Signals and Systems Lab 5

Sait Sarper Özaslan 22002861

Part 1)

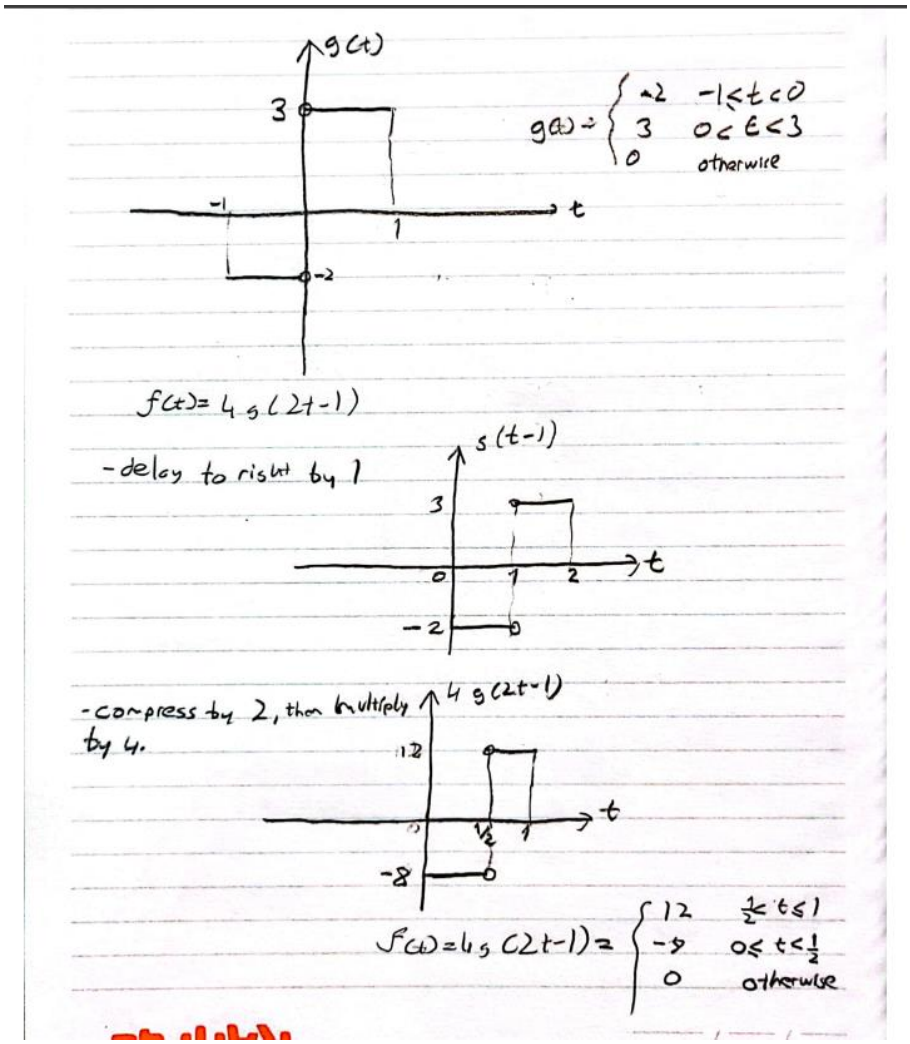
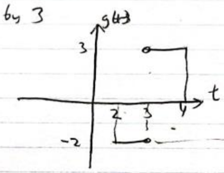


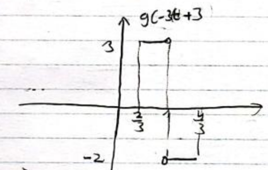
Figure 1: Explanation of Part 1

$h(t) = 3g(-3(t-1)) = 3g(-(3t-3))$

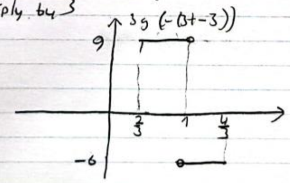
- shift to right by 3



- compress by 3, do time reversal centered on 1 (compressed center)



multiply by 3



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

Figure 2: Explanation of Part 1-2

- We can't recover when $g(t)$ is sampled. The Frequency domain of $g(t)$ can be seen as $g(jw)$, a sinc function w/ Sinc function is not a band limited signal, in other words $G(jw) \neq 0$ for $w > w_c$.

To show this,

$$G(jw) = \int_{-\infty}^{\infty} g(t) e^{-jwt} dt = \int_0^1 3e^{-jwt} dt + \int_{-1}^{0} -2e^{-jwt} dt$$

$$3\frac{e^{-jwt}}{jw}\Big|_1^0 + 2\frac{e^{-jwt}}{jw}\Big|_{-1}^0$$

$$3+2 - \frac{3e^{-jw}}{jw} - 2\frac{e^{jw}}{jw}$$

$$G(jw) = \frac{5}{jw} - \frac{e^{-jw}}{jw} - 4\cos(jw)$$

As one can see $G(jw > jw_m) \neq 0$

Figure 3: Explanation of Part 1

Part 2

Part 2

Let's define $\tilde{x}[n]$, sampled version of $x(t)$ as

$$\tilde{x}[n] = x(nT_s)$$

To achieve that, we use an impulse train with period T_s

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

convolving this function with $p(t)$ we will have our reconstructed signal as essentially we are applying a filter.

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

The last equality is derived from, impulse's sampling property

Hence
$$\boxed{x_R(t) = \sum_{n=-\infty}^{\infty} \tilde{x}[n]\, p(t - nT_s)}$$

☆ If $p(0) = 1$ and $p(kT_s) = 0$ we can only reconstruct at each $n'T_s$ when $t = n'T_s$. In other words

$$x_R(t) = \sum_{n'=-\infty}^{\infty} \tilde{x}[n']\, p(t - n'T_s)$$ Turns into

$$\boxed{x_R(n'T_s) = \tilde{x}[n] \quad \text{when } t = n'T_s,\ p(0) = 1}$$

consistent interpolation

If $t \neq n'T_s$ $p(t - n'T_s) = 0$

Figure 4: Explanation of Part 2

a) $P_Z, P_L, P_I(0) = 1$ as all can be seen from given definition, when $t=0$ is placed we get 1.

b) For $t = kT_s$

$$P_Z(kT_s) = P_L(kT_s) = P_I(kT_s) = 0$$

- For $P_Z(kT_s)$ we get 0 as we repeat $\sin \pi k = 0$ on numerator
- other 2 are just essentially out of defined interval.

c) Since we know $P_Z, P_L, P_I(0) = 1$ if pick a $t = kT$ such that $k = n$

$$X_R(kT_s) = \sum_{-\infty}^{\infty} x[n] \underbrace{p(kT_s - nT_s)}_{p(0)=1}$$

$$X_R(kT_s) = x[n]$$

for if $k \neq n$ we know that $p((k-n)T_s) = 0$. Hence, All of them are consistent.

Figure 5: Explanation of Part 2(a,b,c)

Part 3

Here is the code and plots for part 3:

```
function p = generateInterp(type,Ts,dur)
% dur = mod(22002861,7);% dur = 6
% Ts= dur/5;
Tf=Ts/500;
```

```
t= -dur/2:Tf:dur/2-Tf;
%dur/(Ts/500) =2501 points
k=size(t);
lengh=length(t);
p=zeros(k);
if type ==0
	for i= 1:lengh
		if -Ts/(2)<= t(i) && t(i)<Ts/(2)
			p(i)=1;
		end
	end
end
if type ==1
	for i= 1:lengh
		if -Ts/2<= t(i) && t(i)<Ts/2
			p(i)= 1- abs(t(i))/(Ts/2);
		end
	end
end
if type ==2
	for i= 1:lengh
		if t(i) ~= 0
			p(i)=sin(pi*t(i)/Ts)/(pi*t(i)/Ts);
		else
			p(i)= 1;
		end
	end
end


dur = mod(22002861,7);% dur = 6
```

```
Ts= dur/5;

type = 0;

Tf=Ts/500;

t= -dur/2:Tf:dur/2-Tf;

p= generateInterp(type,Ts,dur);

figure (1),plot(t,p), title("Zero Order Hold"), xlabel("Time
Duration"),ylabel("P_Z(t)"), grid on,

type = 1;

a= generateInterp(type,Ts,dur);

figure (2),plot(t,a), title("Linear Interpolation"), xlabel("Time
Duration"),ylabel("P_L(t)"), grid on,

type = 2;

b= generateInterp(type,Ts,dur);

figure (3),plot(t,b), title("İdeal Bandlimited İnterpolation"), xlabel("Time
Duration"),ylabel("P_I(t)"), grid on,
```

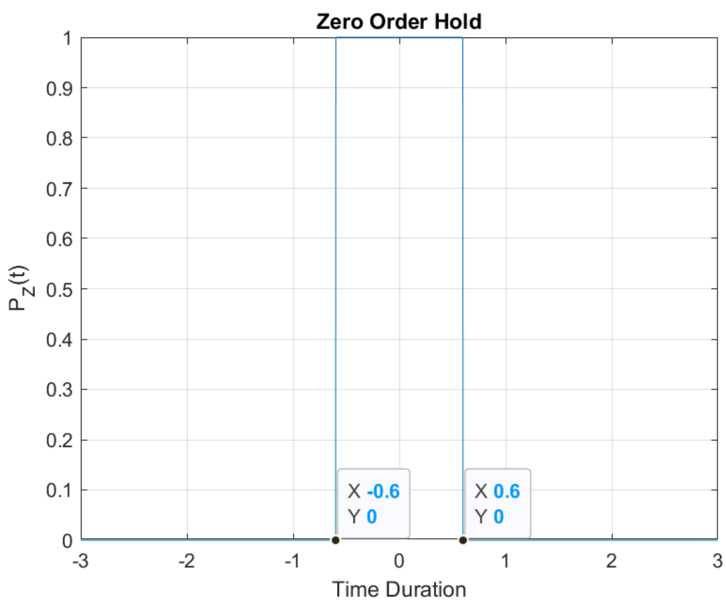


Figure 6: Zero Order Hold Interpolation

it as it is.

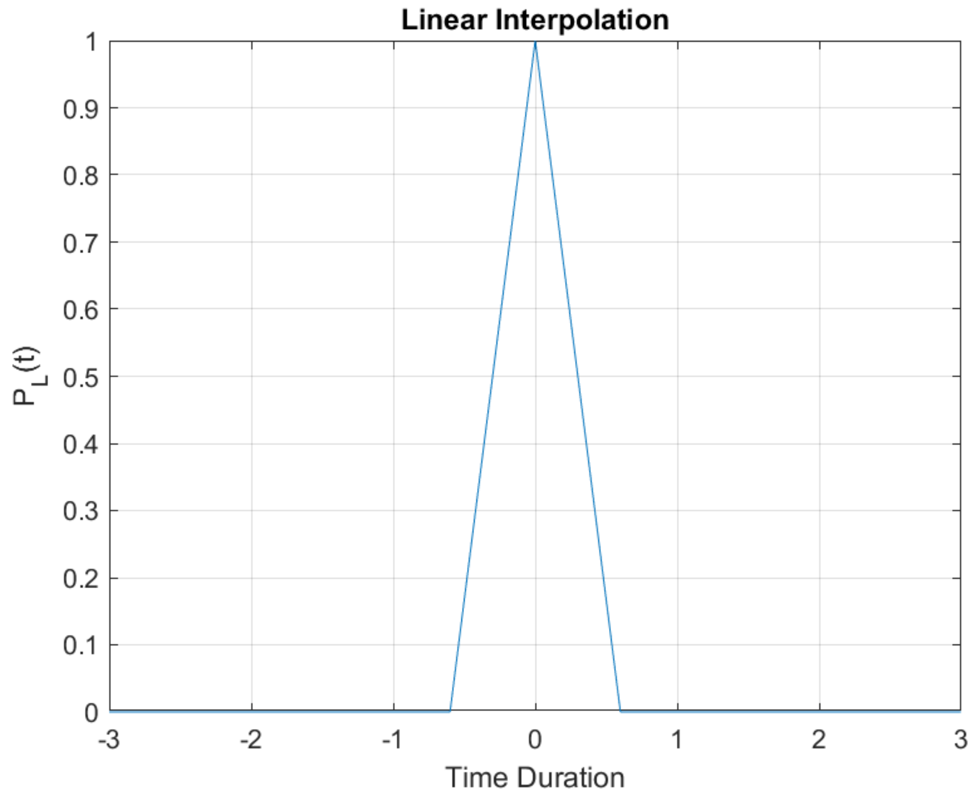


Figure 7: Linear Interpolation

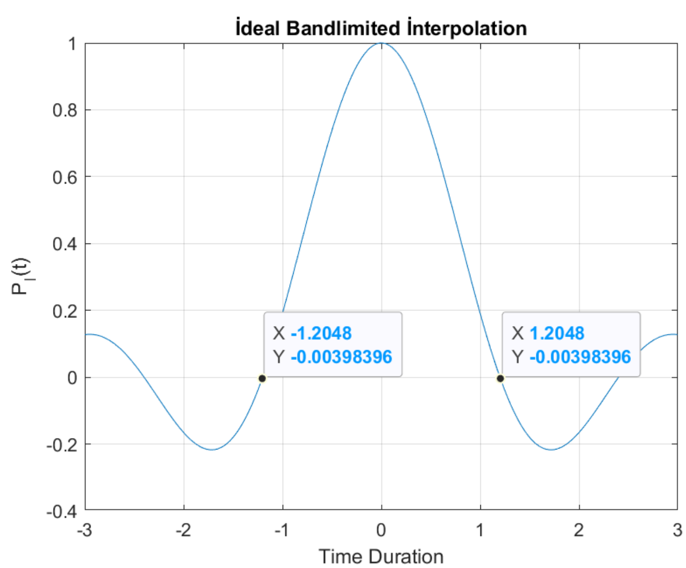


Figure 8: Ideal Bandlimited Interpolation

Looking at the code the values where 0 occurs are as expected. Though I used a for loop to access t array, it can also be done without using a for loop. However, that wasn't necessary for this part, so I kept it as it is.

Part 4

Here is the code for the part:

```
function xR=DtoA(type,Ts,dur,Xn)

Tf= Ts/500

% we define the length of xn as:

n=length(Xn);

interp = generateInterp(type,Ts,dur*Ts);

xR=zeros(1,round((dur+(n-1))*500)); %the matrix length is n-1 samples taken + the one that doesn’t get counted by the other on the left and right

%now we have the relation for the Xr we can write xr as the sum of:

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)*interp;

end
```

Part 5

Code for g(t) and reconstructed function:

```
a=randi([2 6],1);

Ts=1/(20*a);

Tf=Ts/500;

dur=6;%for -3<t<3

%let's define t

t = -dur/2:Ts:dur/2-Ts;%-1 assures that there is a point at t=0

%we have 120a-1 points so the middle point is surely is in 0

k = length(t);

g= zeros(1,k); %let's define the length of gt before assigning values

g(-1<=t & t<0)=-2;

g(0<t & t<=1)=3;

%figure ,plot (g)

figure(4), stem(t,g), title("Stemplot"),
```

```
gR0=DtoA(0,Ts,dur,g);

t0=linspace(-3,3,length(gR0));

figure (5), plot(t0,gR0), title("Reconstructed - Zero Hold"),
xlabel("Time"),ylabel("G_R_O(t)"), 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"),ylabel("G_R_1      "), 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"),ylabel("G_R_2"), grid on,
```

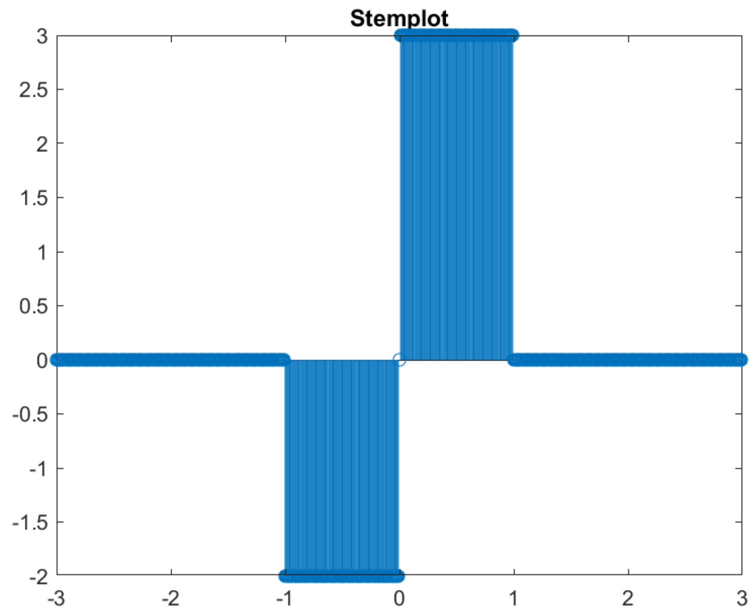


Figure 9: Stem plot of g(nTs)

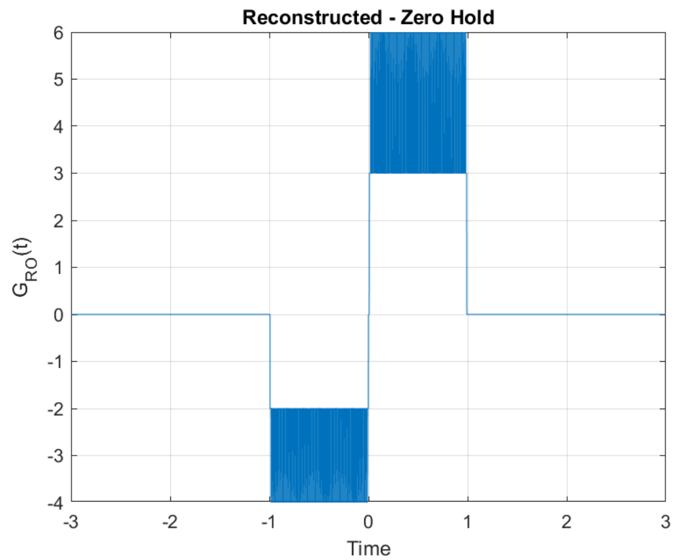


Figure 10: Reconstructed Zero Hold

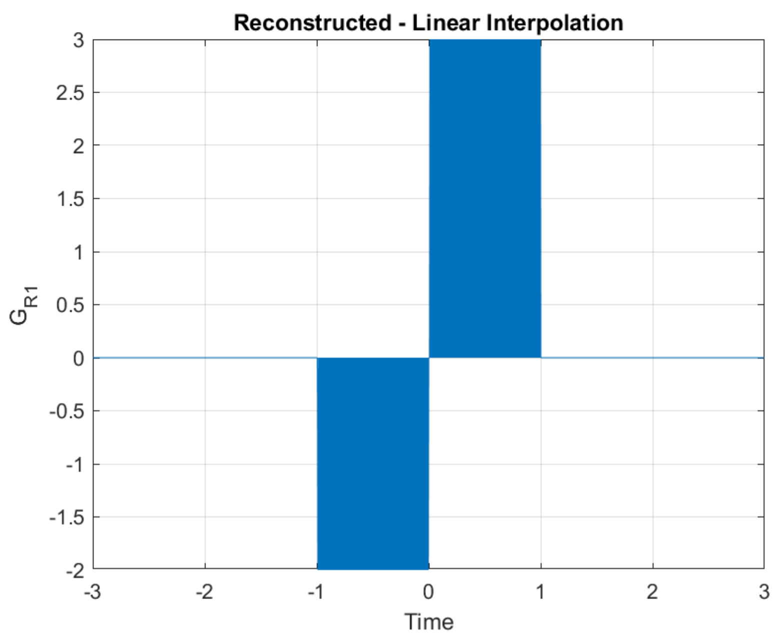


Figure 11: Reconstructed g from Linear Interpolation

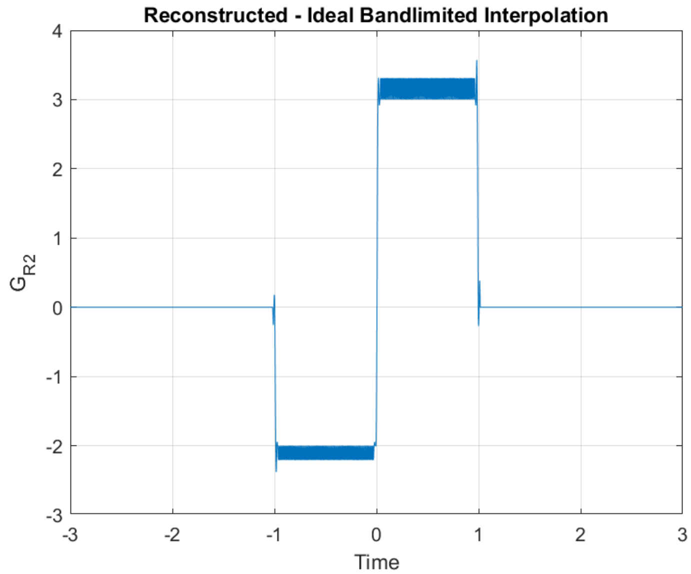


Figure 12: Reconstructed g from Bandlimited Interpolation

There are unexpected spikes in the band-limited filter, the zero-hold reconstruction and linear doesn't give as accurate data as wanted. The latter two are quite different from the g(t). Linear reconstruction, which is a better choice than zero order as it's a better lowpass filter in the frequency domain, still fails to represent g(t) accurately. When it comes to analyzing the ideal bandlimited interpolation's reconstruction, we can say that it is the closest to original signal g(t). However, it should be noted that there are oscillations that are caused by the Gibbs phenomenon.

As a result, the ideal bandlimited interpolation is the most successful one.

As Ts is increased there are less samples being formed and as a result the accuracy of each reconstruction goes down. (Look at appendix for the figures)

Part 6

Here is the code for part 6:

```
D7=rem(22002861,7); % =6

Ts=0.005*(D7+1); % I will change this value for each part

Tf= Ts/500;

dur = 4; % -2<t<2

t= -dur/2:Tf:dur/2-Tf;%analog time

sdt=-dur/2:Ts:dur/2-Ts;%sampled discrete tf

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(8),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));

figure (10) , 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));

figure (11) , plot(tr1,xR1),title('Linear
Interpolation'),xlabel('time'),ylabel('xR(t)')

xRe2= DtoA(2,Ts,dur,sampledxt);

tr2=linspace(-2,2,length(xRe2));

figure (12), plot(tr2,xRe2),title('Bandlimited
Interpolation'),xlabel('time'),ylabel('xR(t)')
```

Part a)

We will compare:

Zero Order Hold Interpolation

Linear Interpolation

Ideal Bandlimited Interpolation

A) `Ts=0.005*(D7+1); %=0.035`

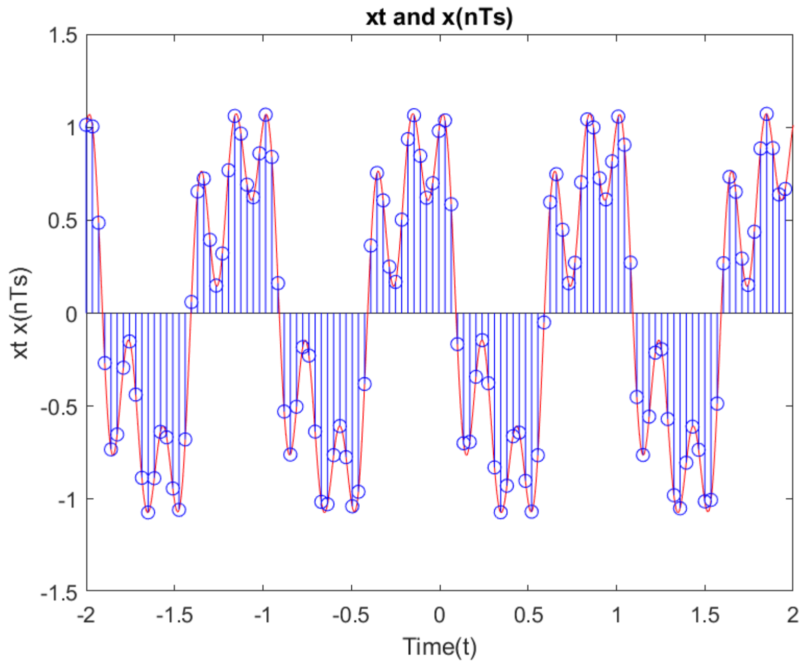


Figure 13: Stemplot of x(nTs) on Plot of xt

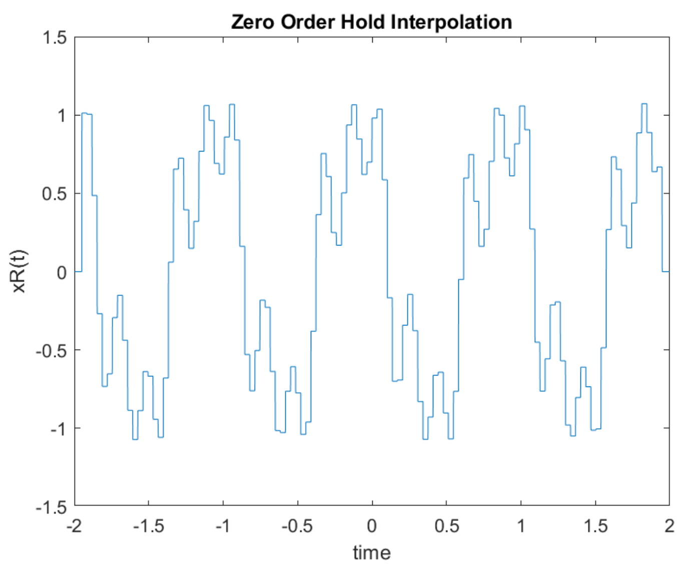


Figure 14: Reconstruction Through Zero Order Hold Interpolation where Ts=0.035

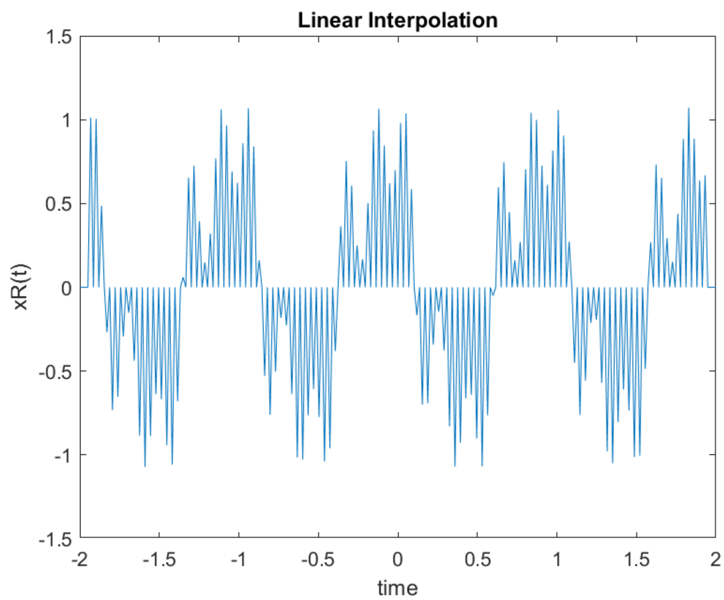


Figure 15: Reconstruction Through Linear Interpolation where Ts=0.035

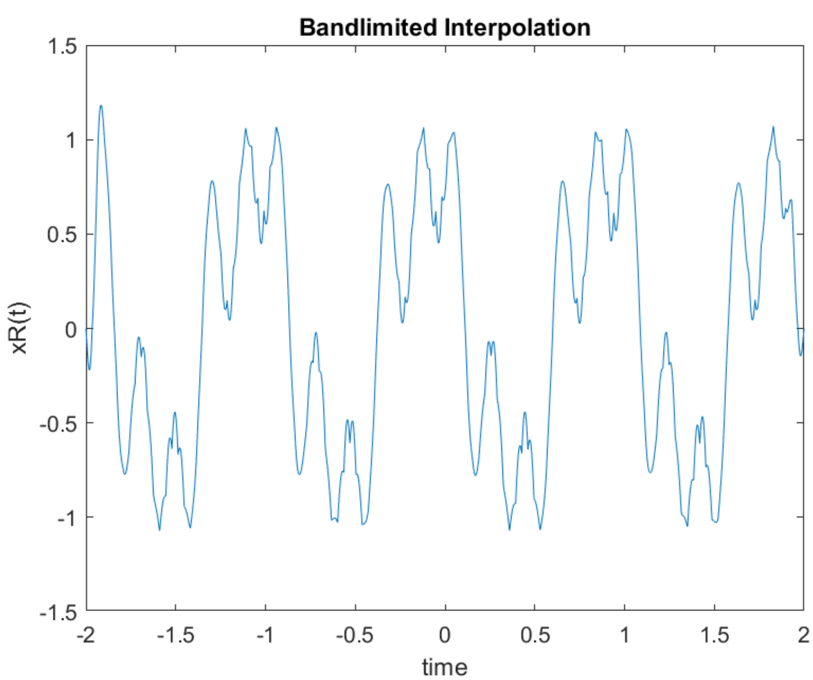


Figure 16: Reconstruction Through Ideal Bandlimited Interpolation where Ts=0.035

Linear Interpolation Reconstruction may look somewhat similar, but it is bad compared to ideal bandlimited one. The zero order adds up a rectangle at each point, so the values look far off as it doesn't reflect the change. While linear still reflects the changes better it is not as good as ideal bandlimited reconstruction.

The ideal band-limited interpolation's reconstruction is the best choice as it is the perfect low pass filter in the ideal conditions. However, we are not in ideal conditions and the reason behind this is explained in the end of part 6.

The original signal can be distinguished from the ideal interpolation even though there are distortions that are caused by the function used in part 4. As we choose to take a finite number of samples to be able to reconstruct the signal, there are bound to be some distortions.

So, if we consider we are in an ideal setting, successes of filter for this time sample are Ideal Bandlimited Interpolation > Linear Interpolation > Zero Order Hold Interpolation.

```
B) Ts=0.25 + 0.01*D7; %=0.32
```

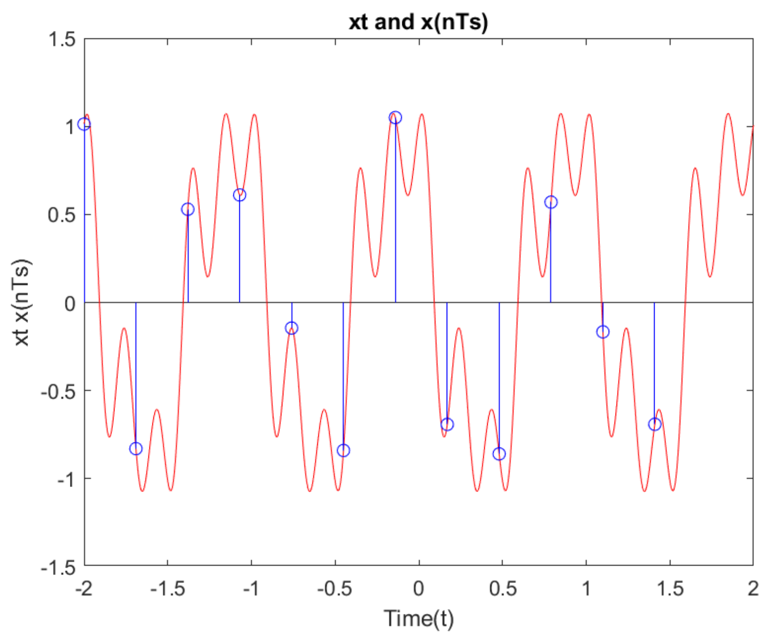


Figure 17: Stemplot of x(nTs) on Plot of xt where Ts=0.32

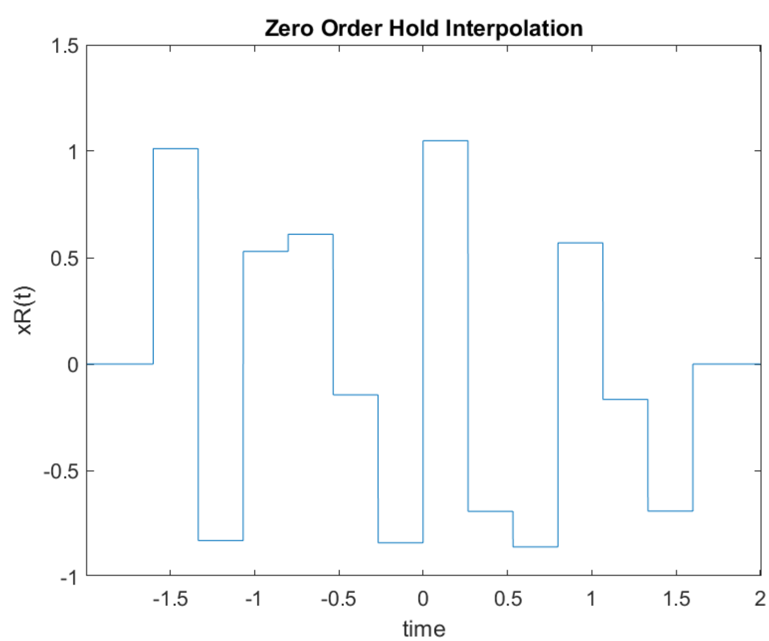


Figure 18: Reconstruction Through Zero Order Hold Interpolation where Ts=0.32

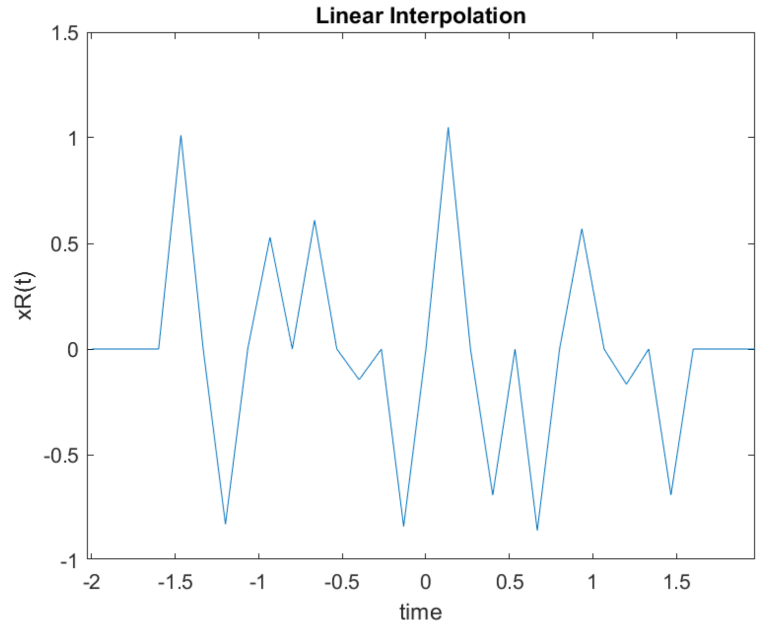


Figure 19: Reconstruction Through Linear Interpolation where Ts=0.32

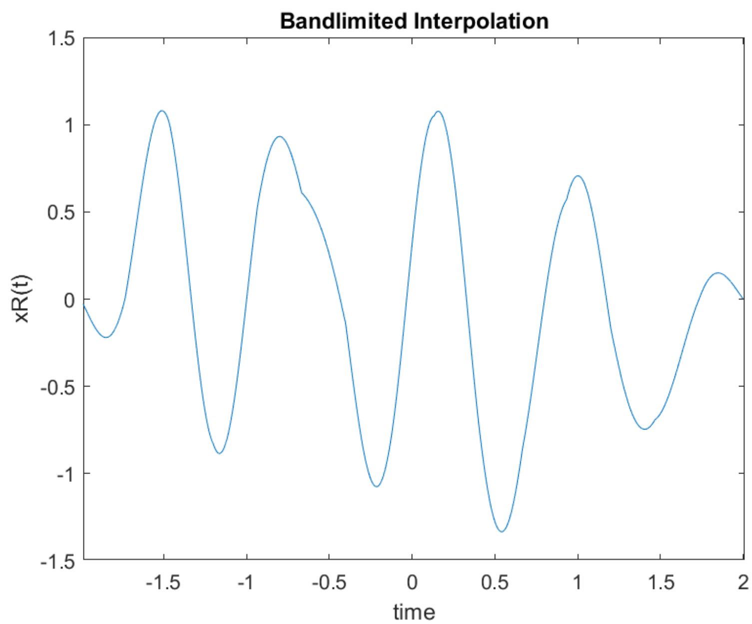


Figure 20: Reconstruction Through Ideal Bandlimited Interpolation where Ts=0.32

The sample rate is too big, so it doesn't make any sense to discuss the successes of each method. If one had to give an answer the Ideal Bandlimited Interpolation looks the closest.

C)
```
Ts=0.18+0.005*(D7+1); %=0.215
```

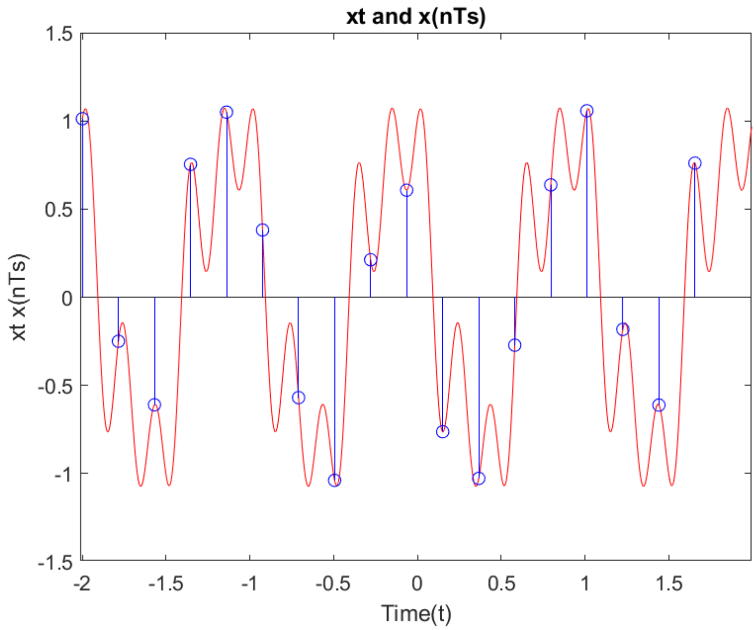


Figure 21: Stemplot of x(nTs) on Plot of xt where Ts=0.215

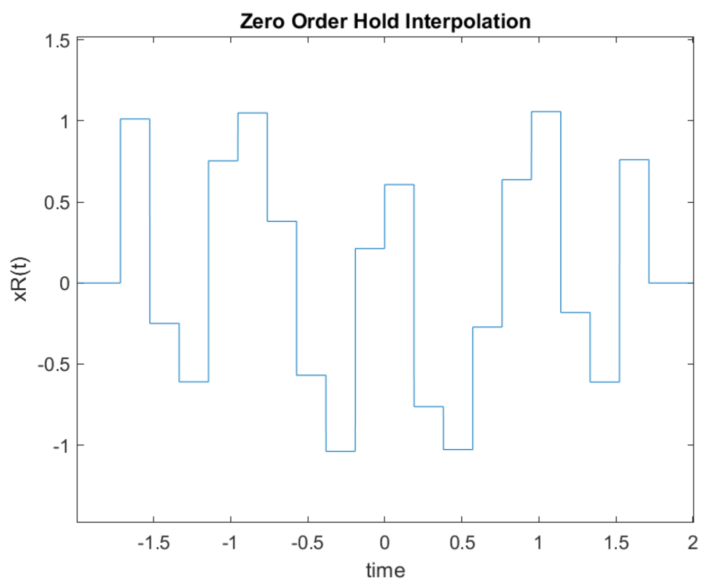


Figure 22: Reconstruction Through Zero Order Hold Interpolation where Ts=0.215

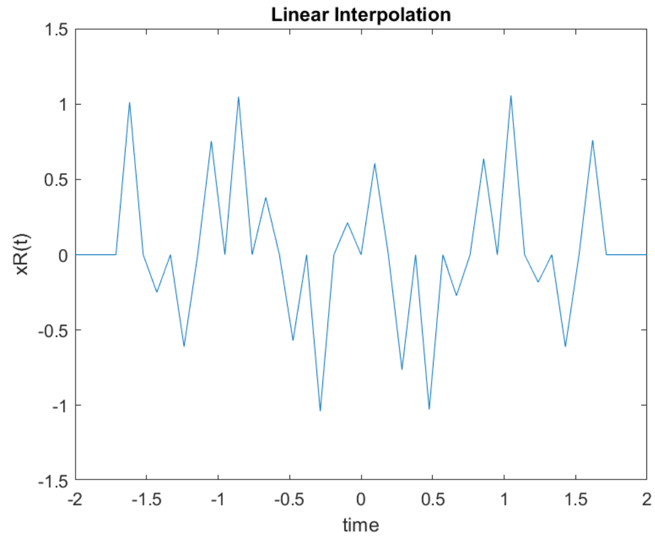


Figure 23: Reconstruction Through Linear Interpolation where Ts=0.215

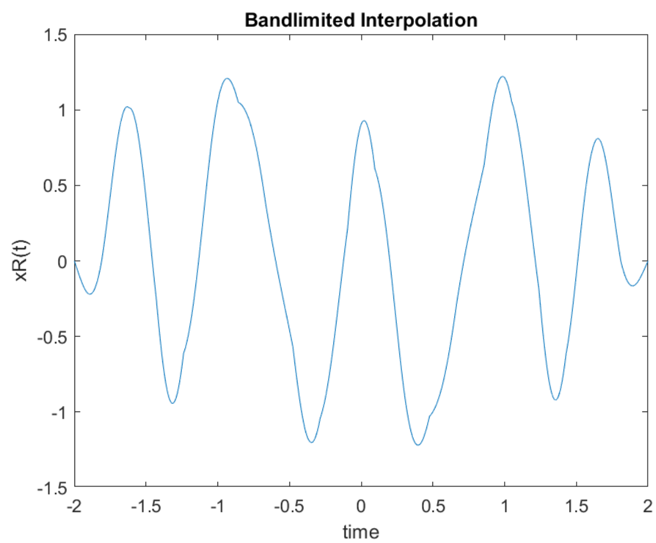


Figure 24: Reconstruction Through Ideal Bandlimited Interpolation where Ts=0.215

The same answer from B, can be given for this part

D) `Ts=0.099;%`

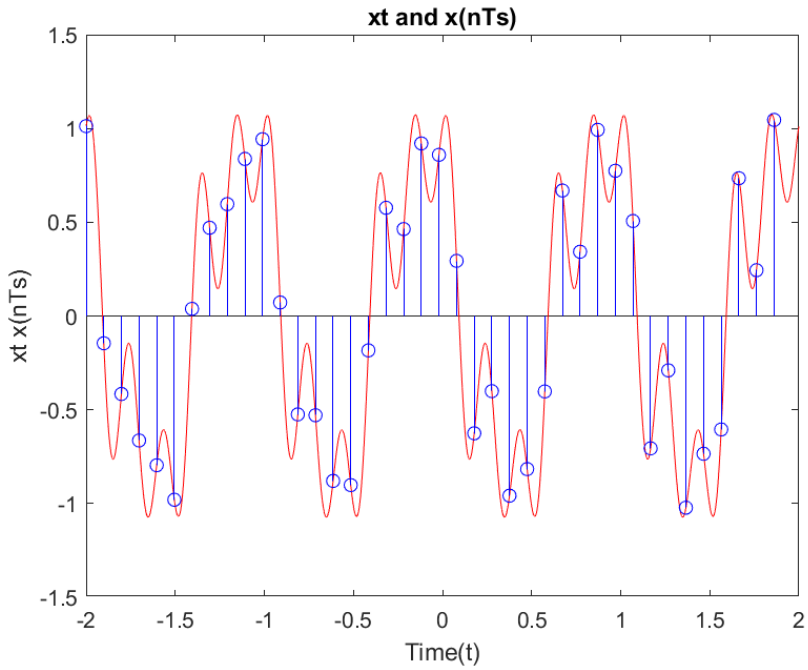


Figure 25: Stemplot of x(nTs) on Plot of xt where Ts=0.099

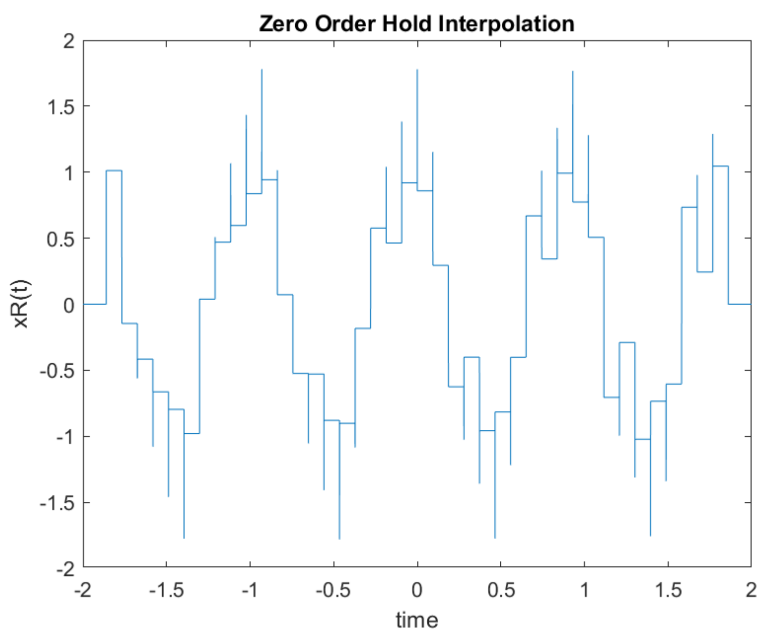


Figure 26: Reconstruction Through Zero Order Hold Interpolation where Ts=0.099

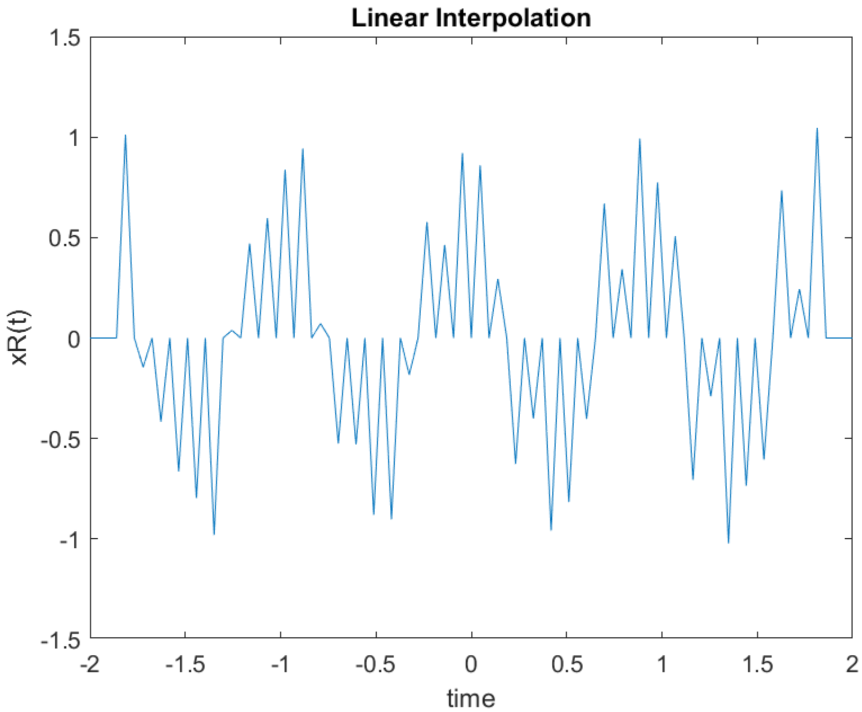


Figure 27: Reconstruction Through Linear Interpolation where Ts=0.099

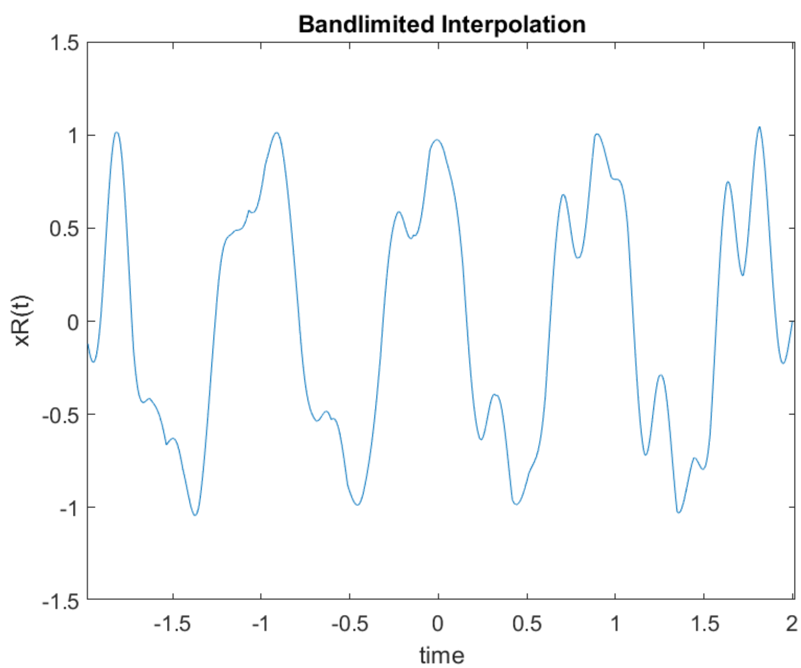


Figure 28: Reconstruction Through Ideal Bandlimited Interpolation where Ts=0.099

In this part the ideal bandlimited looks the most accurate with linear interpolation getting closer to the actual signal.

Looking at all the cases it can be said that as Ts gets smaller, signal gets reconstructed more accurately.

Now let’s reduce the Ts to its limit Ts=0.01 and compare the original plot to reconstructed ones.

- Ts=0.01

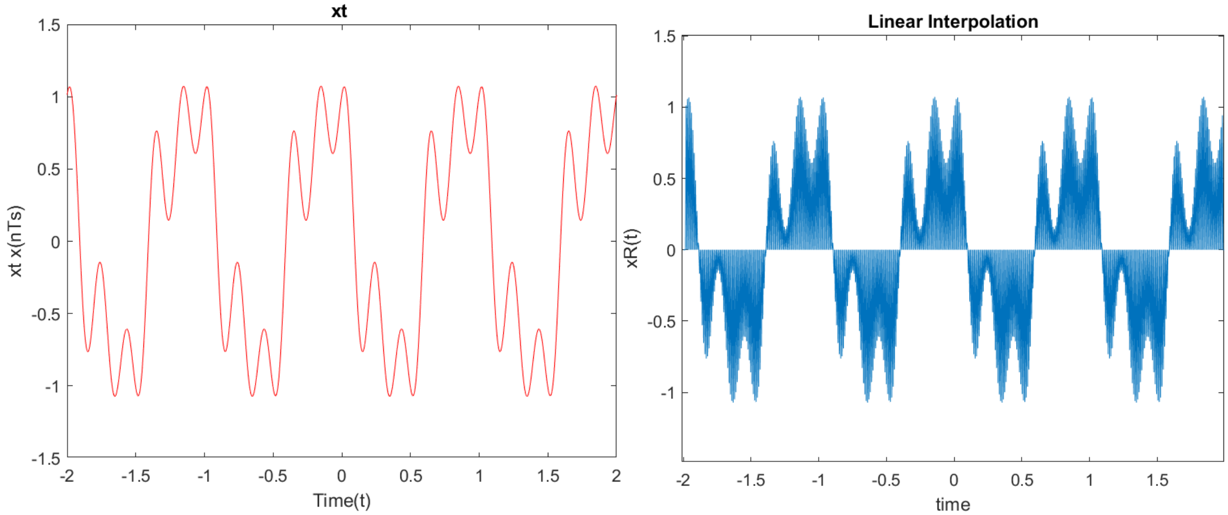


Figure 29&30: Comparison between x(t) and Reconstructed Signal

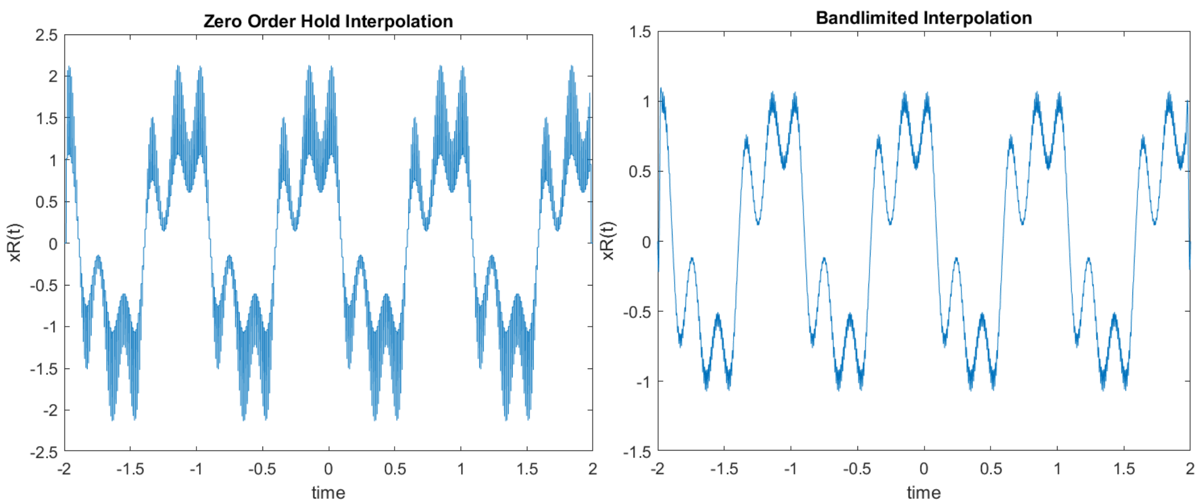


Figure 31 & 32: Zero Order Hold and Ideal Bandlimited Interpolation

Before explaining the plot, it should be noted that there can be some errors in the plot that are caused by MATLABS approximation errors.

For the ideal bandlimited interpolation, the interpolation method ideally will have the best reconstruction since in an ideal setting the signal’s defined range are infinities. However, we cannot create such a function through digital means. As a result, the ideal bandlimited interpolation is not accurate enough and even though it is the best choice linear interpolation seems to give better results especially at lower sample rate.

Earlier it was noted that as the sample rate (sample period) goes down the accuracy of each reconstruction increases. It is smarter to use ideal bandlimited interpolation as the accuracy tends to be higher in the given range, 0.1<t<2. We can confirm this from the from part a as at Ts=0.035 Bandlimited

Reconstruction was close to the original signal, even more than the linear and zero order reconstruction. Thus, Ideal Bandlimited Interpolation is the most successful method.

For the Ts where 0.1<Ts<0.2, the differences between original and reconstructed signal are noticeable as the sample number is not that high and the sum function we use in part 4 is not infinitely many so there are bound to be noticeable differences. Moreover, Nyquist Criteria is not met in this condition.

It should also be noted that even though zero order hold is the least efficient method, the results still get closer to original plot as we decreased the sampling rate. However, it is not an adequate choice for the given range.

As Ts(sample period) gets bigger none of the reconstructions can be considered successful as can be seen from part b and c.

Hence most accurate data is plotted between 0.01<Ts<1 and in this interval and the accuracy goes down extremely between 0.1 < Ts < 0.2 and only the ideal band-limited interpolator is somewhat like the original signal. The reason why the signal is very inaccurate is due to Nyquist criteria, the highest frequency of xt is 5Hz so if we take a Ts larger than 0.1 the Nyquist criteria is not meant so the distortions increase a lot regardless of the number samples.

The calculation for the value can be seen below:

$$\frac{2\pi}{Ts} > 2w_c \ (Nyquist\ Criteria)$$

$$w_c = 2\pi \cdot 5 \ for\ 5Hz$$

$$\frac{1}{10} > Ts \ or\ 0.1 > Ts$$

Appendix

After a was formed once the random part is deleted so that we can get more accurate results in this case a was generated as 5.

- For Ts= 1/10a,

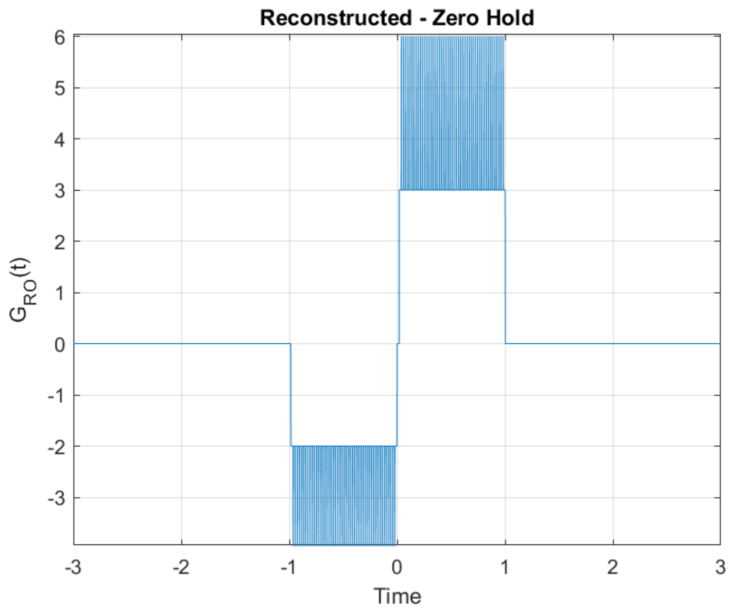


Figure 28: Reconstructed g(t) from Zero Hold for Ts=1/50

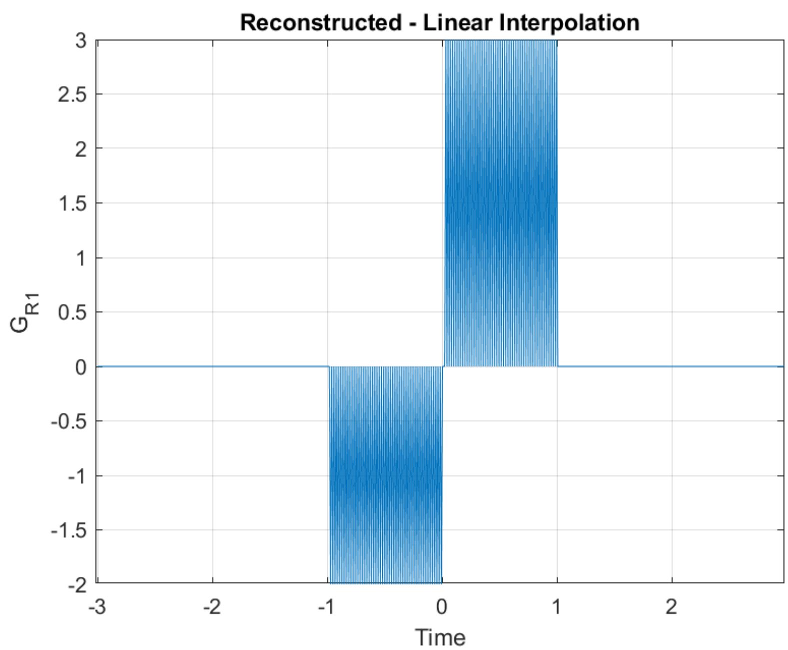


Figure 29: Reconstructed g(t) from Linear Interpolation for Ts=1/50

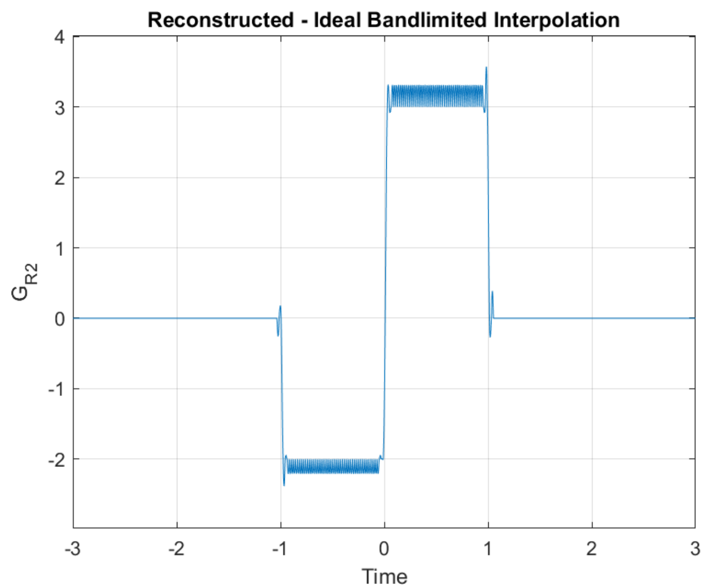


Figure 30: Reconstructed g(t) from Bandlimited Interpolation for Ts=1/50

- Ts=1/5a

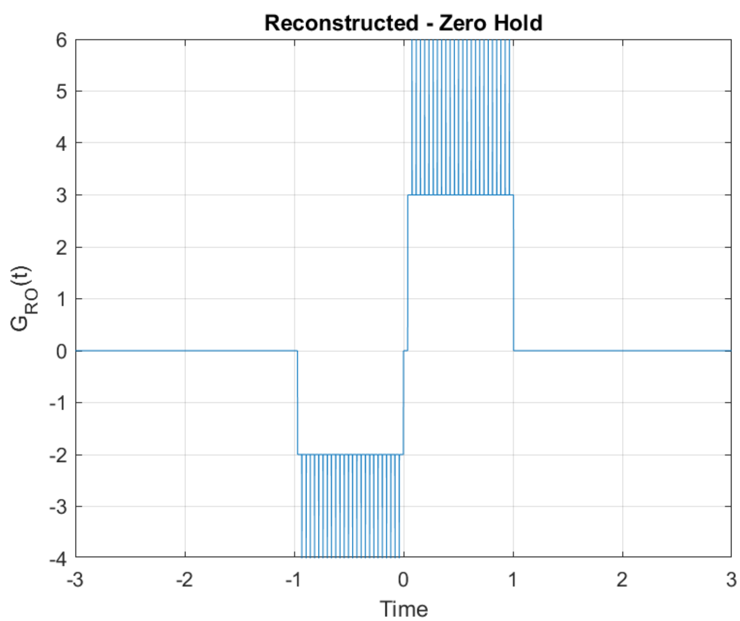


Figure 31: Reconstructed g(t) from Zero hold for Ts=1/25

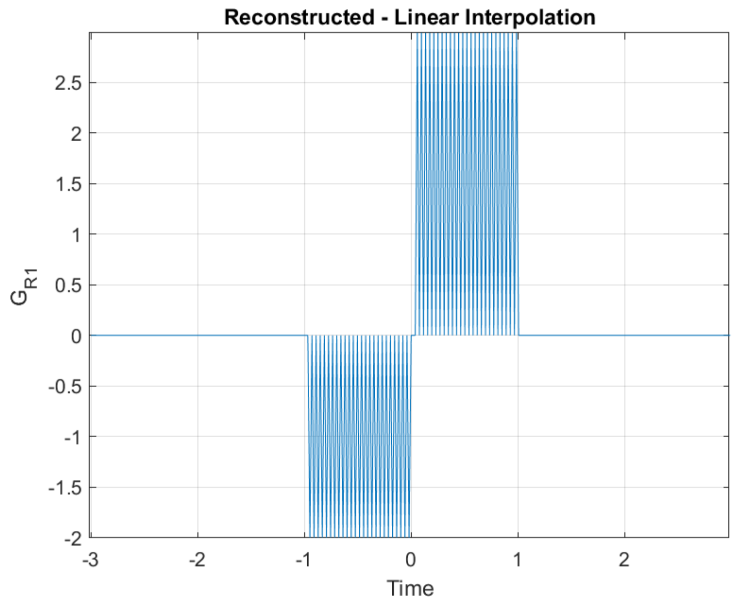


Figure 32: Reconstructed g(t) from Linear Interpolation for Ts=1/25

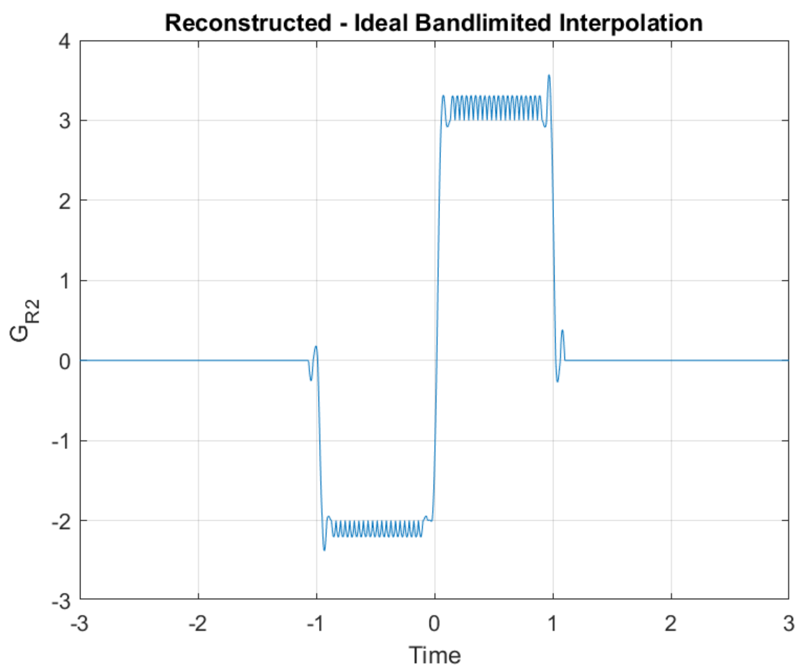


Figure 33: Reconstructed g(t) from Bandlimited Interpolation for Ts=1/25

- Ts=1/a

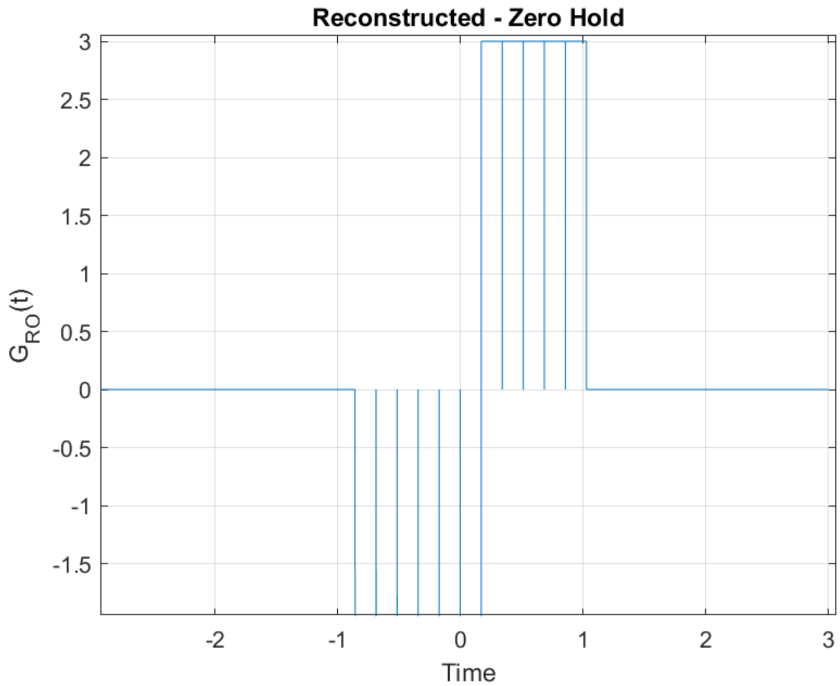


Figure 34: Reconstructed g(t) from Zero Hold for Ts=1/5

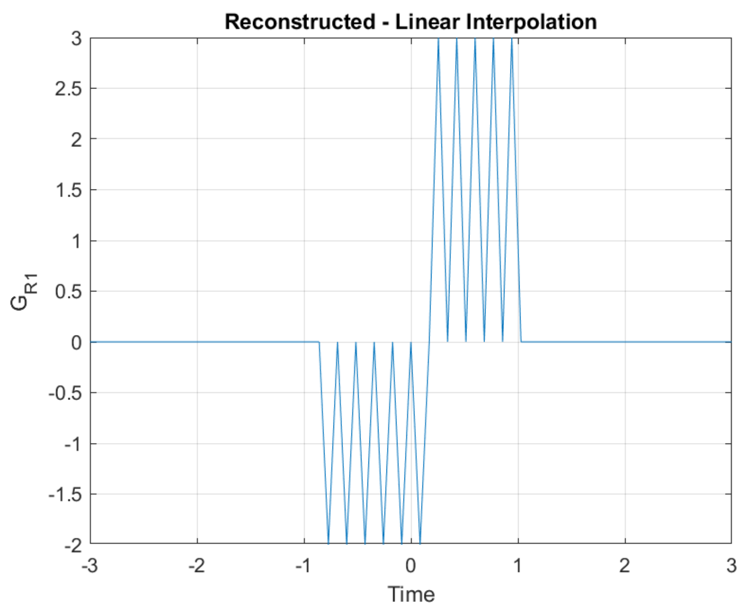


Figure 35: Reconstructed g(t) from Linear Interpolation for Ts=1/5

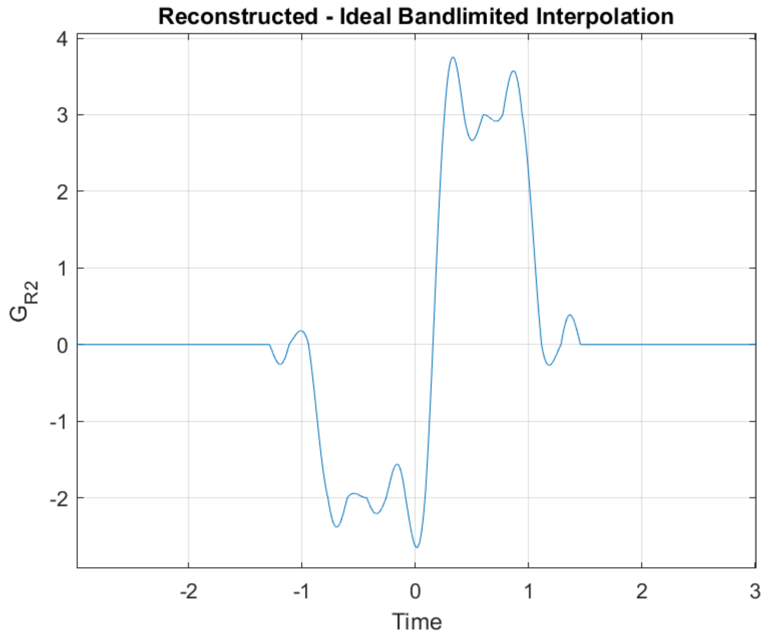


Figure 36: Reconstructed g(t) from Ideal Bandlimited for Ts=1/5