

Demo 6.1 – Sampling and Reconstruction (Introduction)

XMUT220 Signals and Systems

Overview

1. Sampling of signals.
2. Sampling and reconstruction of signals.

Exercise 1 – Sampling of Signals

Perform the following signal sampling tasks in MATLAB:

a. Simulate sampling of a continuous time sinusoidal signal with frequency f_m = 10 kHz by using a sampling rate of f_s = 500 kHz.

b. Create the plots of the sampled signals at sampling frequencies of f_{s1} = 30 kHz and f_{s2} = 50 kHz.

c. Create the plots of the sampled signals at sampling frequencies of f_{s1} = 5 Hz, f_{s2} = 10 Hz, f_{s3} = 20 Hz, f_{s4} = 30 Hz, f_{s5} = 50 Hz, and f_{s6} = 100 Hz.

Answer

For the given continuous time sinusoidal signal:

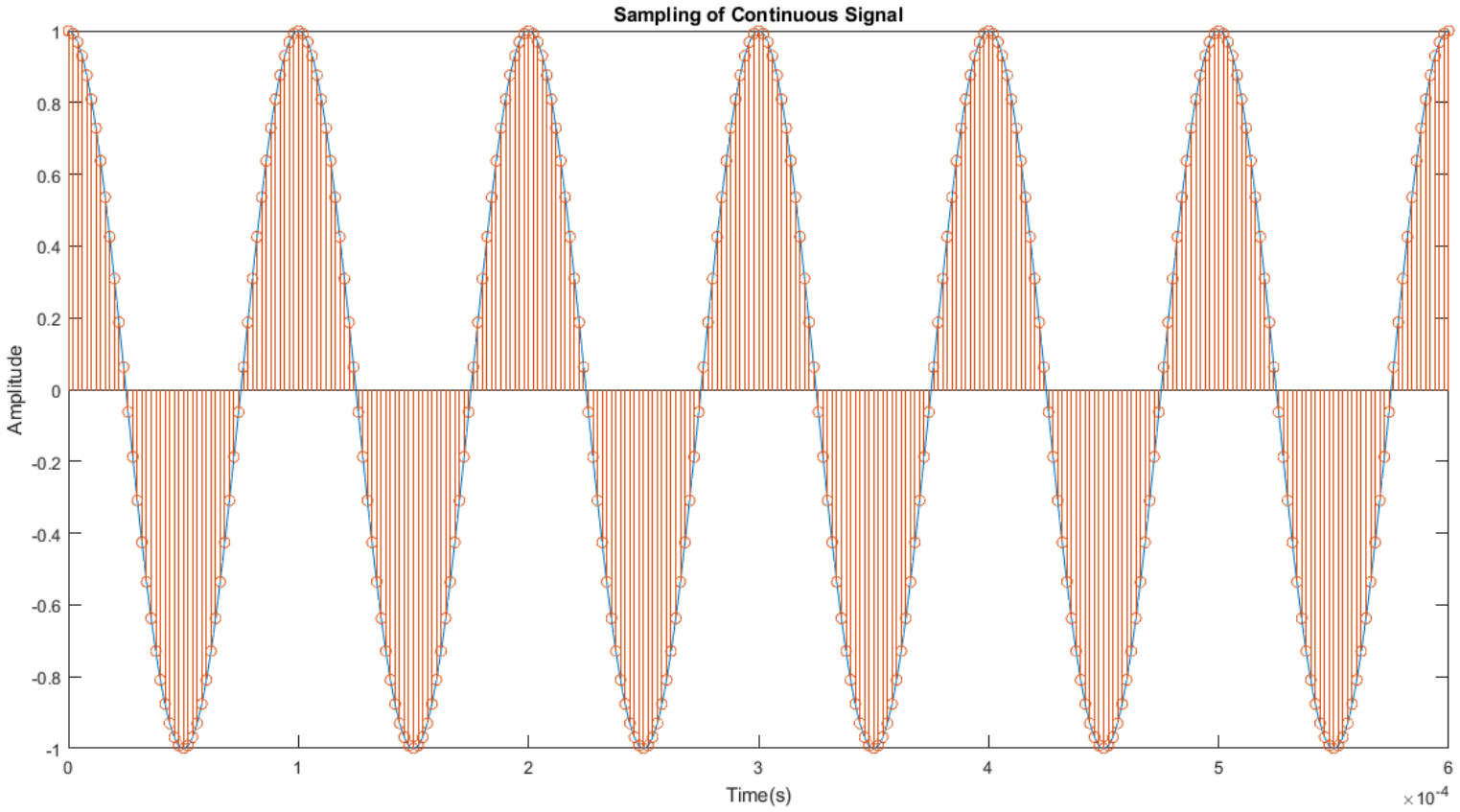
a. Simulate sampling of a simple continuous-like sinusoidal signal with frequency f_m = 10 kHz. In order to make it appear as a continuous signal when plotting, a sampling rate of f_s = 500 kHz is used.

   The following MATLAB code simulates the sampling of the simple continuous time sinusoidal signal.

```
% sampling1.m (plot of signal to be sampled)
nocycle=6; % Number of cycles to be sampled

f=1e4; % Signal frequency of 10 kHz
t=0:1/1e6:nocycle*1/f; % Time index
x=cos(2*pi*f*t); % Signal to be sampled

fs=5e5; % Sampling rate of 500 kHz
ts=0:1/fs:nocycle*1/f; % Time index
xs=cos(2*pi*f*ts); % Sampling signal

plot(t,x); % simulate the signal
hold on;
stem(ts,xs); % simulate the sampled signal
```

```
hold off;

title('Sampling of Continuous Signal');
xlabel('Time(s)');
ylabel('Amplitude');
```

The following plots show the 10 kHz sinusoidal signal which is sampled at frequency = 500 kHz



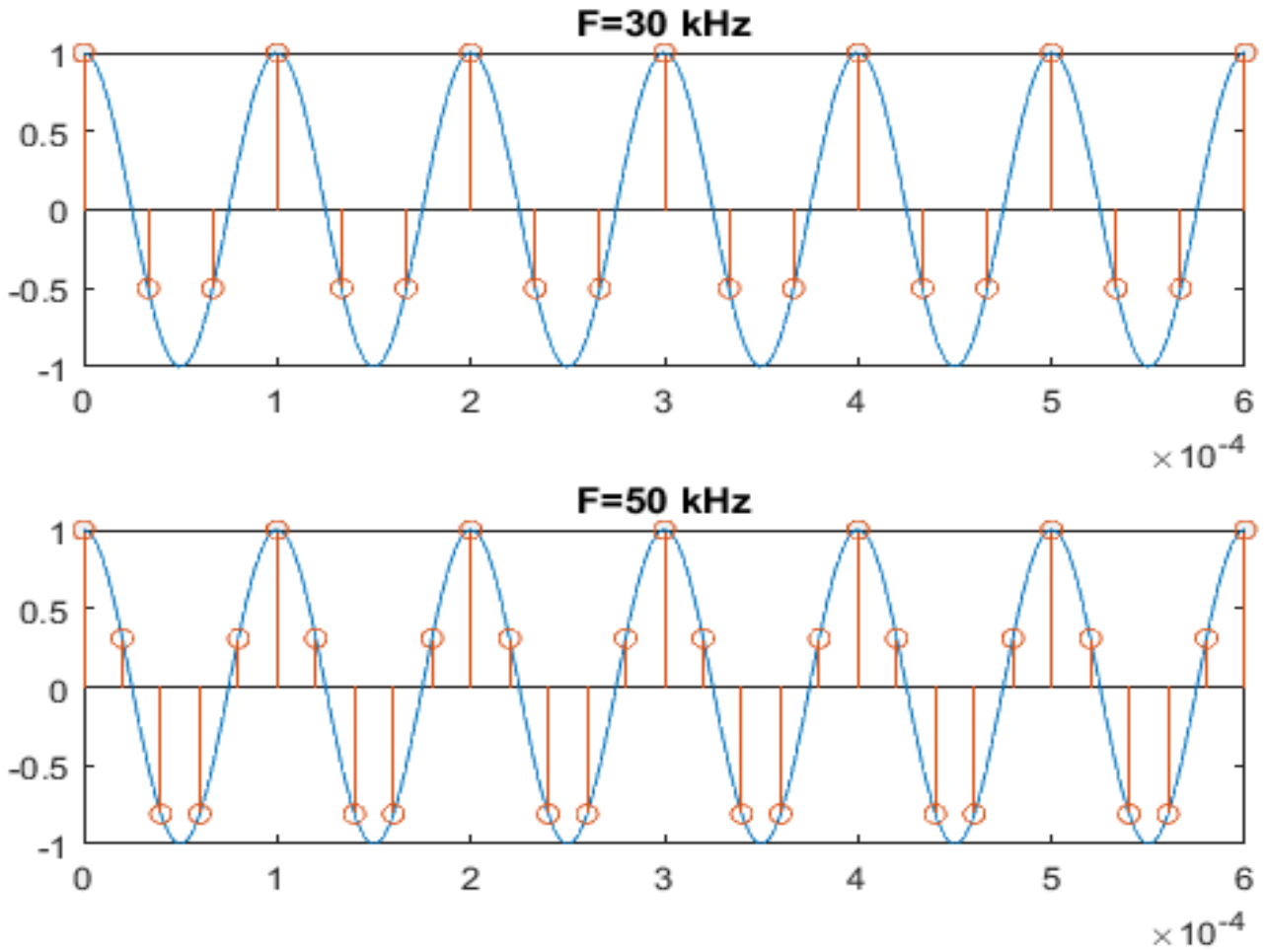
b. After generating the signal as a sinusoidal signal, we would like to convert the signal to discrete-time equivalent by sampling. By sampling theorem, the signal has to be sampled at at-least $f_s = 2*f_m = 20$ kHz and at $f_{s1} = 30$ kHz and then at $f_{s2} = 50$ kHz.

The following MATLAB code samples the signal at 30 kHz and 50 kHz.

```
% sampling2.m (plots of sampled signals)

f=10e3;
nocycle=6;

fs1=30e3; % Sampling rate of 30 kHz
t1=0:1/fs1:nocycle*1/f; %time index
x1=cos(2*pi*f*t1);
```

```
fs2=50e3; % sampling rate of 50 kHz
t2=0:1/fs2:nocycle*1/f; % Time index
x2=cos(2*pi*f*t2);

hold on;
subplot(2,1,1); % Plot first signal
plot(t,x);
stem(t1,x1);
hold off;

hold on;
subplot(2,1,2); % Plot second signal
plot(t,x);
stem(t2,x2);
hold off;
```

The following plots show the signal and its resulting sampled signals at 30 kHz and 50 kHz.



c. Create the plots of the sampled signals and interpolated signals at sampling frequencies of f_{s1} = 5 kHz, f_{s2} = 10 kHz, f_{s3} = 20 kHz, f_{s4} = 30 kHz, f_{s5} = 50 kHz, and f_{s6} = 100 kHz.

The following MATLAB code samples the signal at various sampling frequencies.

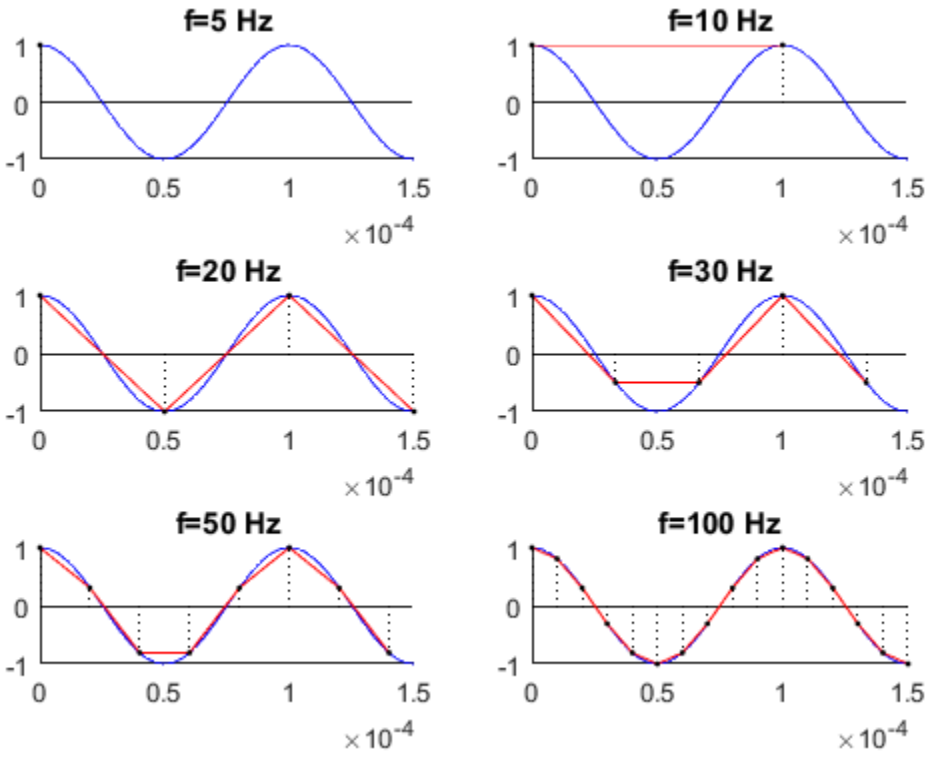
```
% sampling3.m (plot of sampled signals)

nocycle=1.5;

f=10e3; % signal frequency
t=0:1/1e6:nocycle*1/f; % time index of signal at 10 kHz
x=cos(2*pi*f*t);
```

```
fs1=5e3; % sampling rate of 5 kHz
t1=0:1/fs1:nocycle*1/f; % time index of sampled signal
x1=cos(2*pi*f*t1);
subplot(3,2,1); % plot sampled signal
hold on;
plot(t,x,'b');
plot(t1,x1,'r');
stem(t1,x1,'k.:');
title('f=5 Hz');
hold off;

fs2=10e3; % sampling rate of 10 kHz
t2=0:1/fs2:nocycle*1/f; % time index of sampled signal
x2=cos(2*pi*f*t2);
subplot(3,2,2); % plot sampled signal
hold on;
plot(t,x,'b');
plot(t2,x2,'r');
stem(t2,x2,'k.:');
title('f=10 Hz');
hold off;
```

```
fs3=20e3; % sampling rate of 20 kHz
t3=0:1/fs3:nocycle*1/f; % time index of sampled signal
x3=cos(2*pi*f*t3);
subplot(3,2,3); % plot sampled signal
hold on;
plot(t,x,'b');
plot(t3,x3,'r');
stem(t3,x3,'k.:');
title('f=20 Hz');
hold off;

fs4=30e3; % sampling rate of 30 kHz
t4=0:1/fs4:nocycle*1/f; % time index of sampled signal
x4=cos(2*pi*f*t4);
subplot(3,2,4); % plot sampled signal
hold on;
plot(t,x,'b');
plot(t4,x4,'r');
stem(t4,x4,'k.:');
title('f=30 Hz');
hold off;
```

```
fs5=50e3; % sampling rate of 50 kHz
t5=0:1/fs5:nocycle*1/f; % time index of sampled signal
x5=cos(2*pi*f*t5);
subplot(3,2,5); % plot sampled signal
hold on
plot(t,x,'b');
plot(t5,x5,'r');
stem(t5,x5,'k.:');
title('f=50 Hz');
hold off;

fs6=100e3; % sampling rate of 100 kHz
t6=0:1/fs6:nocycle*1/f; % time index of sampled signal
x6=cos(2*pi*f*t6);
subplot(3,2,6); % plot sampled signal
hold on
plot(t,x,'b');
plot(t6,x6,'r');
stem(t6,x6,'k.:');
title('f=100 Hz');
hold off;
```

The following plots show the signal and sampling signal at various sampling frequencies.



Exercise 2 – Sampling and Reconstruction

Reconstruct in the MATLAB the sampled sinusoidal signal back to their respective continuous time signals. This case is a signal with frequency = 200 Hz and sampling rate of frequency of 50 Hz, 100 Hz, 200 Hz, 400 Hz, 500 Hz and 1,000 Hz.

a. Create plots of reconstructed continuous time signals of the sampled signals.

b. Comment on the differences between the reconstructed signals with the original continuous time signals.

Where T_s is the original sampling period and T_r is the reconstruction sampling period.

In the code T_s is given by T_s = 1/500 for sampling frequency of 500 Hz and T_r = 1/10,000 set by your continuous plot period.

Note that you will still have slight errors due to the fact the input signal is not being perfectly bandlimited.

The sinc interpolation formula for reconstruction of the sampled signal is defined as:

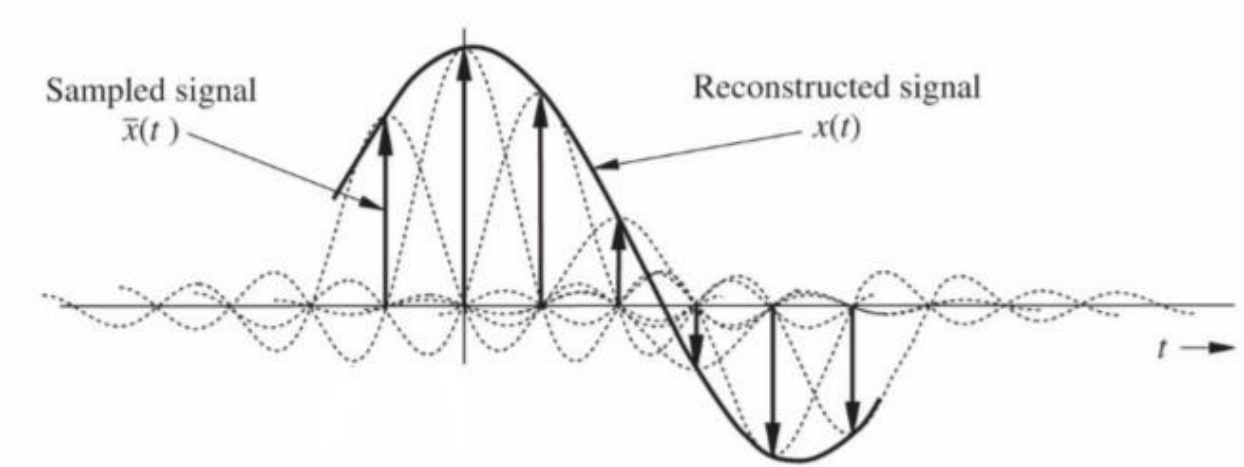
$$x(t) = \sum_{n=-\infty}^{\infty} x_n \, sinc \left[\frac{\pi}{T}(t - nT)\right]$$

Where: T is the sampling period used to determine x_n from the original signal, and x_t is the reconstructed signal.

The function $sinc(x)$ is defined by $sinc(x) = sin(x)/x$ for $\neq 0$, with $sinc(0) = 1$.

The above formula represents a linear convolution between the sequence x_n and scaled and shifted samples of the $sinc$ function.

In MATLAB the $sinc$ function is realised with `sinc()` function.



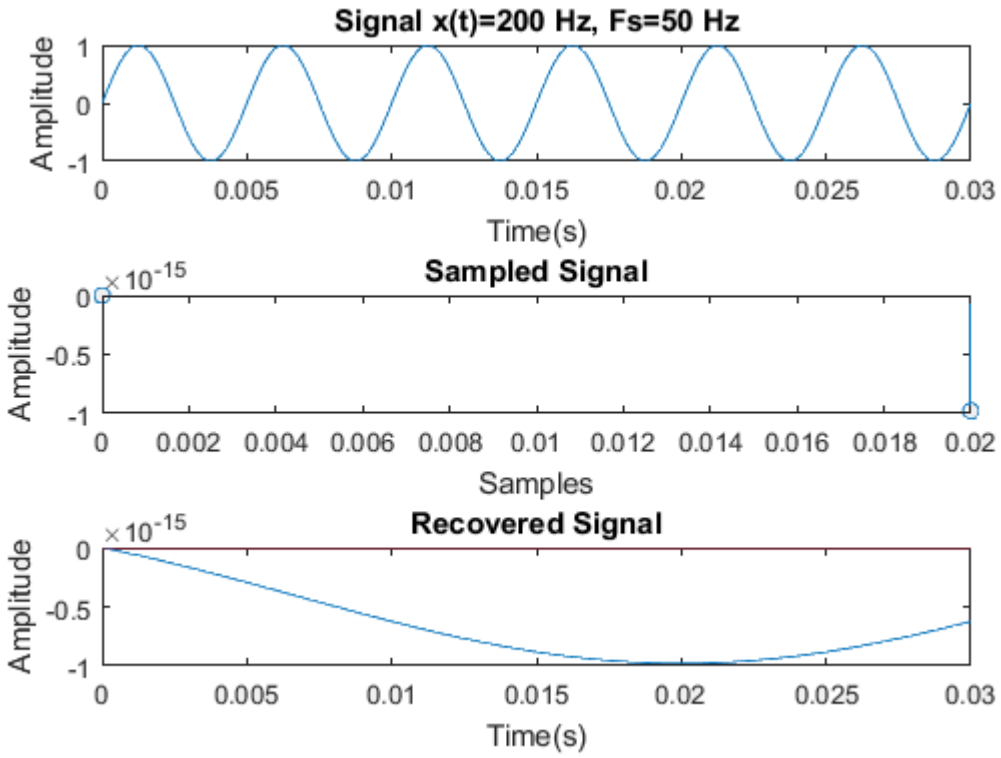
The following MATLAB code shows the sampling and reconstruction process. It also highlights the issues related with sampling whenever inadequate numbers of sampling samples taken in the sampling process.

```
% samplingreconstruction.m

nocycle=6;
f=200;

fs=[50,100,200,400,500,1000]; % sampling frequencies

for i = 1:length(fs)

 Ts = 1/fs(i); % Sampling Period
 t = 0:1e-4:nocycle/f; % time vector for continuous plot
 w = f*2*pi;
 xa = sin(w*t); % signal in continuous form

 t2=(0:Ts:nocycle/f);  % time vector for sampling
 xb = sin(w*t2); % sampled signal
```

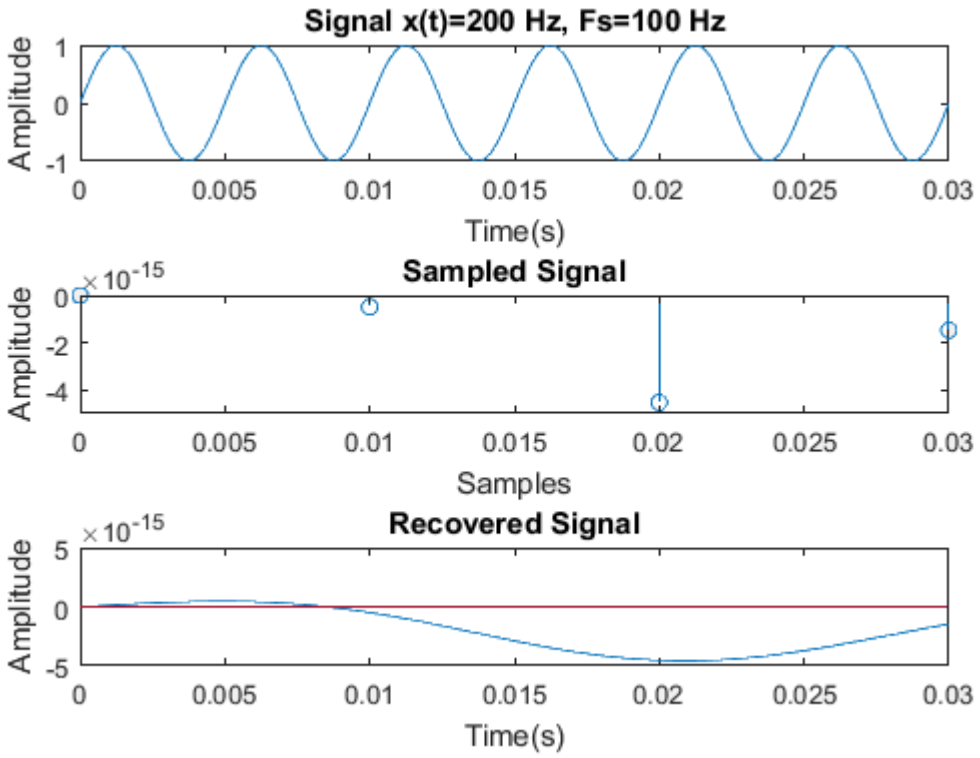
```
figure

subplot(3,1,1);

plot(t,xa) % continuous signal plot

str = ['Signal x(t)='num2str(f)'Hz',',Fs=' num2str(fs(i))'Hz'];
title(str);
ylabel('Amplitude');
xlabel('Time(s)');

subplot(3,1,2);

stem(t2,xb); % sampled signal plot

title('Sampled signal');
ylabel('Amplitude')
xlabel('Samples');
```

```
 xr = zeros(length(t));

 for k=1:length(t)
        for p=1:length(xb)
                xr(k)=xr(k)+xb(p)*sinc(fs(i)*(k-t2(p)));
        end
 end

 subplot(3,1,3);

 plot(t,xr);

 title('Recovered signal');
 ylabel('Amplitude');
 xlabel('Time(s)');

end
```

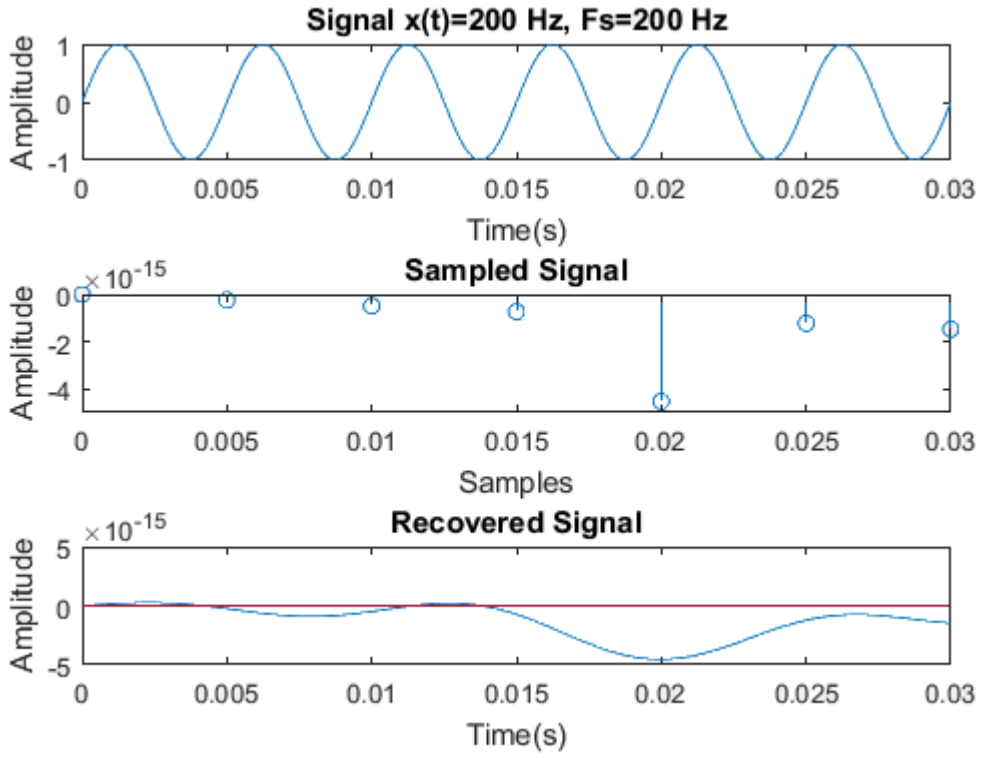
The following plot is the resulting plot for a signal of 200 Hz with a sampling frequency 50 Hz and reconstruction timing, T_r= 0.0001 second.



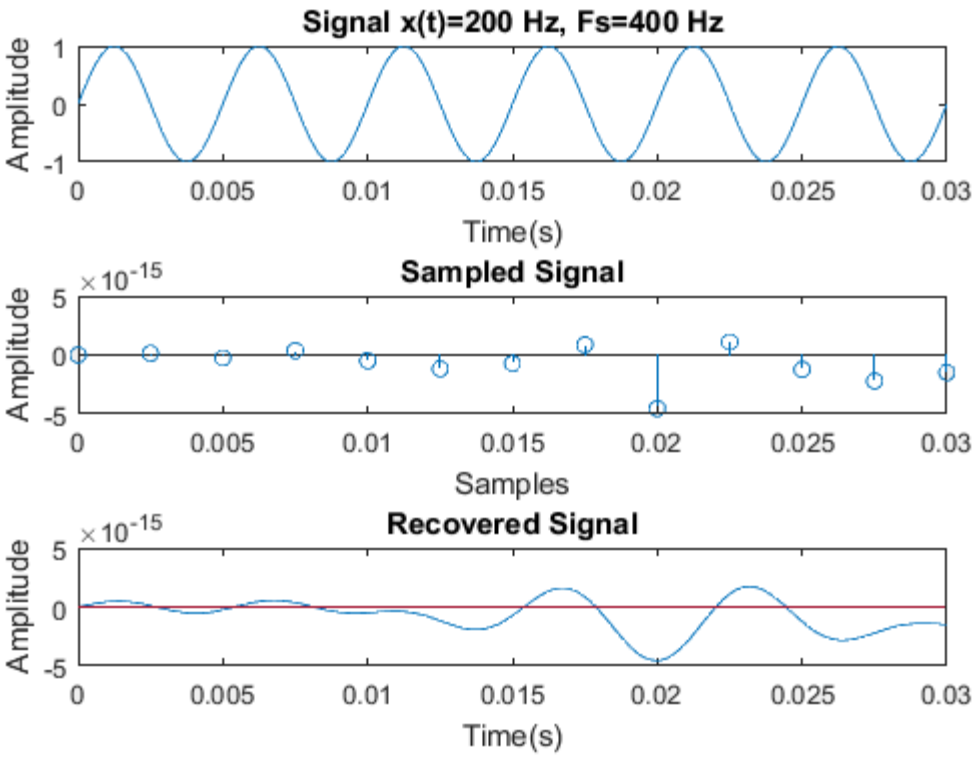
The following plot is the resulting plot for a signal of 200 Hz with a sampling frequency 100 Hz and reconstruction timing, T_r= 0.0001 second.



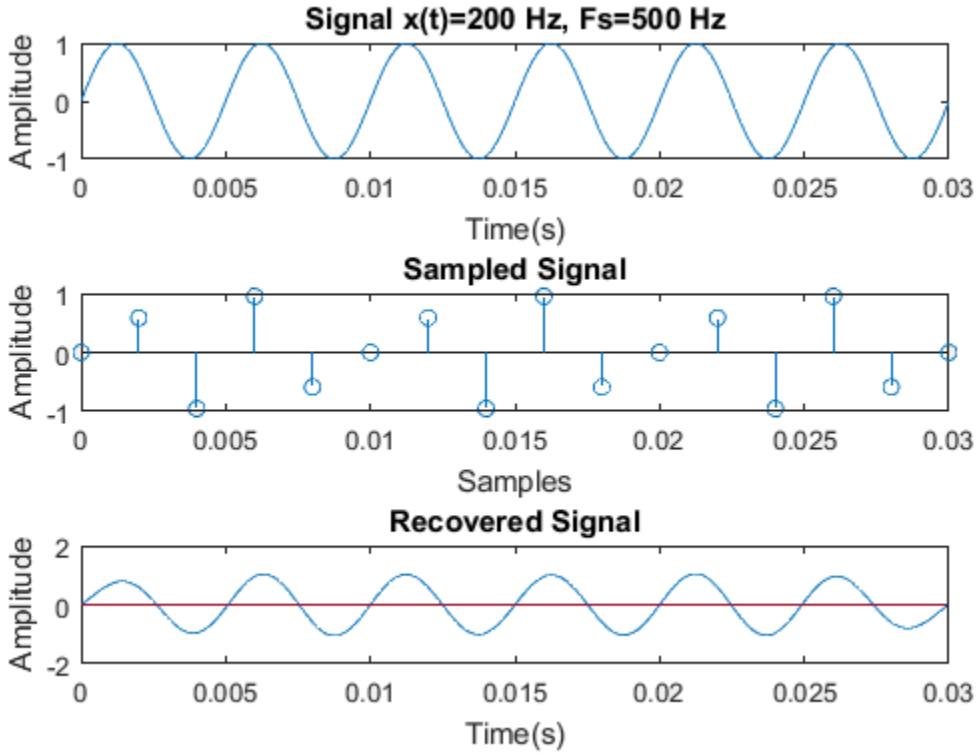
The following plot is the resulting plot for a signal of 200 Hz with a sampling frequency 200 Hz and reconstruction timing, T_r= 0.0001 second.



The following plot is the resulting plot for a signal of 200 Hz with a sampling frequency 400 Hz and reconstruction timing, T_r= 0.0001 second.



The following plot is the resulting plot for a signal of 200 Hz with a sampling frequency 500 Hz and reconstruction timing, T_r= 0.0001 second.



The following plot is the resulting plot for a signal of 200 Hz with a sampling frequency 1000 Hz and reconstruction timing, T_r= 0.0001 second.

