t)|'); xlabel('t
ylim([-0.8 2])
```



$$X(\omega) = \mathscr{F}\{x(t)\}$$

$$= \mathscr{F}\left\{\frac{4}{\omega_m \pi t^2} \cdot \sin^2(\omega_m t)\cos(\omega_m t)\sin(2\omega_m t)\right\}$$

here it is apropriate to use the convolution quality:

$$= \frac{4}{\omega_m \pi}\mathscr{F}\left\{\frac{\sin(\omega_m t)}{t}\right\} * \mathscr{F}\left\{\frac{\sin(\omega_m t)}{t}\right\} * \mathscr{F}\{\sin(2\omega_m t)\} * \mathscr{F}\{\cos(\omega_m t)\}$$

$$= \frac{4}{\pi\omega_m}\left[\Pi\left(\frac{\omega}{2\omega_m}\right) * \Pi\left(\frac{\omega}{2\omega_m}\right)\right] * \left[\pi(\delta(\omega - 2\omega_m) - \delta(\omega + 2\omega_m)) * \pi(-i)(\delta(\omega - \omega_m) + \delta(\omega + \omega_m))\right]$$

notice that convolving with dirac's deltas is equivalent to time shifting:

$$= -\frac{4}{\pi\omega_m} i \left[\Lambda\left(\frac{\omega}{2\omega_m}\right)\right] * \left[-\pi i(\delta(\omega - 2\omega_m) - \delta(\omega + 2\omega_m))\right] * [\pi(\delta(\omega - \omega_m) + \delta(\omega + \omega_m))]$$

$$= \frac{1}{\omega_m}\left[\Lambda\left(\frac{\omega - 2\omega_m}{2\omega_m}\right) - \Lambda\left(\frac{\omega + 2\omega_m}{2\omega_m}\right)\right] * \pi(\delta(\omega - \omega_m) + \delta(\omega + \omega_m))$$

$$X(\omega) = \frac{\pi}{\omega_m}\left[\Lambda\left(\frac{\omega - 3\omega_m}{2\omega_m}\right) + \Lambda\left(\frac{\omega - \omega_m}{2\omega_m}\right) - \Lambda\left(\frac{\omega + 3\omega_m}{2\omega_m}\right) - \Lambda\left(\frac{\omega + \omega_m}{2\omega_m}\right)\right]$$

we got four lambda signals in frequency domain.

---

from the above expression it is clear that $X(\omega)$ ranges from $\omega = -5\pi$ up to 5π in frequancy domain.

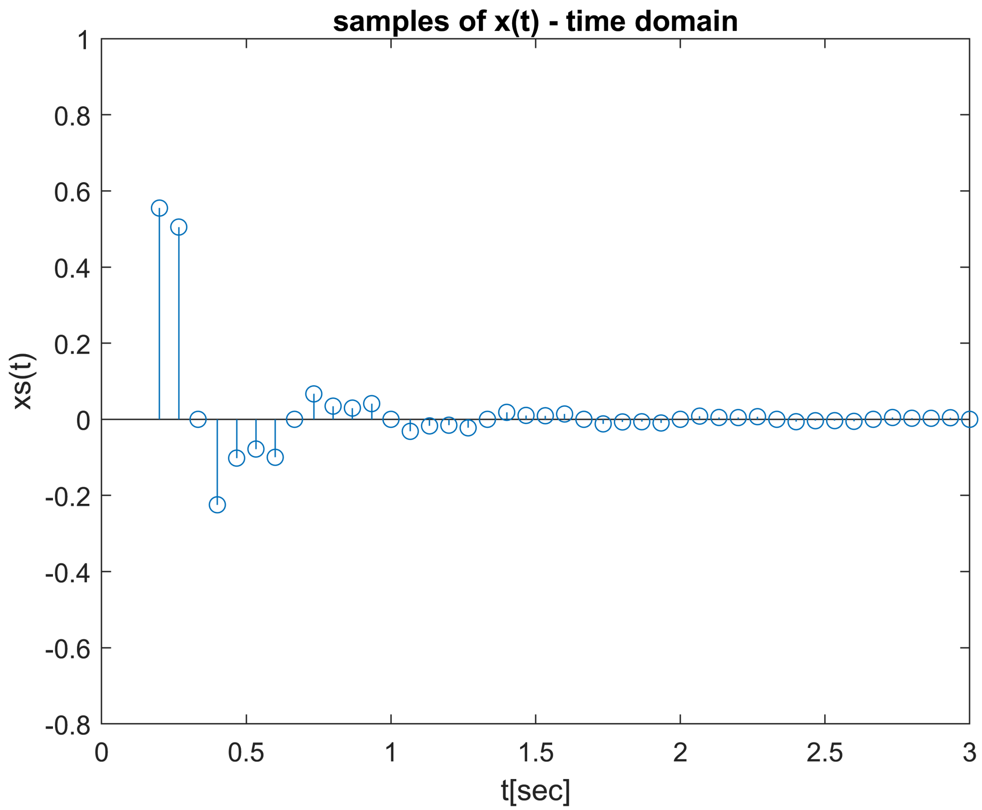
meaning $X(\omega) = 0 \ \forall|\omega| \geq 5\pi$.

said realizaton provides us with sufficient terms for sampling in *Nyquist rate.*

here we will use: $\omega_s = \omega_{\text{nyquist}} = 2\omega_{\max} = 10\omega_m$

let's sample the signal

```
ws = 10 * wm;
Ts = 2*pi/(ws);
pT = 0.2:Ts:3;
N = 0:(floor(3/Ts)+1);

xs = x(pT);
stem(pT,xs) ;title ('samples of x(t) - time domain'); ylabel('xs(t)'); xlabel('t[sec]');
ylim([-0.8 1])
```

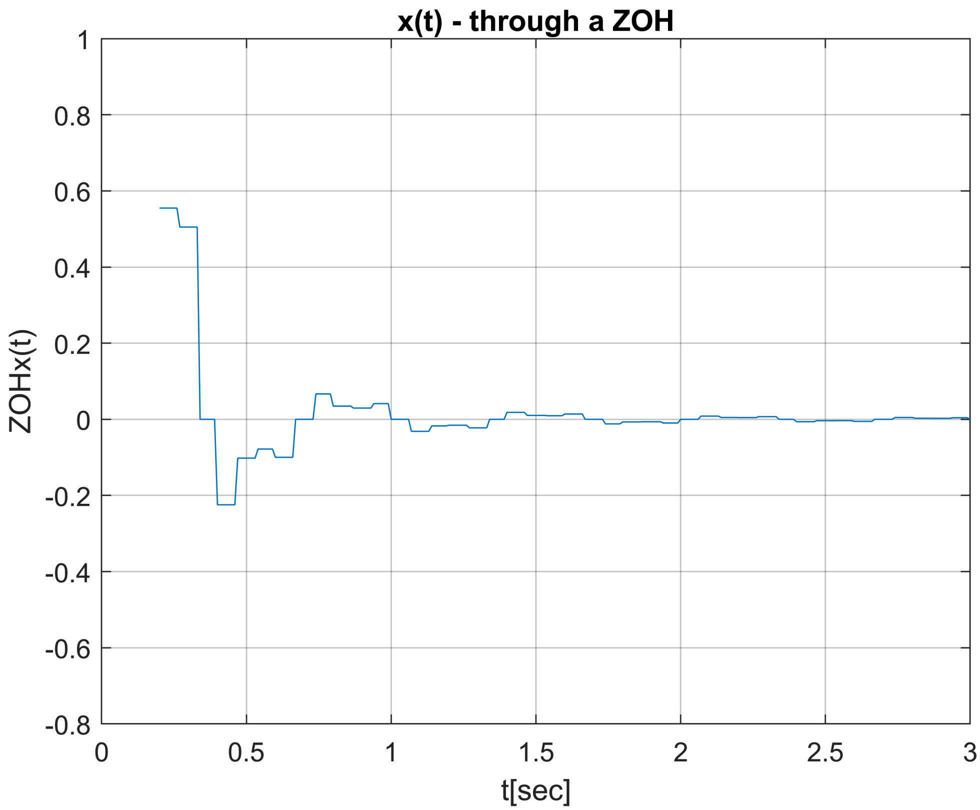


now we will implement a zero order hold:

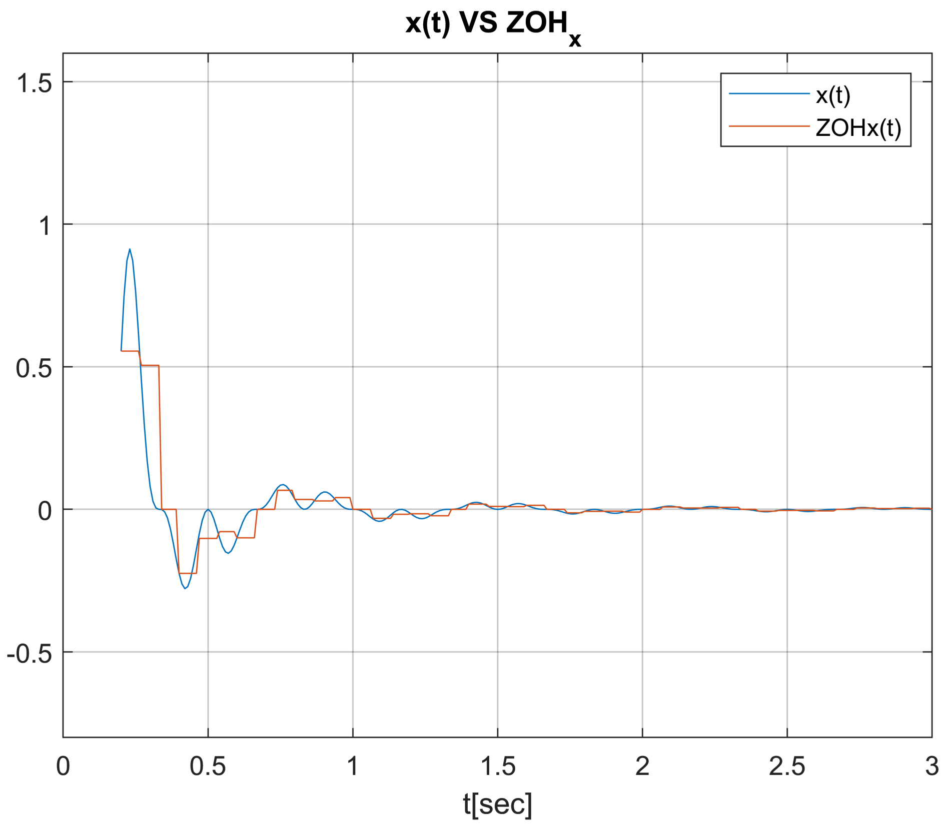
```
x_ZOH = xs(floor((t-0.2)/Ts)+1);
plot(t,x_ZOH); title ('x(t) - through a ZOH'); ylabel('ZOH{x(t)}'); xlabel('t[sec]');
ylim([-0.8 1])
```

we will display them both on one grid:

```
grid on;
```




```
plot(t,x(t));
hold on
plot(t,x_ZOH);
title ('x(t) VS ZOH_x '); legend('x(t)','ZOH{x(t)}'); xlabel('t[sec]');
ylim([-0.8 1.6])
grid on;
hold off;
```



$$X_{\mathrm{ZOH}}(\omega) = \mathscr{F}\{x_{\mathrm{ZOH}}(t)\}$$

$$= \mathscr{F}\left\{ \sum_{n=0}^{\infty} x(n \cdot T_s) \cdot \Pi\left(\frac{t - \left(n + \frac{1}{2} T_s\right)}{T_s} \right) \right\}$$

$$= \sum_{n=0}^{\infty} x(n \cdot T_s) \cdot \mathscr{F}\left\{ \Pi\left(\frac{t - \left(\left(n + \frac{1}{2}\right) T_s\right)}{T_s} \right) \right\}$$

$$= \sum_{n=0}^{\infty} x(n \cdot T_s) \cdot e^{-i\,n\omega}\, \mathscr{F}\left\{ \Pi\left(\frac{t - \frac{1}{2} T_s}{T_s} \right) \right\}$$

$$= \mathscr{F}\left\{ \Pi\left(\frac{t - \frac{1}{2}T_s}{T_s} \right) \right\} \sum_{n=0}^{\infty} x(n \cdot T_s) \cdot e^{-i\,n \cdot T_s \omega}$$

here we may recognize a familliar form:

$\sum_{n=0}^{\infty} x(n \cdot T_s) \cdot e^{-i\,(n \cdot T_s)\omega}$ is no other then then the DTFT of the sampled x(t) signal!

$$= X_P(\omega)$$

$$X_{\mathrm{ZOH}}(\omega) = \mathscr{F}\left\{ \Pi\left(\frac{t - \frac{1}{2}T_s}{T_s} \right) \right\} X_P(\omega)$$

$$= e^{-i \cdot \frac{1}{2}T_s \cdot \omega} \mathscr{F}\left\{ \Pi\left(\frac{t}{T_s} \right) \right\} X_P(\omega)$$

$$= e^{-i \cdot \frac{1}{2}T_s \cdot \omega} T \cdot \mathrm{sinc}\left(\frac{\omega}{\omega_s} \right) X_P(\omega)$$

$$= e^{-i \cdot \frac{1}{2}T_s \cdot \omega} T \cdot \mathrm{sinc}\left(\frac{\omega}{\omega_s} \right) \frac{1}{T} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s)$$

$$= e^{-i \cdot \frac{1}{2}T_s \cdot \omega} \cdot \mathrm{sinc}\left(\frac{\omega}{\omega_s} \right) \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s)$$

without further a due, we shall represent $X_{\mathrm{ZOH}}(\omega)$ in our code.we already have most of the needed input, so it will be a good idea to use this form:

$$X_{\mathrm{ZOH}}(\omega) = = e^{-i \cdot \frac{1}{2}T_s \cdot \omega} \cdot \mathrm{sinc}\left(\frac{\omega}{\omega_s} \right) \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s)$$

```
w = -17*pi:1/100:17*pi;
```

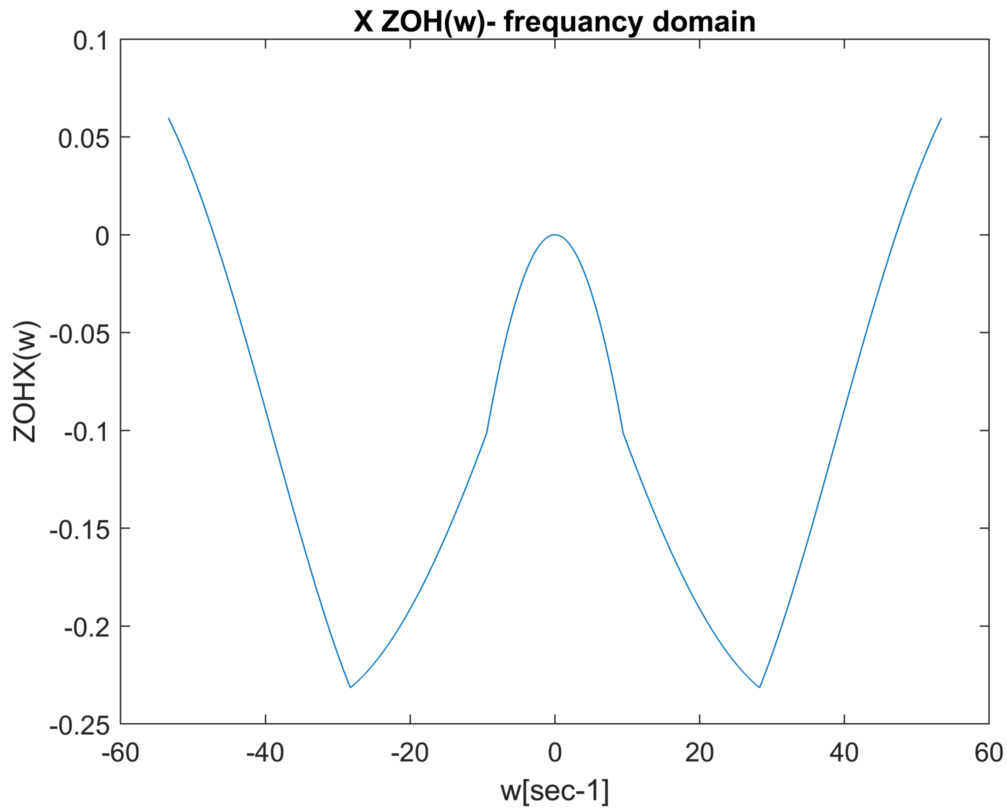
```
TRI = @(w,s) triangularPulse((w+s*wm)/(2*wm));

X = @(w) -1i*(TRI(w,-3)+TRI(w,-1)-TRI(w,1)-TRI(w,3));
%plot(w,X(w)); title ('X(w)- frequancy domain'); ylabel('X(w)'); xlabel('w[sec-1]');
```

for now we have $X(\omega)$ correctly, $X_p(\omega)$ would be duplications of it:

```
X_P = @(w) (pi/wm)*(X(w-ws) +X(w) +X(w+ws));

time_shift = @(w) exp(-1i*0.5*Ts*w);

X_ZOH = time_shift(w).*sinc(w/ws).*X_P(w);

plot(w,X_ZOH); title ('X ZOH(w)- frequancy domain'); ylabel('ZOH{X(w)}'); xlabel('w[sec-1]');
```

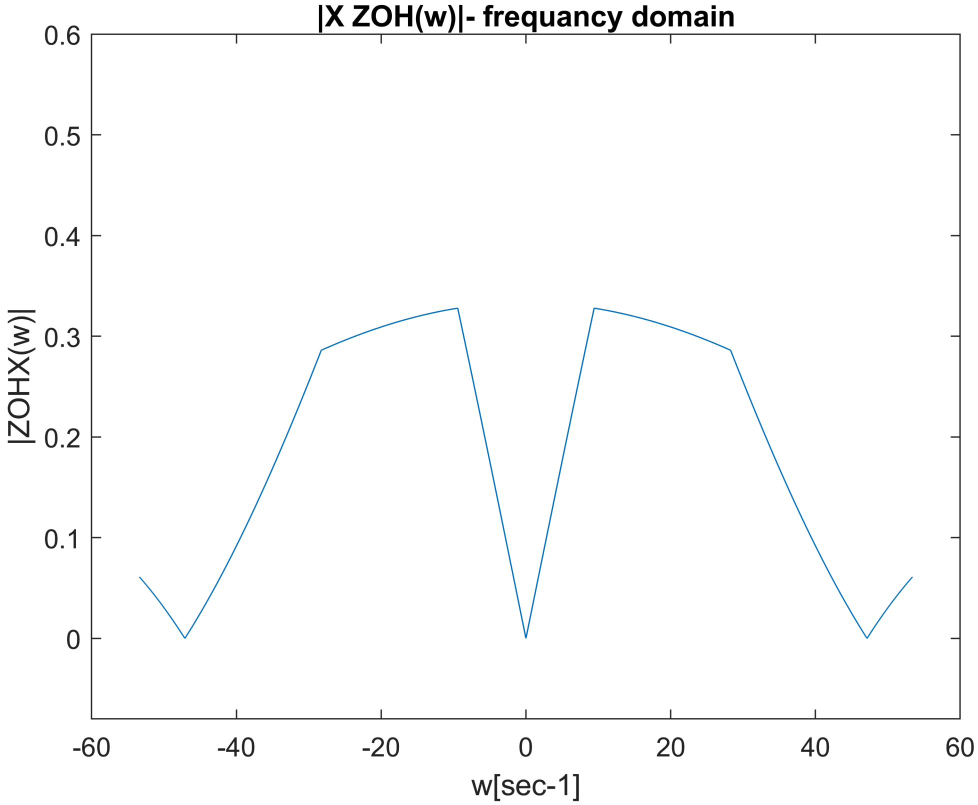
```
Warning: Imaginary parts of complex X and/or Y arguments ignored
```

```
hold off
```



now let's see the absulute value of $X_{\text{ZOH}}(\omega)$

```
plot(w,abs(X_ZOH)); title ('|X ZOH(w)|- frequancy domain'); ylabel('|ZOH{X(w)}|'); xlabel('w[se
ylim([-0.08 0.6])
```



HEA

let us define $H(\omega)$:

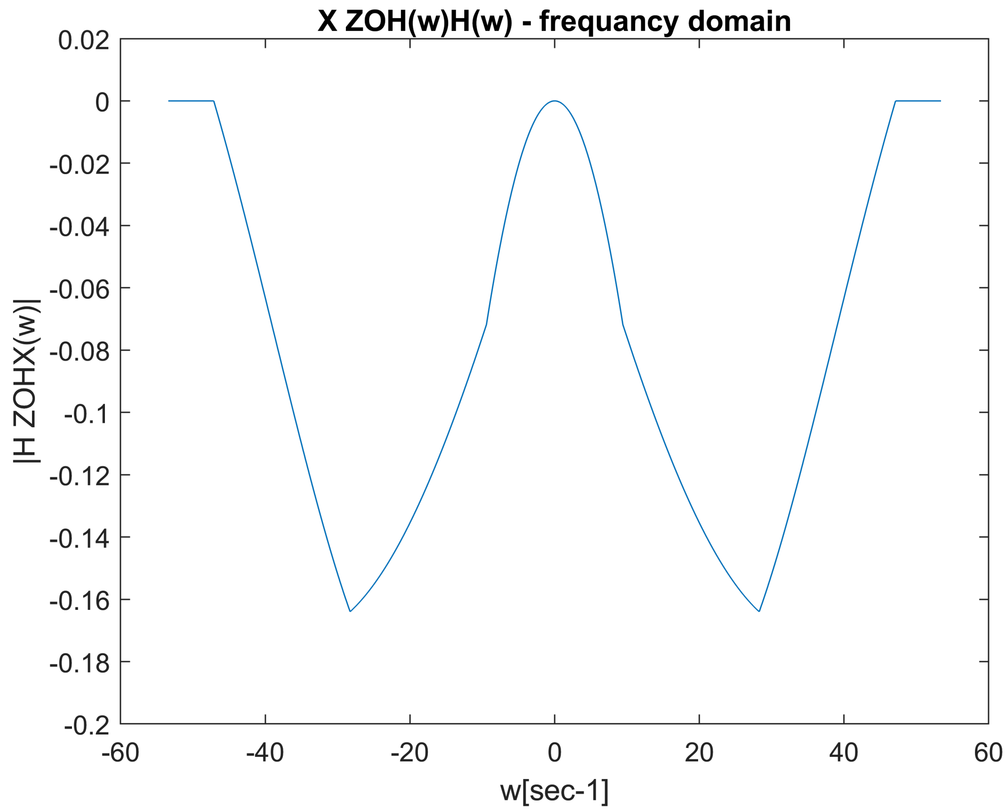
```
filter = @(w) exp(1i*pi*w/ws)/(sinc(w/ws));
H = @(w) (filter(w))*(abs(w) < (ws/2));
```

now is the time to multiply $H(\omega)$ with $X_{\mathrm{ZOH}}(\omega)$ to get the filterd signal:

```
X_rec =  H(w).*X_ZOH;
plot(w, X_rec);title ('X ZOH(w)H(w) - frequancy domain'); ylabel('|H {ZOH{X(w)}}|'); xlabel('w[
```

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

```
ylim([-0.2 0.02])
```

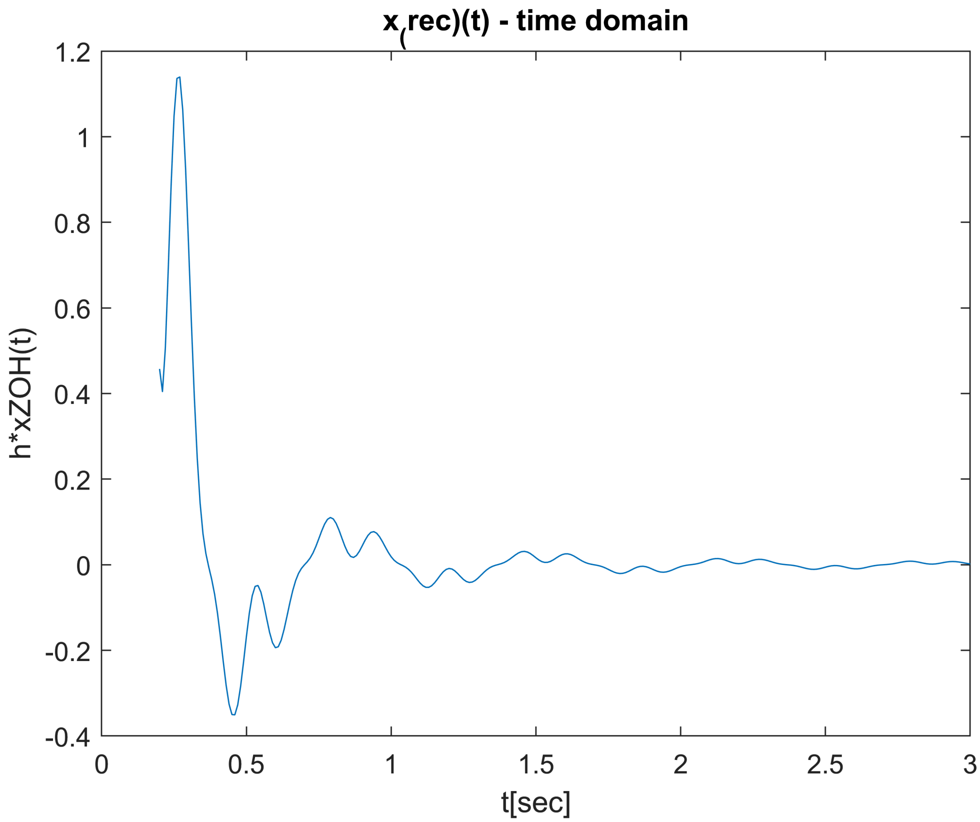


as we can see here, all other clones of $X(\omega)$ are eliminated, and only the one around $\omega = 0$ has an effect.

now, the inverse fourier transfom shuld return the original signal x(t)

```
X_rec_eiwt =@(w,t) X_rec.*exp(1i*w*t);
x_rec =t;
for i = 1:size(t,2)
    x_rec(i) = trapz(w,X_rec_eiwt(w,t(i)));
end

plot(t,x_rec); title ('x_(rec)(t) - time domain'); ylabel('h*xZOH(t)'); xlabel('t[sec]');
```

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

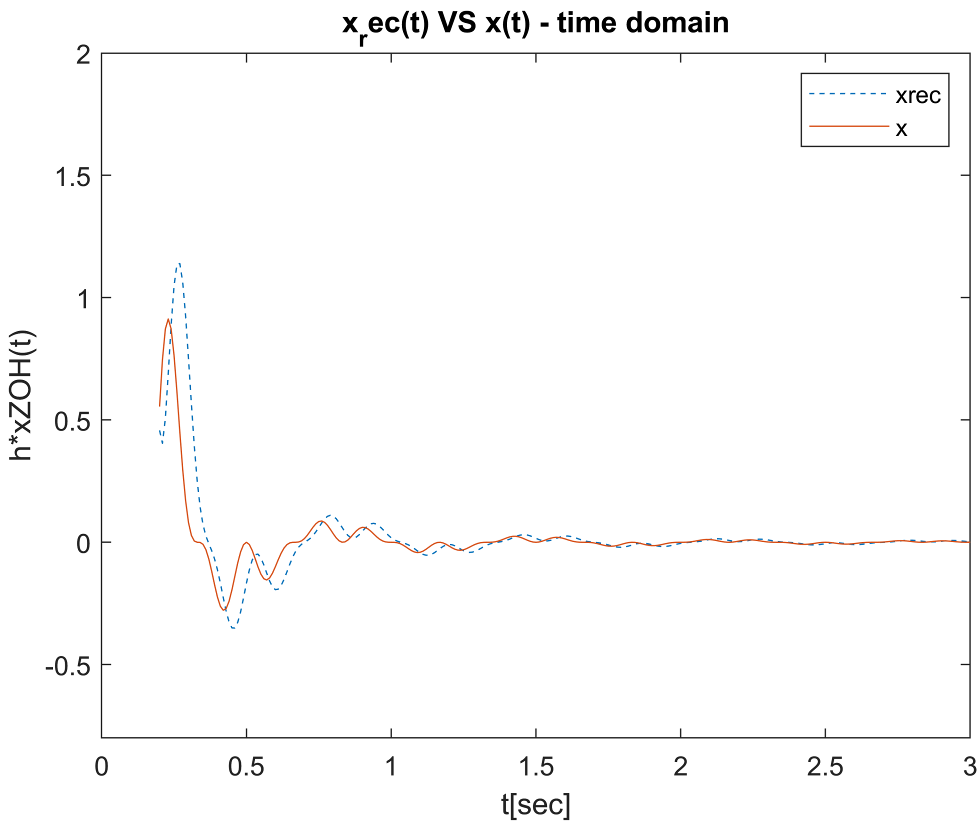


lets plot $x_{\text{rec}}(t)$ and the original $x(t)$ toether:

```
plot(t,x_rec,'--', t, x(t)); title ('x_rec(t) VS x(t) - time domain'); ylabel('h*xZOH(t)'); xla
```

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

```
legend('xrec' , 'x')
ylim([-0.8 2])
```



a little comment deep from the heart might be in place here. the signal *shuld have* mached percisely its recoverd form. both theory, and specific calculations show that, and we brought them bothhere in our work. but after hours and hours of debugging i can say that I lost my left and right trying to find the bug causing the mismatch, without success. my hatred for matlab has meanwhile grew to a ranging fire, that burns with the fierce passion of a million suns. we know what to expact, and are **confident** enough in the studied material to say: there is an error. and it suld be confessed. matlab is not my favourite coding platform.

---

of curse $x(t)$ cannot be recoverd from equaly spaced samples at sampling rate ω_s because such sub-nyquist rate would cause *aliasing* in $X_p(\omega)$

```
ws = 9*wm;
X_P = @(w) (1/3)*(X(w-ws) +X(w) +X(w+ws));

time_shift = @(w) exp(-1i*0.5*(2*pi/ws)*w);
```

```
X_ZOH = time_shift(w).*sinc(w/ws).*X_P(w);

filter = @(w) exp(1i*pi*w/ws)/(sinc(w/ws));
H = @(w) (filter(w))*(abs(w) <= (ws/2));

X_rec =  H(w).*X_ZOH;
X_rec_eiwt =@(w,t) X_rec.*exp(1i*w*t);
x_rec =t;

for i = 1:size(t,2)
    x_rec(i) = trapz(w,X_rec_eiwt(w,t(i)));
end

plot(t,x_rec,'--', t, x(t)); title ('x_rec(t) VS x(t) - time domain'); ylabel('h*xZOH(t)'); xla
```

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

```
legend('xrec' , 'x')
ylim([-0.8 2])
```