EEE 321-Signals and Systems

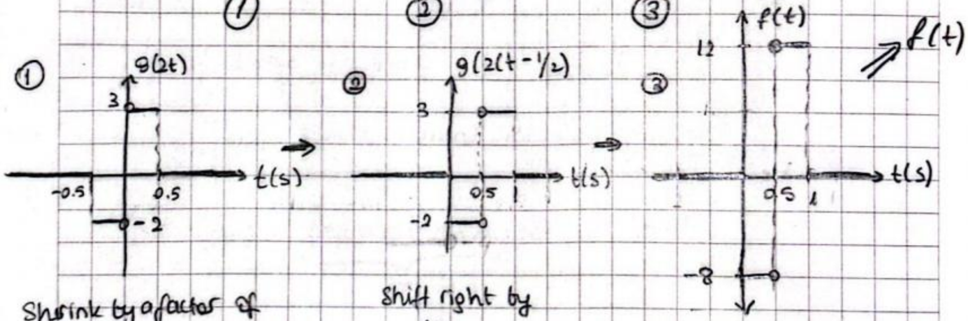
-Lab 05-

Defne Yaz Kılıç
Section 001
22102167
11.24.23

Part 1

$f(t) = 4g(2t-1) = 4g(2(t-\frac{1}{2})):$

$g(t) \rightarrow \underset{①}{g(2t)} \rightarrow \underset{②}{g(2(t-\frac{1}{2}))} \rightarrow \underset{③}{4g(2(t-\frac{1}{2})}$



Shrink by a factor of 2.

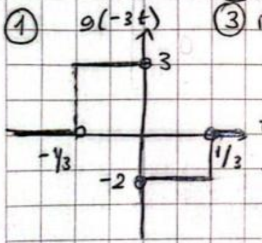
Shift right by 1/2

Multiply by 4

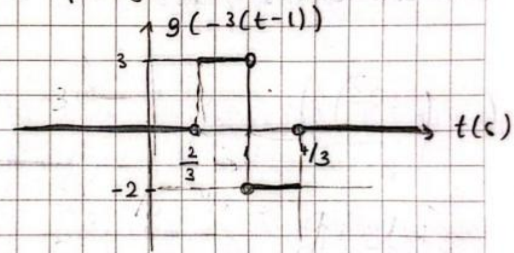
$$f(t) \begin{cases} -8 & 0 \leq t < 0.5 \\ 12 & 0.5 < t \leq 1 \\ 0 & \text{else} \end{cases}$$

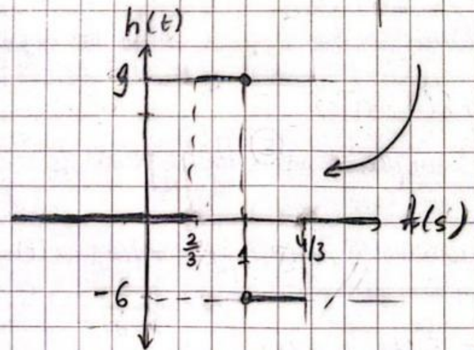
$3g(-3(t-1)) \Rightarrow$ ① Shrink by a factor of 3 and reverse $g(t)$
② Shift right by 1.
③ multiply the corresponding values with 3.

① $g(-3t)$



②





$$h(t) \begin{cases} 9 & \frac{2}{3} \leq t < 1 \\ -6 & 1 < t \leq 4/3 \\ 0 & \text{else} \end{cases}$$

According to Shannon- Nyquist sampling theorem, for a full recovery of the signal, the sampling rate should be at least twice of the maximum frequency of the original signal. Otherwise some information of the signal will be lost, aliasing will happen. Since the signal g(t) contains discontinuities, the corresponding frequencies are infinitely high hence the signal g(t) is not band-limited meaning that there is no cut-off frequency. For full recovery, there should be a cut-off frequency for choosing the sampling rate. G(t) is therefore cannot be fully recovered. If a LPF filter is applied the high frequency content can be eliminated and then maybe the obtained signal can be recovered.

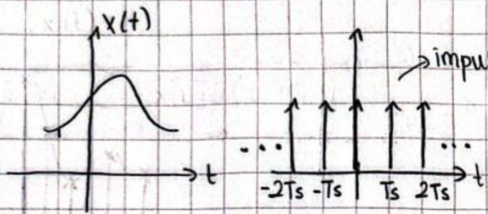
Part 2

Derivation of Part 2 is as follows:

$$\bar{x}(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t-nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\,\delta(t-nT_s)$$

$x(nT_s) = \bar{x}[n]$: discrete version of the signal

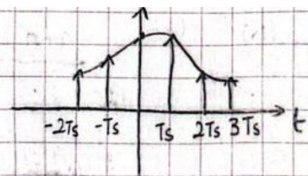
$\bar{x}(t) \longrightarrow \bar{x}[n] \longrightarrow x_R(t)$



impulse train $\sum_{n=-\infty}^{\infty} \delta(t-nT_s)$

if we multiply x(t) and the impulse train we can obtain another impulse train scaled by the values of x(t) for corresponding nTs values.

$$\bar{x}(t) = x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_s) =$$



this is equal to the multiplication of discrete coefficients of x with the impulses:

$$\bar{x}(t) = \sum_{n=-\infty}^{\infty} \underbrace{x(nT_s)}_{\bar{x}[n]} \delta(t - nT_s)$$

$\bar{x}[n]$: sampled x with sampling period T_s.

we can obtain the reconstructed signal by convolving $\bar{x}(t)$ with an interpolating pulse.

$$x_R(t) = \bar{x}(t) * p(t)$$

$$x_R(t) = \sum_{n=-\infty}^{\infty} \bar{x}[n] \, \delta(t - nT_s) * p(t) \qquad (1)$$

$\bar{x}[n]$ → coefficients

the convolution : $\delta(t-nT_s) * p(t)$ shifts $p(t)$ by (nT_s)

$$\int_{-\infty}^{\infty} p(\tau)\, \delta((t-nT_s)-\tau)\, d\tau = p(t-nT_s)$$

Hence, we can write (1) as

$$x_R(t) = \sum_{n'=-\infty}^{\infty} \bar{x}[n']\, p(t-n'T_s) \quad //$$

for $p(0) = 1$ & $p(\underbrace{k}_{k \in \mathbb{Z}-\{0\}} T_s) = 0$

$$x_R(nT_s) = \sum_{n'=-\infty}^{\infty} \bar{x}[n']\, p(nT_s - n'T_s)$$

$$= \sum_{n'=-\infty}^{\infty} \bar{x}[n']\, p(\underbrace{(n-n')T_s})$$

if $n-n' = 0$, $p(0) = 1$

if $\underbrace{n-n'}_{\text{integer since both } n, n' \in \mathbb{Z}} \neq 0$, $p(kT_s) = 0$

Hence the sum is non zero only when $n = n'$,

$$x_R(nT_s) = \underbrace{0+0+0\cdots}_{n' \neq n} + \underbrace{\bar{x}[n']}_{n=n'} + \underbrace{0+0\cdots}_{n' \neq n}$$

$$\boxed{x_R(nT_s) = \bar{x}[n]}$$ consistent interpolation

$p_Z(t) = \text{rect}\left(\frac{t}{T_s}\right)$ zero order hold interpolation

$$\text{rect}(t) \begin{cases} 1 & \text{if } -\frac{1}{2} \le t < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

$p_L(t) = \text{tri}\left(\frac{t}{T_s}\right)$ linear interpolation

$$\text{tri}(t) \begin{cases} 1 - \frac{|t|}{0.5} & \text{if } -0.5 \le t \le 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$p_I(t) : \text{sinc}\left(\frac{t}{T_s}\right)$ Ideal bandlimited interpolation

$$\text{sinc}(t) \begin{cases} 1 & \text{if } t = 0 \\ \frac{\sin(\pi t)}{\pi t} & \text{otherwise} \end{cases}$$

a) $p_Z(0) = 1$, $p_L(0) = 1$, $p_I(0) = 1$

b) $p_Z(kT_s) = 0$, $p_L(kT_s) = 0$, $p_I(kT_s) = \dfrac{\sin(\pi k T_s)}{\pi k T_s} = 0$

c) Since they suit the definition of $p(t)$ for a consistent interpolation, yes they are consistent.

$p(0)=1$ & $p(kT_s) = 0$ for all p_Z, p_L, p_I

Part 3

The function for generating the interpolating pulses is as follows:

```
function p = generateInterp(type,Ts,dur)
new_Ts=Ts/500;
t= -dur/2:new_Ts:dur/2-new_Ts;
p=zeros(size(t));
if type ==0
for i= 1:length(t)
if -Ts/(2)<= t(i) && t(i)<Ts/(2)
p(i)=1;
end
end
end
if type ==1
for i= 1:length(t)
if -Ts/2 <= t(i) && t(i)<Ts/2
p(i)= 1- abs(2*t(i))/Ts;
end
end
end
if type ==2
for i= 1:length(t)
if t(i) == 1
p(i)= 1;
else
p(i)=sin(pi*t(i)/Ts)/(pi*t(i)/Ts);
end
end
end
end
```

Using the function above the following functions are obtained.

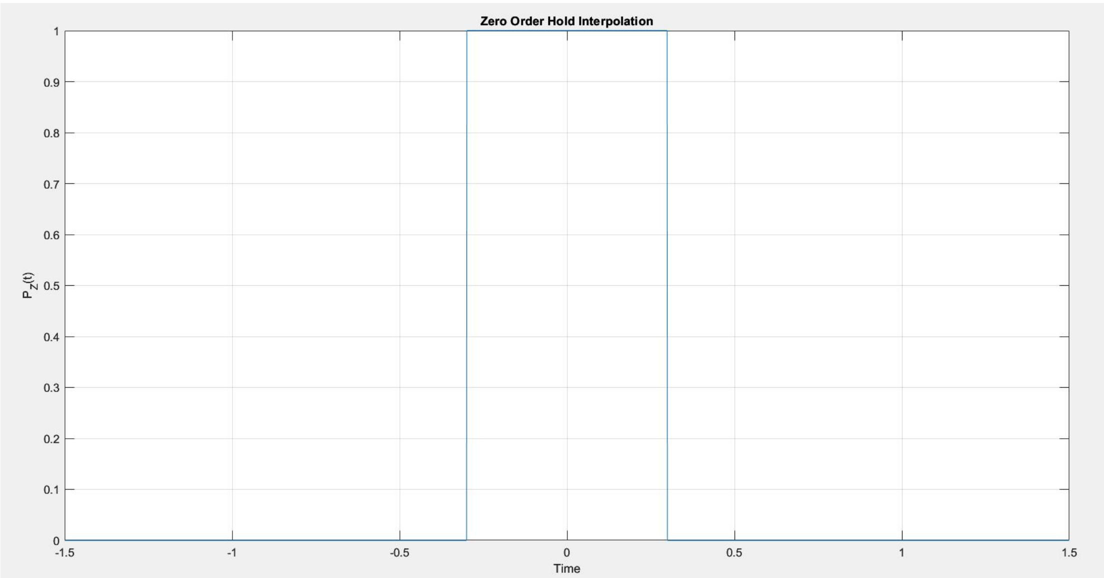


Fig.1 Zero Order Hold Interpolation

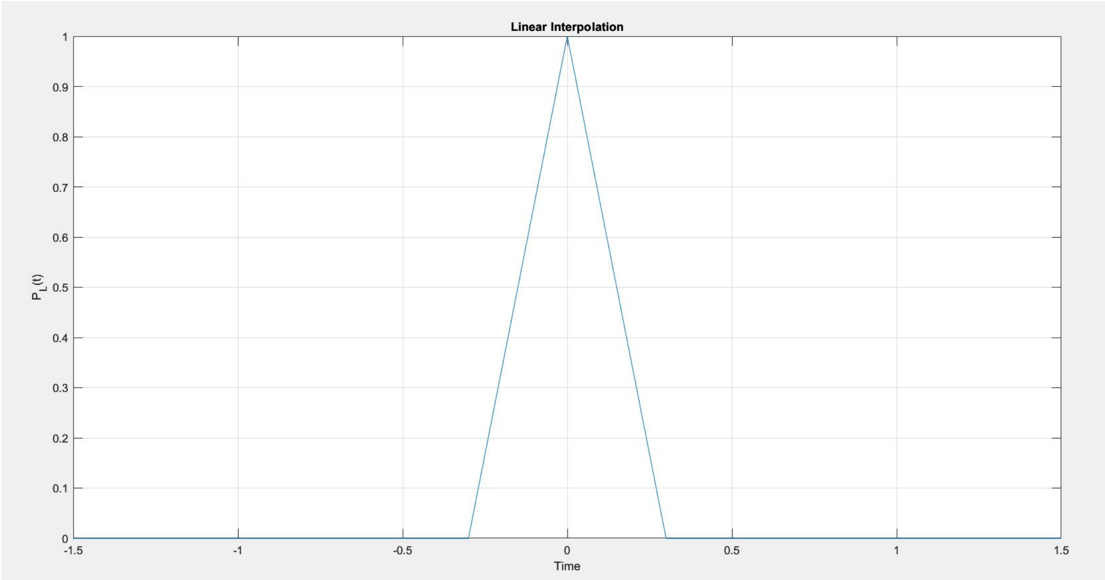


Fig.2 Linear Interpolation

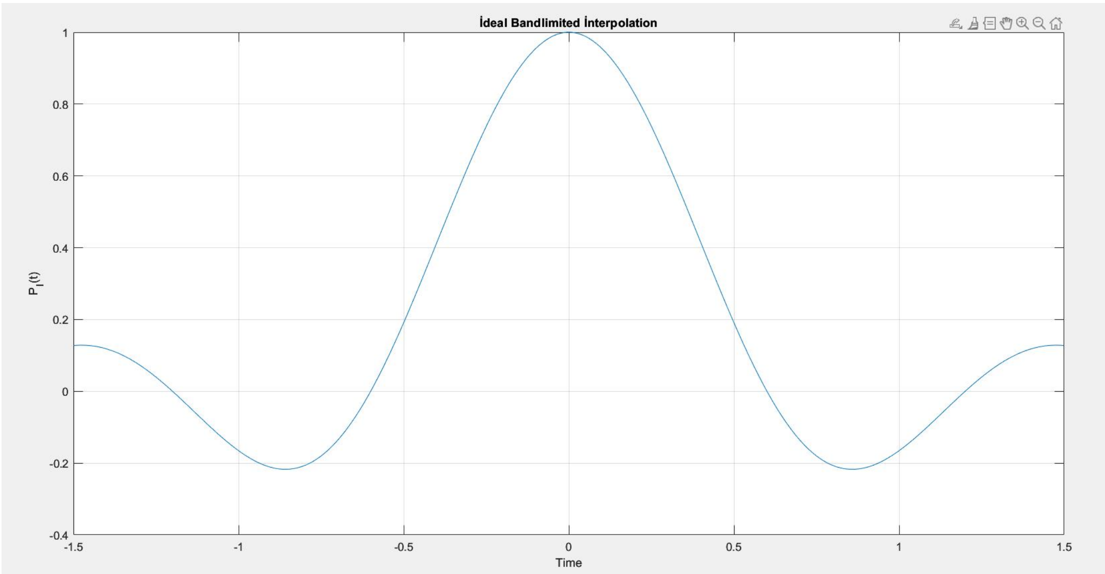


Fig.3 Ideal Bandlimited Interpolation

Part 4

The function for interpolation is as follows:

```
function xR=DtoA(type,Ts,dur,Xn)
dur=dur*Ts;
Tf=Ts/500;
n=length(Xn);
pulse = generateInterp(type,Ts,dur);
xR=zeros(1,round((dur+(n-1)*Ts)/Tf));
for i=0:n-1
xR(1+round(i*Ts/Tf):round((dur+i*Ts)/Tf))=xR(1+round(i*Ts/Tf):round((dur+i*Ts)/Tf))+Xn(i+1)*pulse;
end
end
```

Part 5

The code for generating reconstructed signals of g(t) for each interpolating pulse type is as follows:

```
a=randi([2 6],1);
Ts=1/(20*a);
Tf=Ts/500;
dur=6;% -3<t<3
time = -dur/2:Ts:dur/2-Ts;
g= zeros(1,length(time));
%construction of G(t)
g(-1<=time & time<0)=-2;
g(0<time & time<=1)=3;

figure(4),
stem(time,g);
title("Stem plot"),
gR0=DtoA(0,Ts,dur,g);
t0= linspace(-3, 3, length(gR0));
figure (5),
plot(t0,gR0)
title("Reconstructed - Zero Hold Interpolation"),
xlabel ({'Time'},'Interpreter','latex');
ylabel ({'$G_r0$'},'Interpreter','latex');
grid on;
gR1=DtoA(1,Ts,dur,g);
t1= linspace(-3, 3, length(gR1));
figure (6), plot(t1,gR1), title("Reconstructed - Linear Interpolation"),
xlabel ({'Time'},'Interpreter','latex');
ylabel ({'$G_r1$'},'Interpreter','latex');
grid on;
gR2=DtoA(2,Ts,dur,g);
t2= linspace(-3, 3, length(gR2));
figure (7), plot(t2,gR2),title("Reconstructed - Ideal Bandlimited
Interpolation"),
xlabel ({'Time'},'Interpreter','latex');
ylabel ({'$G_r2$'},'Interpreter','latex');
grid on;
```

Here are the obtained plots:

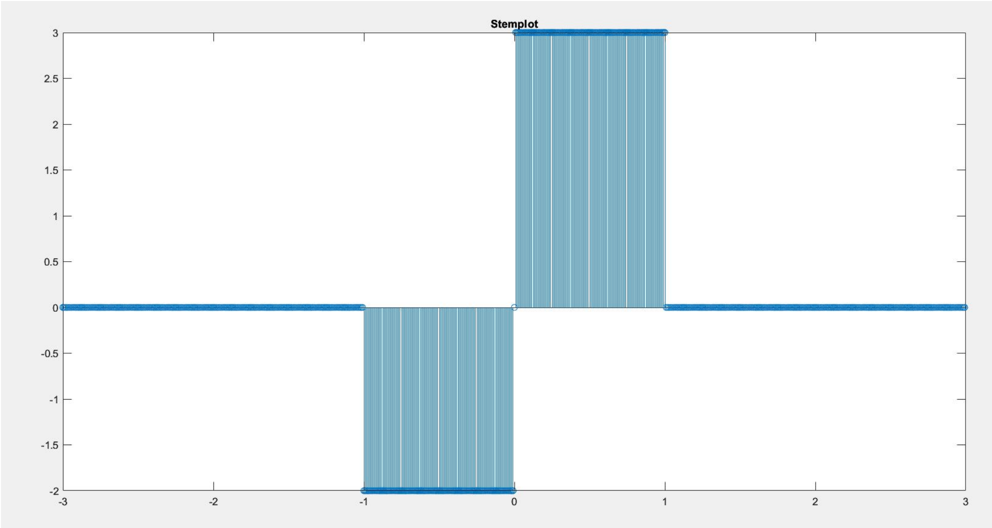


Fig.4 Stem Plot

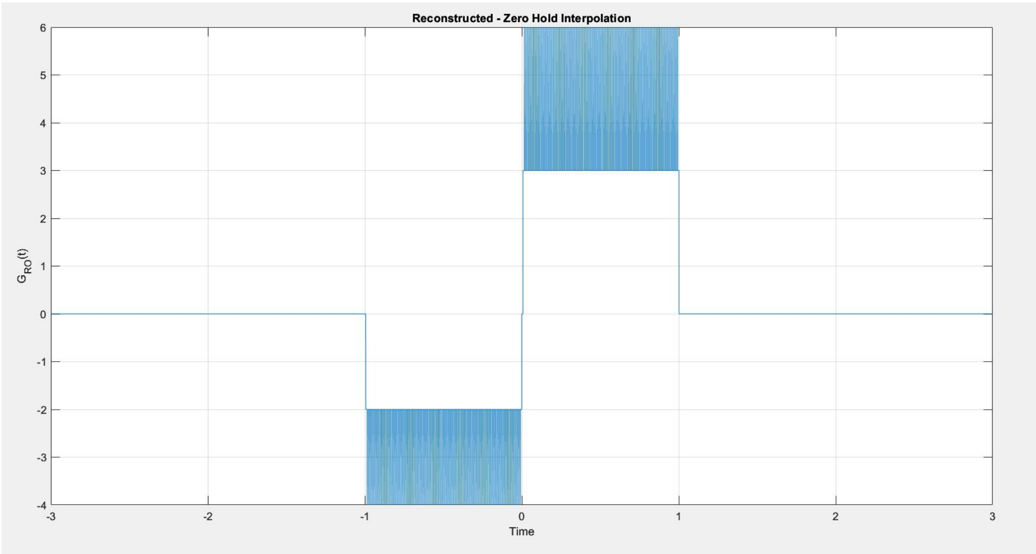


Fig.5 Reconstruction with Zero Order Hold Interpolation

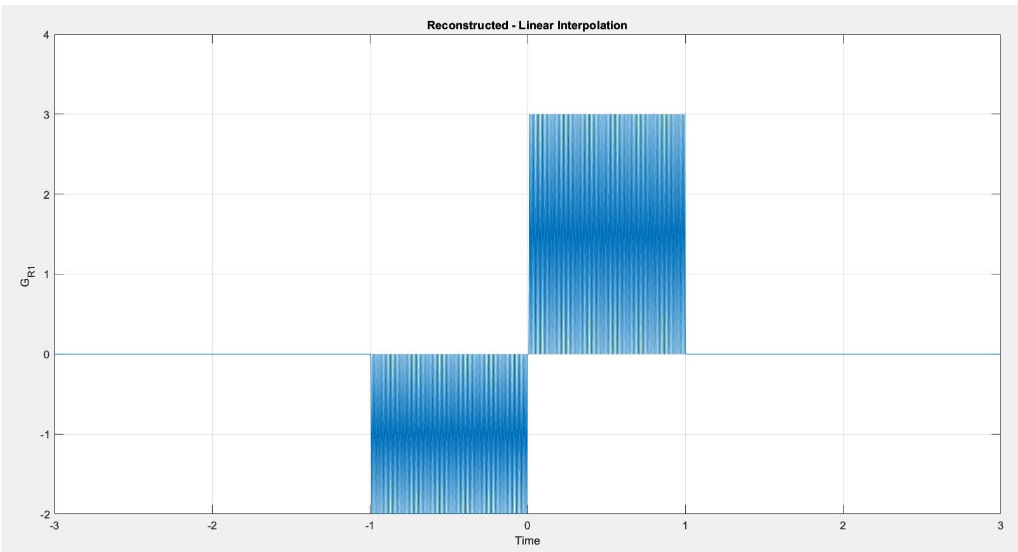


Fig.6 Reconstruction with Linear Interpolation

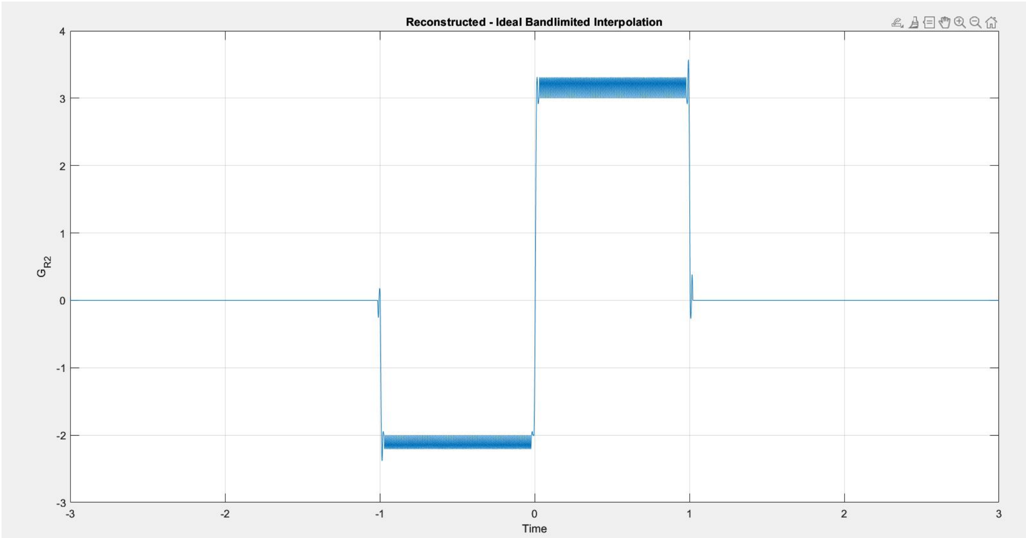


Fig.7 Reconstruction with Ideal Bandlimited Interpolation

When the obtained plots are compared, although due to high frequency content around the discontinuities there is Gibbs phenomenon; it can be seen that the Band limited Interpolation is more accurate then the other pulse types. Looking at the shape of the plots, reconstruction with linear interpolation gives the best shape however there are unnecessary lines filling the shape. When these lines are made bigger it is seen that they are small triangles going up and down. For the Zero Hold Interpolation since g(t) has very high frequency content around discontinuities due to the overlapping during the convolution the reconstructed signal is higher than the original one. The values are doubled.

As Ts s sampling rate gets smaller hence less information is gathered from the original signal this causes less accurate recovered signals. Aliasing occurs. As Ts decreases more samples are being taken therefore the plots become more accurate however even for very large sampling rates due to the discontinuities during the convolution, overlapping happens which causes additional filled areas on top of the squares.

Part 6

For different sampling rates the following plots are obtained:

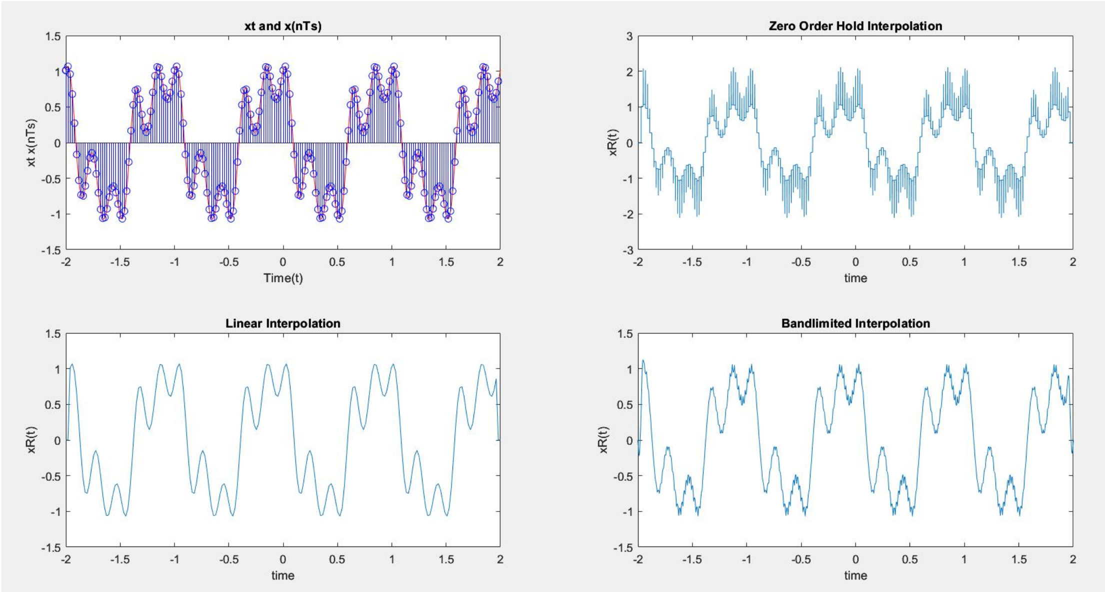


Fig.7 Reconstruction for `Ts=0.005*(D7+1)`

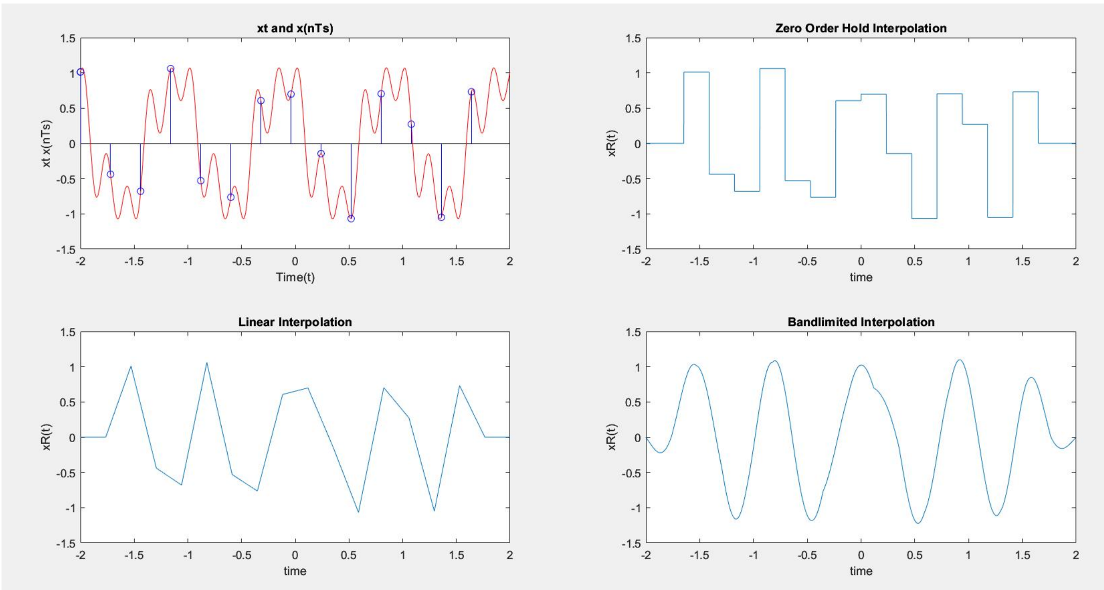


Fig.8 Reconstruction for `Ts=0.25+ D7*0.01`

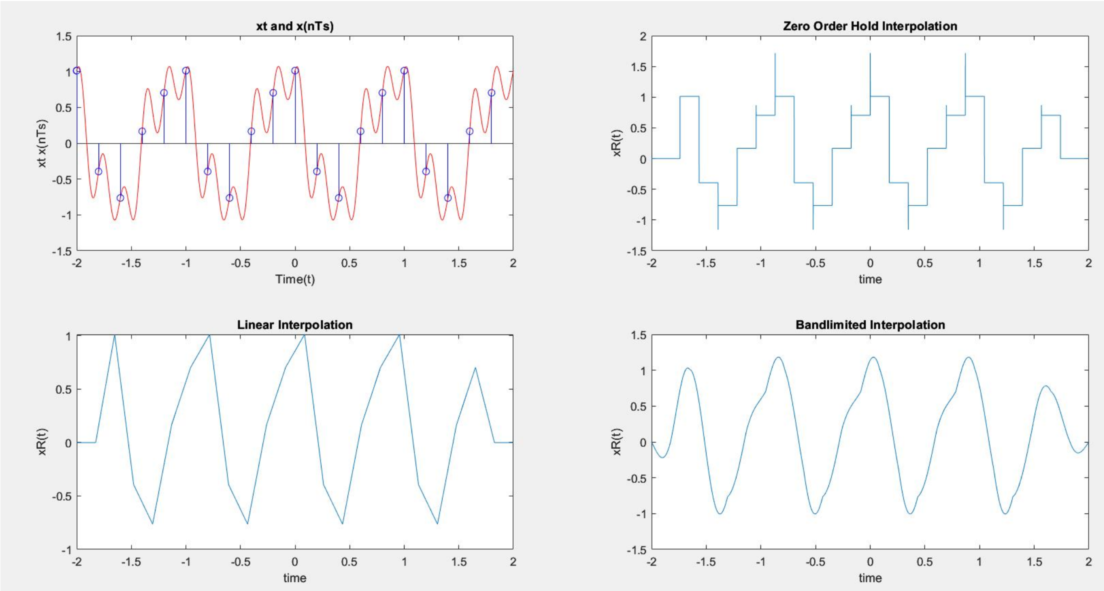


Fig.9 Reconstruction for `Ts=0.25+ D7*0.01`

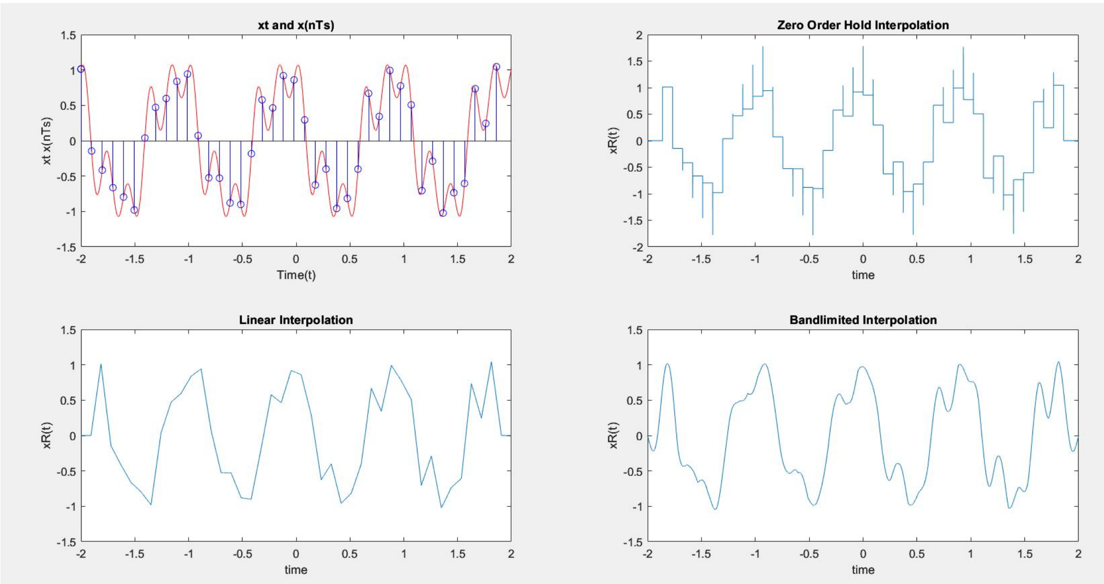


Fig.10 Reconstruction for `Ts=0.099`

Once again Ideal Bandlimited Interpolation gives the best results for this part since sinc function is and ideal low pass filter in the fourier domain. Since N samples are taken rather than infinite amount the reconstructed signal is a little distorted. For interpolation processes choosing the sampling rate has great importance, it should be optimized for not loosing data. The maximum frequency of x(t) is 5Hz therefore due to shannon Nyquist criteria the sampling rate should be at least twice of this value that is 10 Hz. Since $f=1/T_s$, when $0.01 < T_s<0.1$, this criteria is satisfied do the reconstructed signal is very similar to the original x(t). As explained in the previous part, as T_s increases the sampling rate decreases therefore for $0.1< T_s<0.2$ the reconstructed signals are very distorted and don't resemble to the original signal.

Appendix

Part 3:

```
dur = mod(22102167,7);% dur = 3
Ts= dur/5;
Tf=Ts/500;
time= -dur/2:Tf:dur/2-Tf;
p= generateInterp(0,Ts,dur);
figure (1)
plot(time,p)
title("Zero Order Hold Interpolation")
xlabel ({'Time'},'Interpreter','latex');
ylabel ({'$P_z(t)$'},'Interpreter','latex');
grid on,
a= generateInterp(1,Ts,dur);
figure (2)
plot(time,a)
title("Linear Interpolation")
xlabel ({'Time'},'Interpreter','latex');
ylabel ({'$P_L(t)$'},'Interpreter','latex');
grid on,
b= generateInterp(2,Ts,dur);
figure (3)
plot(time,b)
title("İdeal Bandlimited İnterpolation")
xlabel ({'Time'},'Interpreter','latex');
ylabel ({'$P_I(t)$'},'Interpreter','latex');
grid on,
```

Part 6:

```
D7=rem(22102167,7); % =3
Ts=0.099;
Tf= Ts/500;
dur = 4;
t= -dur/2:Tf:dur/2-Tf;
sdt=-dur/2:Ts:dur/2-Ts;
sampledxt=0.25*cos(2*pi*3*sdt+pi/8)+0.4*cos(2*pi*5*sdt-
1.2)+0.9*cos(2*pi*sdt+pi/4);
xt=0.25*cos(2*pi*3*t+pi/8)+0.4*cos(2*pi*5*t-1.2)+0.9*cos(2*pi*t+pi/4);
figure
subplot(2,2,1);
plot(t,xt,'r'),title('xt and x(nTs)'),xlabel('Time(t)'),ylabel('xt
x(nTs)'),hold on
stem(sdt,sampledxt,'b')
hold off
%
%a)
xRe0= DtoA(0,Ts,dur,sampledxt);
tr0=linspace(-2,2,length(xRe0));
subplot(2,2,2), plot(tr0,xRe0),title('Zero Order Hold
Interpolation'),xlabel('time'),ylabel('xR(t)')
xRe1= DtoA(1,Ts,dur,sampledxt);
tr1=linspace(-2,2,length(xRe1));
subplot(2,2,3) , plot(tr1,xRe1),title('Linear
Interpolation'),xlabel('time'),ylabel('xR(t)')
xRe2= DtoA(2,Ts,dur,sampledxt);
```

```
tr2=linspace(-2,2,length(xRe2));
subplot(2,2,4), plot(tr2,xRe2),title('Bandlimited
Interpolation'),xlabel('time'),ylabel('xR(t)')
```