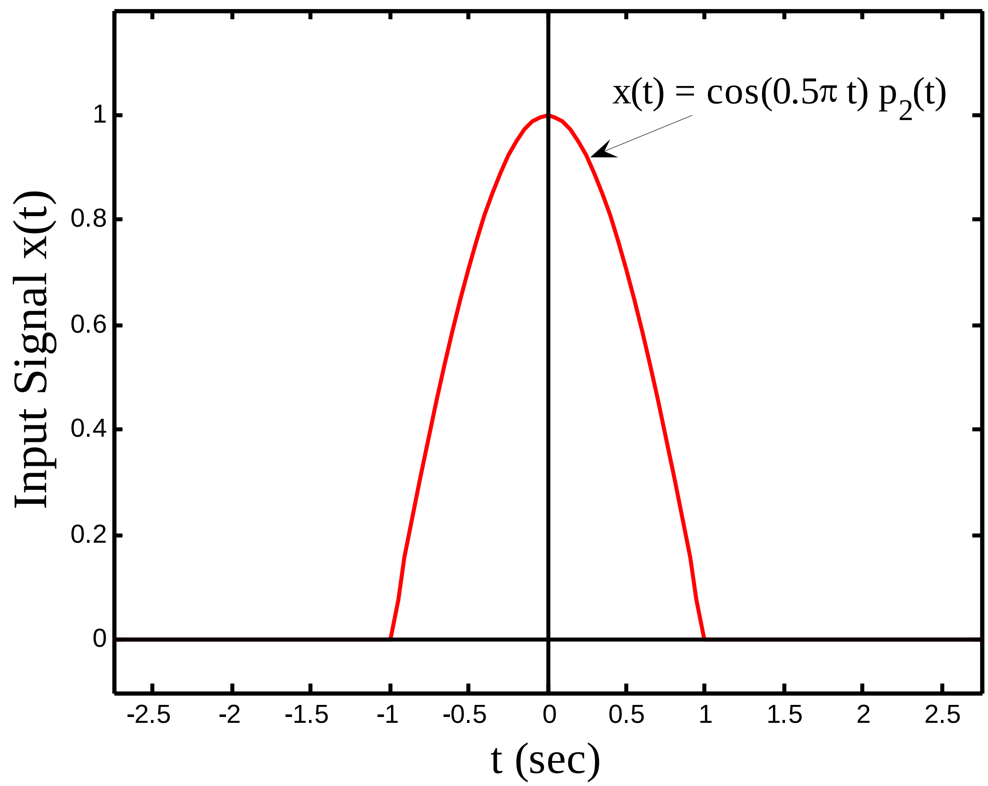
Consider the input signal $x(t)$, a half-cycle sinusoid pulse, shown below. The input signal can be mathematically expressed as $x(t) = \cos(0.5\pi t)\, p_2(t)$.




```
% chap5_cosine_halfcycle.m
%
% Interpolation Formula and sampling example
%
% Plot cosine half-cycle x(t)=cos(0.5*pi*t) -1<t<1 sec
%
clear;clc;close all;
t = -2.75:0.05:2.75;
x = cos(0.5*pi*t);
for i=1:length(t), % set x(t) < 0 to x(t) = 0
    if(x(i)<0),
        x(i)=0;
    end
end
plot(t,x,'r',[-2.75,2.75],[0,0],'k',[0,0],[-0.1,1.2],'k'),
axis([-2.75 2.75 -0.1 1.2]),
ylabel(['Input Signal x(t)'],'fontsize',18,...
    'fontname','times'),
%title('Interpolation Formula and Sampling example',...
%     'fontsize',18,'fontname','times');
xlabel('t (sec)','fontsize',16,'fontname','times'),
text(0.1,1.1,'x(t) = cos(0.5\pi_{ }t)p_{2}(t)','HorizontalAlignment',...
    'left','VerticalAlignment','top','fontsize',14,'fontname','times')
set(findobj('type','line'),'linewidth',1.5)
set(findobj('type','line'),'markersize',18)
set(findobj('type','axes'),'linewidth',2)
```

The Fourier transform of this signal $X(\omega)$ can be obtained using the property

multiplication by $\cos(\omega_0 t)$: $x(t)\cos(\omega_0 t) \leftrightarrow \frac{1}{2}\left[X(\omega+\omega_0)+X(\omega-\omega_0)\right]$,

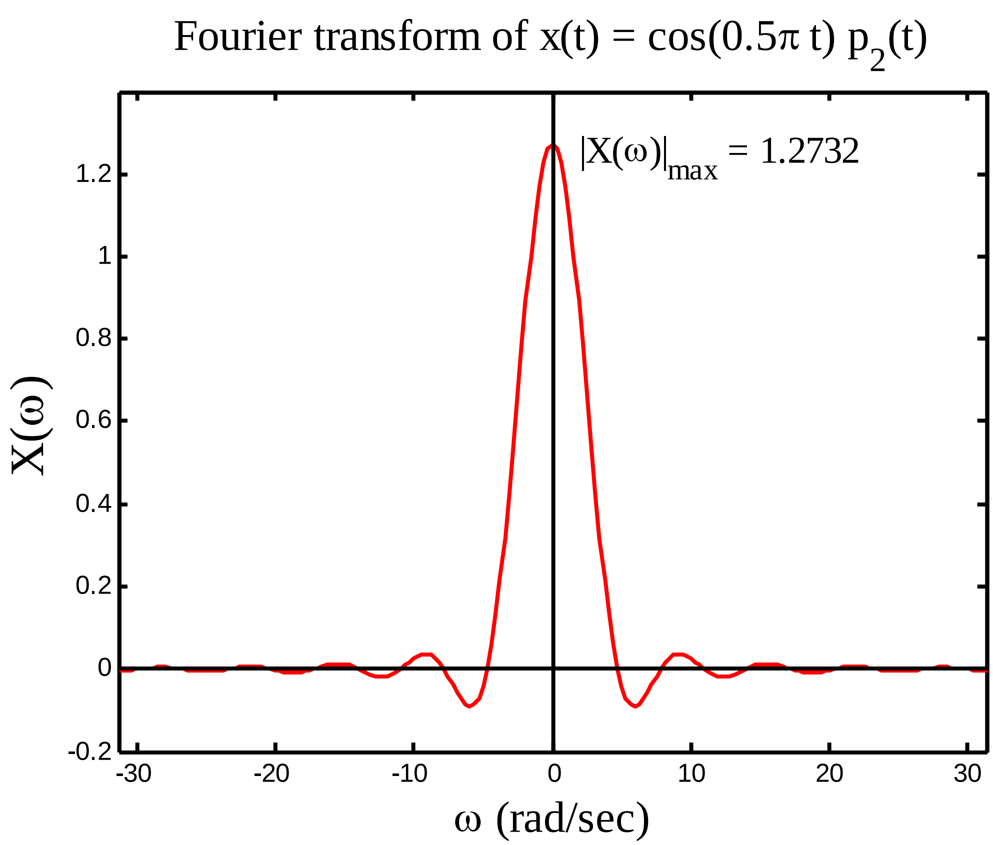
and the transform pair

$$p_\tau(t) \leftrightarrow \tau\,\mathrm{sinc}\left(\frac{\tau\omega}{2\pi}\right)$$

yielding

$$\begin{aligned} X(\omega) &= \frac{1}{2}\left[2\,\mathrm{sinc}\left(\frac{2(\omega+0.5\pi)}{2\pi}\right)+2\,\mathrm{sinc}\left(\frac{2(\omega-0.5\pi)}{2\pi}\right)\right] \\ &= \mathrm{sinc}\left(\frac{\omega+0.5\pi}{\pi}\right)+\mathrm{sinc}\left(\frac{\omega-0.5\pi}{\pi}\right) \end{aligned}.$$

Since $X(\omega)$ is real, the frequency spectrum can be plotted directly as shown.



Note that the frequency response is infinite in extent. Therefore, we will experience some aliasing when sampling this signal regardless of how quickly the signal is sampled. The question is “How bad will the aliasing be?”

```
% chap5_cosine_halfcycle_FT.m
%
% Interpolation Formula and sampling example
%
% For the cosine half-cycle
%     x(t)=cos(0.5*pi*t) -1<t<1 sec
% plot its Fourier transform
%     X(w) = sinc[(w+0.5pi)/pi)] + sinc[(w-0.5pi)/pi)].
%
clear;clc;close all;
w = -10*pi:pi/10:10*pi;
X = sinc((w+pi/2)/pi)+sinc((w-pi/2)/pi);
plot(w,X,'r',[-10*pi,10*pi],[0,0],'k',[0,0],[-0.2,1.4],'k'),
axis([-10*pi 10*pi -0.2 1.4]),
title('Fourier transform of x(t) = cos(0.5\pi_{ }t) p_{2}(t)',...
    'fontsize',16,'fontname','times'),
ylabel(['X(\omega)'],'fontsize',18,'fontname','times'),
xlabel('\omega (rad/sec)','fontsize',16,'fontname','times'),
text(2,1.32,['|X(\omega)|_{max} = ',num2str(max(X))],...
    'HorizontalAlignment','left','VerticalAlignment','top',...
    'fontsize',14,'fontname','times')
set(findobj('type','line'),'linewidth',1.5)
set(findobj('type','line'),'markersize',18)
set(findobj('type','axes'),'linewidth',2)
```

Next, we'll sample the input signal with the sampling periods

$$T_1 = 0.5 \text{ s with } f_1 = \frac{1}{T_1} = 2 \text{ Hz}, \ \omega_{s1} = \frac{2\pi}{T_1} = 4\pi \text{ rad/s}, \ \& \ B_1 = \frac{\omega_{s1}}{2} = 2\pi \text{ rad/s}$$

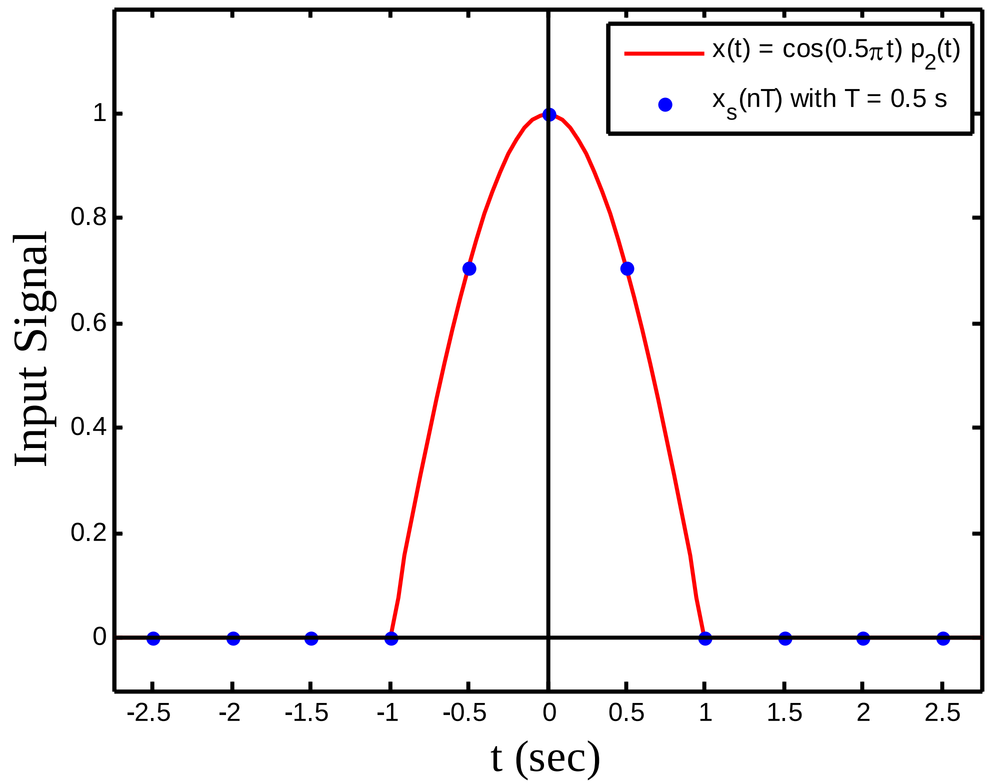
and

$$T_2 = 0.25 \text{ s with } f_2 = \frac{1}{T_2} = 4 \text{ Hz}, \ \omega_{s1} = \frac{2\pi}{T_1} = 8\pi \text{ rad/s}, \ \& \ B_2 = \frac{\omega_{s2}}{2} = 4\pi \text{ rad/s}.$$

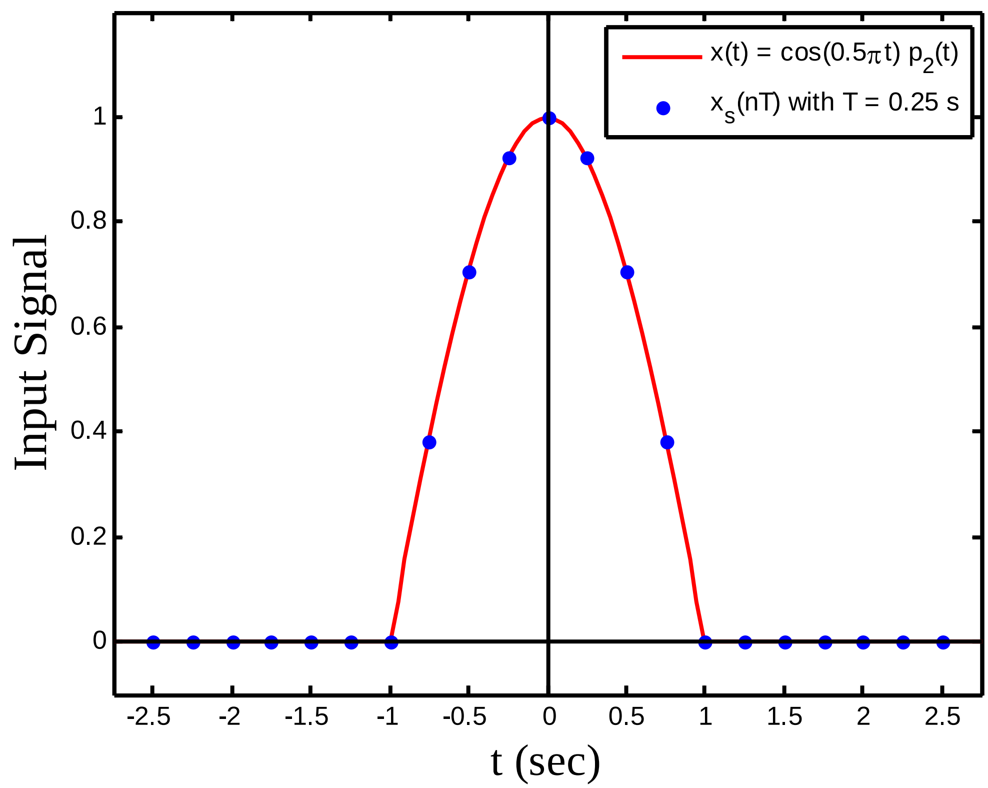
Remember, we mathematically represent sampled signals as $x(nT) = x(t)p(t)$ where $p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$ which yields

$$\begin{aligned} x(nT) &= \sum_{n=-\infty}^{\infty} x(t)\delta(t-nT) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t-nT) \\ &= \sum_{n=-\infty}^{\infty} \cos(0.5\pi nT)\, p_2(nT)\delta(t-nT) \end{aligned}.$$

Note: Due to the rectangular pulse function $p_2(nT)$, the sampling summation will only have a few non-zero values (depending on sampling period T).



- Note that there are only three (3) non-zero samples.



- Note that there are seven (7) non-zero samples.

```
% chap5_sampled_cosine_halfcycle.m
% Interpolation Formula and sampling example
%
% Plot cosine half-cycle x(t)=cos(0.5*pi*t) -1<t<1 sec
% and sampled version xs(t) = x(nT)
%
clear;clc;close all;
%
T = 0.25; % sampling period
t = -2.75:0.05:2.75;
x = cos(0.5*pi*t);
n = 0:1:20;
Tn = (n-10)*T;
xs = cos(0.5*pi*Tn);
for i=1:length(t), % setting values less than zero to zero
    if(x(i)<0),
        x(i)=0;
    end
end
for i=1:length(Tn),
    if(xs(i)<0),
        xs(i)=0;
    end
end
plot(t,x,'r',Tn,xs,'b.',[-2.75,2.75],[0,0],'k',[0,0],[-0.1,1.2],'k'),
axis([-2.75 2.75 -0.1 1.2]),
ylabel(['Input Signal'],'fontsize',18,'fontname','times'),
xlabel('t (sec)','fontsize',16,'fontname','times'),
legend('x(t) = cos(0.5\pi_{ }t) p_{2}(t)',...
    ['x_{s}(nT) with T = ',num2str(T),' s']),
set(findobj('type','line'),'linewidth',1.5)
set(findobj('type','line'),'markersize',17)
set(findobj('type','axes'),'linewidth',2)
```

The Fourier transform of the sampled signals is

$$X_S(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T} X(\omega - k\omega_S).$$

For the selected sampling periods, we then get

$$X_{S1}(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T_1} X(\omega - k\omega_{S1}) = \sum_{k=-\infty}^{\infty} 2\, X(\omega - k4\pi) \text{ for } T_1 = 0.5 \text{ s}$$

and

$$X_{S2}(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T_2} X(\omega - k\omega_{S2}) = \sum_{k=-\infty}^{\infty} 4\, X(\omega - k8\pi) \text{ for } T_2 = 0.25 \text{ s}.$$

Before plotting $X_{S1}(\omega)$ and $X_{S2}(\omega)$, how bad will the aliasing be in each case?

Using MATLAB, we'll characterize the aliasing in terms of the largest magnitude value of $X(\omega)$ for the frequencies where $X_S(\omega)$ would overlap, i.e., $|\omega| \geq B = \frac{\omega_s}{2}$ for the k = 0 term of $X_S(\omega)$.

For $T_1 = 0.5$ s with $B_1 = 2\pi$ rad/s, $\left|X(\omega \geq B_1)\right|_{\max} = 0.0849$. So, the aliasing is

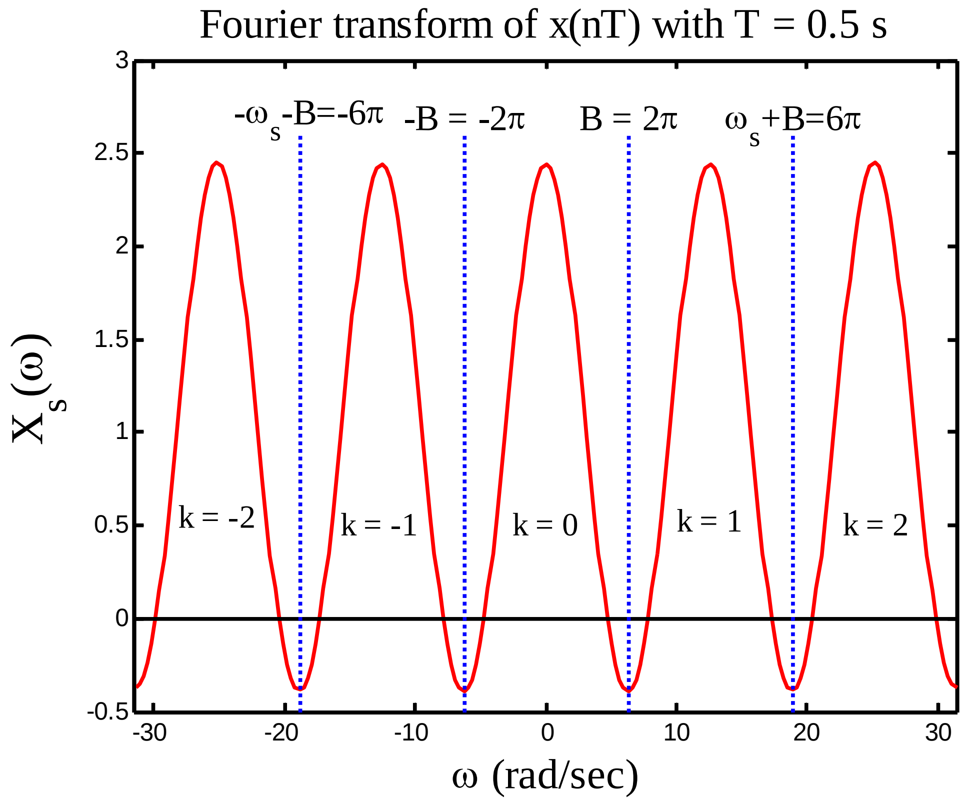
$$\frac{\left|X(\omega \geq B_1)\right|_{\max}}{\left|X(\omega)\right|_{\max}} = \frac{0.0849}{1.2732} * 100\% = 6.67\% = -23.5 \text{ dB}.$$

For $T_2 = 0.25$ s with $B_2 = 4\pi$ rad/s, $\left|X(\omega \geq B_2)\right|_{\max} = 0.0202$, and the aliasing is

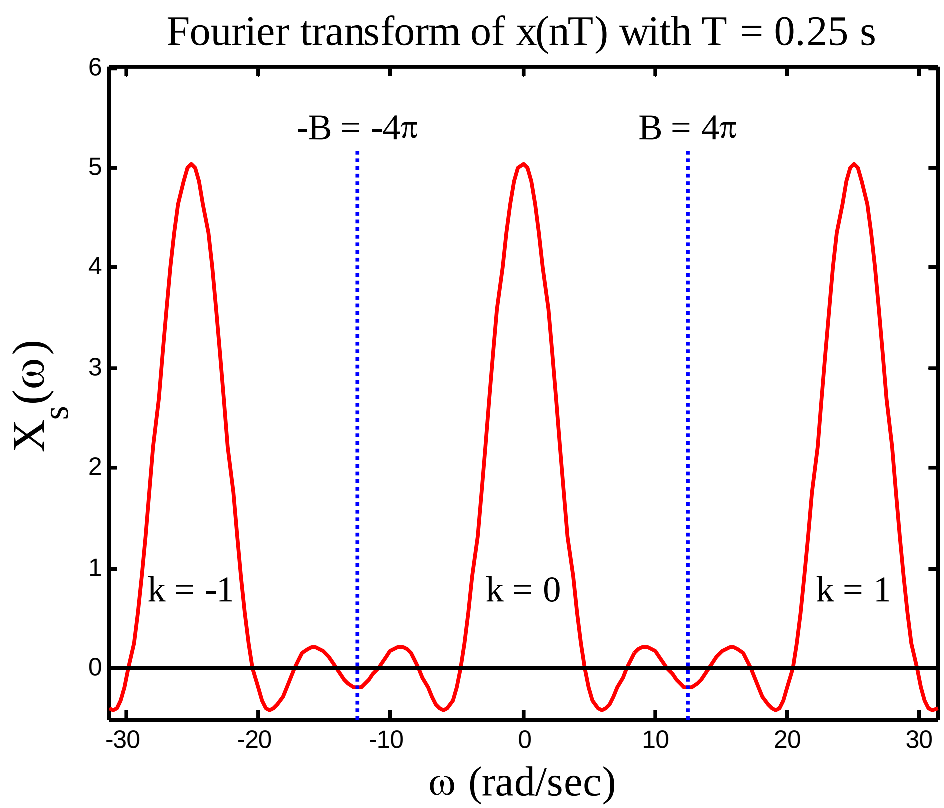
$$\frac{\left|X(\omega \geq B_2)\right|_{\max}}{\left|X(\omega)\right|_{\max}} = \frac{0.0202}{1.2732} * 100\% = 1.59\% = -36 \text{ dB}.$$

When plotting $X_{S1}(\omega)$ and $X_{S2}(\omega)$, I'll restrict the range of k values to something numerically reasonable (i.e., -3 ≤ k ≤ 3). So, what is plotted is actually

$$X_{S1}(\omega) = \sum_{k=-3}^{3} 2\, X(\omega - k4\pi) \text{ and } X_{S2}(\omega) = \sum_{k=-3}^{3} 4\, X(\omega - k8\pi).$$



- Obviously, there is a great deal of overlap in the sampled spectrum (aliasing) and the sidelobes, seen in the plot of $X(\omega)$, are essentially "wiped-out".



- Here, there is less overlap in the sampled spectrum (i.e., less aliasing) and the first two sidelobes, seen in the plot of $X(\omega)$, are essentially untouched.

```
% chap5_sampled_cosine_halfcycle_FT.m
% Interpolation Formula and sampling example
%
% For the sampled cosine half-cycle
%     x(nT)=cos(0.5*pi*n*T) -1<t<1 sec
% plot its Fourier transform
%     Xs(w) = sum[ (1/T)X(w-k*ws) ]
% where the unsampled Fourier transform is
%     X(w) = sinc[(w+0.5pi)/pi)] + sinc[(w-0.5pi)/pi)].
%
clear;clc;close all;
T = 0.25; ws = 2*pi/T; % sampling period & frequency
B = ws/2;
w = -10*pi:pi/10:10*pi; % frequency range for plotting
% Calculate frequency-shifted components of Xs(w)
% Since X(w) dies out fairly rapidly, will only compute the
% finite range of k from -3 to +3
Xsm3 = sinc((w+3*ws+pi/2)/pi)+sinc((w+3*ws-pi/2)/pi); % k=-3
Xsm2 = sinc((w+2*ws+pi/2)/pi)+sinc((w+2*ws-pi/2)/pi); % k=-2
Xsm1 = sinc((w+ws+pi/2)/pi)+sinc((w+ws-pi/2)/pi);     % k=-1
Xs0 = sinc((w+pi/2)/pi)+sinc((w-pi/2)/pi);            % k=0
Xs1 = sinc((w-ws+pi/2)/pi)+sinc((w-ws-pi/2)/pi);      % k=1
Xs2 = sinc((w-2*ws+pi/2)/pi)+sinc((w-2*ws-pi/2)/pi);  % k=2
Xs3 = sinc((w-3*ws+pi/2)/pi)+sinc((w-3*ws-pi/2)/pi);  % k=3
Xs = 1/T*(Xsm3+Xsm2+Xsm1+Xs0+Xs1+Xs2+Xs3);
plot(w,Xs,'r',[-10*pi,10*pi],[0,0],'k',...
    [-B,-B],[-0.5,5.2],'b:',[B,B],[-0.5,5.2],'b:'),
axis([-10*pi 10*pi -0.5 6]),
xlabel('\omega (rad/sec)','fontsize',16,'fontname','times'),
ylabel(['X_{s}(\omega)'],'fontsize',18,'fontname','times'),
title(['Fourier transform of x(nT) with T = ',num2str(T),' s'],...
    'fontsize',16,'fontname','times'),
text(B,1.41/T,['B = ',num2str(B/pi),'\pi'],...
    'HorizontalAlignment','center','VerticalAlignment','top',...
    'fontsize',14,'fontname','times')
text(-B,1.41/T,['-B = -',num2str(B/pi),'\pi'],...
    'HorizontalAlignment','center','VerticalAlignment','top',...
    'fontsize',14,'fontname','times')
text(-ws,1,'k = -1','HorizontalAlignment','center',...
    'VerticalAlignment','top','fontsize',14,'fontname','times')
text(0,1,'k = 0','HorizontalAlignment','center',...
    'VerticalAlignment','top','fontsize',14,'fontname','times')
text(ws,1,'k = 1','HorizontalAlignment','center',...
    'VerticalAlignment','top','fontsize',14,'fontname','times')
set(findobj('type','line'),'linewidth',1.5)
set(findobj('type','line'),'markersize',18)
set(findobj('type','axes'),'linewidth',2)
```

Lastly, we'll use the interpolation formula

$$x(t) = \frac{BT}{\pi} \sum_{n=-\infty}^{\infty} x(nT) \operatorname{sinc}\left[\frac{B}{\pi}(t - nT) \right]$$

to recreate our signal from the sampled (discrete) data points.

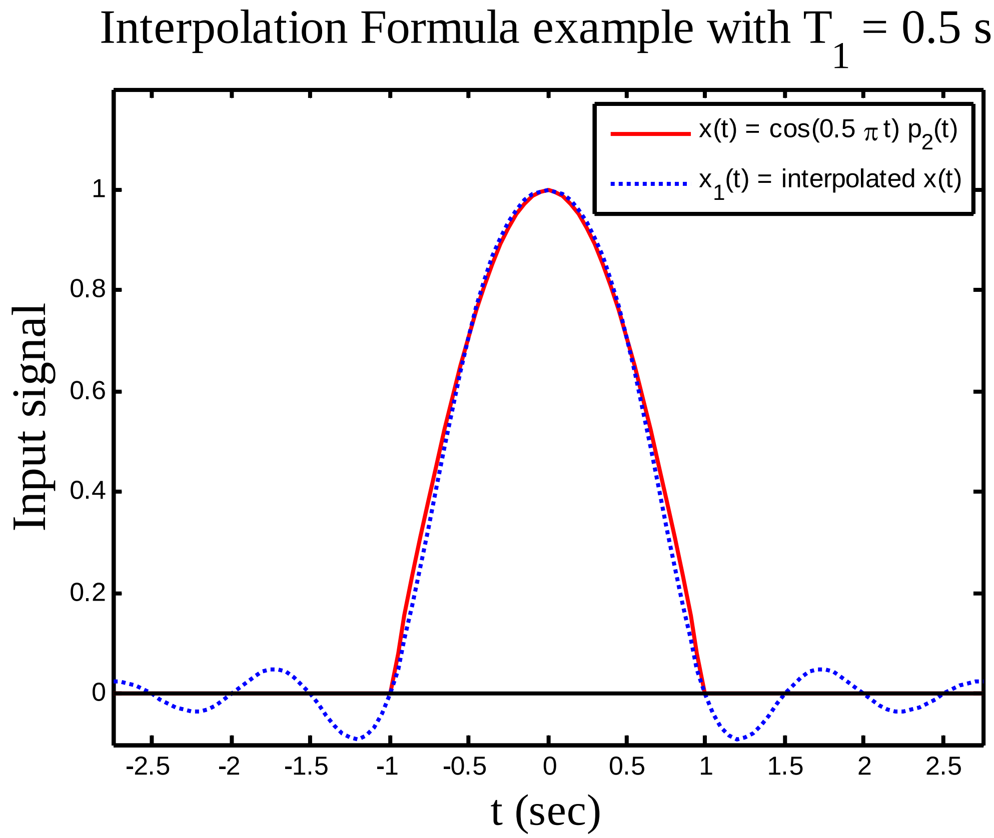
For $T_1 = 0.5$ s with $B_1 = 2\pi$ rad/s, let $x(t) = x_1(t)$. There are only three non-zero sampled data points [see plot of $x(nT_1)$]. Therefore, the interpolation formula is

$$\begin{aligned} x_1(t) &= \frac{2\pi(0.5)}{\pi} \sum_{n=-1}^{1} \cos(0.5\pi\, n 0.5) \operatorname{sinc}\left[\frac{2\pi}{\pi}(t - n0.5) \right] \\ &= \sum_{n=-1}^{1} \cos(0.25\pi\, n) \operatorname{sinc}\left[2t - n \right] \\ &= \cos(-0.25\pi) \operatorname{sinc}\left[2t + 1 \right] + \operatorname{sinc}\left[2t \right] + \cos(0.25\pi) \operatorname{sinc}\left[2t - 1 \right] \\ &= 0.707 \operatorname{sinc}\left[2t + 1 \right] + \operatorname{sinc}\left[2t \right] + 0.707 \operatorname{sinc}\left[2t - 1 \right] \quad -\infty < t < \infty \end{aligned}.$$

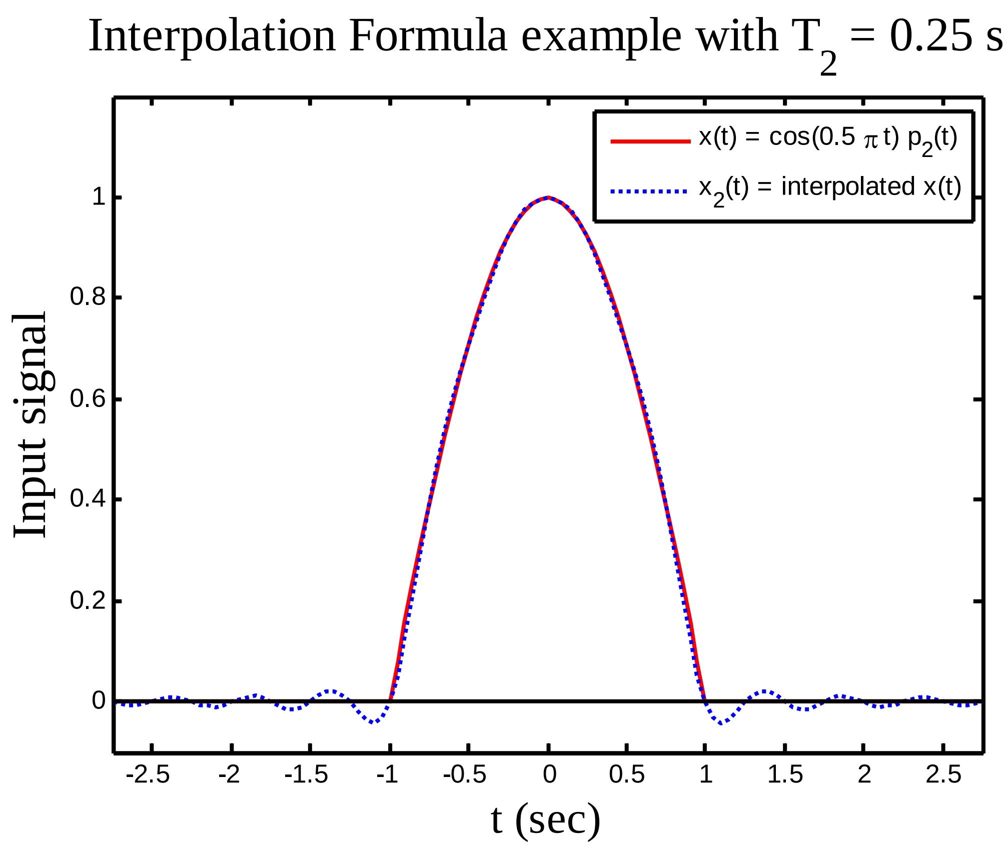
For $T_2 = 0.25$ s with $B_2 = 4\pi$ rad/s, let $x(t) = x_2(t)$. In this case, there are seven non-zero sampled data points [see plot of $x(nT_2)$]. Therefore, the interpolation formula is

$$\begin{aligned} x_2(t) &= \frac{4\pi(0.25)}{\pi} \sum_{n=-3}^{3} \cos(0.5\pi\, n 0.25) \operatorname{sinc}\left[\frac{4\pi}{\pi}(t - n0.25) \right] \\ &= \sum_{n=-3}^{3} \cos(0.125\pi\, n) \operatorname{sinc}\left[4t - n \right] \quad -\infty < t < \infty \end{aligned}.$$

For comparison, the exact signal $x(t)$ is also plotted.



- The error in the recreated signal, due to aliasing, is obvious (roughly 9%).



- The error in the recreated signal, due to aliasing, is smaller (roughly 4.1%).

```
% chap5_cosine_halfcycle_interpolated_T2.m
% Interpolation Formula and sampling example
%
% Recreate x(t) from samples using Interpolation formula
%
clear;clc;close all;
T = 0.25; ws = 2*pi/T; % sampling period & frequency
B = ws/2;
%
xnm3 = cos(0.5*pi*-3*T); % sample at n=-3
xnm2 = cos(0.5*pi*-2*T); % sample at n=-2
xnm1 = cos(0.5*pi*-1*T); % sample at n=-1
xn0 = cos(0.5*pi*0*T);   % sample at n=0
xn1 = cos(0.5*pi*1*T);   % sample at n=1
xn2 = cos(0.5*pi*2*T);   % sample at n=2
xn3 = cos(0.5*pi*3*T);   % sample at n=3
% all other samples are equal to zero
t = -2.75:0.05:2.75;
% Calculate exact expression for x(t)
x = cos(0.5*pi*t);
for i=1:length(t),   % set values less than zero to zero
    if(x(i)<0),
        x(i)=0;
    end
end
xm3 = B*T/pi*xnm3*sinc(B/pi*(t-(-3)*T)); % contrib. from n=-3 sample
xm2 = B*T/pi*xnm2*sinc(B/pi*(t-(-2)*T)); % contrib. from n=-2 sample
xm1 = B*T/pi*xnm1*sinc(B/pi*(t-(-1)*T)); % contrib. from n=-1 sample
x0 = B*T/pi*xn0*sinc(B/pi*(t-(0)*T));    % contrib. from n=0 sample
x1 = B*T/pi*xn1*sinc(B/pi*(t-1*T));      % contrib. from n=1 sample
x2 = B*T/pi*xn2*sinc(B/pi*(t-2*T));      % contrib. from n=2 sample
x3 = B*T/pi*xn3*sinc(B/pi*(t-3*T));      % contrib. from n=3 sample
xint = xm3+xm2+xm1+x0+x1+x2+x3;          % x2(t) interp. from samples
plot(t,x,'r',t,xint,'b:',[-2.75,2.75],[0,0],'k'),
axis([-2.75 2.75 -0.1 1.2]),
ylabel(['Input signal'],'fontsize',18,'fontname','times'),
title('Interpolation Formula example with T_2 = 0.25 s',...
    'fontsize',18,'fontname','times');
xlabel('t (sec)','fontsize',16,'fontname','times'),
legend('x(t) = cos(0.5 \pi t) p_{2}(t)','x_{2}(t)=interpolated x(t)'),
set(findobj('type','line'),'linewidth',1.5)
set(findobj('type','text'),'fontname','times')
set(findobj('type','line'),'markersize',18)
set(findobj('type','axes'),'linewidth',2)
```