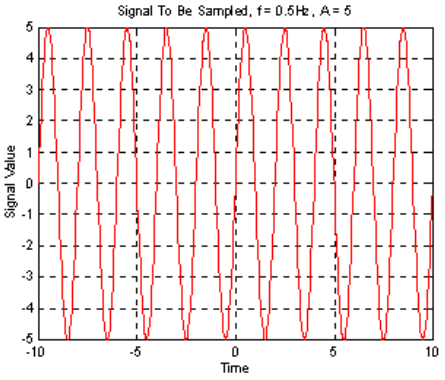
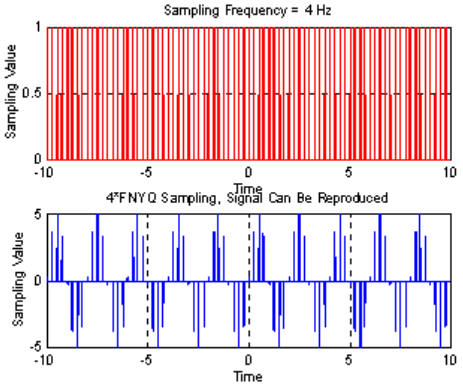
Appendix D: Supplementary Problems on Pulse Modulation

Example D.1.1: Sampling at Nyquist rate, under sampling and over sampling. Our task is this problem is to sample a 0.5Hz Sine wave by means of Natural Sampling. Pulses of varying frequencies are used to sample the signal. The signal varies from +5 V to -5 V. The Nyquist Rate of the signal is 1Hz.

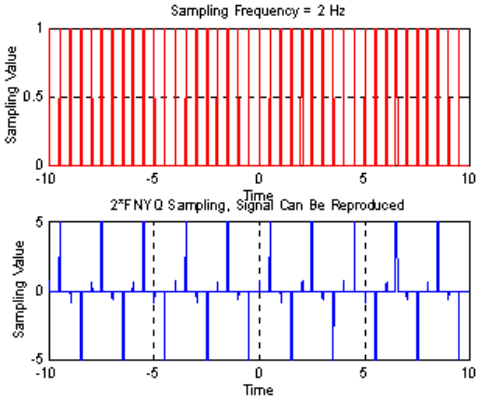
Original Signal to be sampled



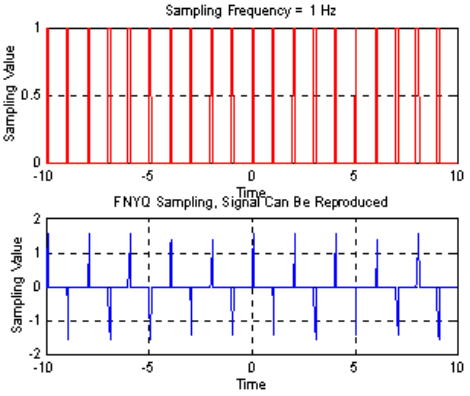
Sampling at 4.0Hz (Oversampled at 4*Nyquist Rate)



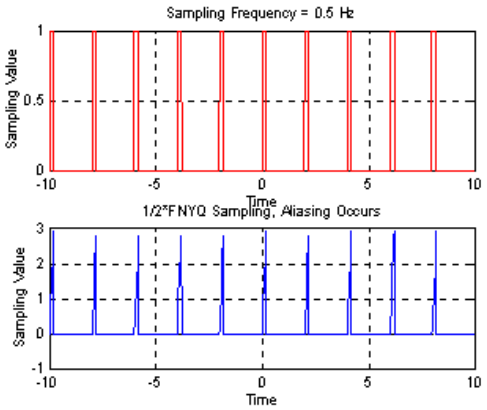
Sampling at 2.0Hz (Oversampled at 2*Nyquist Rate)



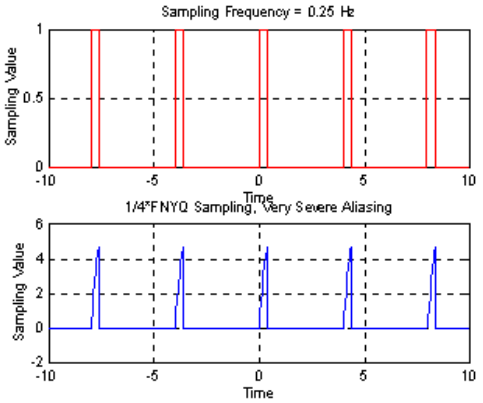
Sampling at 1.0Hz (Nyquist Rate)



Sampling at 0.5Hz (Nyquist Rate/2)



Sampling at 0.25Hz (Nyquist Rate/4)

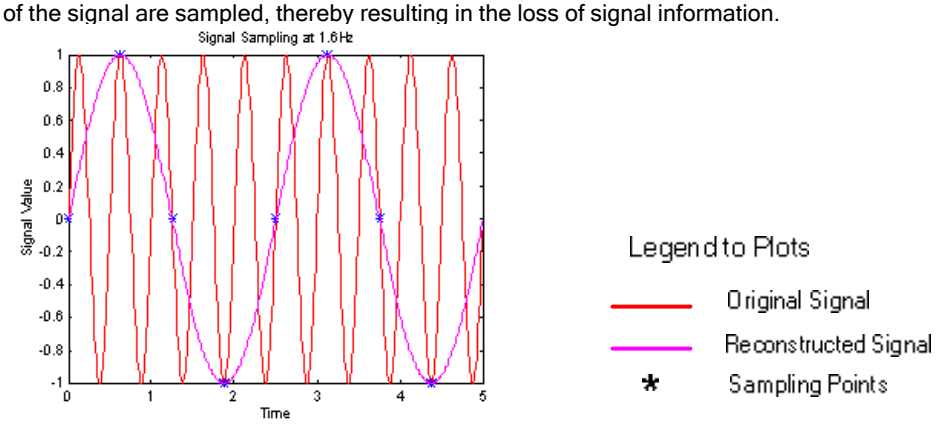


As we can deduce from this example, signals can exactly be reconstructed from their samples when the sampling rate is at least twice the signal frequency. Sampling the signal any lower than its Nyquist

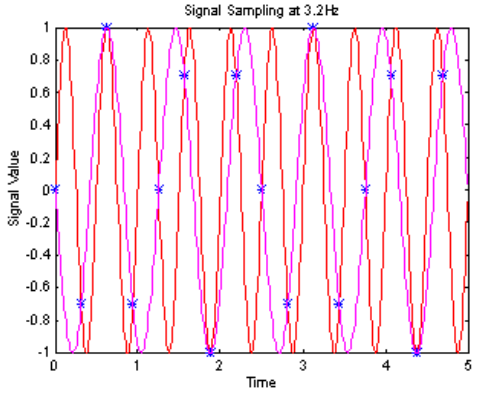
Rate would result in loss of signal information and aliasing (spectral fold-over distortion) is unavoidable.

Example D .1.2: In this simple illustration of aliasing, we sample a 2Hz Sine wave at 1.6Hz, 3.2Hz and 6.4Hz over a 5 second interval (A/D Stage). In the reconstruction process (D/A Stage), the signal is then synthesized at the sample resolution which clearly shows the loss of signal information when a signal is undersampled.

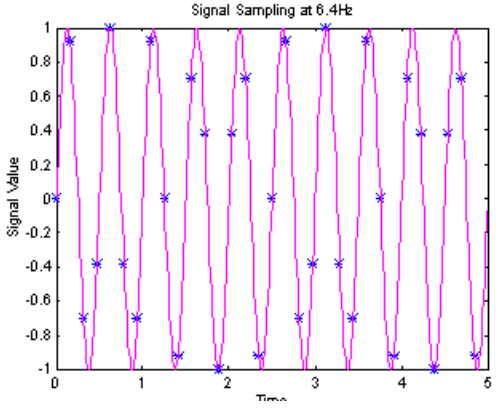
Case 1: Sampling at 1.6 Hz. Here, the samples actually result in a 0.4 Hz signal and not the 2Hz signal that we want to see. The number of samples is actually too small to be able to exactly recover the 2.0 Hz signal. It can also be seen that not all cycles of the signal are sampled, thereby resulting in the loss of signal information.



Case 2: Sampling at 3.2 Hz. Although the number of samples is larger than the previous case, we are still sampling below the Nyquist Rate. Thus there is some loss of information but it is not as severe as in the 0.8Hz case.

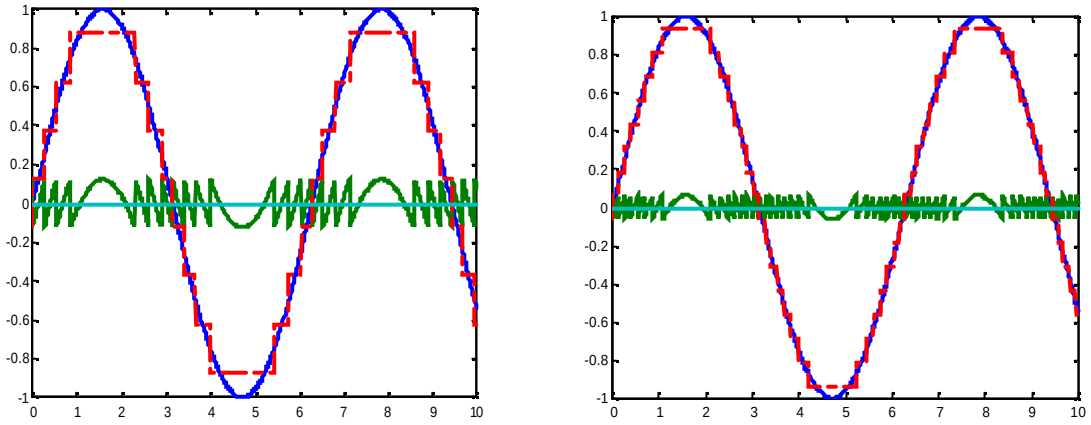


Case 3: Sampling at 6.4 Hz. Here, we have managed to reproduce the exact curve. Notice the original and the sampled version overlap exactly. Also, as you can see from the plots, there are more than two samples for each cycle of the signal. By increasing the sampling rate, we increase the number of samples which would result in a more accurately reproduced signal.



Example D.1.3: In this example we demonstrate the Matlab implementation of uniform quantizers used in PCM systems for a sinusoidal signal with an amplitude 1.0 and $w=1$ rad/s. Use 8-level and 16-level quantizers and plot input/output signals and the quantization error.

```
% MATLAB Source Code for  Uniform Quantization
echo on;  t=[0:0.01:10];
a=sin(t);
[sqnr8,aquan8,code8]=u_pcm(a,8);
[sqnr16,aquan16,code16]=u_pcm(a,16);
pause;                  % Press a key for SQNR of N = 8.
sqnr8

pause                   % Press a key for SQNR of N = 16.
sqnr16

% Plots for input/output & error signal.
error1 = a-aquan8; L=length(t);
plot(t,a,t,error1, '-',t,aquan8, '-.',t,zeros(1,L));
error2 = a-aquan16; figure;
plot(t,a,t,error2, '-',t,aquan16,'-. ',t,zeros(1,L));
```

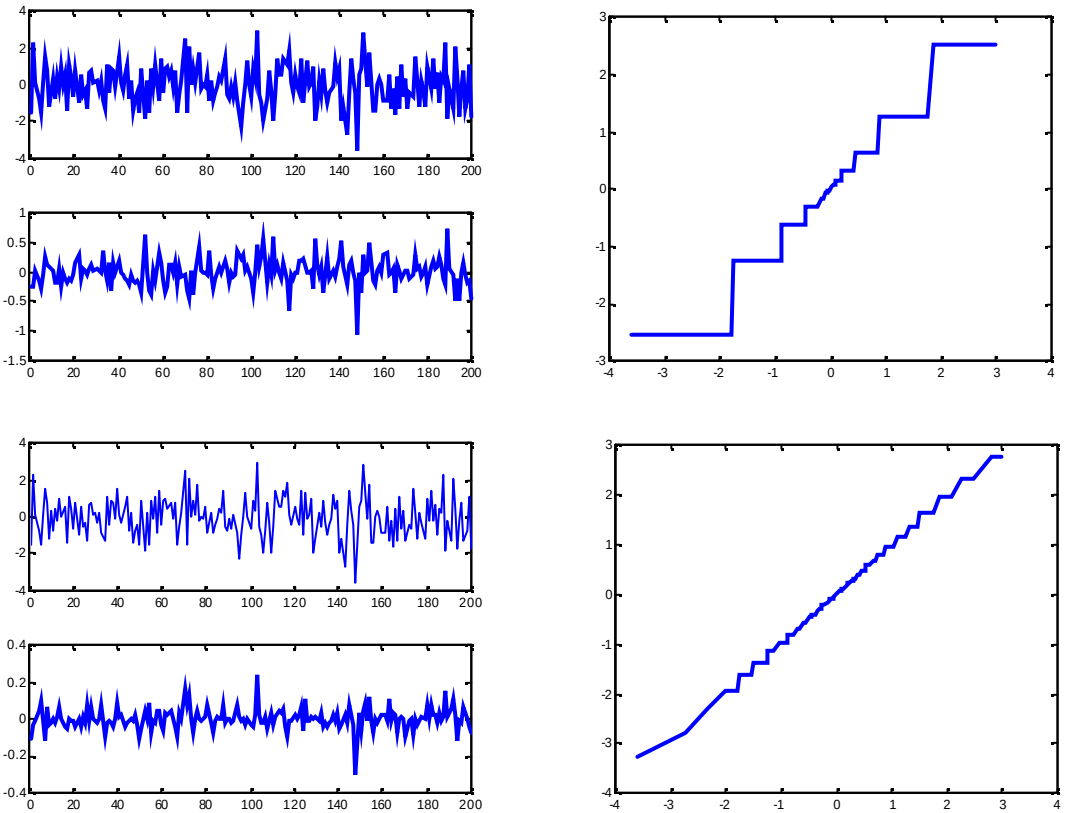


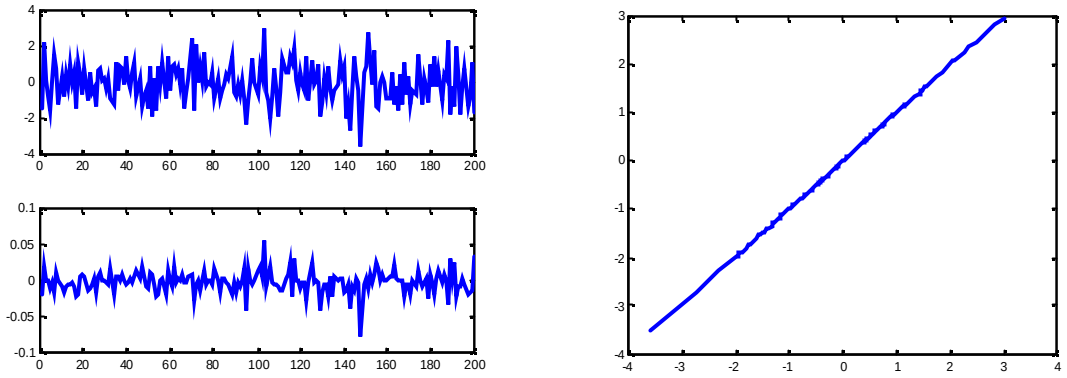
Example D.1.4: In this example we study Non-Linear Quantizers used in PCM Systems.

```
% Non-linear quantizers
t=1:200; a=randn(1,200);
[sqnr,a_quan,code]=mula_pcm(a,16,255);

pause;          % Press a key for SQNR of N = 16.
Sqnr

% Plots for input & error signal
error16=a-a_quan; subplot(2,1,1); plot(t,a);
subplot(2,1,2); plot(t,error16); figure;
[Y,I]=sort(a); plot(Y,a_quan(I))

[sqnr,a_quan,code]=mula_pcm(a,64,255);
pause;          % Press a key for SQNR of N = 64.
```

```
sqnr

% Plots for input & error signal
error64=a-a_quan; figure; subplot(2,1,1); plot(t,a);
subplot(2,1,2); plot(t,error64); figure;
[Y,I]=sort(a); plot(Y,a_quan(I))

[sqnr,a_quan,code]=mula_pcm(a,256,255);
pause;               % Press a key for SQNR of N = 256.
sqnr

% Plots for input & error signal
error256=a-a_quan; figure;
subplot(2,1,1); plot(t,a);
subplot(2,1,2); plot(t,error256);
figure;
[Y,I]=sort(a); plot(Y,a_quan(I))
```

Results from the execution of the code:
SQNR16 = 12.37 dB; SQNR64= 25.6 dB; SQNR= 37.5 dB





Example D.1.5: In this Matlab demanstration of Linear (LM) and ADM systems as per Example 6.9 in the text.

```
%Example on Linear and Adaptive Delta Modulation
% Modified an example from Haykin: Communication Systems, 4th Edition, Wiley, 2001

% Generating sine wave as input signal and Initialization
t=[0:2*pi/100:2*pi];
a=10*sin(t);
n=length(a); dels=1;
xhat(1:n)=0; x(1:n)=a; d(1:n)=0;

%CASE 1: Linear Delta Modulation
for k=1: n
   if (x(k)-xhat(k)) > 0
      d(k)=1;
   else d(k)=-1;
   end %if
   xtilde(k)=xhat(k)+d(k)*dels;
    xhat(k+1)=xtilde(k);
end %

%Plots
figure(1); hold on;
plot(a); plot(xhat,'-.');plot(d-15);
xlabel('Number of iterations'); ylabel('Amplitude');
title('Linear DM Case Top: Input/Approximation and Bottom: Coded Output');
axis([0 100 -20 20])

%CASE 2: Adaptive Delta Modulation with Eqn: 6.41
mindels=1; dels(1:n)=mindels;
xhat(1:n)=0;
x(1:n)=a; d(1:n)=1;
for k=2:n
   if ((x(k)-xhat(k-1)) > 0 )
      d(k)=1;
   else d(k)=-1;
   end %if
```

```
        if k==2
        xhat(k)=d(k)*mindels+xhat(k-1);
        end
        if ((xhat(k)-xhat(k-1)) > 0)
           if (d(k-1)  == -1 &d(k) ==1)
                 xhat(k+1)=xhat(k)+0.5*(xhat(k)-xhat(k-1));
           elseif (d(k-1)  == 1 &d(k) ==1)
                 xhat(k+1)=xhat(k)+1.15*(xhat(k)-xhat(k-1));
           elseif (d(k-1)  == 1 &d(k) ==-1)
                 xhat(k+1)=xhat(k)-0.5*(xhat(k)-xhat(k-1));
           elseif (d(k-1)  == -1 &d(k) ==-1)
                 xhat(k+1)=xhat(k)-1.15*(xhat(k)-xhat(k -1));
           end
        else
          if (d(k-1)  == -1 &d(k) ==1)
                 xhat(k+1)=xhat(k)-0.5*(xhat(k)-xhat(k-1));
          elseif (d(k-1)  == 1 &d(k) ==1)
                 xhat(k+1)=xhat(k)-1.15*(xhat(k)-xhat(k -1));
          elseif (d(k-1)  == 1 &d(k) ==-1)
                 xhat(k+1)=xhat(k)+0.5*(xhat(k)-xhat(k-1));
          elseif (d(k-1)  == -1 &d(k) ==-1)
                 xhat(k+1)=xhat(k)+1.15*(xhat(k)-xhat(k-1));
      end
        end
end

%Plots
figure(2);hold on;
plot(a); plot(xhat,'-.'); plot(d-15)
xlabel('Number of iterations); ylabel('Amplitude');
title('Adaptive DM Case Top: Input/Approximation and Bottom: Coded Output);
```