3.2-4 A *first-order hold* (FOH) circuit can also be used to reconstruct a signal $x(t)$ from its samples. The impulse response of this circuit is

$$h(t) = \Lambda\left(\frac{t}{2T}\right),$$

where T is the sampling interval.

(a) Using an impulse-sampled signal $x_{\tilde{\delta}}(t)$, show that the FOH circuit performs *linear interpolation*. In other words, the filter output consists of samples connected by straight-line segments.

a) $x_{\tilde{\delta}}(t)$ will be scaled impulses of $x(t)$ separated by the sampling period T_s. Then,

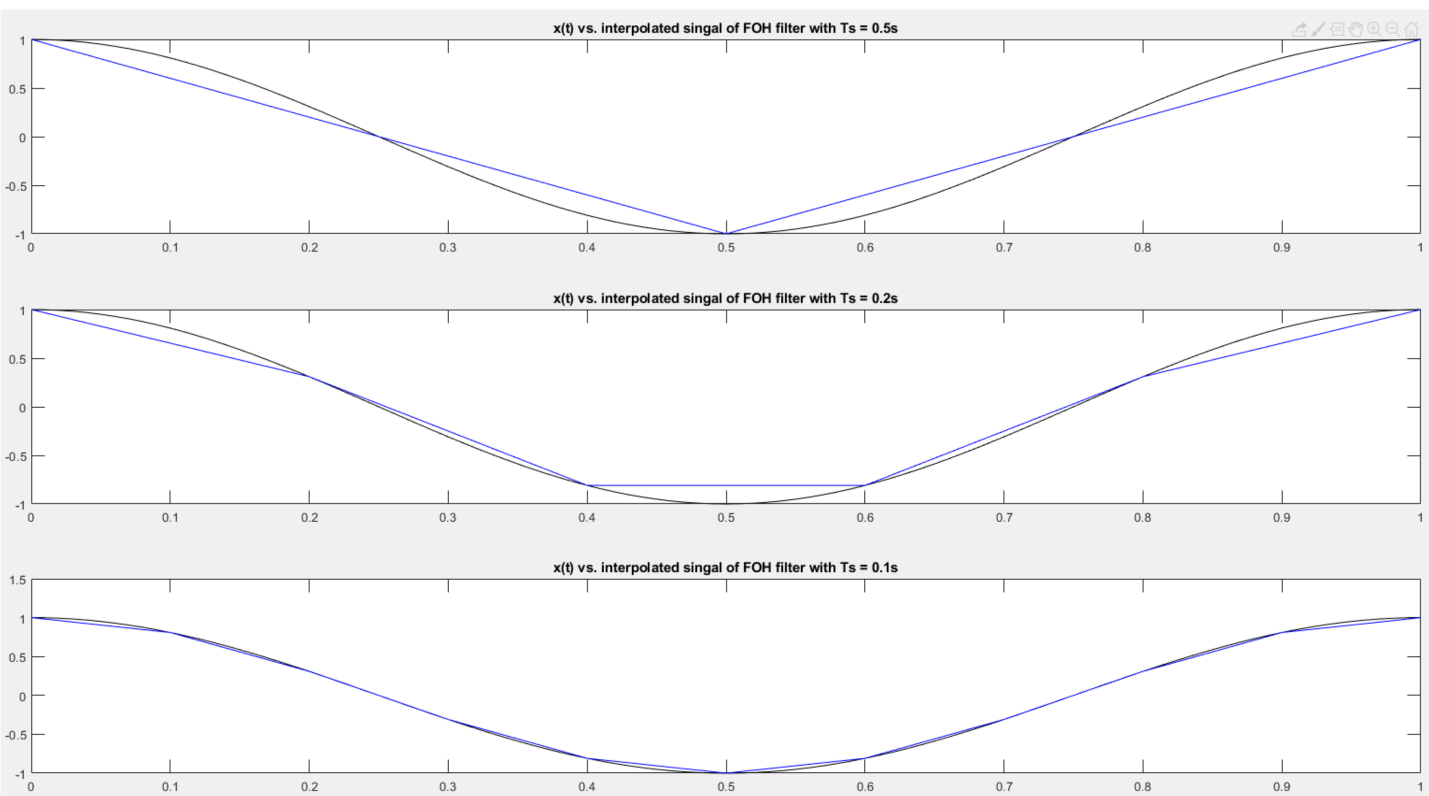
$$y(t) = x_{\tilde{\delta}}(t) * h(t) = \left(\sum_{n=-\infty}^{\infty} x(nT)\right) * h(t) = \sum_{n=-\infty}^{\infty} x(nT)\,\Lambda\left(\frac{t-nT}{2T}\right) \qquad (see\ pg. 165)$$

Stealing your code from pg. 166 in the textbook, we can see what this looks like in MATLAB with sampling periods of 0.5s, 0.2s and 0.1s

C3_2_4.m

```
close all
triangle = @(t) (1-2.*abs(t)).*(abs(t) <= 0.5);
x = @(t) cos(2*pi*t);
t = 0:0.01:1;
T = [0.5 0.2 0.1];
for k = 1:3
    xhatN = 0;
    for n = -10:10
        xhatN = xhatN + x(n*T(k))*triangle((t-n*T(k))/(2*T(k)));
    end
    subplot(3,1,k);
    plot(t,x(t),'k'); hold on; plot(t,xhatN,'b');
    title("x(t) vs. interpolated singal of FOH filter with Ts = " + T(k) + "s");
    dt = t(2)-t(1); RMSerror = sqrt(sum((xhatN-x(t)).^2)*dt);
    RMSerror
end
```

```
RMSerror =

    0.1509

RMSerror =

    0.0985

RMSerror =

    0.0253
```



(b) Determine the frequency and magnitude responses of the FOH filter. Compare these responses with both the ideal and ZOH reconstruction filters.

$$h(t) = \Lambda\left(\frac{t}{2T}\right)$$

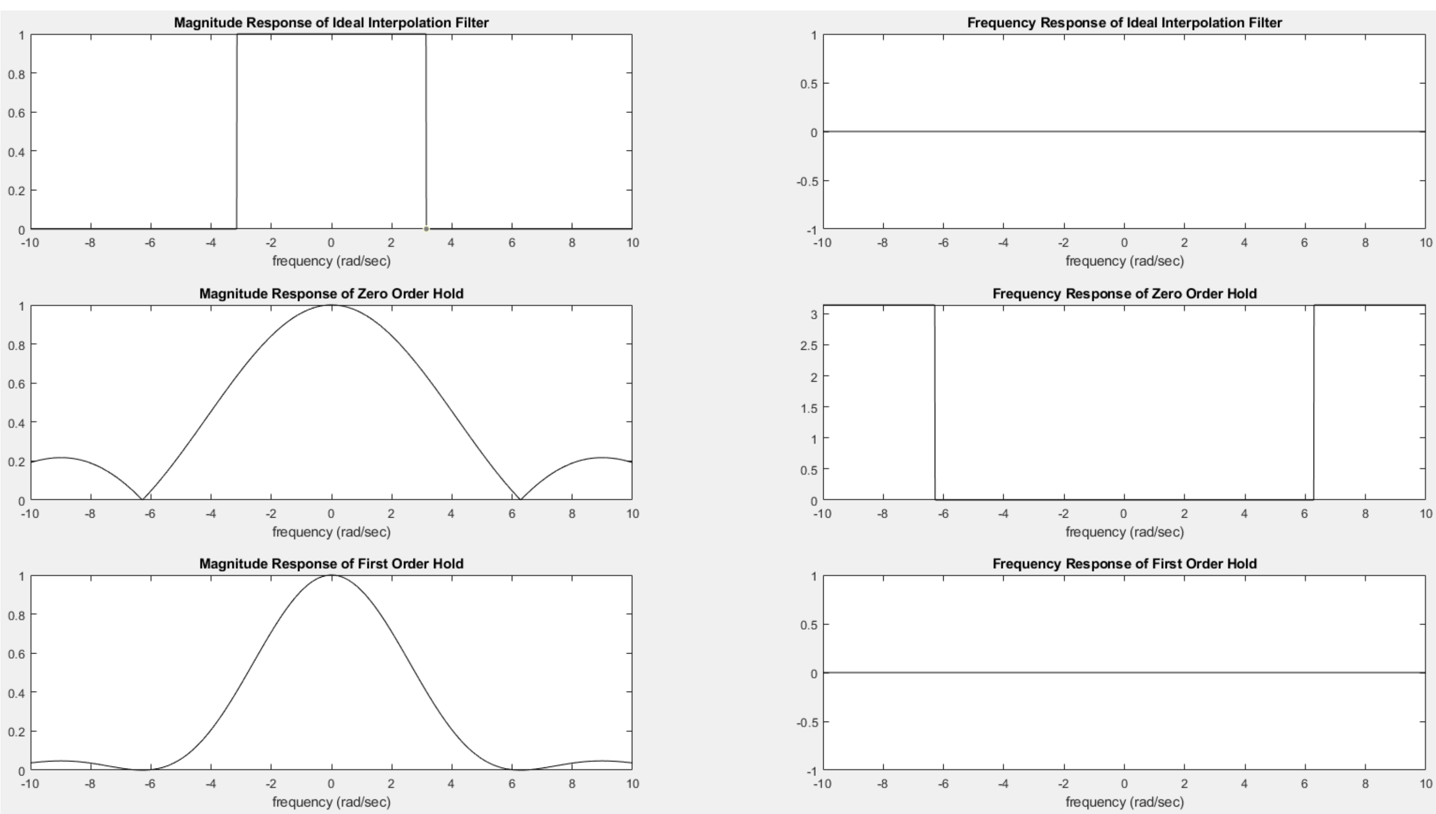
Using transform pair: $\Lambda\left(\frac{t}{\tau}\right)$ $\qquad$ $\frac{\tau}{2}\text{sinc}^2\left(\frac{\tau\omega}{4\pi}\right)$

Then, $H(\omega) = T sinc^2\left(\frac{\omega T}{2\pi}\right)$ where $sinc = \frac{\sin(\pi t)}{\pi t}$

We can steal the frequency domain equations for ZOH and ideal reconstruction filters from pg. 167 and 164 respectively

Then in MATLAB,

```
%% part b
close all
T = 1;
unit_gate = @(w) (abs(w) < 0.5) + 0.5.*(abs(w) == 0.5);
% H{1} = ideal, H{2} = ZOH, H{3} = FOH
H = {@(w) T.*unit_gate((w.*T)./(2*pi)); ...
    @(w) T.*sinc((w.*T)./(2*pi)); ...
    @(w) T.*(sinc((w.*T)./(2*pi))).^2 };
name = ["Ideal Interpolation Filter"; "Zero Order Hold"; "First Order Hold"];
w = -10:0.01:10;

for k = 1:3
    subplot(3,2,2*k-1);
    plot(w,abs(H{k}(w)),'k'); xlabel("frequency (rad/sec)");
    title("Magnitude Response of " + name{k});

    subplot(3,2,2*k);
    plot(w,angle(H{k}(w)),'k'); xlabel("frequency (rad/sec)");
    title("Frequency Response of " + name{k});
end
```



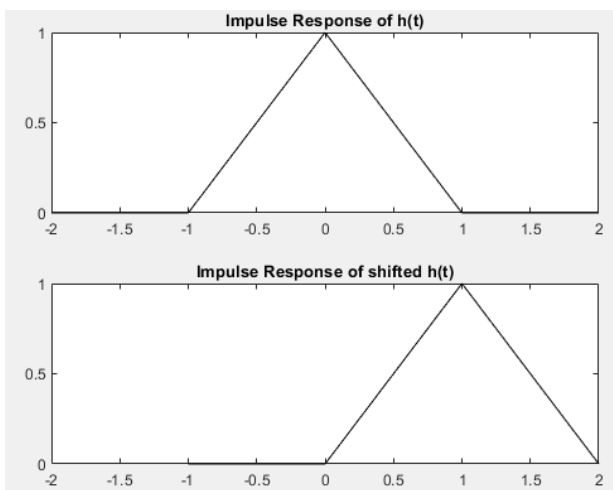
(c) This filter, being noncausal, is unrealizable. By delaying its impulse response, the filter can be made realizable. What is the minimum delay required to make it realizable? How does this delay affect the reconstructed signal and the filter frequency response?

Currently, the impulse response is a Triangle model, centered on t = 0, with limbs that go out to from -T to T. Then the minimum delay necessary to make this filter realizable (causal) is one period "T".

$$x_{FOH_{Delayed}}(t) = \sum_{n=-\infty}^{\infty} x(nT)\,\Lambda\left(\frac{t-T-nT}{2T}\right)$$

If we delay $h(t)$ by one sampling period it will be identical to the FOH except that the output will be delayed by one sample period. In MATLAB if we let T = 1,

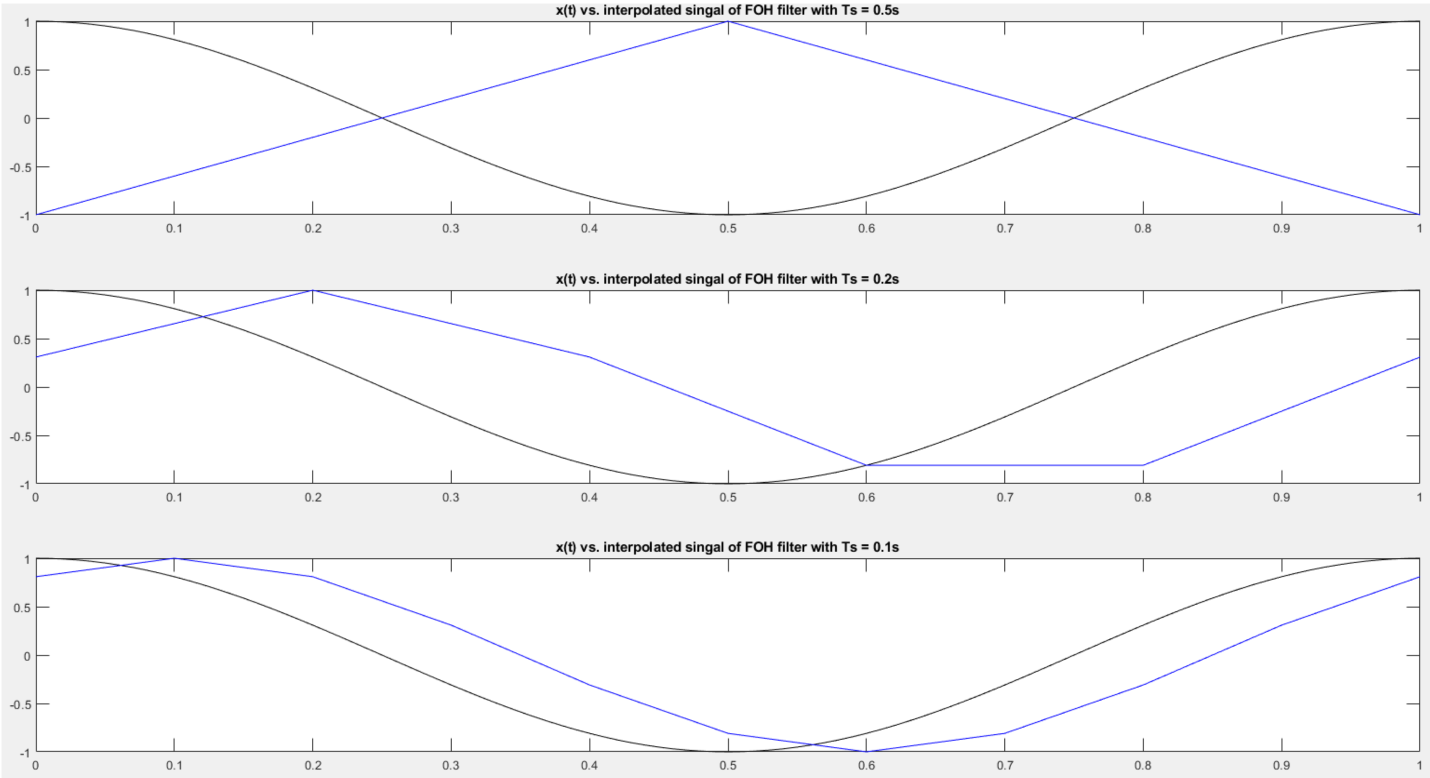
```
%% part c
close all
T = 1;
triangle = @(t) (1-2.*abs(t)).*(abs(t) <= 0.5);
t = -2:0.1:2;

subplot(2,1,1);
plot(t,triangle(t./(2*T)),'k'); title("Impulse Response of h(t)");
axis([-2 2 0 1]);
subplot(2,1,2);
plot(t+T,triangle(t./(2*T)),'k'); title("Impulse Response h(t) by T");
axis([-2 2 0 1]);
```



Or in MATLAB using our same code from part a,

```
%% part c part 2
close all
triangle = @(t) (1-2.*abs(t)).*(abs(t) <= 0.5);
x = @(t) cos(2*pi*t);
t = 0:0.01:1;
T = [0.5 0.2 0.1];
for k = 1:3
    xhatN = 0;
    for n = -10:10
        xhatN = xhatN + x(n*T(k))*triangle((t-T(k)-n*T(k))/(2*T(k)));
    end
    subplot(3,1,k);
    plot(t,x(t),'k'); hold on; plot(t,xhatN,'b');
    title("x(t) vs. interpolated singal of FOH filter with Ts = " + T(k) + "s");
    dt = t(2)-t(1); RMSerror = sqrt(sum((xhatN-x(t)).^2)*dt);
    RMSerror
end
```

```
RMSerror =

    1.2979


RMSerror =

    0.7870


RMSerror =

    0.4311
```



(d) Show that the causal FOH circuit in part (c) can be realized by a cascade of two ZOH circuits, where each ZOH is constructed as shown in Fig. P3.2-3.

$$H_{ZOH}(\omega) = T sinc\left(\frac{\omega T}{2\pi}\right) \quad see\ pg.167$$

Cascaded systems multiply in the frequency domain, so:

$$H_{ZOH_{twice\ cascaded}}(\omega) = T sinc\left(\frac{\omega T}{2\pi}\right) T sinc\left(\frac{\omega T}{2\pi}\right) = T^2 sinc^2\left(\frac{\omega T}{2\pi}\right)$$

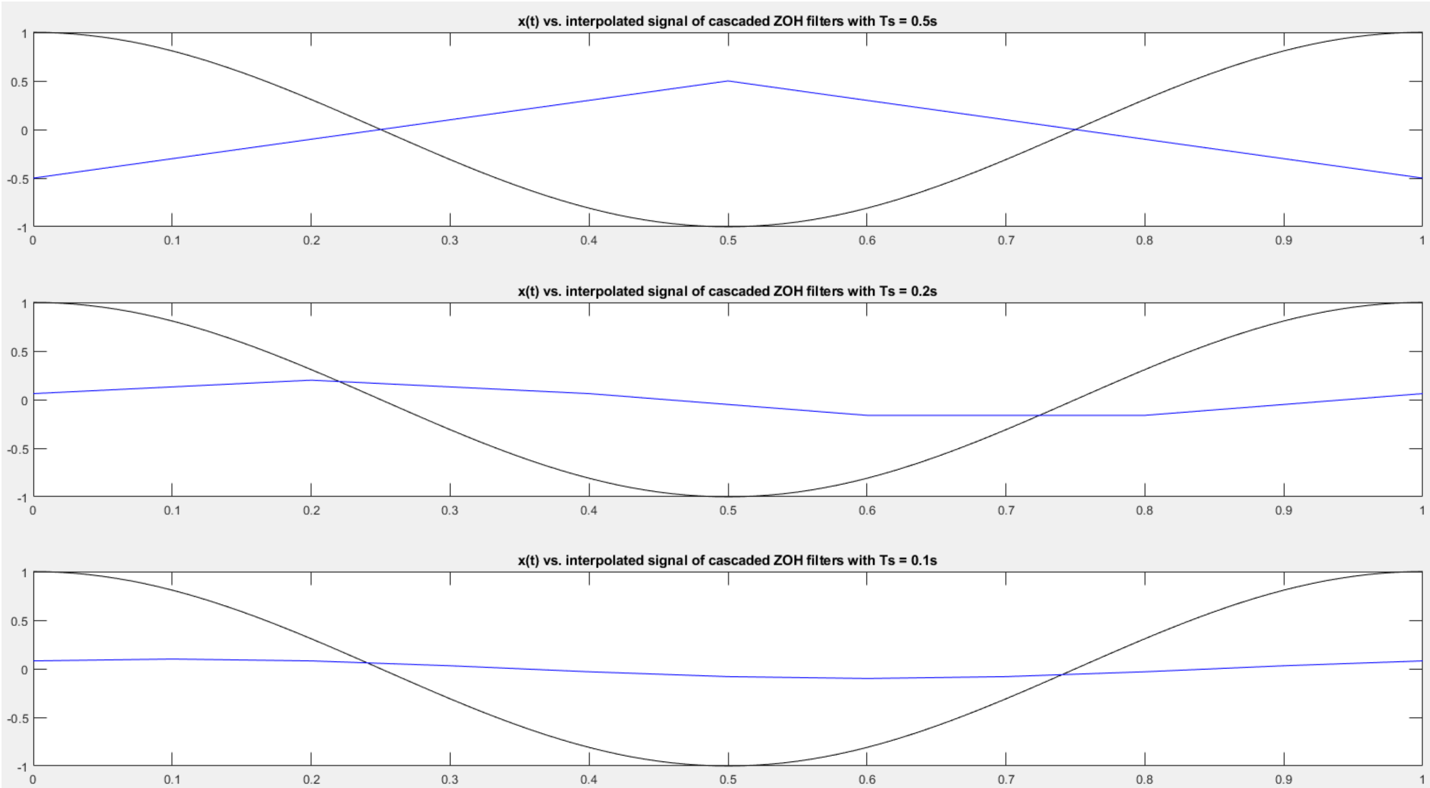
From the FOH transform pair, we know that in the time-domain this is:

$$h_{ZOH_{twice\ cascaded}}(t) = T\Lambda\left(\frac{t}{2T}\right)$$

This will be the same as FOH circuit, except scaled by T. In our case if we use the same $T = 0.5, 0.2, 0.1$ then in MATLAB:

```
%% part d
close all
triangle = @(t) (1-2.*abs(t)).*(abs(t) <= 0.5);
x = @(t) cos(2*pi*t);
t = 0:0.01:1;
T = [0.5 0.2 0.1];
for k = 1:3
    xhatN = 0;
    for n = -10:10
        xhatN = xhatN + x(n*T(k))*T(k)*triangle((t-T(k)-n*T(k))/(2*T(k)));
    end
    subplot(3,1,k);
    plot(t,x(t),'k'); hold on; plot(t,xhatN,'b');
    title("x(t) vs. interpolated singal of FOH filter with Ts = " + T(k) + "s");
    dt = t(2)-t(1); RMSerror = sqrt(sum((xhatN-x(t)).^2)*dt);
    RMSerror
end
```

Thomas Smallarz ECE444 Homework 4



These are the same values as our FOH filter except scaled by T.