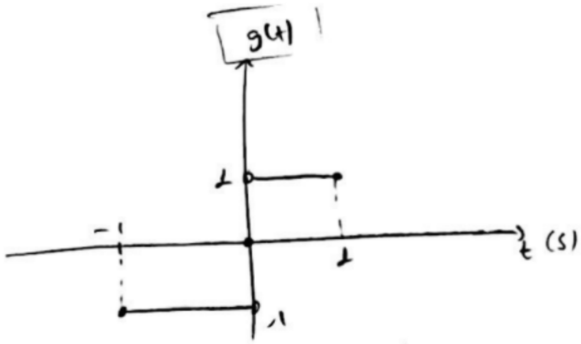
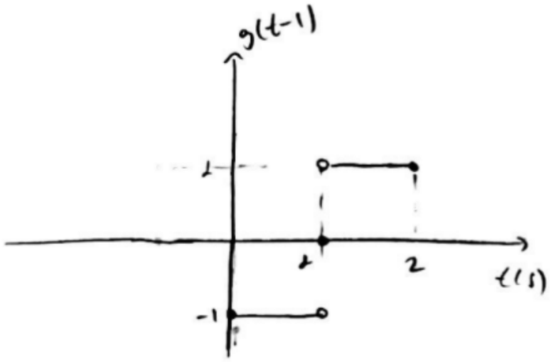
Lab 05

Mehmet Berk Şahin
21703190

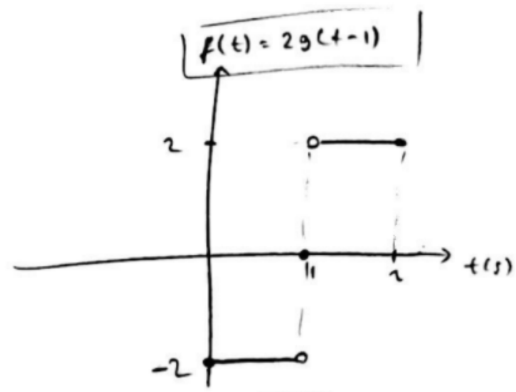
Part 1

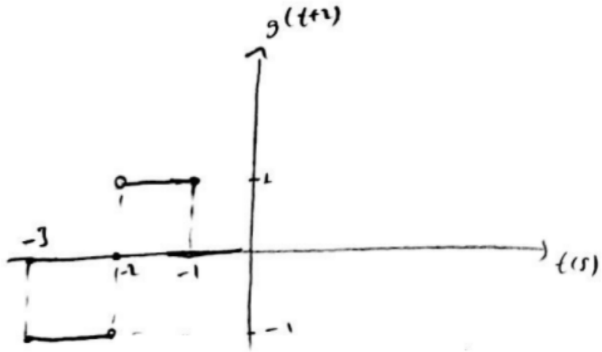
$f(t) = g(-2t+2)$



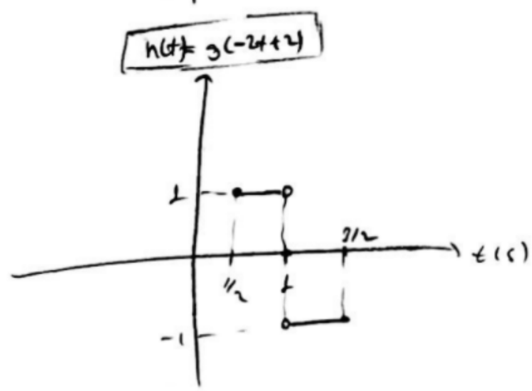


$\Rightarrow$





$\Rightarrow$



(*) It is not possible to perfectly recover the signal g(t) from its samples because g(t) is not a band-limited signal, $G(j\omega) \neq 0$ for $|\omega| > \omega_m$. This contradicts the first criteria of sampling theorem.

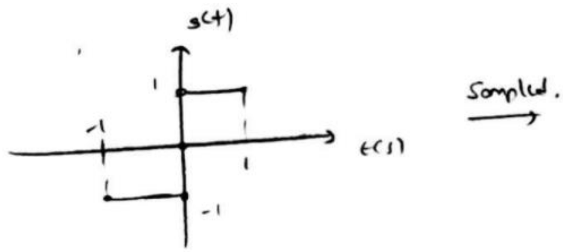
$$G(j\omega) = \int_{-\infty}^{+\infty} g(t)\, e^{-j\omega t}\, dt = \int_0^1 e^{-j\omega t}\, dt + \int_{-1}^{0} (-e^{-j\omega t})\, dt = \frac{1}{j\omega}\left(1 - e^{-j\omega} + 1 - e^{+j\omega}\right)$$

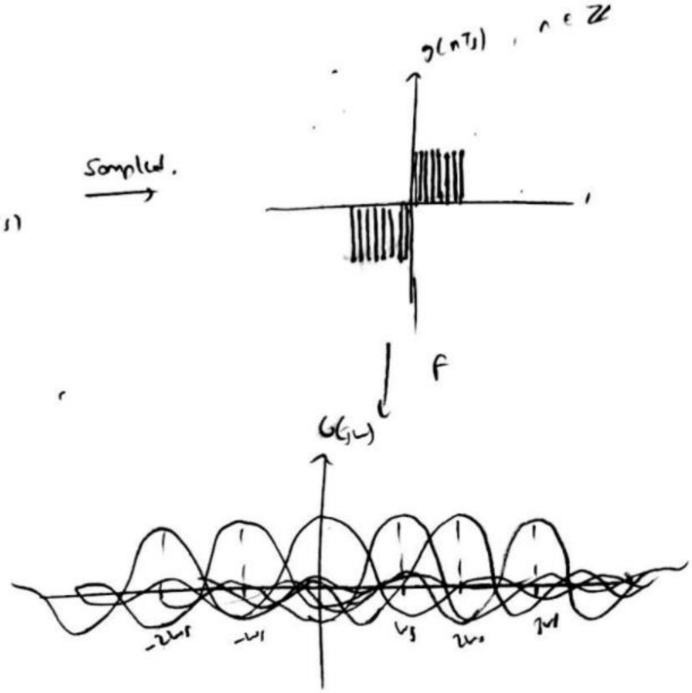
$$\left.\frac{e^{-j\omega t}}{-j\omega}\right|_0^1 = \frac{e^{-j\omega}-1}{-j\omega} \qquad \left.\frac{+e^{-j\omega t}}{j\omega}\right|_{-1}^{0} = \frac{1-e^{+j\omega}}{j\omega}$$

$$\Rightarrow G(j\omega) = \frac{1}{j\omega}\left(2 - 2\cos(j\omega)\right)$$

$G(j\omega) \neq 0$ for $|\omega| > \omega_m$

All waves are overlapped and distorted because all of them goes to infinity, thus, we cannot distinguish them making T_s smaller and making ω_s larger.





② $\tilde{x}(t) * p(t) = x_R(t)$

$\tilde{x}(t) = \sum_{n=-\infty}^{+\infty} x(nT_s)\,\delta(t - nT_s)$ and $\tilde{x}[n] = x(nT_s)$ for $n \in \mathbb{Z}, -\infty < n < \infty$

$$\Rightarrow \quad \tilde{x}(t) = \sum_{n=-\infty}^{+\infty} \tilde{x}[n]\,\delta(t-nT_s)$$

Shifting Property

② $\Rightarrow$ $x_R(t) = \sum_{n'=-\infty}^{+\infty} \tilde{x}[n']\,p(t - n'T_s)$ //

$$x_R(nT_s) = \sum_{n'=-\infty}^{+\infty} \tilde{x}[n']\,p(T_s(n-n')) \quad \Rightarrow \quad k = n - n'$$

if $p(0) = 1$ and $p(kT_s) = 0$ for all nonzero integer k

$\Rightarrow \quad x_R(nT_s) = \tilde{x}[n]$ // ✓ → Consistent interpolation.

Answers:

a) $P_Z(0) = 1$, $P_L(0) = 1$, $P_I(0) = 1$

b) $P_Z(kT_s) = 0$, $P_L(kT_s) = 0$, $P_I(kT_s) = 0$ for $k \neq 0$, $k \in \mathbb{Z}$

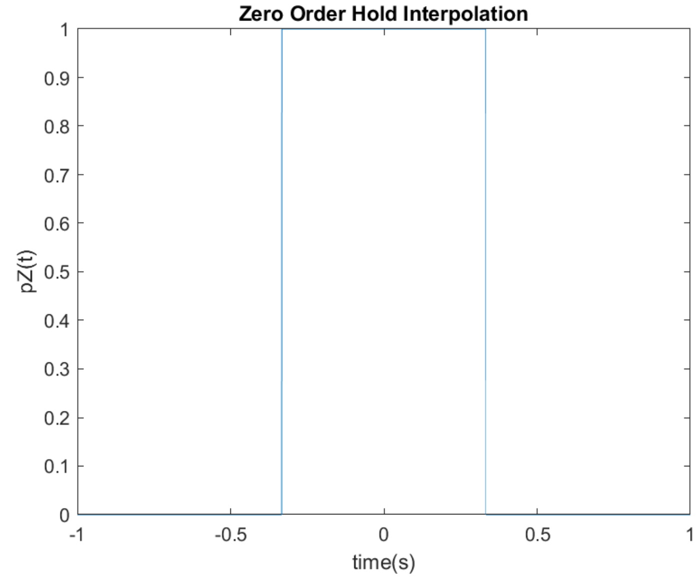
c) Yes, all interpolations are consistent.

Part 3:

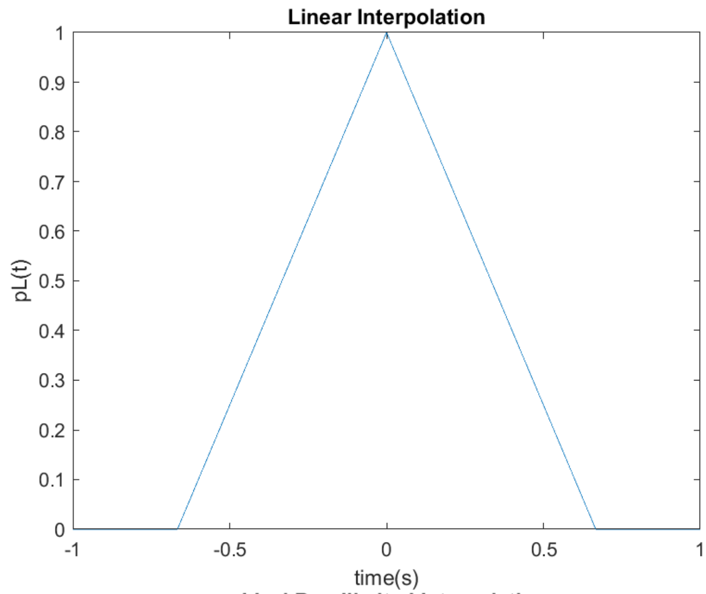
Matlab Code:

```
dur = rem(21703190,5);
if dur == 0
    dur =2
end
Ts= dur/3;
t=[-dur/2:Ts/1000:dur/2-Ts/1000];
%Zero Order Hold Interpolation
pz = generateInterp(0,Ts,dur);
plot(t,pz);
title("Zero Order Hold Interpolation");
xlabel("time(s)");
ylabel("pZ(t)");
figure;
%Linear Interpolation
pl = generateInterp(1,Ts,dur);
plot(t,pl);
title("Linear Interpolation");
xlabel("time(s)");
ylabel("pL(t)");
figure;
%Ideal Bandlimited Interpolation
pI = generateInterp(2,Ts,dur);
plot(t,pI);
title("Ideal Bandlimited Interpolation");
xlabel("time(s)");
ylabel("pI(t)");

function p = generateInterp(type,Ts,dur)
    Ti = Ts/1000;
    t=[-dur/2:Ts/1000:dur/2-Ts/1000];
    p=zeros(1,length(t));
    if type == 0
        p(-Ts/2 <= t & t < Ts/2)=1;
    elseif type == 1
        p(-Ts <= t & t <= Ts) = 1-abs(t(-Ts <= t & t <= Ts))/Ts;
    elseif type == 2
        p=sin(pi*t/Ts)./(pi*t/Ts);
        p(t==0)=1;
    else
        disp("invalid type")
    end
end
```

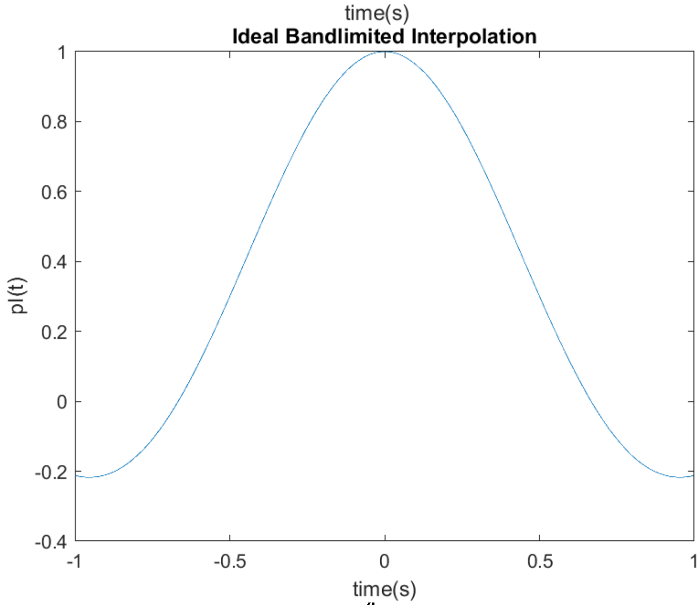
Plots:



Zero Order Hold Interpolation
pZ(t)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.5
0
0.5
1
time(s)



Linear Interpolation
pL(t)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.5
0
0.5
1
time(s)



Ideal Bandlimited Interpolation
pI(t)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-1
-0.5
0
0.5
1
time(s)

Part 4:

Matlab Code:

```
function xR=DtoA(type,Ts,dur,Xn)
    Ti=Ts/1000;
    dur =dur*Ts
    ta=[-dur/2:Ti:dur/2-Ti];
    N = length(Xn);
    p=generateInterp(type,Ts,dur);
    xR=zeros(1,round((dur+(N-1)*Ts)/Ti));
    for n = 0:N-1
      xR(1+round(n*Ts/Ti):round((dur+n*Ts)/Ti)) = xR(1+round(n*Ts/Ti):round((dur+n*Ts)/Ti))+Xn(n+1)*p;
    end
end
```

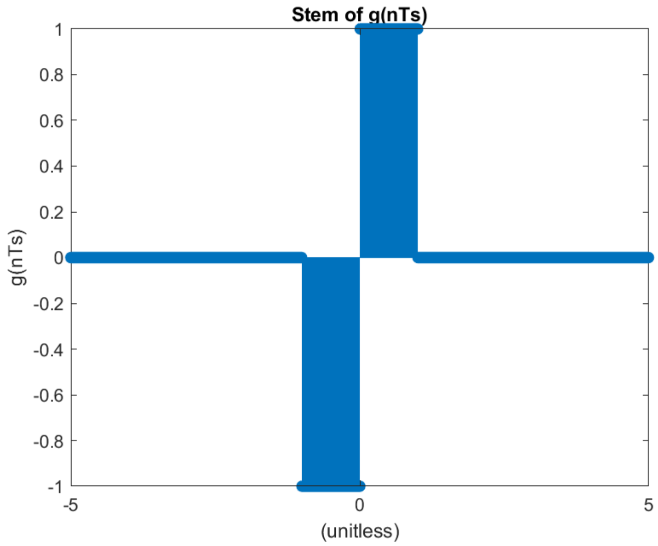
Part 5:

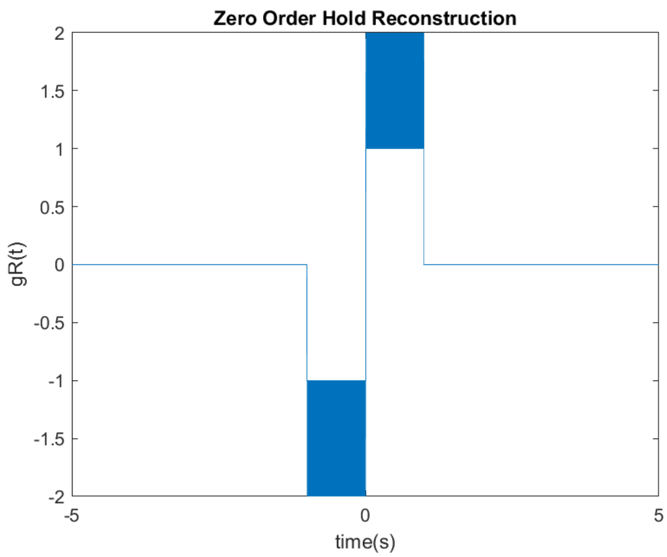
Matlab Code:

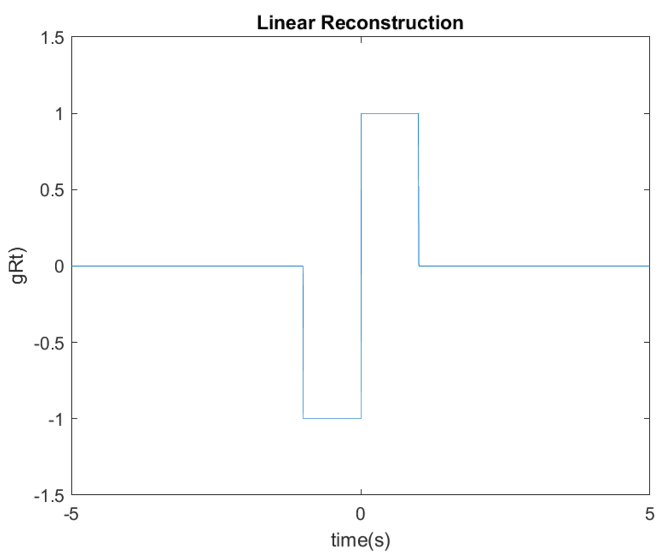
```
a = randi([2 6],1);
Ts = 1/(25*a);
dur=10;
Ti=Ts/1000;
t = [-dur/2:Ts:dur/2-Ts]; %for digital signal
ta=[-dur/2:Ts/1000:dur/2-Ts/1000];%for analog signal
g=zeros(1,length(t));
g(-1<=t & t<0) = -1;
g(0<t & t<=1) = 1;

%Sampled signal
stem(t,g);
title("Stem of g(nTs)");
figure;
%Generation of gR(t) for each interpolating method
gR1=DtoA(0,Ts,dur,g);
plot(linspace(-5,5,length(gR1)),gR1);
title("Zero Order Hold Reconstruction");
figure;
gR2=DtoA(1,Ts,dur,g);
plot(linspace(-5,5,length(gR2)),gR2);
title("Linear Reconstruction");
figure;
```

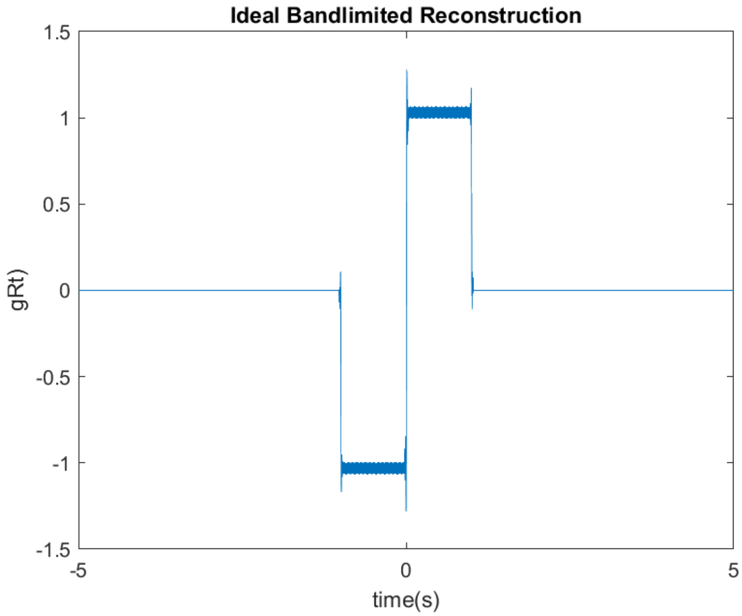
```
gR3=DtoA(2,Ts,dur,g);
plot(linspace(-5,5,length(gR3)),gR3);
title("Ideal Bandlimited Reconstruction");

function p = generateInterp(type,Ts,dur)
   Ti = Ts/1000;
   t=[-dur/2:Ts/1000:dur/2-Ts/1000];
   p=zeros(1,length(t));
   if type == 0
      p(-Ts/2 <= t & t < Ts/2)=1;
   elseif type == 1
      p(-Ts <= t & t <= Ts) = 1-abs(t(-Ts <= t & t <= Ts))/Ts;
   elseif type == 2
      p=sin(pi*t/Ts)./(pi*t/Ts);
      p(t==0)=1;
   else
      disp("invalid type")
   end
end
function xR=DtoA(type,Ts,dur,Xn)
   Ti=Ts/1000;
   dur =dur*Ts;%Interesting ask this to assistant
   ta=[-dur/2:Ti:dur/2-Ti];
   N = length(Xn);
   p=generateInterp(type,Ts,dur);
   xR=zeros(1,round((dur+(N-1)*Ts)/Ti));
   for n = 0:N-1
      xR(1+round(n*Ts/Ti):round((dur+n*Ts)/Ti)) =
xR(1+round(n*Ts/Ti):round((dur+n*Ts)/Ti))+Xn(n+1)*p;
   end
end
```

Plots:


Stem of g(nTs)
g(nTs)
(unitless)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-5
0
5


Zero Order Hold Reconstruction
gR(t)
time(s)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-5
0
5


Linear Reconstruction
gRt)
time(s)
1.5
1
0.5
0
-0.5
-1
-1.5
-5
0
5



(*) It seems the signal reconstructed by linear interpolation is the most succesfull one in the approximation of g(t). The plot looks almost the same with g(t). However, in terms of the information reconstructed, ideal bandlimited interpolation is the best one because other two filters are approximation of that filter.

(*) As I increase T_s gradually, reconstruction becomes less successful because as T_s increases the number of samples decreases so approximation becomes less accurate.

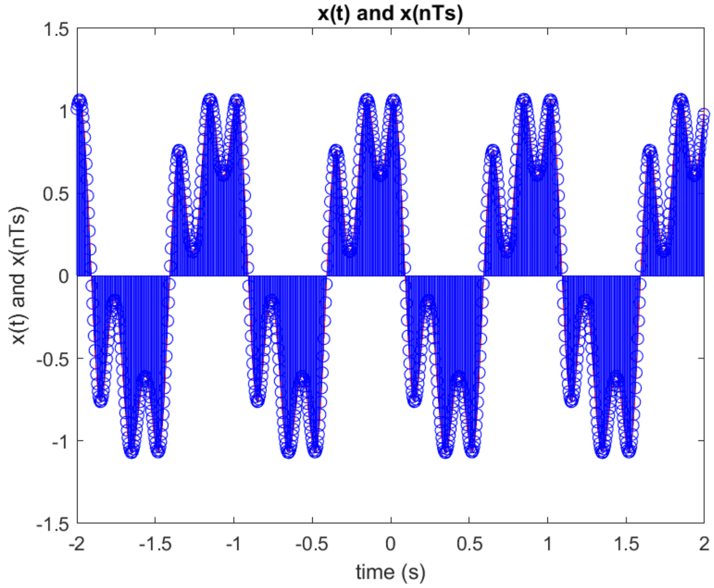
Part 6:

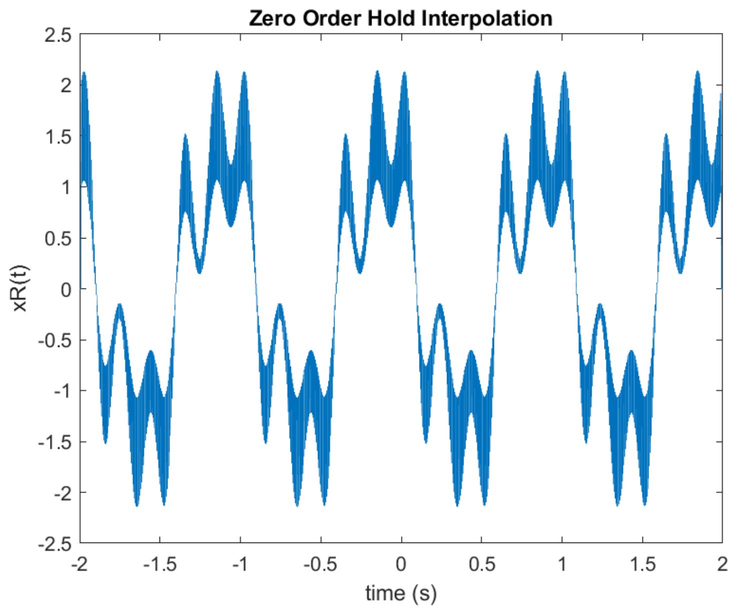
Matlab Code:

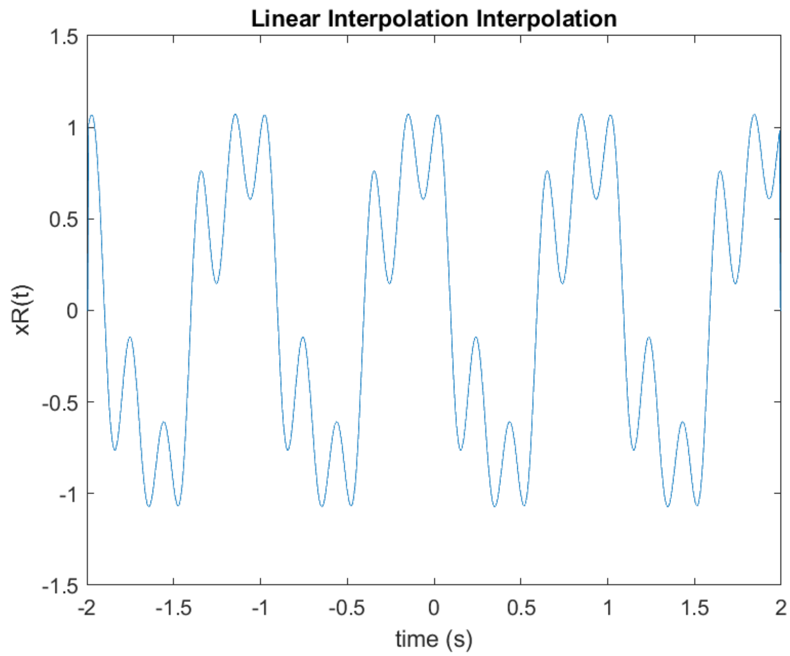
```
D5 = rem(21703190,5);
Ts = 0.005*(D5+1);
dur=4;
ta=[-dur/2:Ts/1000:dur/2-Ts/1000]; %analog (continous-time) signal
td=[-dur/2:Ts:dur/2-Ts];%digital (discrete-time) signal
xa=0.25*cos(2*pi*3*ta+pi/8)+0.4*cos(2*pi*5*ta-1.2)+0.9*cos(2*pi*ta+pi/4);
xd=0.25*cos(2*pi*3*td+pi/8)+0.4*cos(2*pi*5*td-1.2)+0.9*cos(2*pi*td+pi/4);

plot(ta,xa,'r');
title("x(t) and x(nTs)");
xlabel("time (s)");
ylabel("x(t) and x(nTs)");
hold on;
stem(td,xd,'b');
hold off;
figure;
xR1=DtoA(0,Ts,dur,xd);
plot(linspace(-dur/2,dur/2,length(xR1)),xR1);
title("Zero Order Hold Interpolation");
xlabel("time (s)");
ylabel("xR(t)");
figure;
xR2=DtoA(1,Ts,dur,xd);
plot(linspace(-dur/2,dur/2,length(xR2)),xR2);
title("Linear Interpolation Interpolation");
xlabel("time (s)");
ylabel("xR(t)");
figure;
xR3=DtoA(2,Ts,dur,xd);
plot(linspace(-dur/2,dur/2,length(xR3)),xR3);
title("Bandlimited Interpolation");
xlabel("time (s)");
ylabel("xR(t)");
```

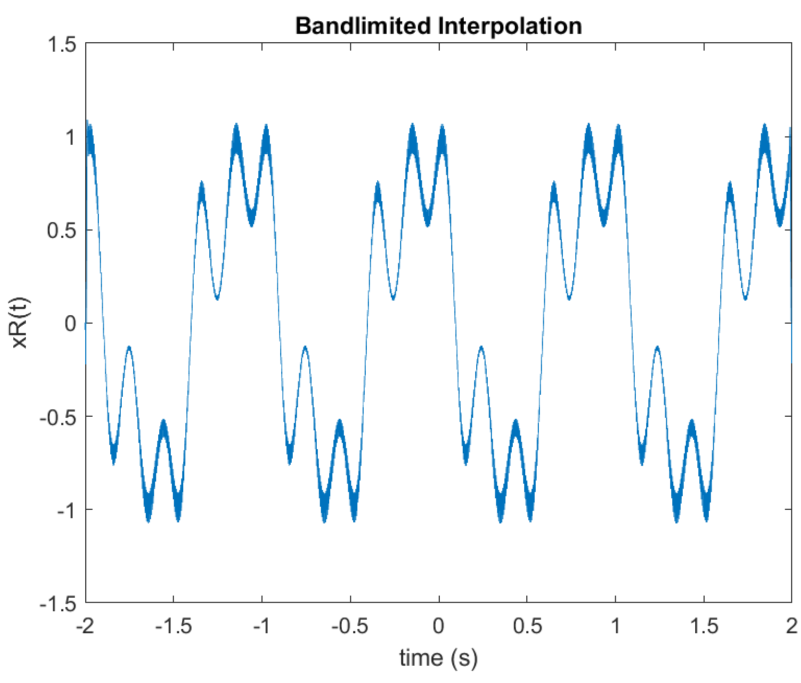
Plots:

$T_s = 0.005(D_5+1)$

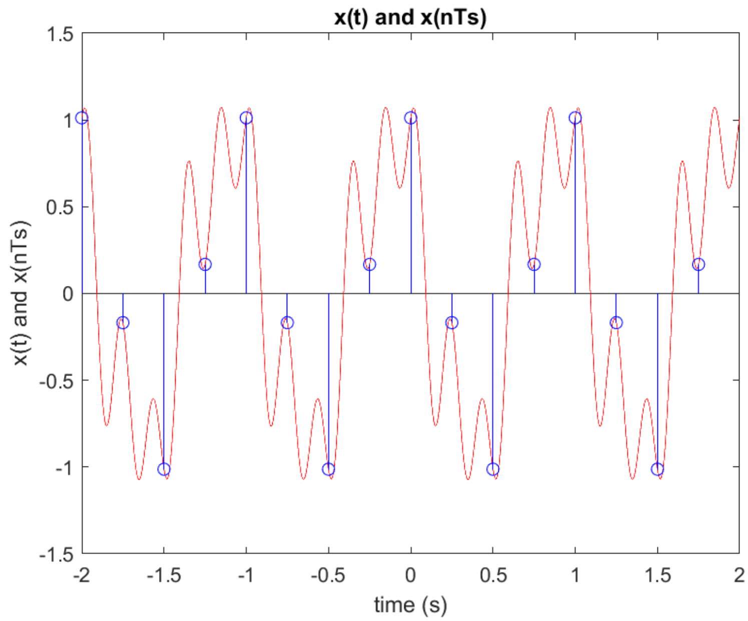




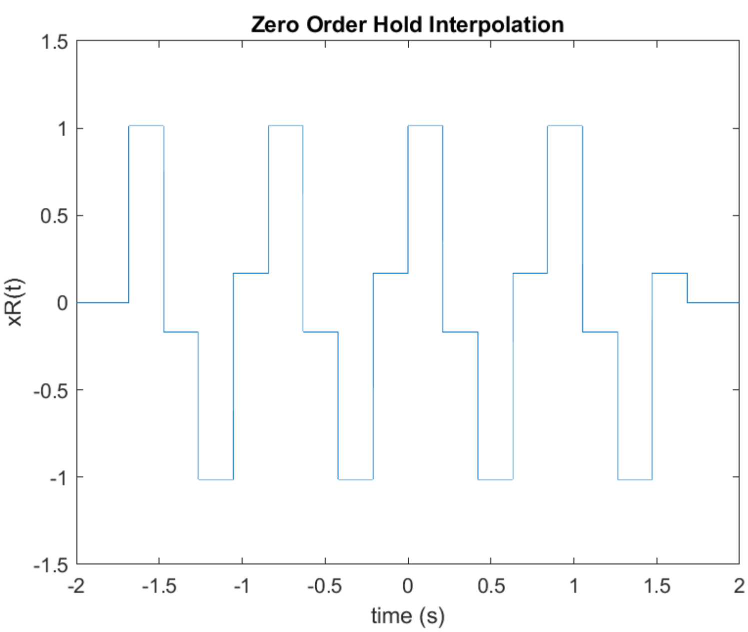

Linear Interpolation Interpolation
1.5
1
0.5
0
-0.5
-1
-1.5
xR(t)
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
time (s)


Bandlimited Interpolation
1.5
1
0.5
0
-0.5
-1
-1.5
xR(t)
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
time (s)

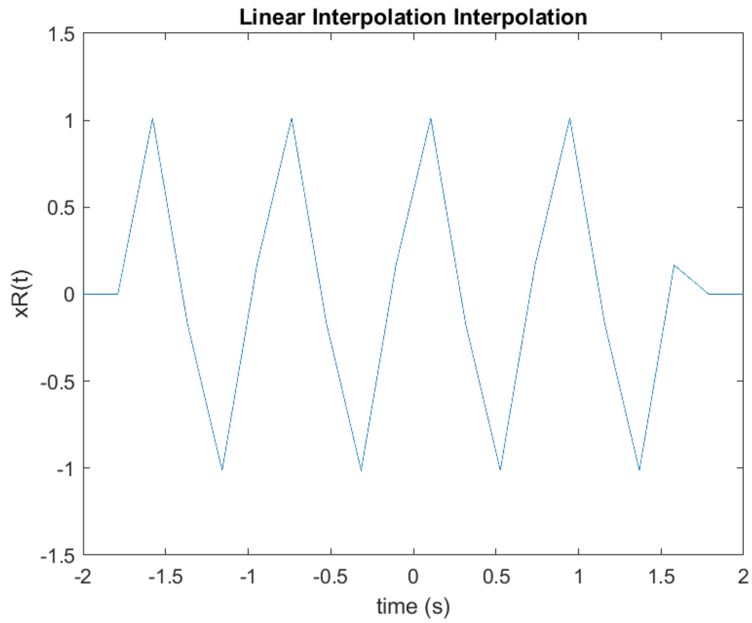
$T_s = 0.25 + 0.01 D_5$

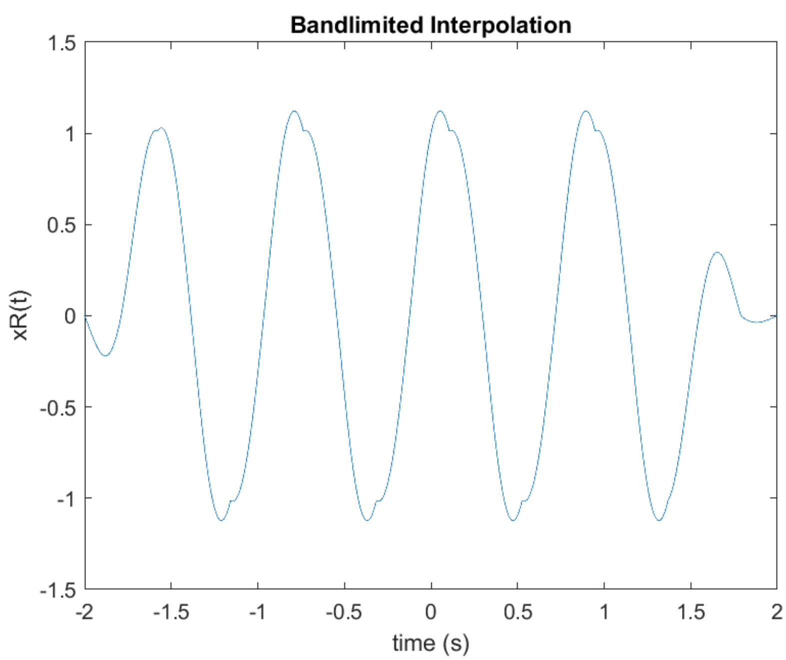


x(t) and x(nTs)
x(t) and x(nTs)
time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

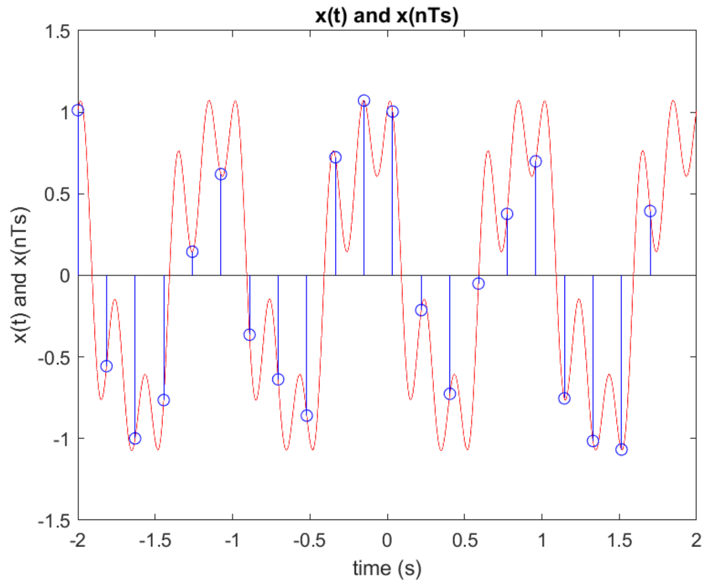


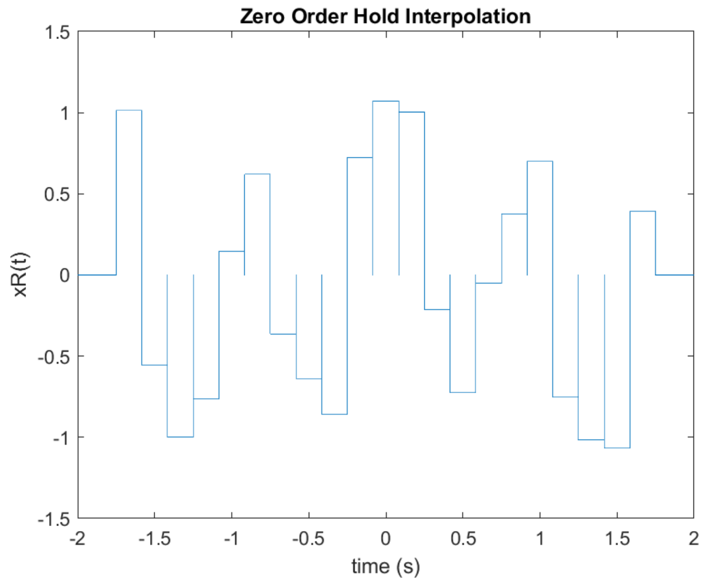
Zero Order Hold Interpolation
xR(t)
time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

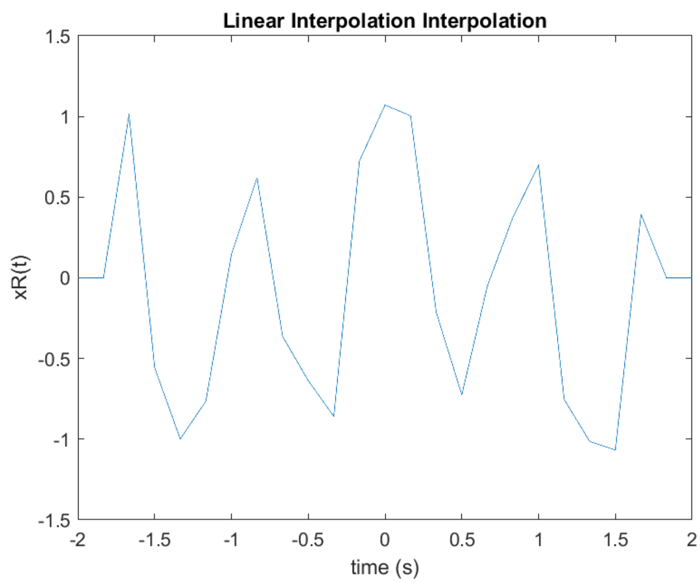

Linear Interpolation Interpolation
1.5
1
0.5
0
-0.5
-1
-1.5
xR(t)
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
time (s)

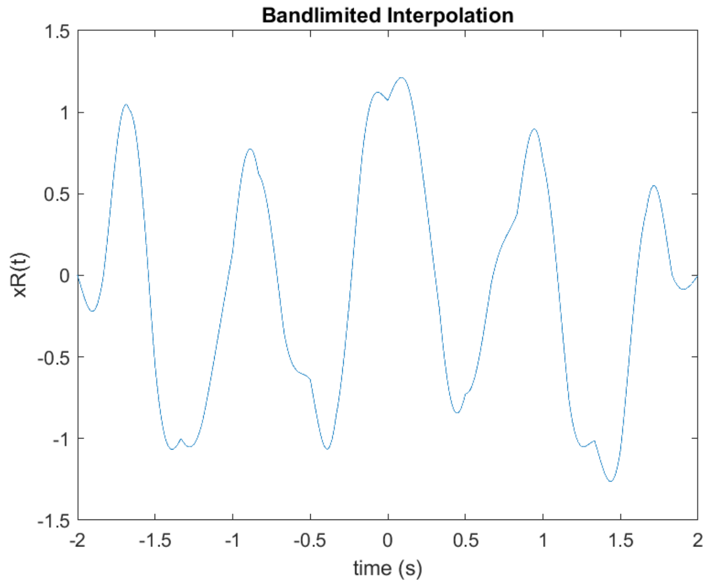

Bandlimited Interpolation
1.5
1
0.5
0
-0.5
-1
-1.5
xR(t)
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
time (s)

$Ts = 0.18+0.005(D5+1)$

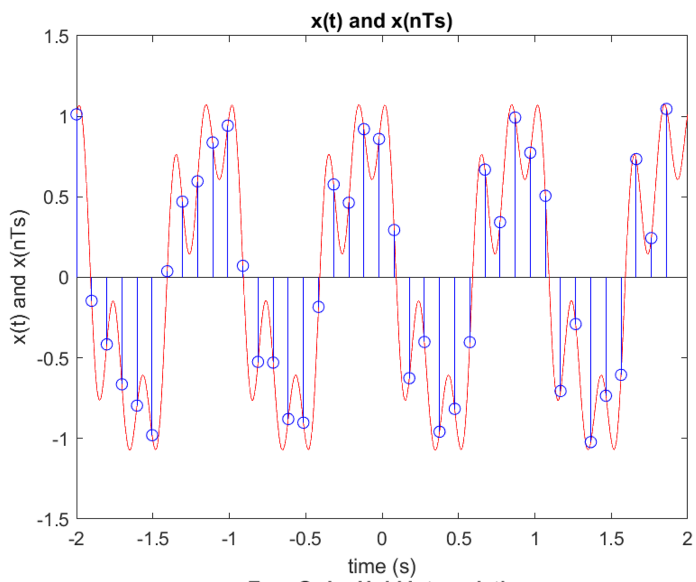


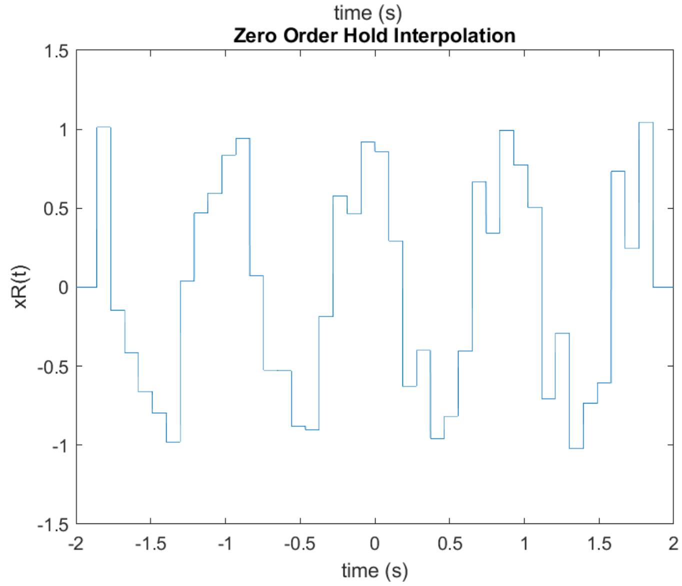


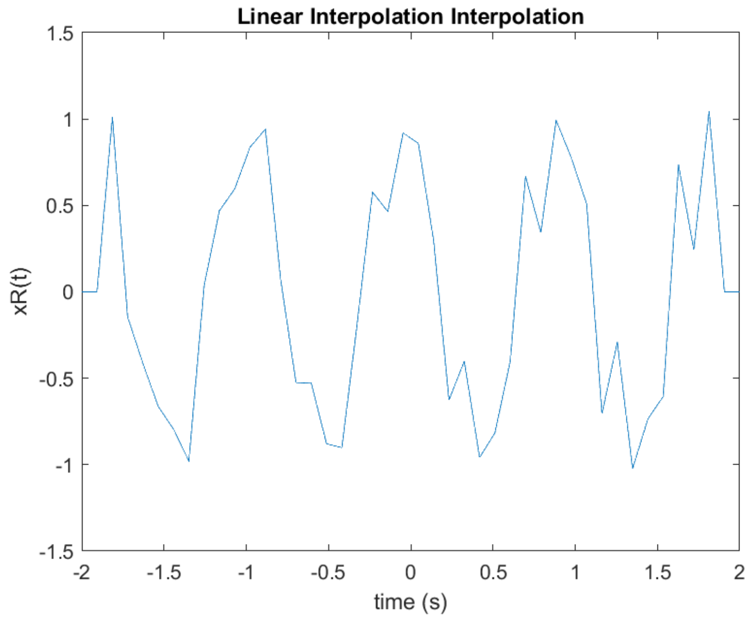


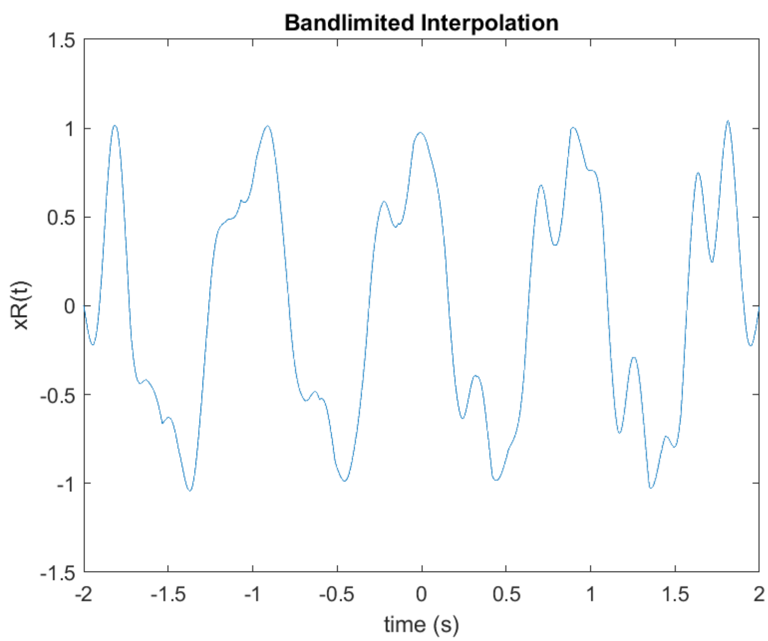

Bandlimited Interpolation
xR(t)
time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

$Ts = 0.099$


x(t) and x(nTs)
x(t) and x(nTs)
time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Zero Order Hold Interpolation
xR(t)
time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Linear Interpolation Interpolation
1.5
1
0.5
0
-0.5
-1
-1.5
xR(t)
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
time (s)


Bandlimited Interpolation
1.5
1
0.5
0
-0.5
-1
-1.5
xR(t)
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
time (s)

Part 6:

a) Linear interpolator seems the most successful one when graphs are considered, because it simply connects the sample points with straight lines. Since there are lots of sample points, graph looks smooth. However, the most successful interpolator is ideal band-limited interpolation because its Fourier transform is ideal low-pass filter. $H(j\omega) = a$ for $\omega < |\omega_c|$ and $H(j\omega) = 0$ for $\omega > |\omega_c|$, where a is constant and ω_c is cut-off frequency.

The success of interpolators:

Ideal Band-limited > Linear > Zero-Order Hold.

→ I can distinguish the original signal from its reconstruction of the ideal interpolator. The reason for that is The difference is there are some distortions in the reconstructed signal. equation (4) is impossible to exactly implement due to the infinite number of samples. Thus, we choose N sample and that finite number of samples cause distortions in the original signal.

Second Part:

As T_s gets smaller, the accuracy becomes higher. When $T_s = 0.01$, it is impossible to distinguish the difference between linear interpolation case and the original signal. Due to finite precision of MATLAB, there are some small errors, which is displaying the same values in zero order hold case. Also, since the sinc function in the time domain does not go to infinity, perfect reconstruction could not be acquired. However, if it went to infinity, I would obtain the exact reconstruction. Still, Ideal interpolator is the closest filter implemented in MATLAB to the ideal low-pass filter in theory, so it is the most successful one.

As T_s gets bigger, accuracy becomes lower. Since the maximum frequency of the signal $x(t)$ is 5Hz Nyquist rate is 10 Hz so $T_s < 0.1s$ should be satisfied for exact and successful recovery.

$\omega_s > 2\omega_m \Rightarrow T_s < T_m/2$

Hence, for $0.25 \geq T_s > 0.1s$, good approximation of the original signal becomes impossible since we lost serious information of the original signal. Even if we were able to let the convolutional sum in equation (4) go to infinity, exact reconstruction would not be possible in that range.