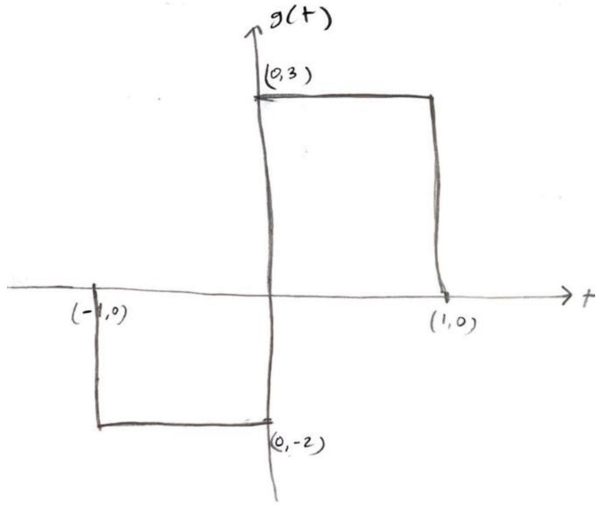
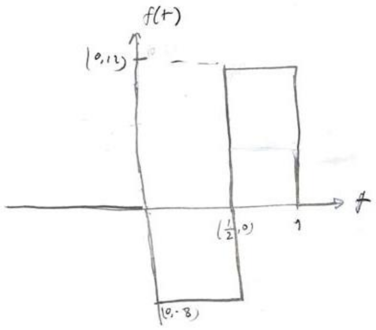
Eray Gündoğdu 22101962

Signals and Systems Lab 5

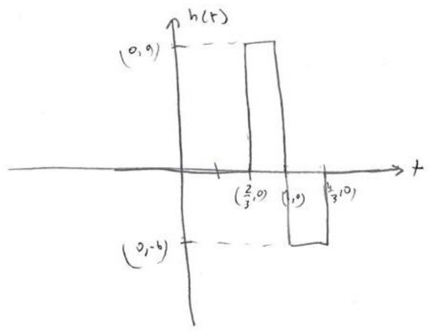
Part 1



$$4g(2t-1) = f(t) = \begin{cases} -8 & 0 \le t < \frac{1}{2} \\ 12 & \frac{1}{2} < t \le 1 \\ 0 & \text{otherwise} \end{cases}$$



$$3g(-3t+3) = h(t) = \begin{cases} -6 & 1 < t \le \frac{4}{3} \\ 9, & \frac{2}{3} \le t < 1 \\ 0 & \text{otherwise} \end{cases}$$



T_s should be smaller then the maximum change in frequency. Refer ΔW (change in frequency) as B

$T_s < \frac{2\pi}{w_c}$.

Fourier Transform of the signal is:

$$G(jw) = \int_{-1}^{0} (-2).e^{-jwt}.dt + \int_{0}^{1} 3.e^{-jwt}$$

$$G(jw) = \frac{2}{jw} e^{-jwt}\Big]_{-1}^{0} - \frac{3}{jw} e^{-jwt}\Big]_{0}^{1}$$

$$G(jw) = \frac{5 - 2e^{jw} - 3e^{-jw}}{jw}$$

Part 2

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

Convolution with $\delta(t-nT_s)$ samples $p(t)$ hence $x_R(t)$ can be written as $x_R(t) = \sum_{n=-\infty}^{\infty} x(nT_s).p(t-nT_s)$

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

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

$p((n-n')T_s) = 0$ for all $n \neq n'$, $p((n-n')T_s) = 1$ for all $n = n'$. Hence, for all $n = n'$ from the eq. above

$x_R(nT_s) = \bar{x}[n]$.

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

(b) $P_Z(kT_s) = rect(k) = 0$

$P_L(kT_s) = tri(k) = 0$

$P_I(kT_s) = sinc(k) = \frac{\sin(\pi k)}{\pi k} = 0$

(c) All interpolation functions equal to 1 when $t=0$ and 0 for $t = kT_s$ where $k \neq 0$ therefore $x_R(nT_s) = \bar{x}[n]$ is obtained for all interpolations.

Part 3

The MATLAB code for the function that generates interpoles is accordingly:

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

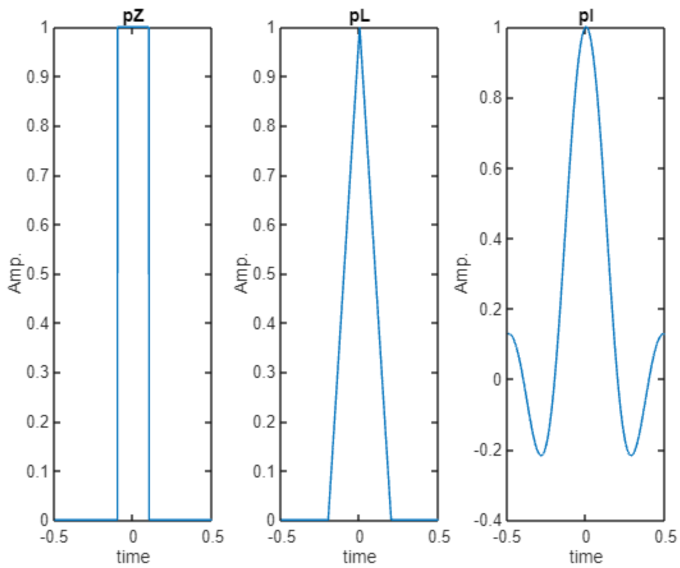


Fig1.1 interpolating functions

Part 4

The MATLAB code for the function that generates a reconstructed signal by using sampled signal is written as:

```
function xR=DtoA(type,Ts,dur,Xn)
    I_pole = generateInterp(type,Ts,dur);
    size_reconstructed = length(Xn)*500+length(I_pole);
    xR = zeros(1,  size_reconstructed);
    for n = 0:length(Xn)-1
        xR(500*n+1:500*n+length(I_pole)) =  xR(500*n+1:500*n+length(I_pole)) +
Xn(n+1)*I_pole;
    end
    xR = xR(250*length(Xn)+1: end-250*length(Xn));
end
```

Part 5

In this part, it is desired to compare interpolation methods in terms of efficiency hence $g(t)$ is sampled and reconstructed by using three different interpolation methods.

```
dur=6;
a = randi([2,6]);
Ts = 1/(20*a);%Ts
t = [-3 : Ts: 3-Ts];%t
g=zeros(1,length(t));
g(t>=-1/Ts & t<0)=-2;
g(t>0 & t<=1/Ts)= 3;
g(t>1)=0;
g(t<-1)=0;
g(t==0)=0;
stem(t,g);title('g(nTs)');xlabel('t');ylabel('Amp.');
gr0 = DtoA(0,Ts,6,g);
plot(linspace(-3,3,length(gr0)), gr0); title('Zero Order Hold
Interpolation');xlabel('t');ylabel('Amp.');
gr1 = DtoA(1,Ts,6,g);
plot(linspace(-3,3,length(gr1)), gr1); title('Linear
Interpolation');xlabel('t');ylabel('Amp.');
gr2 = DtoA(2,Ts,6,g);
plot(linspace(-3,3,length(gr2)), gr2); title('Ideal Bandlimited
Interpolation');xlabel('t');ylabel('Amp.');
```

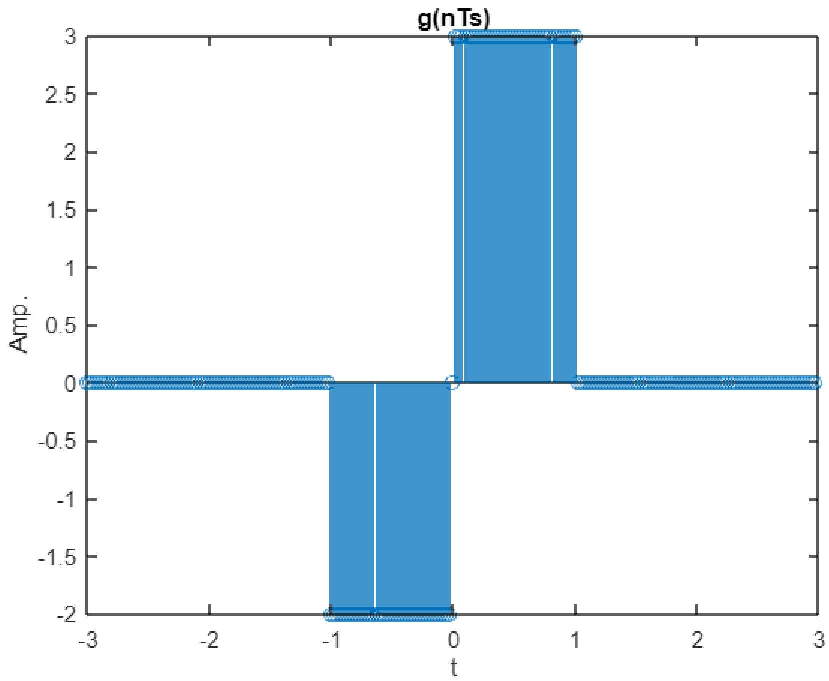


Fig2.1 $g(nT_s)$

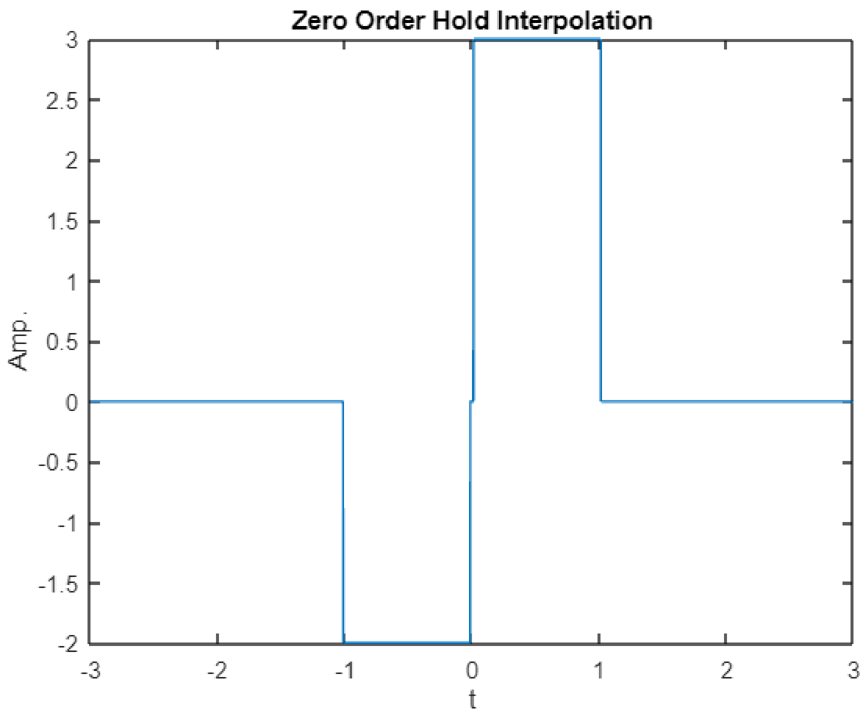


Fig2.2 Zero Order Hold Interpolation of $g(t)$

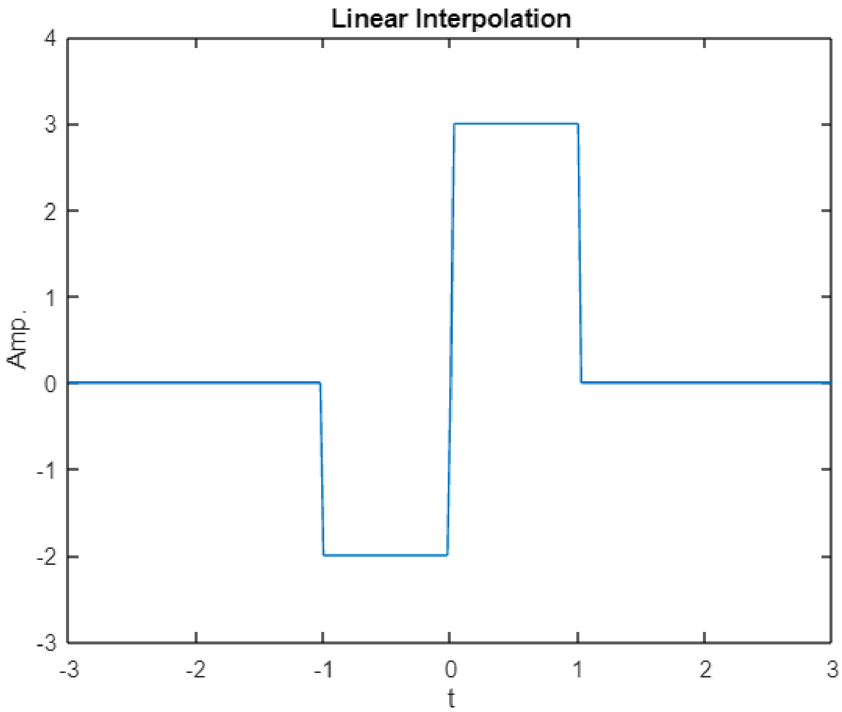


Fig2.3 Linear Interpolation of $g(t)$

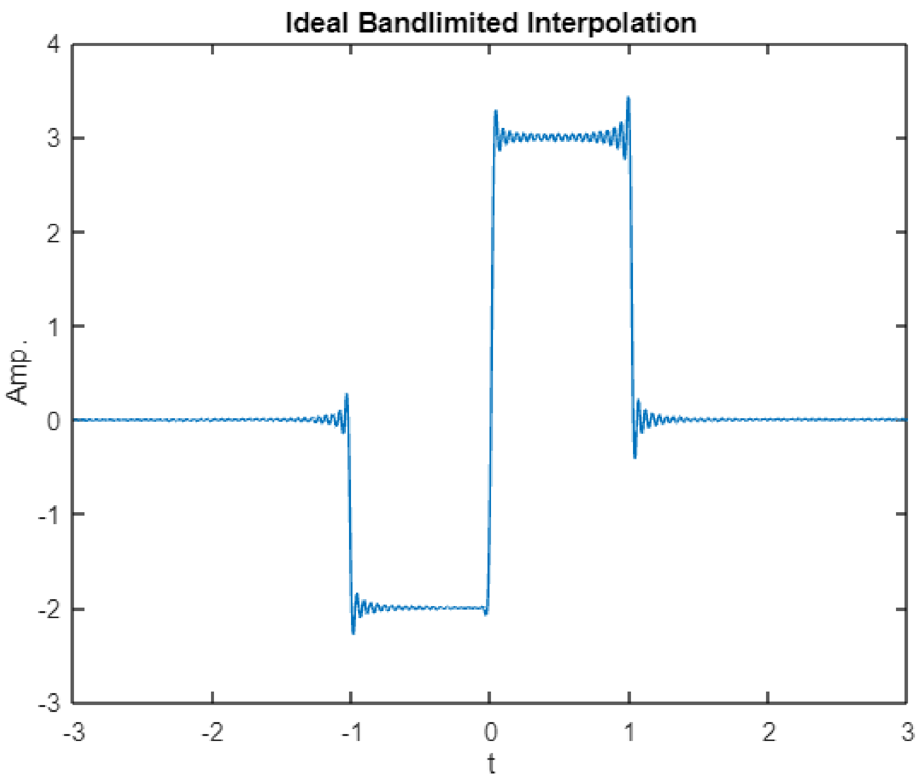


Fig2.4 Ideal Bandlimited Interpolation of $g(t)$

After observing the three methods for reconstructing the original signal, linear interpolation suits the best for reconstructing $g(t)$. It is desired to investigate reconstructed signals by increasing T_s gradually. As T_s increases the amount of sampling of the original signal decreases meaning that if T_s increases to the point where it is larger than $\frac{2\pi}{\omega_c}$ hence, the signal can not be reconstructed with hundred percent accuracy. As a conclusion, it can be said that as T_s increases the reconstruction accuracy decreases.

Part 6

Code for sampling the given function is accordingly:

```
D=22191962;
D7=rem(D,7);
Ts=0.15;
t=[-2:Ts:2-Ts];
x=0.25*cos(2*pi*3*t+pi/8)+0.4*cos(2*pi*5*t-1.2)+0.2*cos(2*pi*t+pi/4);
%plot(t,x,'b');
%title('x(t) vs x(nTs)');
%xlabel('t');
%ylabel('Amp');
%%hold on;
%stem(t,x,'r');
%hold off;
x0=DtoA(0,Ts,4,x);
x1=DtoA(1,Ts,4,x);
x2=DtoA(2,Ts,4,x);
figure;
subplot(3,1,1);
plot(linspace(-2,2,length(x0)),x0);
title('Zero Order Hold Interpolation');xlabel('t');ylabel('Amp.');
subplot(3,1,2);
plot(linspace(-2,2,length(x1)),x1);
 title('Linear Interpolation');xlabel('t');ylabel('Amp.');
 subplot(3,1,3);
plot(linspace(-2,2,length(x2)),x2);
title('Ideal Bandlimited Interpolation');xlabel('t');ylabel('Amp.');
```

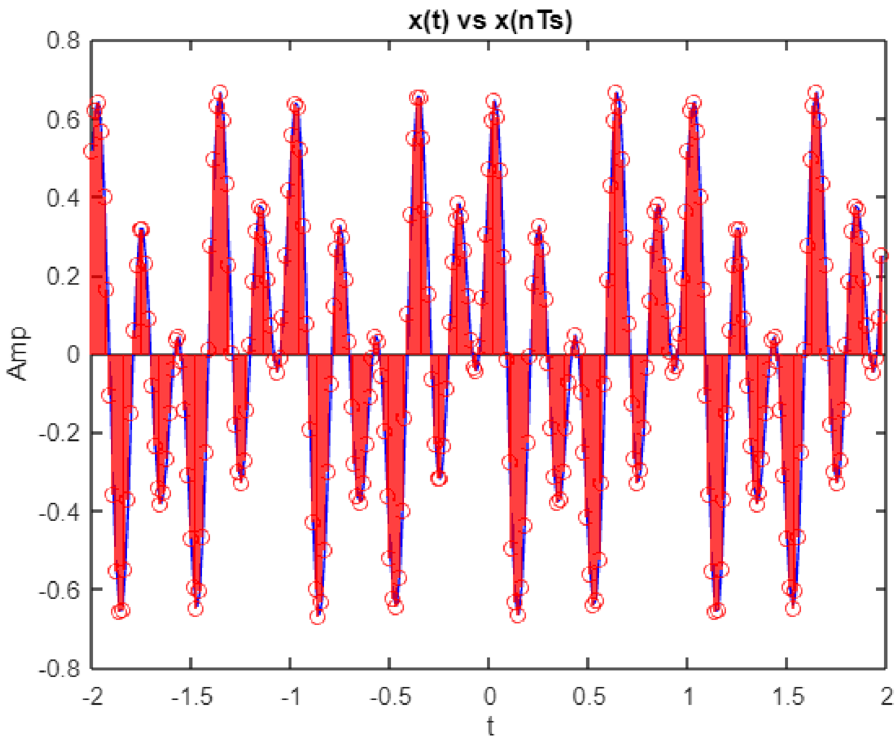


Fig3.1 $x(t)\ vs\ x(nT_s)$

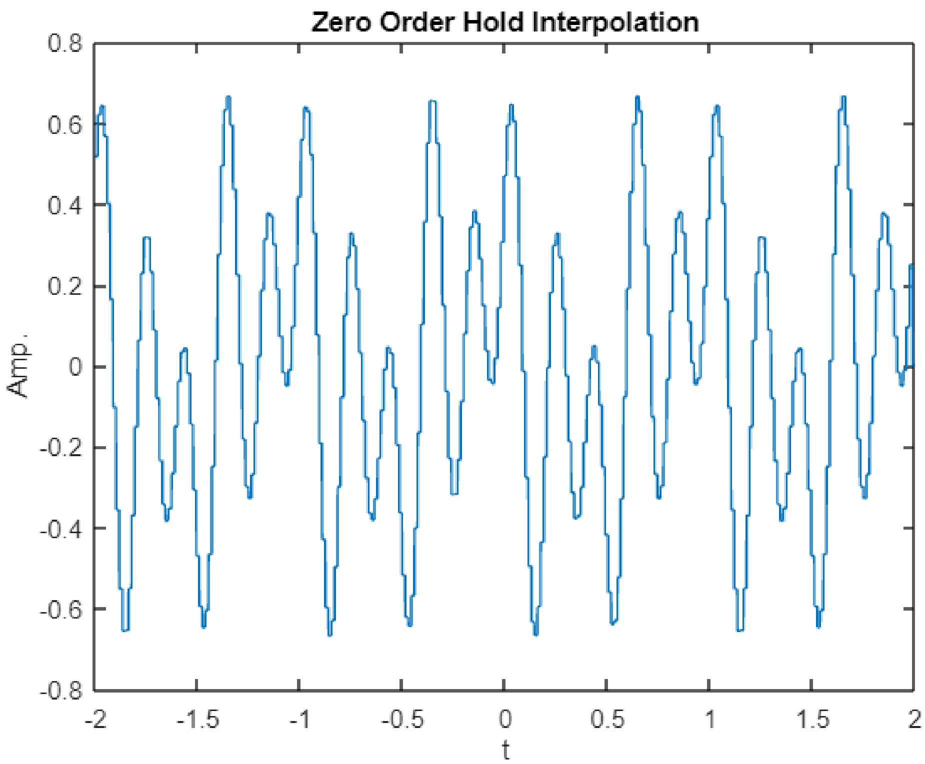


Fig3.2 Zero Order Hold Interpolation of $x(t)$, $T_s = 0.005(D_7 + 1)$

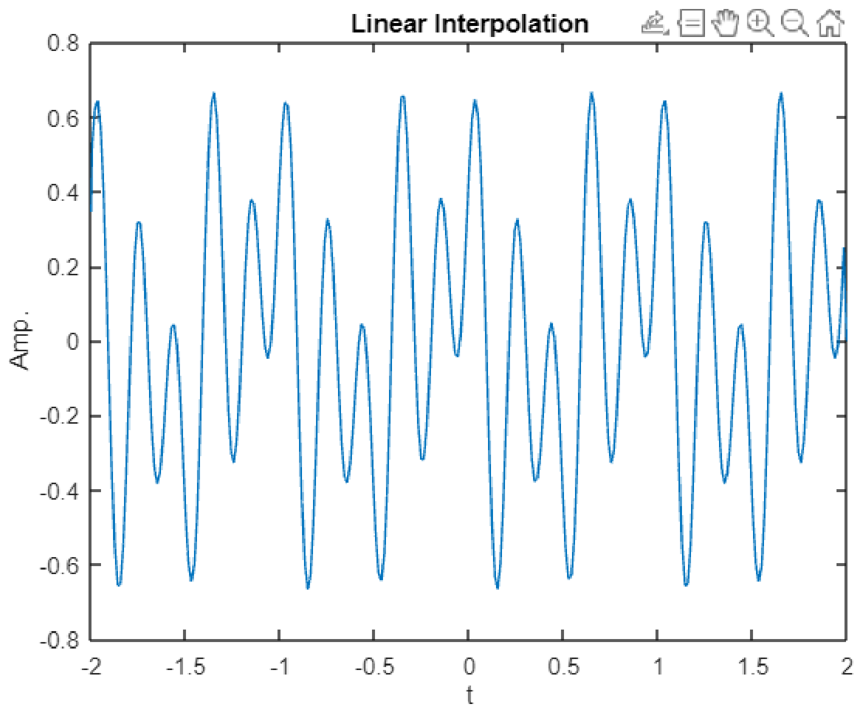


Fig3.3 Linear Interpolation of $x(t)$, $T_s = 0.005(D_7 + 1)$

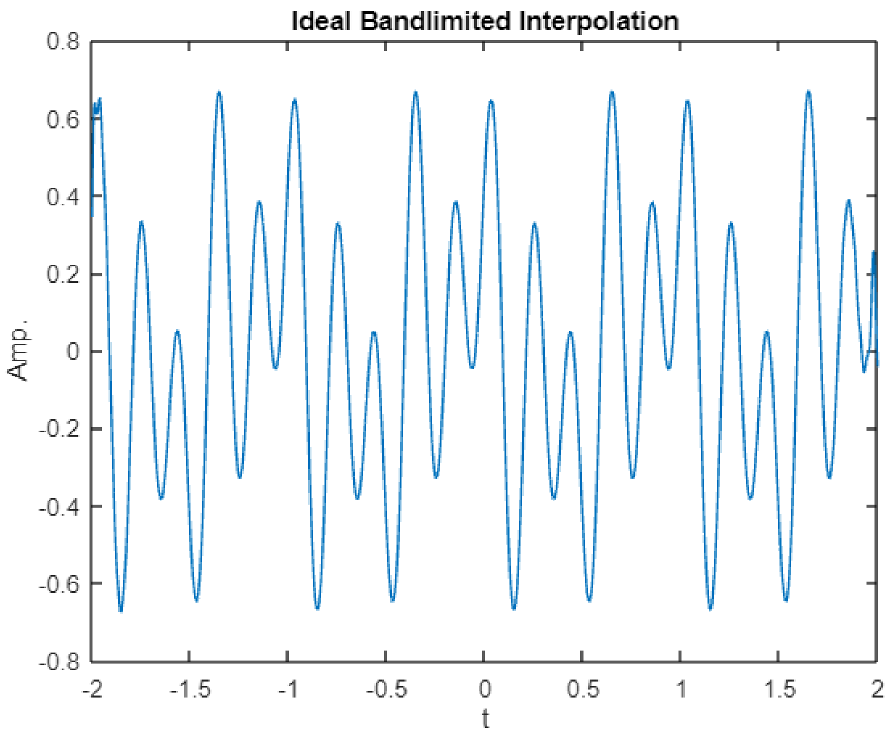


Fig3.4 Ideal Bandlimited Interpolation of $x(t)$, $T_s = 0.005(D_7 + 1)$

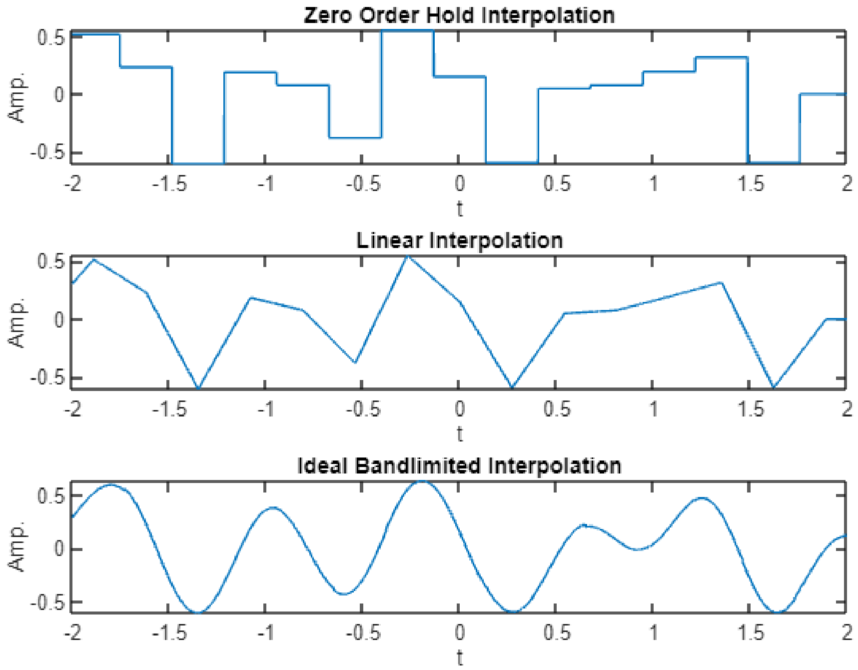


Fig3.5 $T_s = 0.25 + 0.01 D_7$

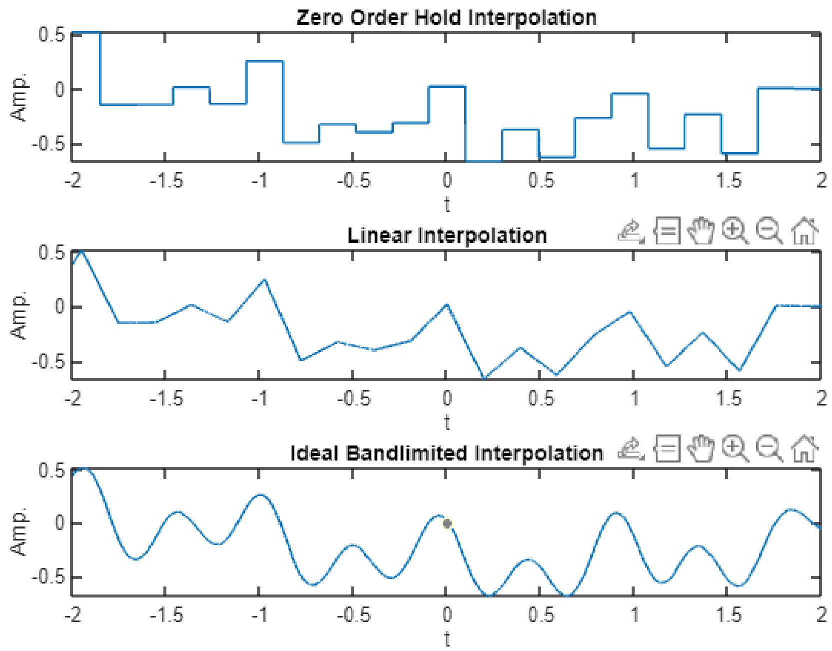


Fig3.6 $T_s = 0.18 + 0.005(D_7 + 1)$

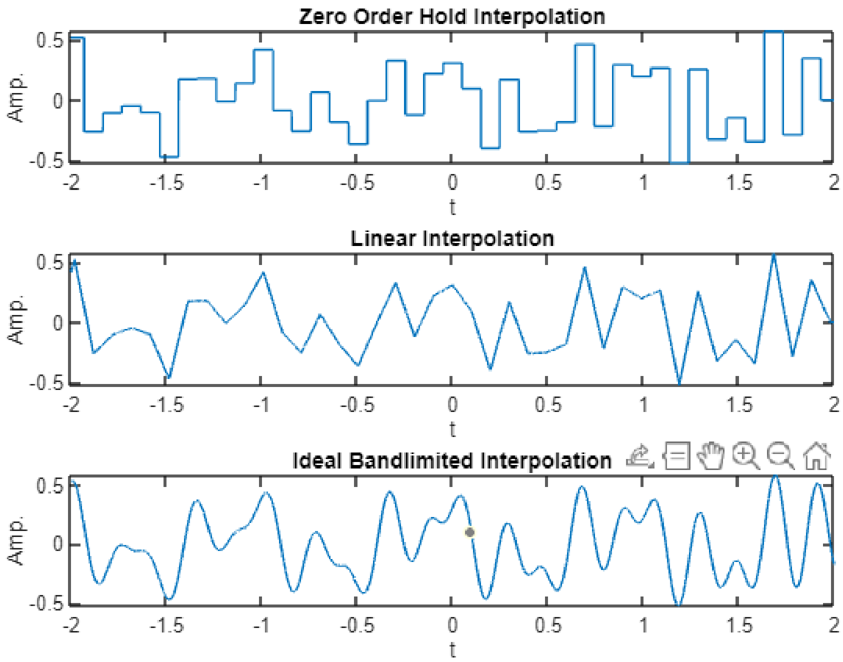


Fig3.7 T_s =0.099

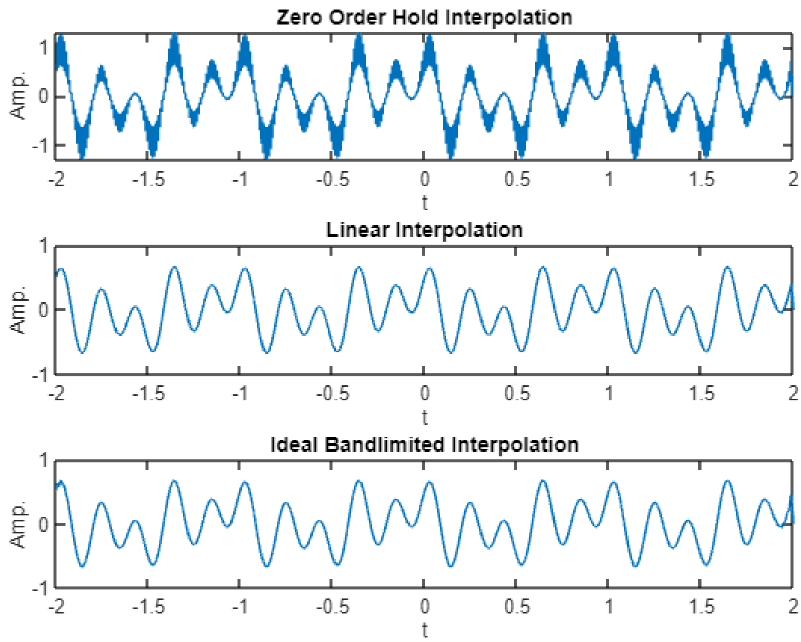


Fig3.8 T_s =0.01

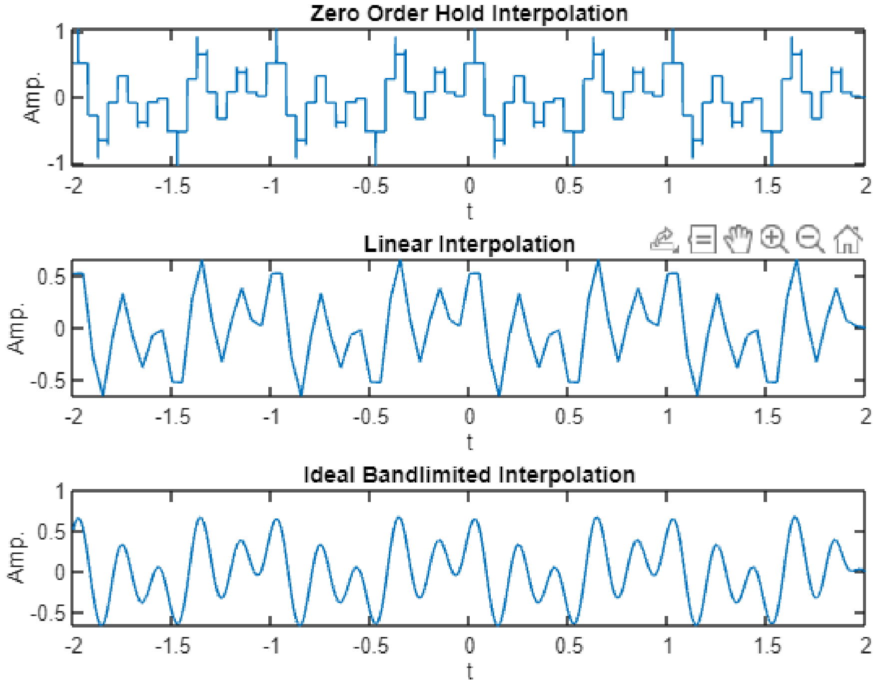


Fig3.9 T_s =0.015

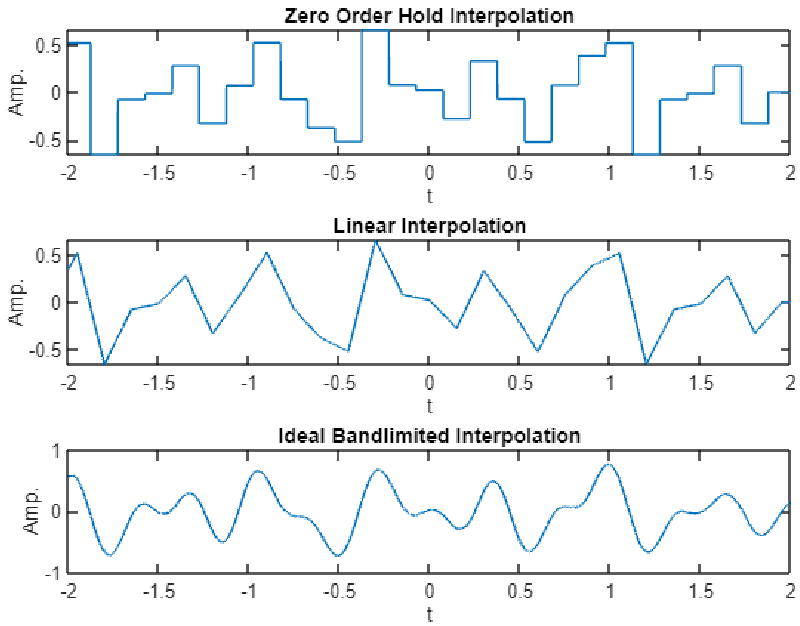


Fig3.10 T_s =0.5

After comparing Fig3.2,3.3,3,4 it can be observed that reconstruction of $x(t)$ is the most accurate with Ideal Bandlimited Interpolation and there is no significant difference between the original signal and reconstructed signal. After making this observation, $x(t)$ is reconstructed by using different T_S values. As explained previously, in some cases where 0.1 ≤ T_S ≤ 0.2 the signal can not be reconstructed due to T_S exceeding $\frac{2\pi}{\omega_c}$. This inequilty is explained in Part 1. In addition accuracy of the reconstructed signal increases as T_S approaches to 0.01 within the interval s 0.01 < T_S < 0.1.