EEE - 321: Signals and Systems

Lab Assignment 5

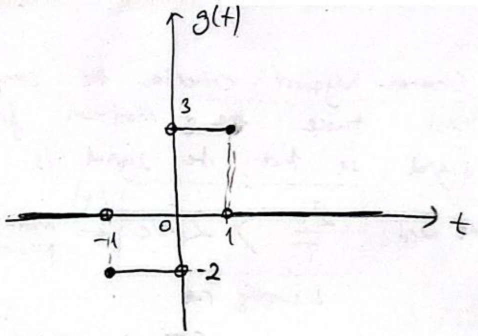
Yiğit Narter

22102718

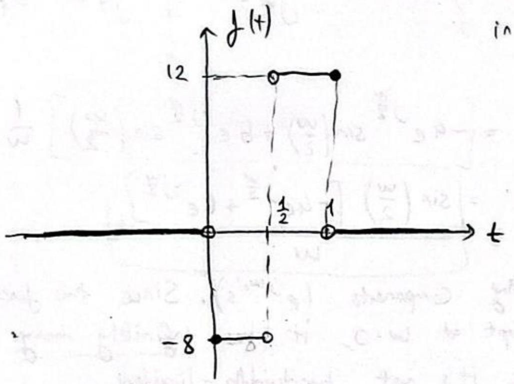
Section 02

Part 1

$$g(t) = \begin{cases} -2, & -1 \leq t < 0 \\ 3, & 0 < t \leq 1 \\ 0, & \text{o.w} \end{cases}$$

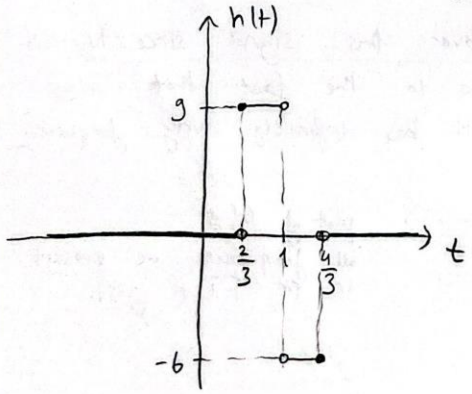


$f(t) = 4g(2t-1) = 4g\left(2\left(t-\frac{1}{2}\right)\right) \rightarrow$ First scale by $\frac{1}{2}$ in the t-axis and shift to the right by $\frac{1}{2}$. Then scale by 4 in the y-axis;



$$\Rightarrow f(t) = \begin{cases} -8, & 0 \leq t < \frac{1}{2} \\ 12, & \frac{1}{2} < t \leq 1 \\ 0, & \text{o.w} \end{cases}$$

$h(t) = 3g(-3(t-1)) \rightarrow$ First scale by $\frac{-1}{3}$ (reverse wrt y-axis due to minus sign) and shift to the right in the x-axis, then scale by 3 in the y-axis.



$$\Rightarrow h(t) = \begin{cases} 9, & \frac{2}{3} \leq t < 1 \\ -6, & 1 < t \leq \frac{4}{3} \\ 0, & \text{o.w} \end{cases}$$

For Shannon-Nyquist criteria, the sampling rate of the signal should be at least twice the maximum frequency (or bandwidth) component in the signal so that the signal is reconstructed accurately:

$$\text{we need: } \boxed{\underbrace{\frac{2\pi}{T}}_{\text{sampling rate}} > 2\,w_c} \ (1) \longrightarrow \text{maximum frequency present in the signal's Fourier transform}$$

However, consider the FT of $g(t)$:

$$G(w) = \int_{-\infty}^{\infty} \underbrace{g(t)}_{\substack{\text{non-zero for} \\ -1 \le t < 0 \text{ and } 0 < t \le 1}} e^{-jwt}\,dt = \int_{-1}^{0} (-2)\,e^{-jwt}\,dt + \int_{0}^{1} (3)\,e^{-jwt}\,dt = \frac{2}{jw}\left[e^{-jwt}\right]_{-1}^{0} + \frac{3}{jw}\left(e^{-jwt}\right]_{1}^{0}$$

$$= \frac{-2e^{j\frac{w}{2}}}{jw}\underbrace{\left(e^{j\frac{w}{2}} - e^{-j\frac{w}{2}}\right)}_{2j.\sin\left(\frac{w}{2}\right)} + \frac{3}{jw}e^{-j\frac{w}{2}}\underbrace{\left(e^{j\frac{w}{2}} - e^{-j\frac{w}{2}}\right)}_{2j.\sin\left(\frac{w}{2}\right)} = \left[-4e^{j\frac{w}{2}}\sin\left(\frac{w}{2}\right) + 6e^{-j\frac{w}{2}}\sin\left(\frac{w}{2}\right)\right].\frac{1}{w}$$

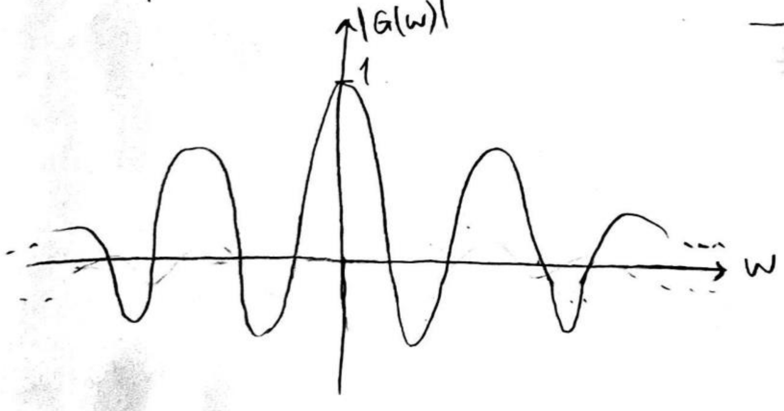
$$= \boxed{\frac{\sin\left(\frac{w}{2}\right)\left[-4e^{j\frac{w}{2}} + 6e^{-j\frac{w}{2}}\right]}{w}} \quad \text{when } w \neq 0$$

when $w=0$, $G(0) = 1$ ⟵

This is a sinc function with shifting components (e^{jw_0}'s). Since this function is continuous for $-\infty < w < \infty$ ⟹ $G(w)$ has infinitely many frequency components, in other words, it's not bandwidth-limited.

So, $w_c = \infty$ (the function $G(w)$ doesn't die out until $w_c \to \infty$, in other words no w_c exists such that $G(w) = 0$ for $w > w_c$)

⟹ It is not possible to fully recover this signal since Nyquist criteria cannot be satisfied (due to the fact that $G(w)$ is not bandwidth limited, meaning it has infinitely many frequency components).



⟶ plot of $G(w)$
all frequencies are present in the FT of $g(t)$.

Part 2

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

Let $x_R(t) = \tilde{x}(t) * p(t) \Rightarrow x_R(t) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(nT_s)\,\delta(\tau-nT_s)\,p(t-\tau)\,d\tau$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) \int_{-\infty}^{\infty} p(t-\tau)\,\underbrace{\delta(\tau-nT_s)}_{\substack{\text{non-zero for} \\ \tau=nT_s}}\,d\tau$$

sifting prop. of impulse

$$= \sum_{n=-\infty}^{\infty} x(nT_s)\,p(t-nT_s) \underbrace{\int_{-\infty}^{\infty} \delta(\tau-nT_s)\,d\tau}_{1}$$

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

$$\Rightarrow \boxed{x_R(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\,p(t-nT_s) = \sum_{n=-\infty}^{\infty} \tilde{x}[n']\,p(t-nT_s)}$$

discrete version sampled at $t=nT_s$

- Since $x_R(t) = \underbrace{\sum_{n=-\infty}^{\infty} x(nT_s)\,p(t-nT_s)}_{\text{Let } n=k \text{ (change of variable)}} \Rightarrow x_R(nT_s) = \sum_{k=-\infty}^{\infty} x(kT_s)\,p(nT_s-kT_s)$

$$\Rightarrow x_R(nT_s) = \underbrace{x(nT_s)\,p(0)}_{\text{at } k=n} + \sum_{k=-\infty}^{n-1} x(kT_s)\,p((n-k)T_s) + \sum_{k=n+1}^{\infty} x(T_s)\,p((n-k)T_s)$$

$n-k \neq 0$ since $k \neq n$ in the summations. Let $n-k=m$, m is therefore a non-zero integer!

Now, if $p(0)=1$ and $p(mT_s)=0$ for $m \neq 0, m \in \mathbb{Z}$:

$$\Rightarrow x_R(nT_s) = \underbrace{x(nT_s)}_{\substack{\| \\ \tilde{x}[n],\ \text{sampled version in eq. 4}}}\ \underbrace{p(0)}_{\substack{\| \\ 1}} + \underbrace{\sum_{k=-\infty}^{n-1} x(kT_s)\cdot 0}_{0} + \underbrace{\sum_{k=n+1}^{\infty} x(kT_s)\cdot 0}_{0}$$

$\Rightarrow \boxed{x_R(nT_s) = \tilde{x}[n]}$ prov. that $p(0)=1$ and $p(mT_s)=0$ for any nonzero integer m.

a) $p_Z(0) = \text{rect}(0) = 1$
$p_L(0) = \text{tri}(0) = 1 - |0| = 1$
$p_I(0) = 1$

$\Big\}$ by the definitions.

b) For p_Z, $\underset{\text{rect}(k)}{\underset{\|}{p_Z(kT_s)}} = \begin{cases} 1 & \text{if } -\frac{1}{2} \leq k \leq \frac{1}{2} \\ 0, & \text{o.w} \end{cases}$

Only non-negative integer in these ranges is <u>zero</u>.

For $p_L(kT_s) = \text{tri}(k) = \begin{cases} 1 - \frac{|k|}{0.5} & \text{if } -0.5 \leq k \leq 0.5 \\ 0, & \text{o.w} \end{cases}$

for $p_I(kT_s) = \text{sinc}(k) = \begin{cases} 1 & \text{if } k = 0 \\ \frac{\sin(\pi k)}{\pi k}, & \text{o.w.} \end{cases}$

c) $\underbrace{\begin{aligned} &\text{At } t=0,\ p_Z(0)=1 \text{ and} \\ &\text{At } t=0,\ p_L(0)=1 \text{ and} \\ &\text{At } t=0,\ p_I(0)=1 \text{ and} \end{aligned}}_{\text{shown in pt. b and a}}$ $\begin{aligned} &p_Z(nT_s)=0 \text{ for other } n, (n \in \mathbb{Z} - \{0\}) \\ &p_L(nT_s)=0 \text{ for other } n, (n \in \mathbb{Z} - \{0\}) \\ &p_I(nT_s)=0 \text{ for other } n, (n \in \mathbb{Z} - \{0\}) \end{aligned}$

(since $\sin(\pi n) = 0$ for $n \in \mathbb{Z}$)

$\Rightarrow$ All the interpolations are consistent since $p(0)=1$ and $p(kT_s)=0$ for n (n is a nonzero integer) $\Rightarrow x_R(nT_s) = \bar{x}[n]$

Part 3

The function generateInterp:

```
function p = generateInterp(type, Ts, dur)
    t = -dur/2:Ts/500:dur/2;
    p = zeros(size(t));
    if type == 0 %zero-order interpolation
        p(abs(t) <= Ts/2) = 1;
    elseif type == 1 % linear interpolation
        p = max(1 - abs(t)/(Ts * 0.5), 0);
    elseif type == 2 %ideal interpolation
        p = sinc(t/Ts);
        p(t==0) = 1;
    end
end
```

The code to generate the plot:

```
dur = mod(22102718, 7);
Ts = dur/5;

pZ = generateInterp(0, Ts, dur);
pL = generateInterp(1, Ts, dur);
pI = generateInterp(2, Ts, dur);
t = -dur/2:Ts/500:dur/2;

subplot(3, 1, 1)
plot(t, pZ)
title('Zero-Order Interpolation')
xlabel('t')
ylabel('pZ(t)')

subplot(3, 1, 2)
plot(t, pL)
title('Linear Interpolation')
xlabel('t')
ylabel('pL(t)')

subplot(3, 1, 3)
plot(t, pI)
title('Ideal Interpolation')
xlabel('t')
ylabel('pI(t)')
```

The output plot is presented in Figure 1.

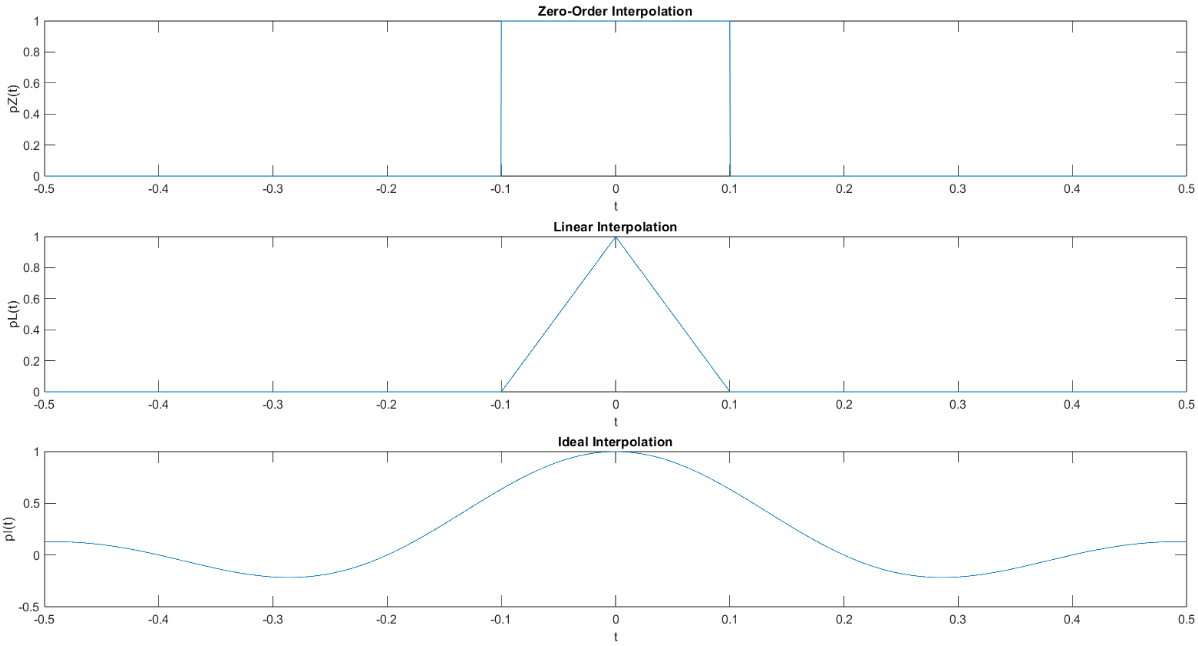


Figure 1: Plots for p_Z (t), p_L(t) and p_I(t)

Part 4

The function DtoA:

```
function xR=DtoA(type,Ts,dur,Xn)
    t = -dur/2 : Ts/500 : dur/2;
    p = generateInterp(type,Ts,dur);
    len = length(t) + length(Xn)*500; %find length of the convolved array xR
    xR = zeros(1, len);

    for n = 0:length(Xn)-1
        xR(500*n+1:500*n+length(p)) = xR(500*n+1:500*n+length(p)) + Xn(n+1)*p;
    end

    xR = xR(250*length(Xn)+1: end-250*length(Xn)); %crop accordingly
end
```

Part 5

The code used to generate a sampled version of g(t) -called g(nT_s)- and g_R(t) for each interpolating method is given below.

```matlab
% Generate sampled version of g(t)
a = randi([2, 6]);
Ts = 1 / (20 * a);
t = -3:Ts:3;
n = t/Ts;

g = zeros(size(t));
g((-1 <= t) & (t < 0)) = -2;
g((0 < t) & (t <= 1)) = 3;
figure;
stem(n, g, 'filled')
title('Sampled Signal g(nTs)')
xlabel('n')
ylabel('g(nTs)')


dur = 6; %for -3<t<3
xR0 = DtoA(0, Ts, dur, g); %zero-order
xR1 = DtoA(1, Ts, dur, g); %linear
xR2 = DtoA(2, Ts, dur, g); %ideal


% Plot the reconstructed signals
tR = linspace(-dur/2, dur/2, numel(xR0));
figure;
subplot(3, 1, 1)
plot(tR, xR0)
title('Reconstructed Signal using Zero-Order Interpolation')
xlabel('t')
ylabel('xR(t)')
subplot(3, 1, 2)
plot(tR, xR1)
title('Reconstructed Signal using Linear Interpolation')
xlabel('t')
ylabel('xR(t)')
subplot(3, 1, 3)
plot(tR, xR2)
title('Reconstructed Signal using Ideal Interpolation')
xlabel('t')
ylabel('xR(t)')
```

The resulting plots are given in Figures 2-3.

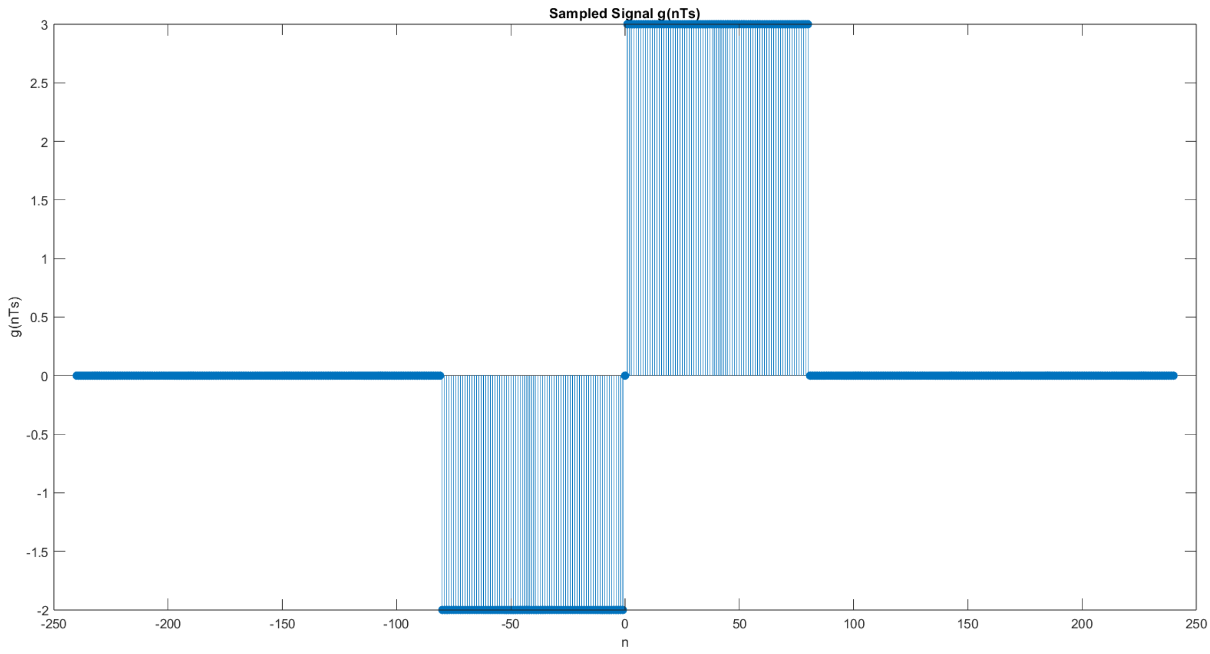


Figure 2: Stem plot of g (nT_s)

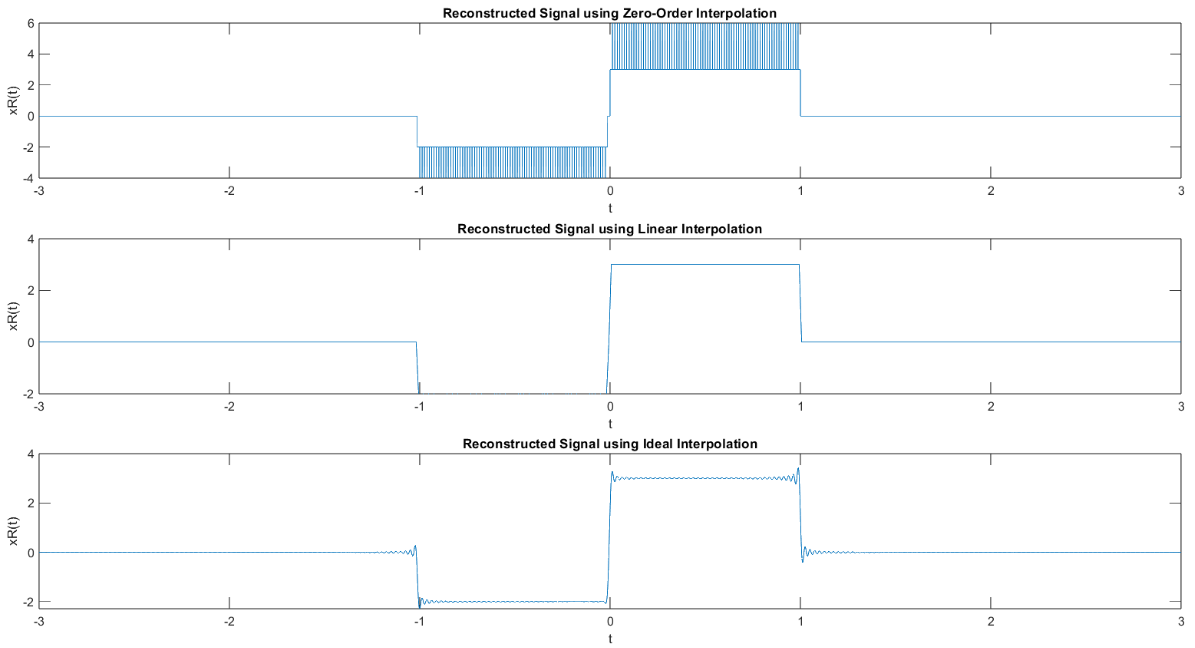


Figure 3: Plots of $g_R(t)$ for each interpolating method

When the resulting plots are observed, it can be seen that the zero-order method, which uses a constant value between samples to reconstruct the signal, is prone to stair-step artifacts and aliasing effects. It can be seen clearly in the upper parts of the reconstructed signal. This makes it the least successful method out of the three.

On the other hand, linear interpolation introduces a linear effect between samples, making it more successful than the zero-order method. With linear interpolation, the reconstructed signal is smoother and involves fewer discontinuities. However, there are still discontinuities and fracture points in the reconstructed signal, despite being less obvious than those of the zero-order method. The vertical lines are also not perfectly straight and again involve fracture points; at points that are close to the discontinuities of g(t) (0, 1, and -1), the g(t) values don't match. For example, at points smaller than -1, g(t) is 0 but in the reconstructed signal at some points smaller than (but close to) -1 $x_R(t)$ is not 0.

The ideal bandlimited interpolation, which introduces a sinc function, proves to be the most successful method out of the three. Despite the presence of some oscillations around the discontinuities, the reconstructed signal resembles g(t) the most, and the vertical lines have minimal fracture. It has the potential to perfectly reconstruct a signal, however in this case since g(t) has infinitely many frequency components, the result is not perfect, though successful.

As T_S is increased gradually, the sampling rate decreases, and the reconstructed signals become smoother since fewer samples are taken in the same time interval, resulting in smoother lines. However, with fewer samples, the accuracy is harmed and finer details of the signals are lost. Hence the reconstruction becomes less successful in terms of accuracy. This effect can be observed for different increased values of T_S, as presented in Figure 4.

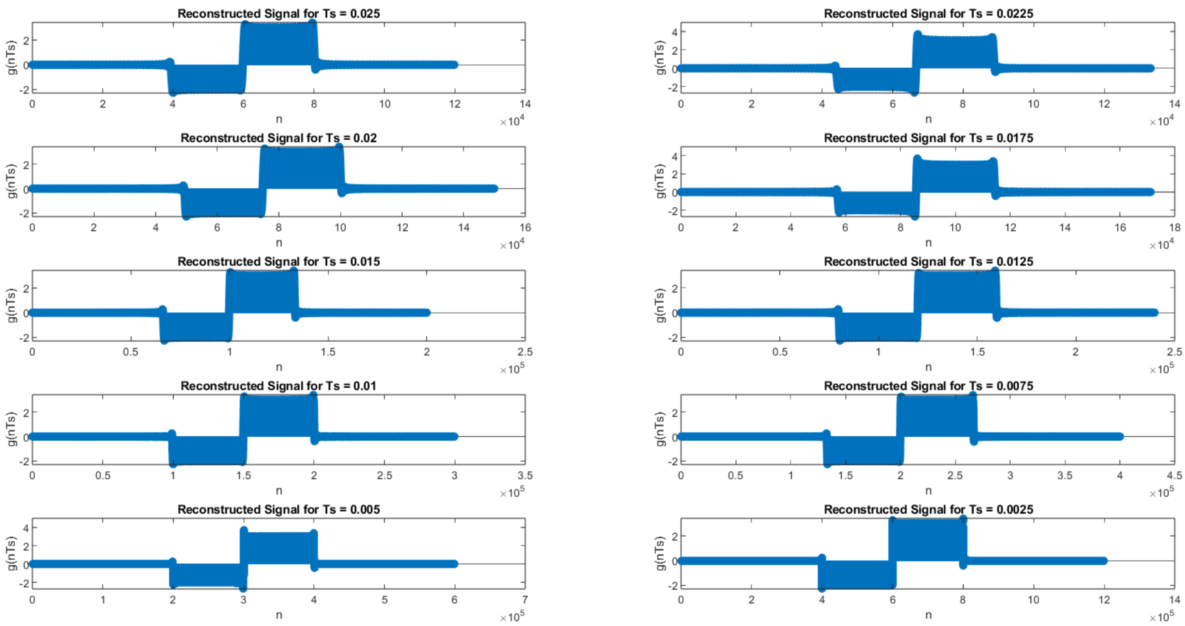


Figure 4: As T_S is increased, the accuracy decreases but signals become smoother

Part 6

The code used in part 6:

```
D = 22102718;
D7 = rem(D, 7);
Ts = 0.005 * (D7 + 1);
Ts2 = 0.25+ 0.01 * D7;
Ts3 = 0.18 + 0.005 * (D7 + 1);
Ts4 =0.099;

t = linspace(-2, 2, 1000);
tx = -2:Ts:2;
n = tx/Ts;

x = 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);
Xn = 0.25 * cos(2*pi*3*n*Ts + pi/8) + 0.4 * cos(2*pi*5*n*Ts -
1.2) + 0.9 * cos(2*pi*n*Ts + pi/4);
figure;
plot(t, x, 'r', 'LineWidth', 1.5);
hold on;
stem(n*Ts, Xn, 'b', 'filled');
title('Continuous Signal x(t) and Sampled Signal x(nTs)');
xlabel('t');
ylabel('x(t), x(nTs)');
legend('Continuous Signal x(t)', 'Sampled Signal x(nTs)');
xR0 = DtoA(0, Ts, 4, Xn);
xR1 = DtoA(1, Ts, 4, Xn);
xR2 = DtoA(2, Ts, 4, Xn);

tR = linspace(-2, 2, numel(xR0));

figure;
subplot(3, 1, 1);
plot(tR, xR0, 'LineWidth', 1.5);
title('Zero-Order Interpolation');
xlabel('t');
ylabel('xR(t)');
subplot(3, 1, 2);
plot(tR, xR1, 'LineWidth', 1.5);
title('Linear Interpolation');
xlabel('t');
ylabel('xR(t)');
subplot(3, 1, 3);
plot(tR, xR2, 'LineWidth', 1.5);
title('Ideal Band-Limited Interpolation');
xlabel('t');
ylabel('xR(t)');
```

a) $T_s = 0.005(D_7 + 1)$

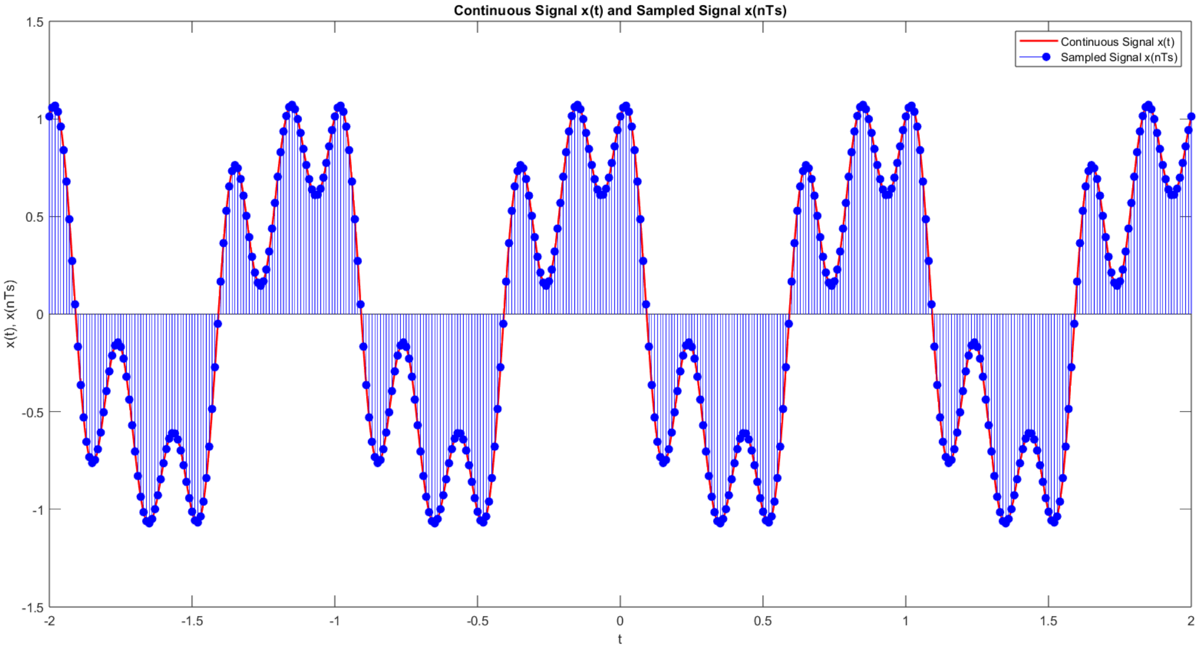


Figure 5: x(t) and x(nTs) for $T_s = 0.005(D_7 + 1)$

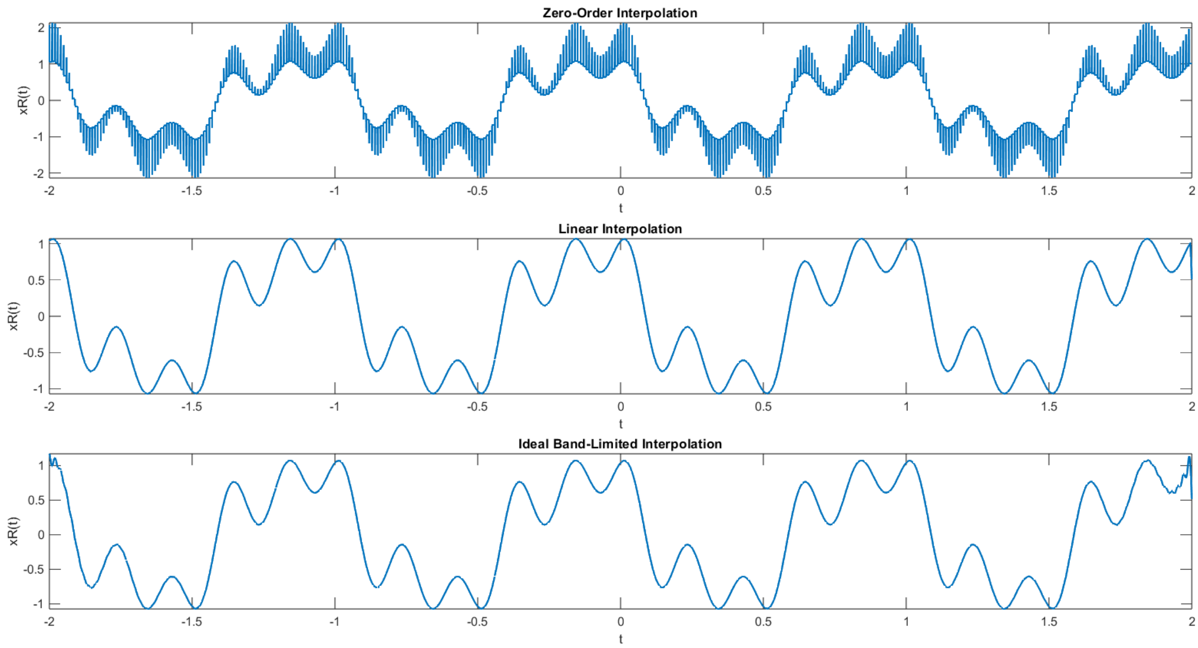


Figure 6: Reconstructed signals for $T_s = 0.005(D_7 + 1)$

b) $T_s = 0.25+0.01D_7$

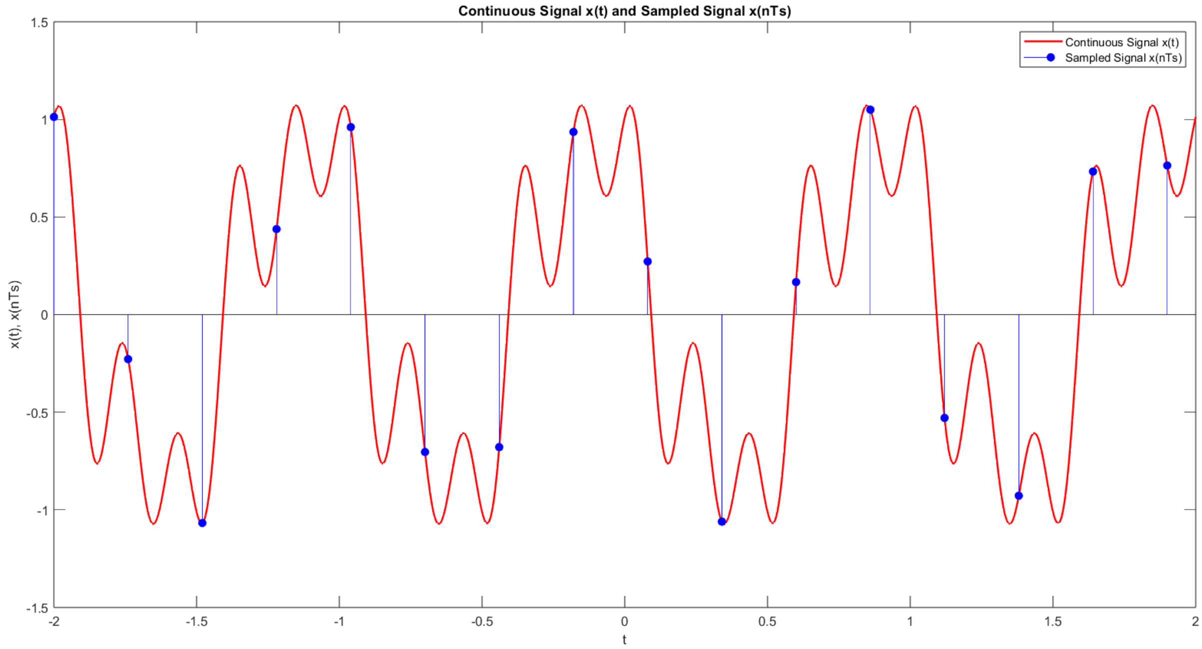


Figure 7: x(t) and $x(nT_S)$ for $T_s = 0.25+0.01D_7$

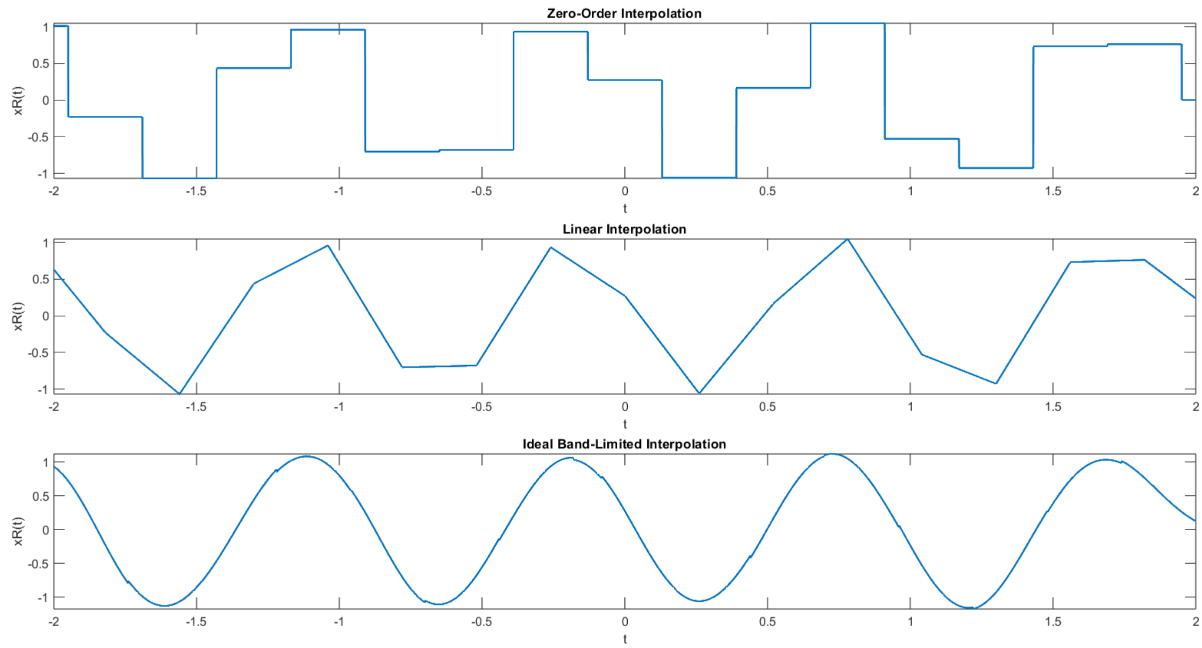


Figure 8: Reconstructed signals for $T_s = 0.25+0.01D_7$

c) $T_s = 0.18+0.005(D_7 + 1)$

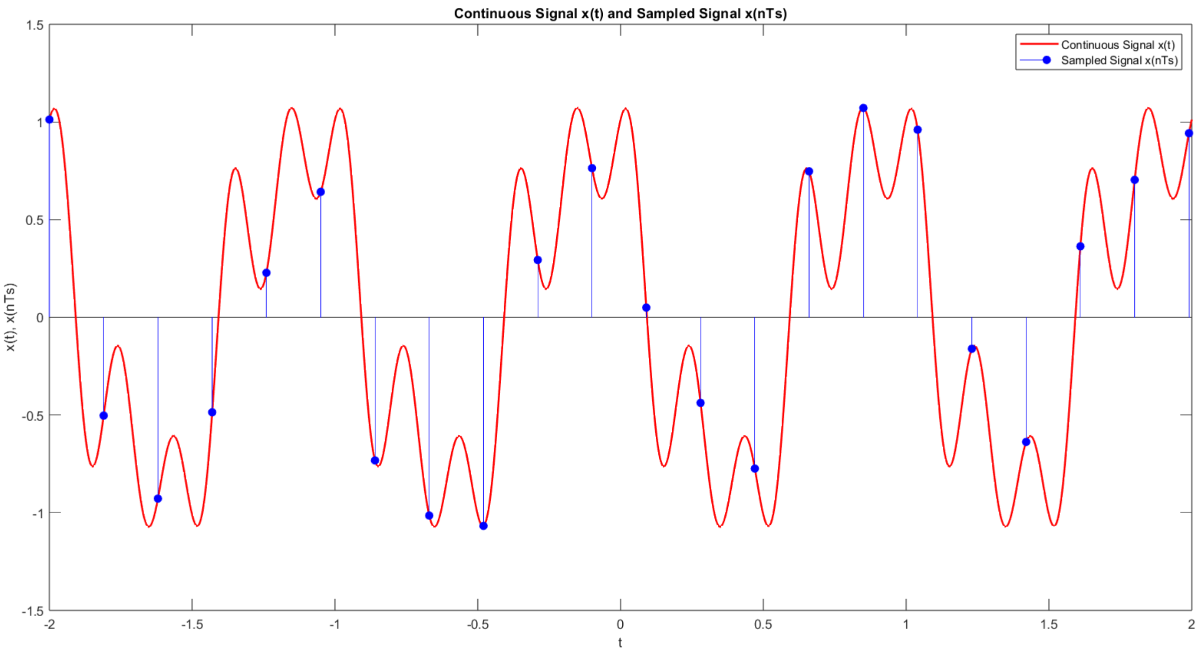


Figure 9: x(t) and x(nTs) for $T_s = 0.18+0.005(D_7 + 1)$

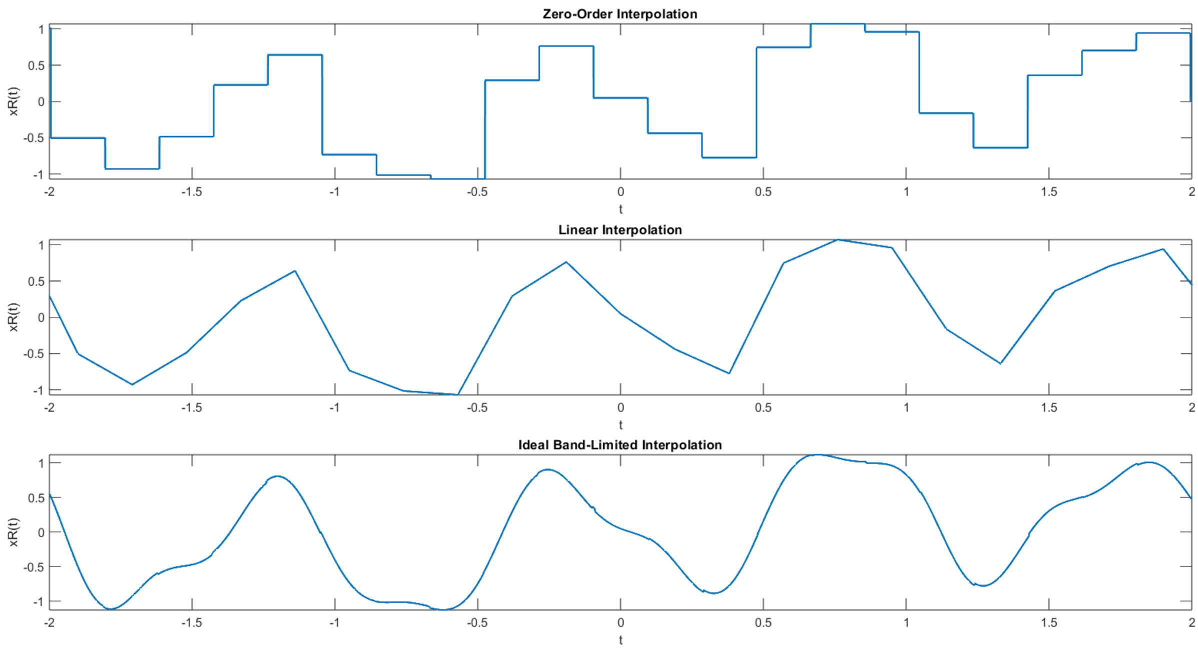


Figure 10: Reconstructed signals for $T_s = 0.18+0.005(D_7 + 1)$

d) $T_s = 0.099$

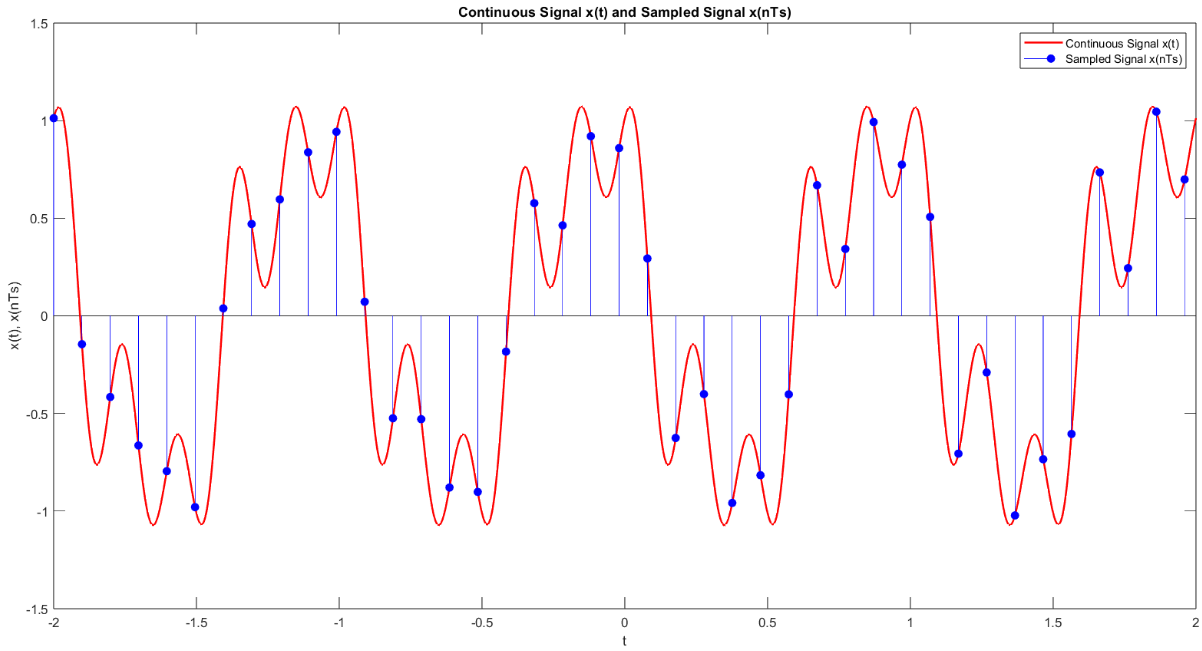


Figure 11: x(t) and x(nTS) for $T_s = 0.099$

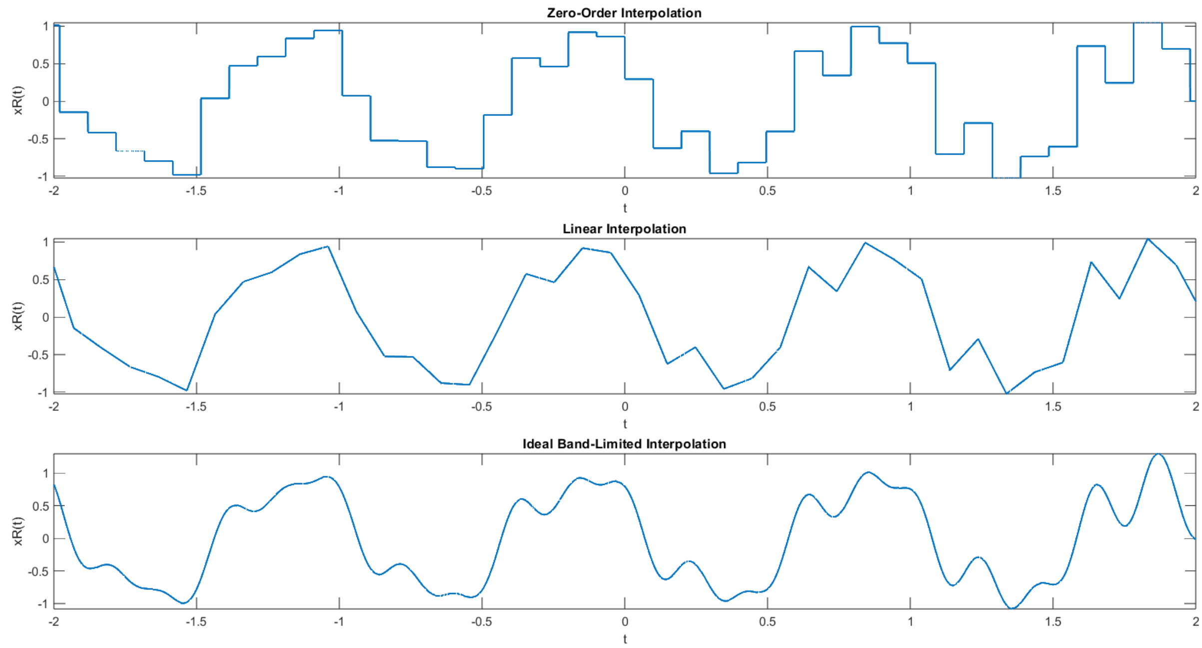


Figure 12: Reconstructed signals for $T_s = 0.099$

When the results for 6a are examined, it can be seen that the ideal bandlimited interpolation turns out to be the most successful method. The reason is that its reconstructed signal involves minimal distortions and also maintains the fine details. The ideal bandlimited interpolation's reconstruction resembles the original signal closely, in fact, they are nearly identical. Still, there exists a minimal difference due to the process of finite sampling, though this difference can be considered negligible.

After the code was run with different T_S values, it was seen that as T_S increased, the reconstructed signals resembled the original lesser and became less accurate. For zero-order and linear interpolations, the increase in T_S caused the reconstructed signal to become different and inaccurate. This can be seen clearly in the reconstructed signal plots of 6b (Figure 8), where $0.25+0.01D_7$ is the highest value among the parts. For ideal interpolation also, the mentioned minimal difference becomes more noticeable, and the details of the signal are lost. Again, examining Figure 8, the ideal interpolation reconstruction is not so similar to the original signal. Hence for increased T_S, the accuracy of the ideal interpolation decreases. Still, among the three, it is still the most resembling one therefore also the most successful.

For T_S values between 0.01 and 0.1, the difference between the original and the reconstructed signal is minimal and hardly recognizable. The two signals are nearly identical. Hence the performance of the method is successful. However, for $0.1 \leq T_s \leq 0.2$, some distortions and inaccuracies are observed, and the reconstructed signal deviates significantly from the original. Hence among the parts b, c, and d; d was the most successful one.

This is because of the aliasing effect, due to the fact that the sampling rate fails to fulfill the Nyquist criteria mentioned earlier (since the rate decreases as Ts increases). This also causes a loss of information in the original signal. Hence as T_S increases, the success of the ideal interpolation decreases since the sampling rate decreases and becomes less than the original signal's bandwidth, becoming insufficient to capture all the information in the original signal. This happens due to the Nyquist criteria mentioned in part 1, which states that the sampling rate ($2\pi/T_s$) should be at least twice the maximum frequency component of the signal. This results in the fact that with increased T_s, the sampling rate decreases and eventually fails to fulfill the criteria, causing alias effects, distortions, inaccuracies, and loss of information.