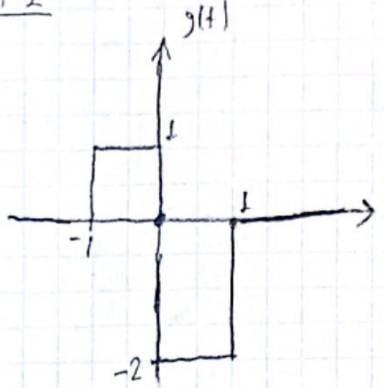
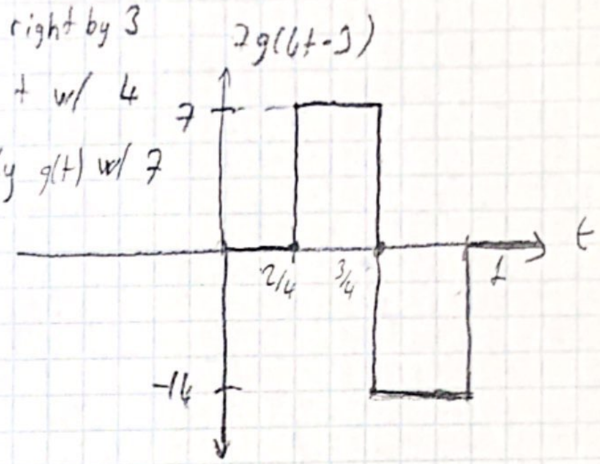
Atahan İyakic'
21801985

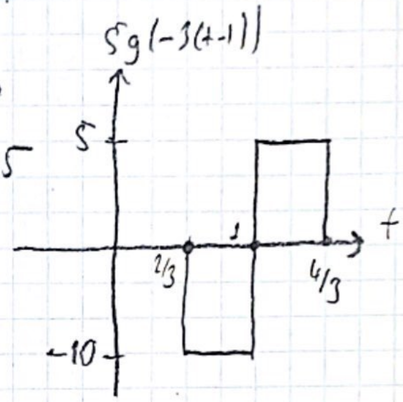
Part 1



$\Rightarrow$ Shift right by 3
Scale t w/ 4
Multiply g(t) w/ 7



Shift right by 1
Scale t w/ -3
$\Rightarrow$ Multiply g(t) w/ 5



g(t) can't be recovered when it is sampled. I cannot be recovered because g(jw) is a sinc function. Sinc function is not band limited function. Therefore, it is not possible to find a Ts value to satisfy Nyquist Criteria.

Part 2

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

$$= \sum_{n=-\infty}^{\infty} x(nT_s)\,\delta(t - nT_s) \longrightarrow \underbrace{x(nT_s)}_{\text{sampled Signal}} = \underbrace{\tilde{x}[n]}_{\text{Discrete time function}}$$

- $\tilde{x}[n] \xrightarrow{\text{from interpolation}} x_R(t)$ As we want $x(t) = x_R(t)$ this equals $x(t)$ $\Rightarrow x_R(t) = x_{sampled}(t) * p(t)$

Thus, $x_R(t) = \int_{-\infty}^{\infty} \underbrace{\sum_{n=-\infty}^{\infty} x(nT_s)\,\delta(t'-nT_s)}_{x_{sampled}(t)}\, p(t-t')\,dt' = \sum_{n=-\infty}^{\infty} x(nT_s)\int_{-\infty}^{\infty} \delta(t'-nT_s)\,p(t-t')\,dt'$

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

$p(0)=1$, $p(kT_s)=0 \rightarrow x_R(nT_s) = \sum_{n=-\infty}^{\infty} x(kT_s)\,p((n-k)T_s)$, $x(kT_s) = x_R(nT_s)$ when $p(0)=1$

$x_R(nT_s) = 0$ when $p(kT_s)=0$ because $(p(n-k)T_s) = 0$. Thus, $x_R(nT_s) = x(nT_s)$ when $p(0)=1$ and $p(kT_s)=0$ $\forall_k \in Z^+$

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

c) All of them are consisten as they satisfy $p(0)=1$ and $p(kT_s)=0$

Part 5

* Zero-Order Hold interpolation recovery looks almost exactly same as the original signal

* Likewise in the first one, in linear interpolation recovery looks almost exactly same as the original signal.

* Ideal Band Limited interpolation is not as efficient as expected. There are some oscillations after jumps.

* The most efficient one is zero-order hold interpolation. Also, it can be said that zero-order hold interpolation is the best interpolation compared to others. When Ts is increased, recovery process seems like becoming less efficient compared to others. This might be because of we get less sample points when Ts increases.

Part 6

* Ideal Band Limited interpolation is the most efficient recovery in this part. Original signal can obtained by this interpolation. In the other ones, it is not efficient as in ideal band limited interpolation because there is loss of data and some incorrect data occurs.

* When $T_s = 0.01$, where $0.1 \leq T_s \leq 0.2$, the most efficient recovery is done. The sketch is similar to the graph of the sum of weighted sinc function which is repeats itself periodically.

* If there is more data, or, more sample points the recovery process would be more efficient and the signals obtained (recovery signals) would be much closer to the original signal.

Atahan İyiekici
21801985
EEE321-001
Lab Report 5

EEE321 Lab Report 5

Part 3

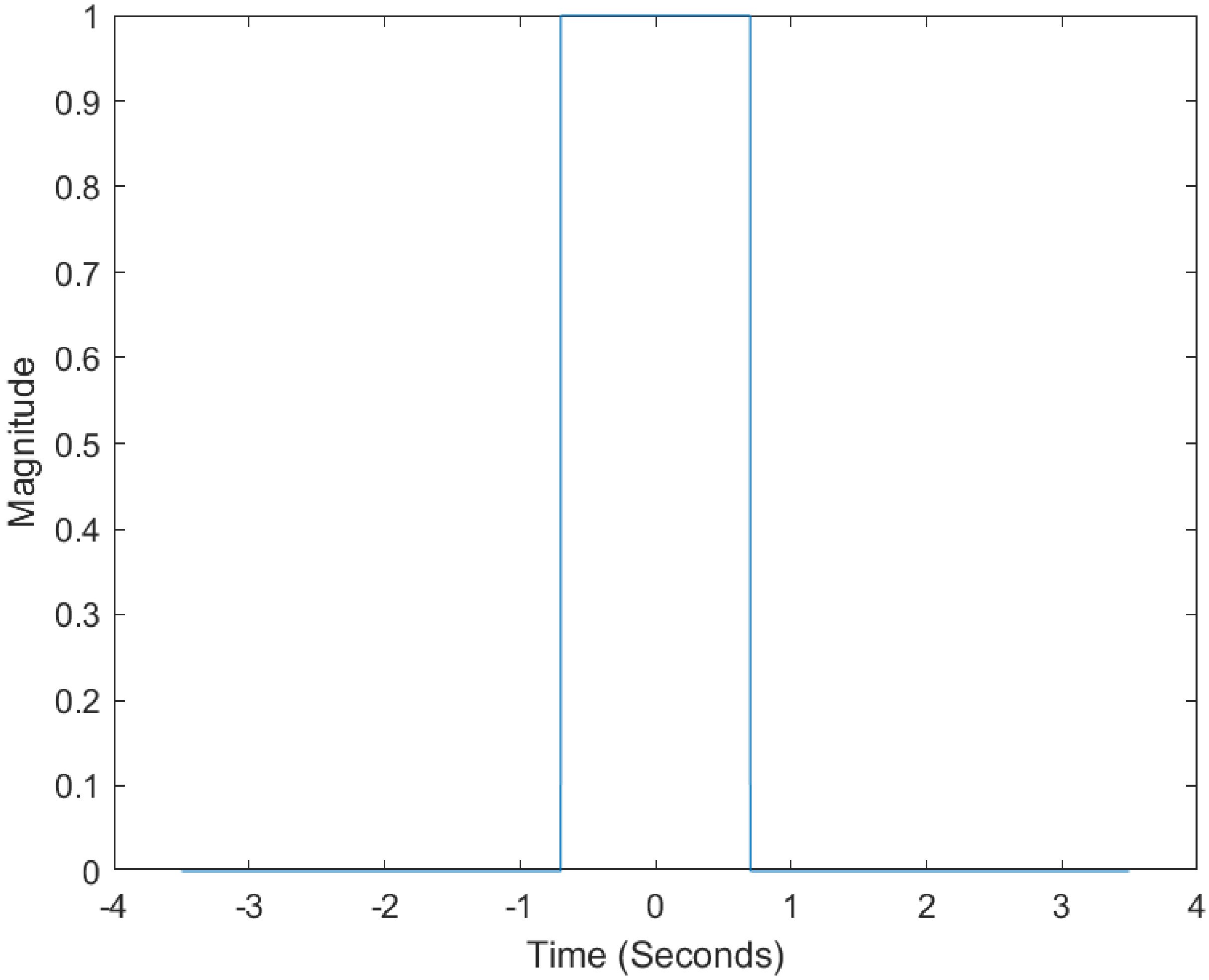


Figure 1: Plot of p_rect function

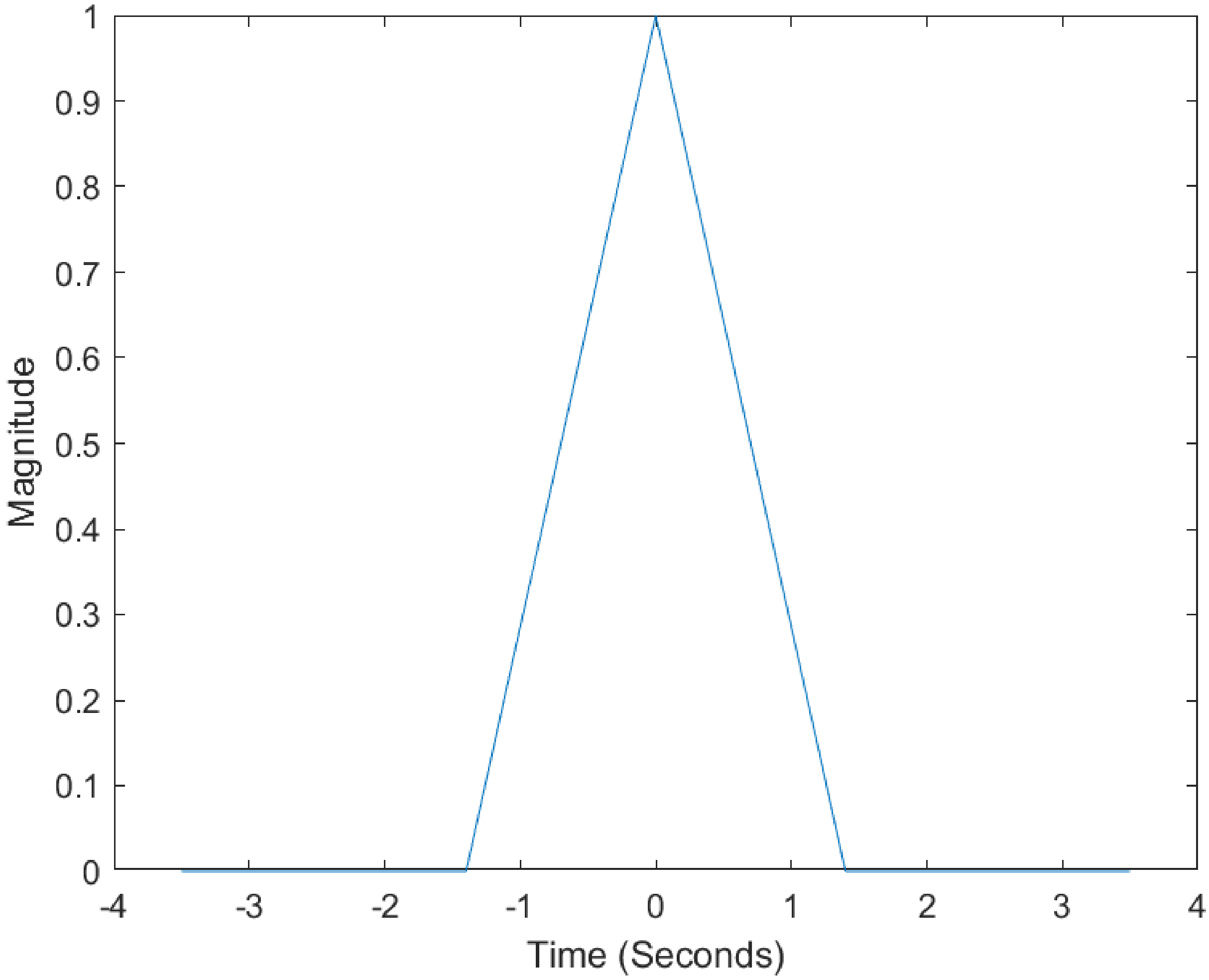


Figure 2: Plot of p_tri function

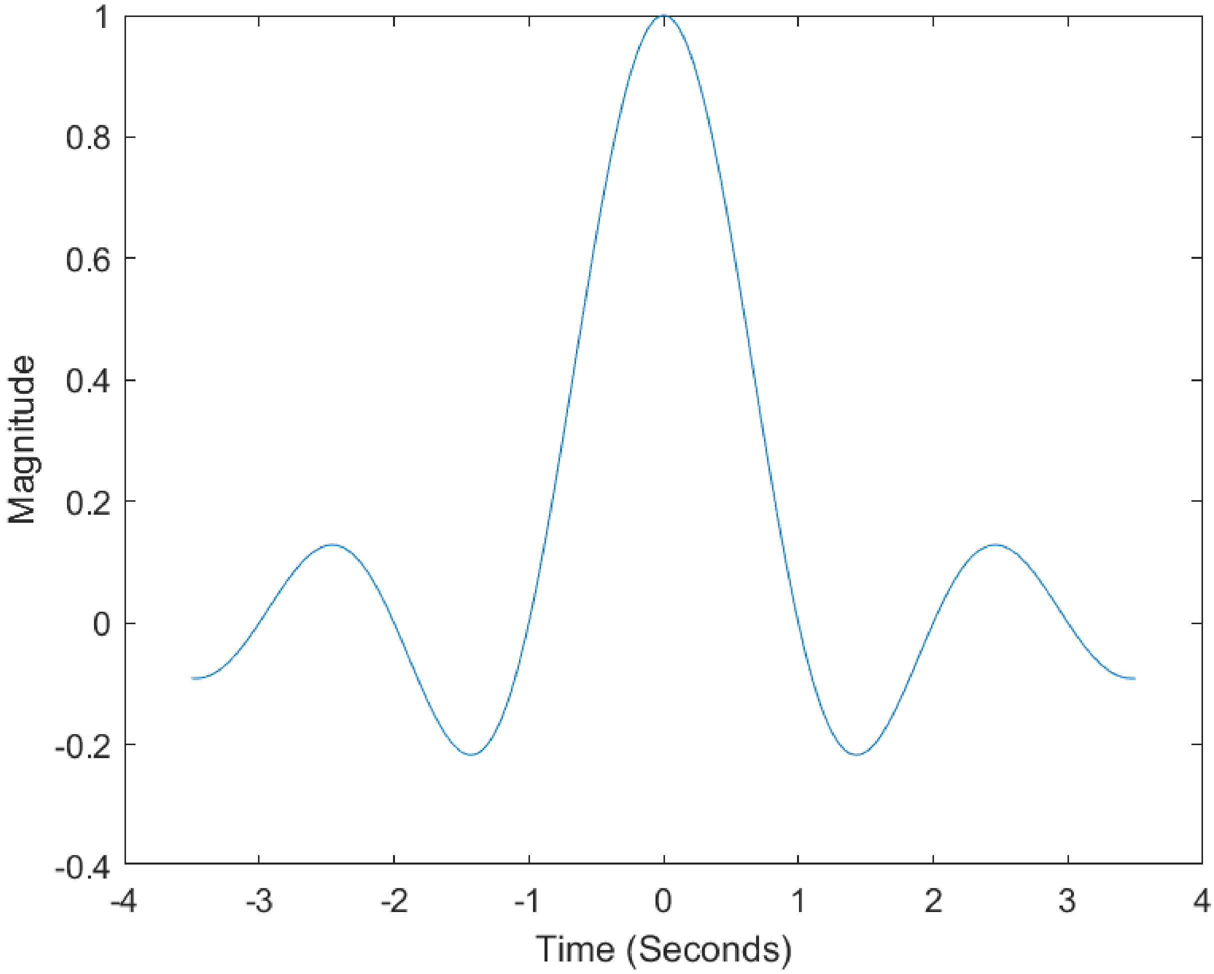


Figure 3: Plot of p_sinc function

Atahan İyiekici
21801985
EEE321-001
Lab Report 5

Part 5

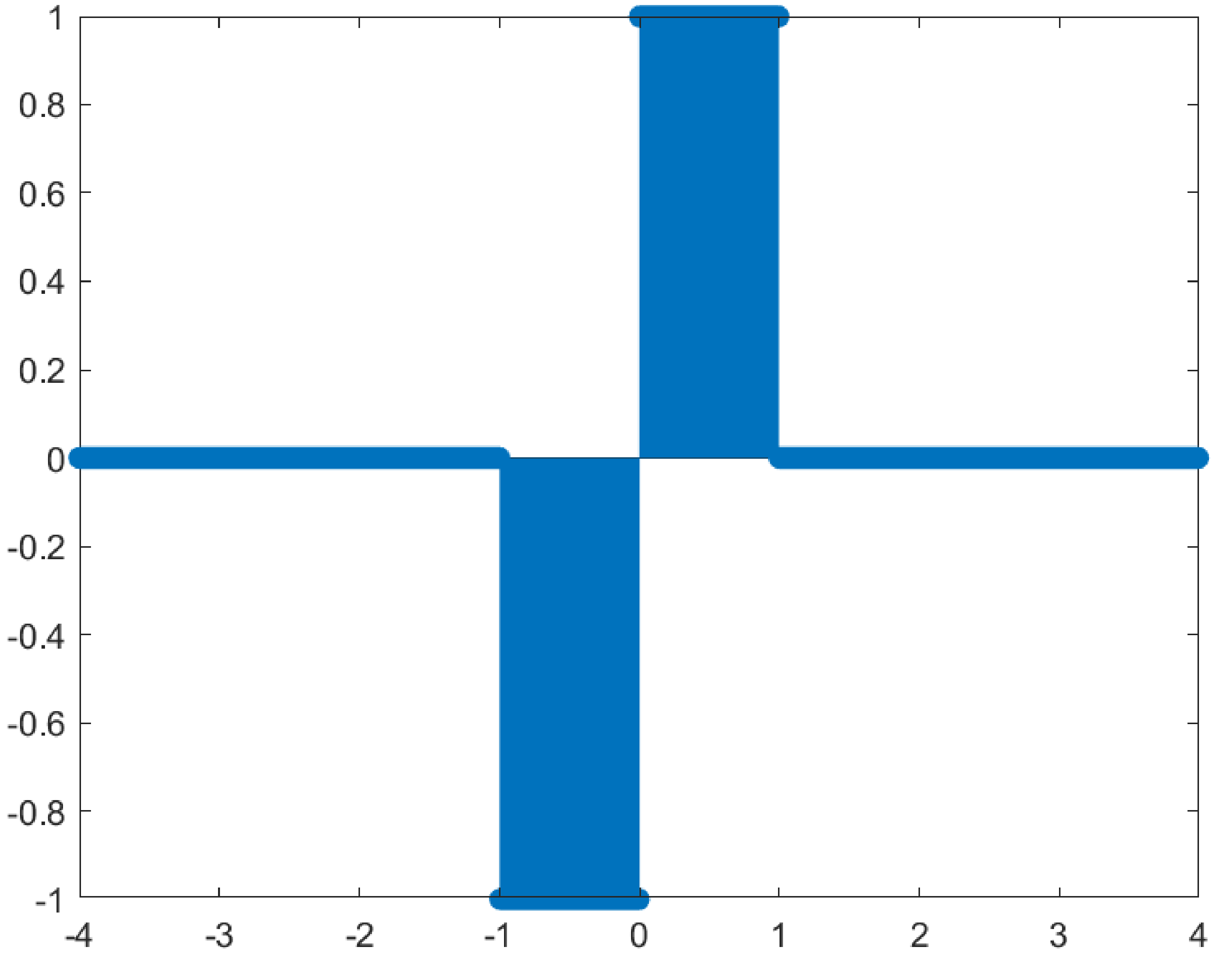


Figure 4: Plot of $g_R(t)$

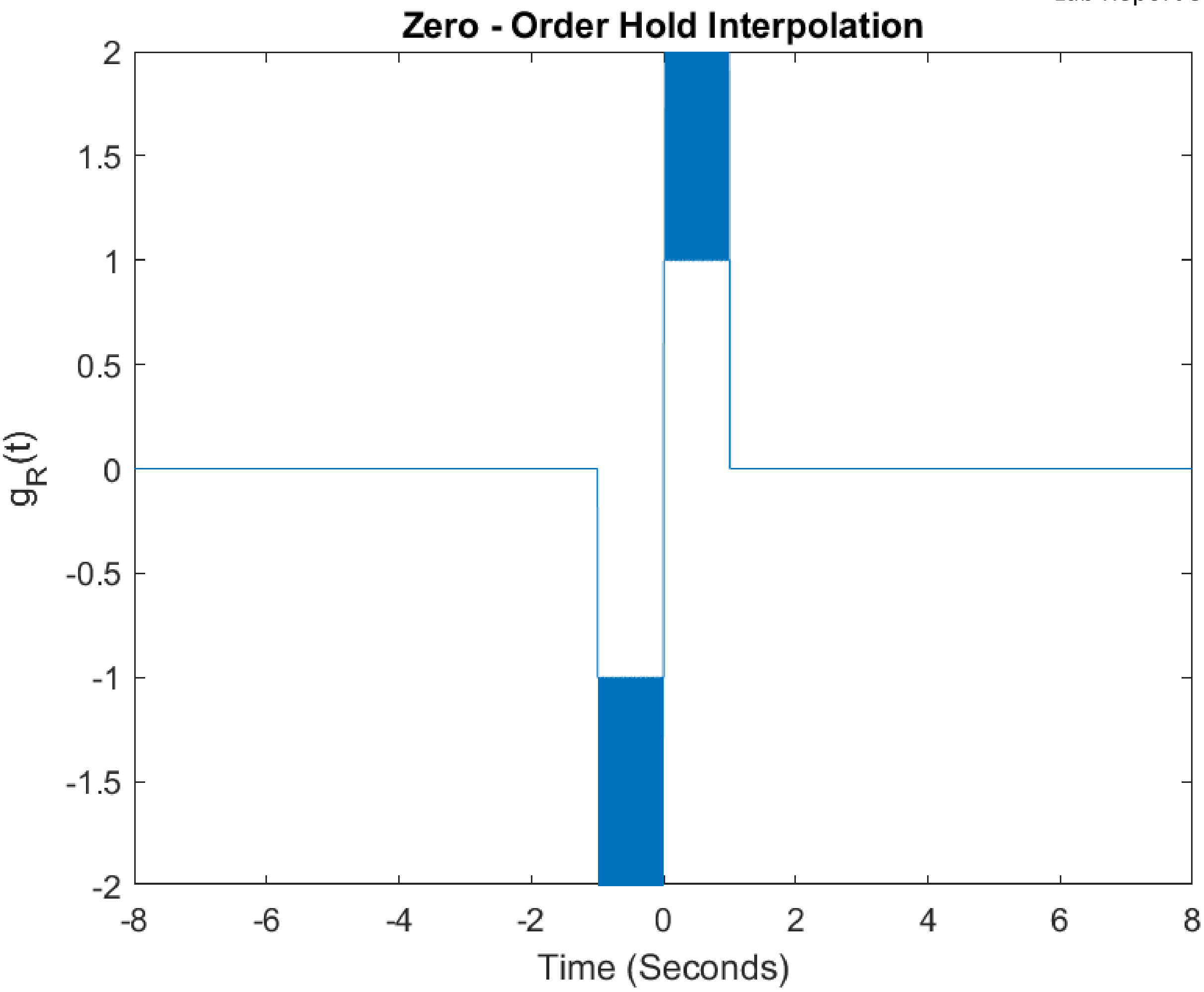


Figure 5: Zero-Order Hold Interpolation plot of $g_R(t)$

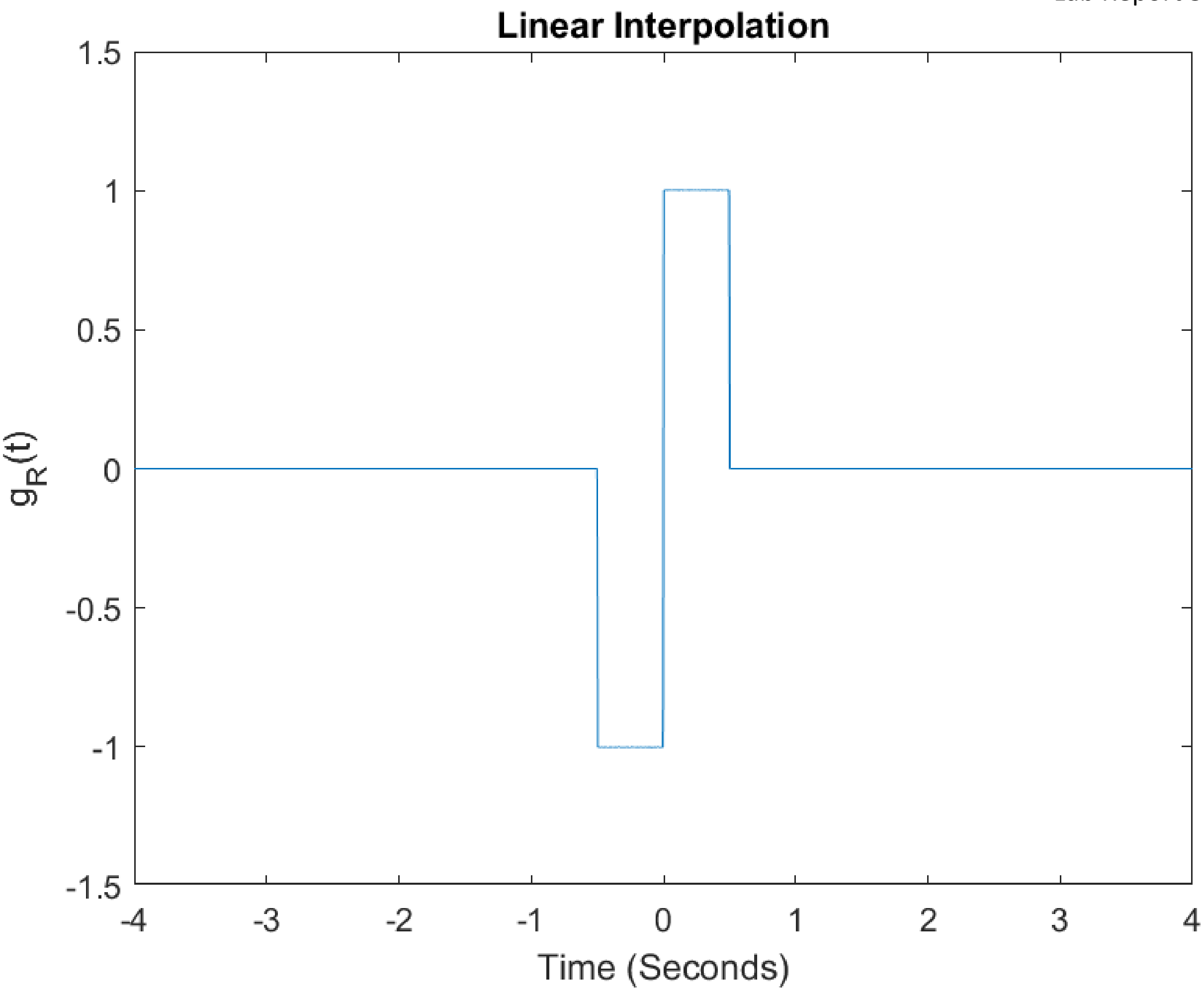


Figure 6: Linear Interpolation plot of $g_R(t)$

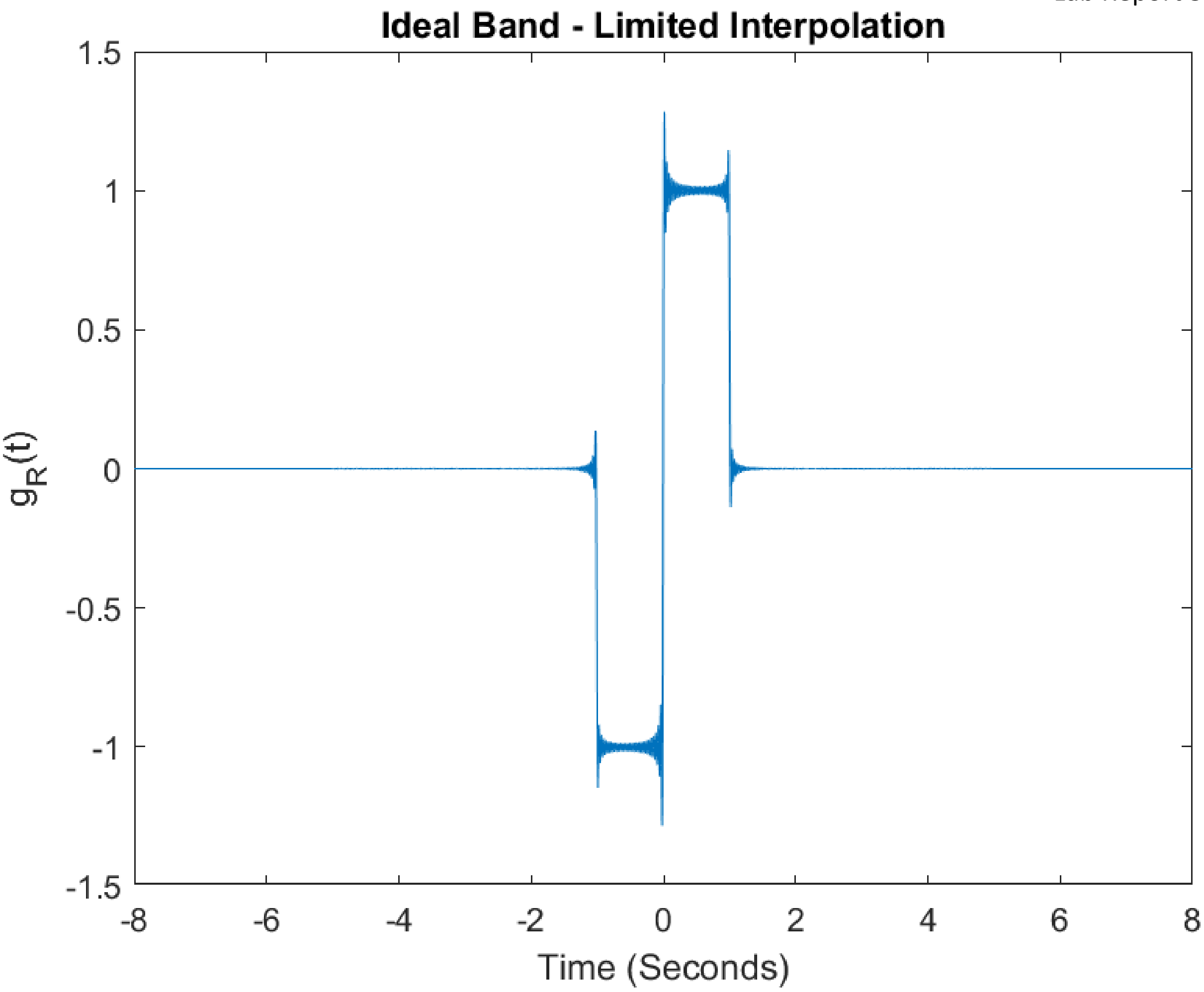


Figure 7: Ideal band – Limited Interpolation plot of $g_R(t)$

Part 6

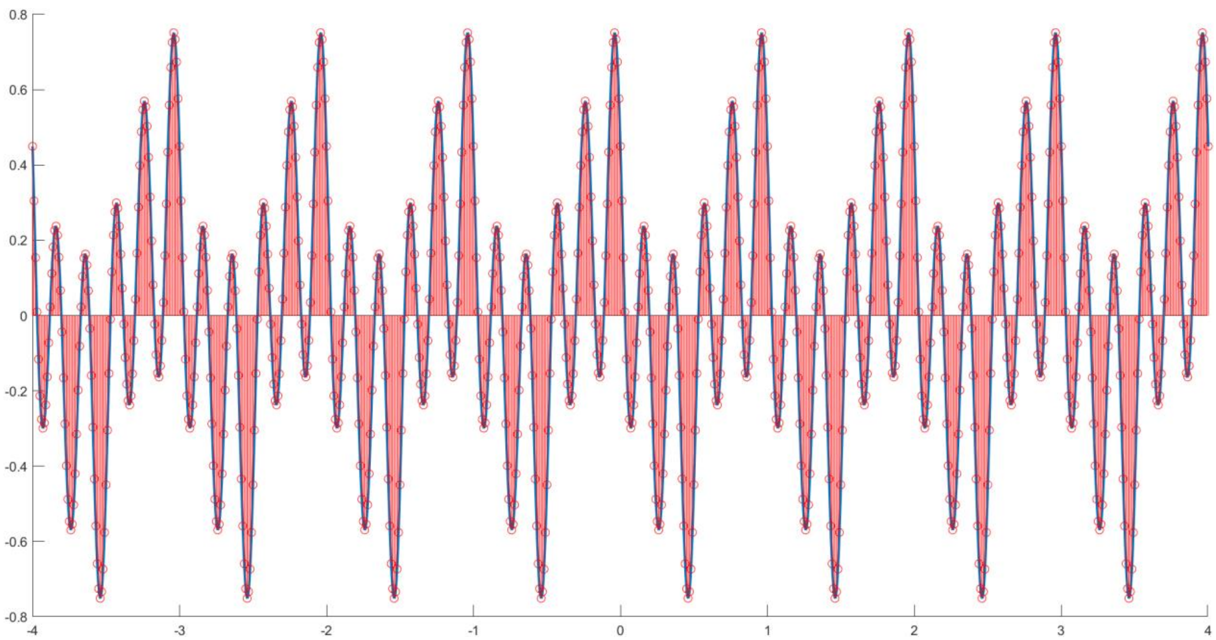


Figure 8: Plot of x(t) and $x_{sampled}(t)$ in The Same Graph when Ts = 0.005(D6 + 1)

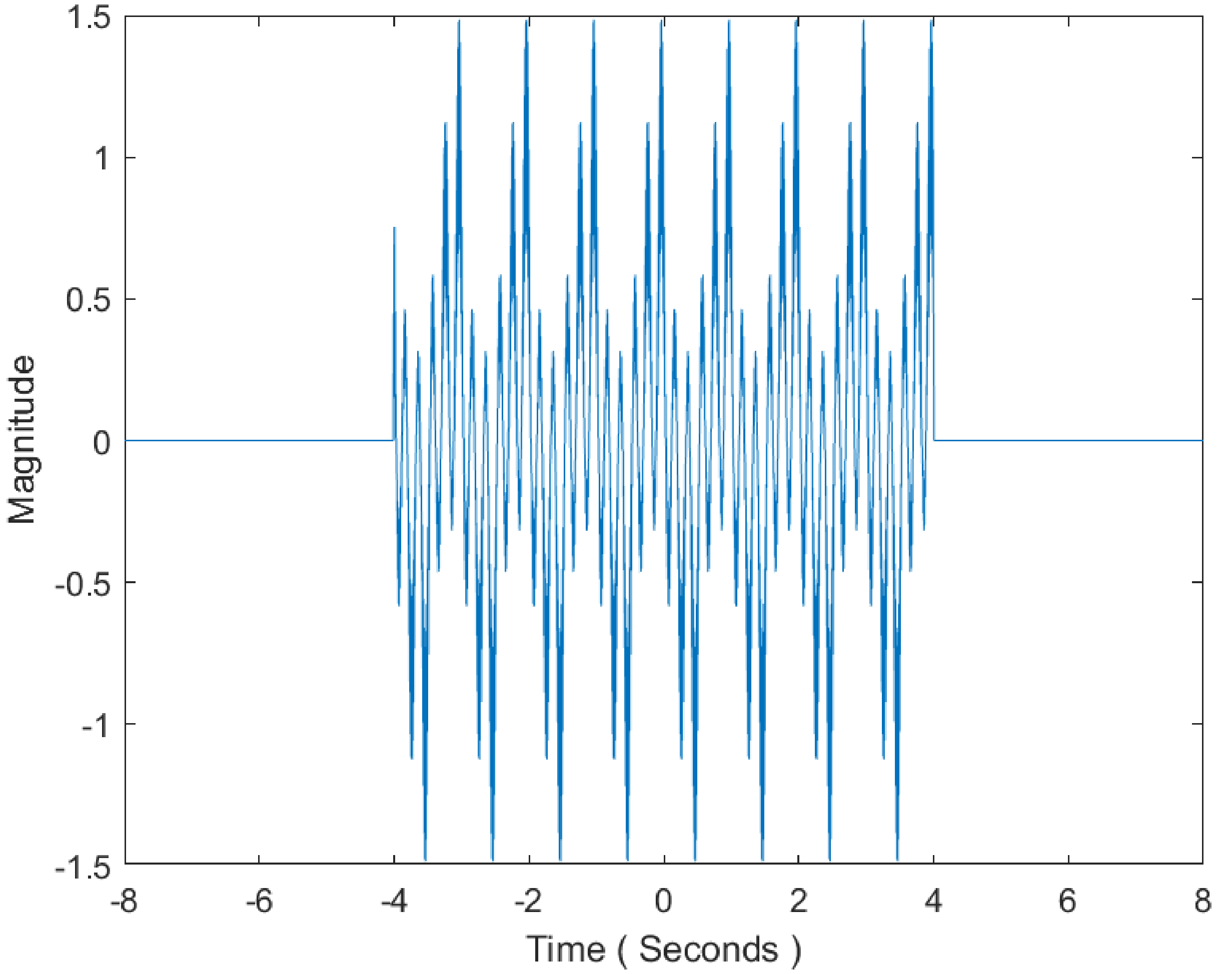


Figure 9: Plot of $x_R(t)$ After Using Zero - Order Hold Interpolation when Ts = 0.005(D6 + 1)

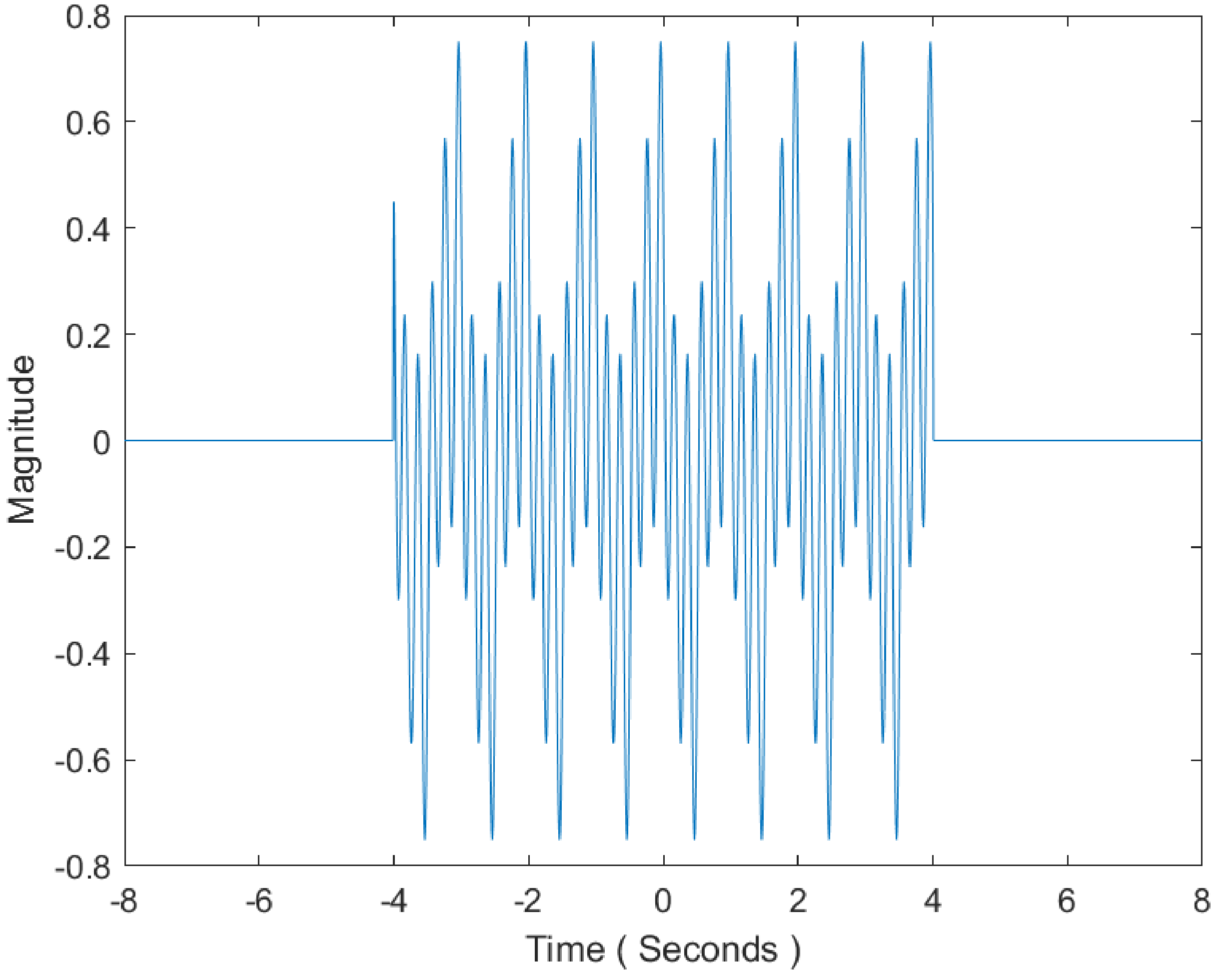


Figure 10: Plot of $x_R(t)$ After Using Linear Interpolation when Ts = 0.005(D6 + 1)

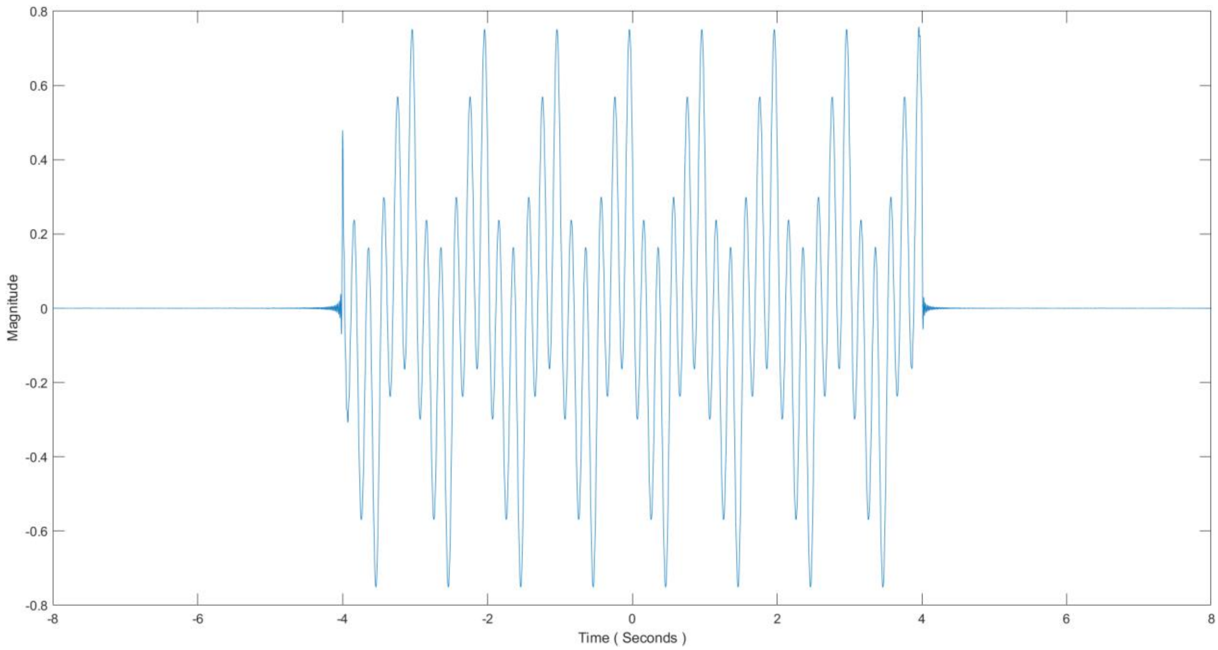


Figure 11: Plot of $x_R(t)$ After Using Ideal Band - Limited Interpolation when Ts = 0.005(D6 + 1)

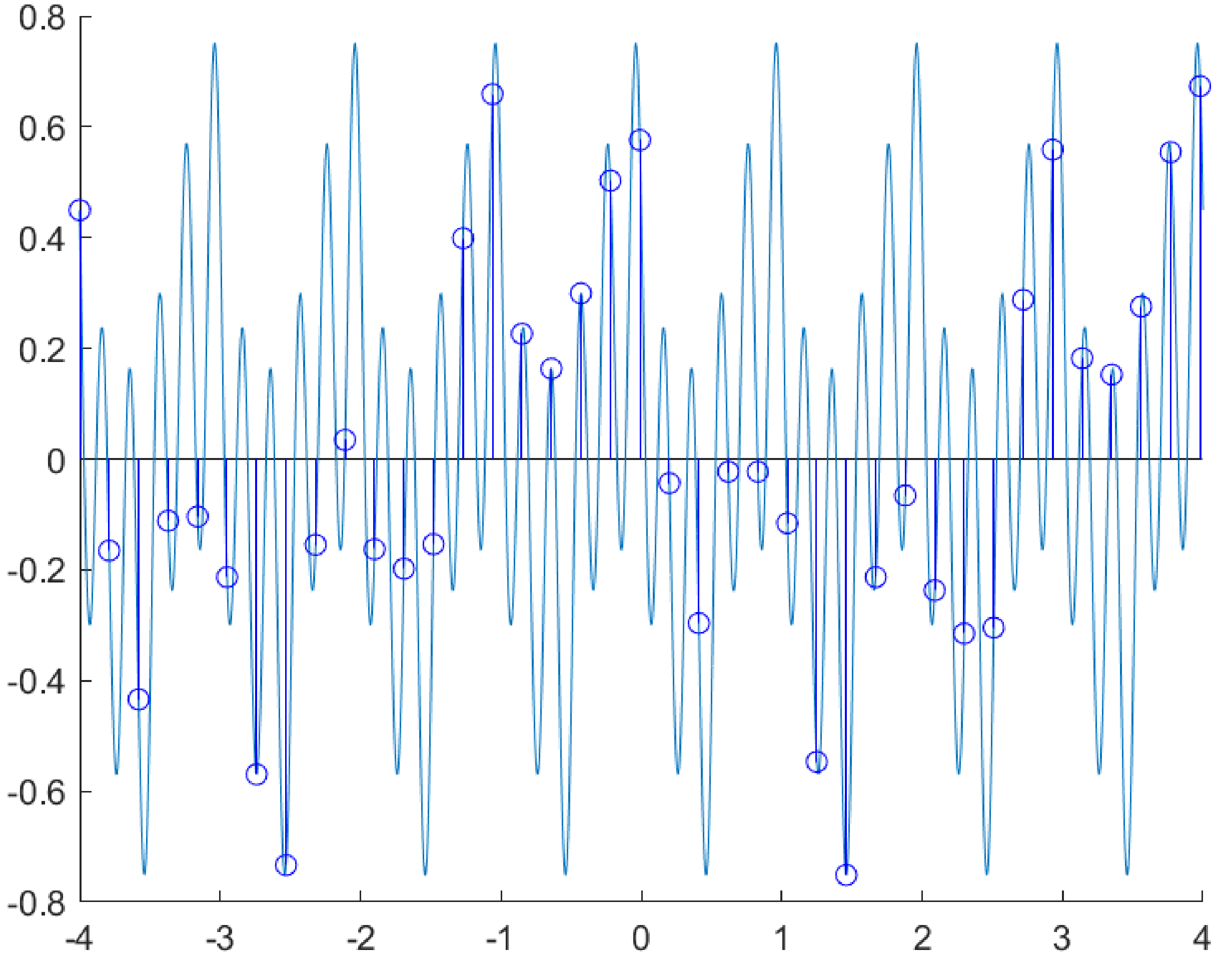


Figure 12: Plot of x(t) and $x_{sampled}(t)$ in The Same Graph when Ts = 0.2+0.01*D6.

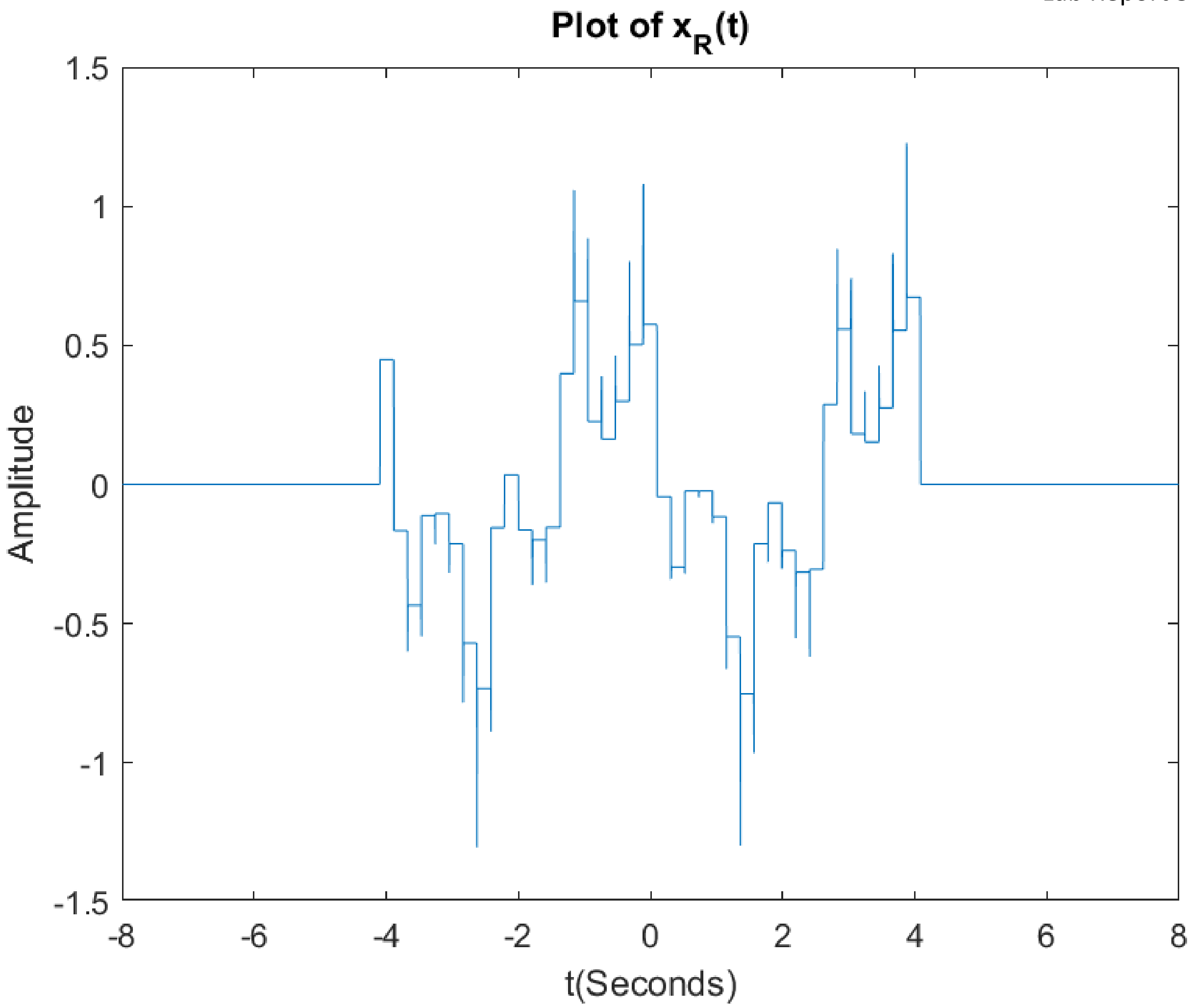


Figure 13: Plot of $x_R(t)$ After Using Zero - Order Hold Interpolation when Ts = 0.2+0.01*D6.

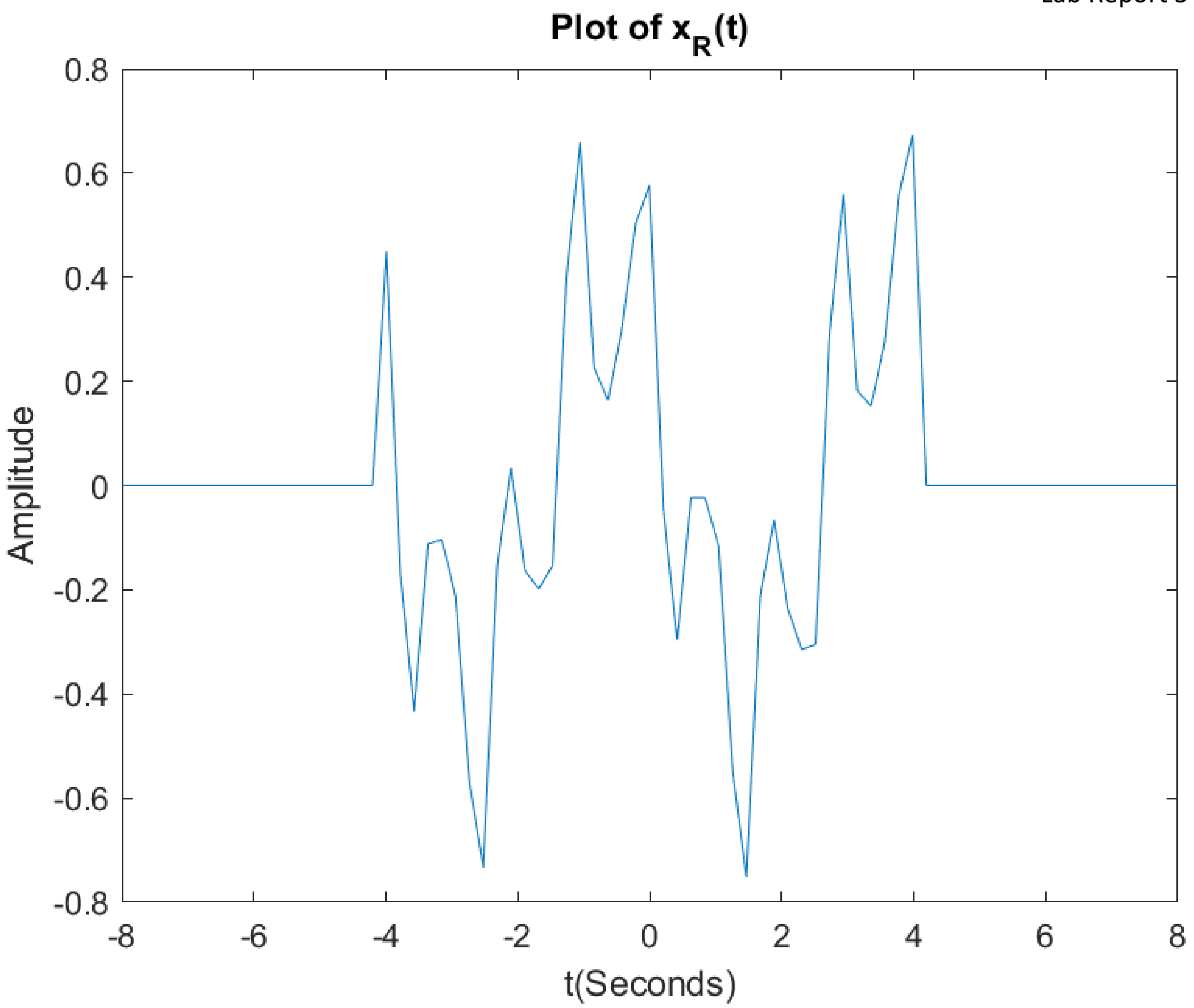


Figure 14: Plot of $x_R(t)$ After Using Linear Interpolation when Ts = 0.2+0.01*D6.

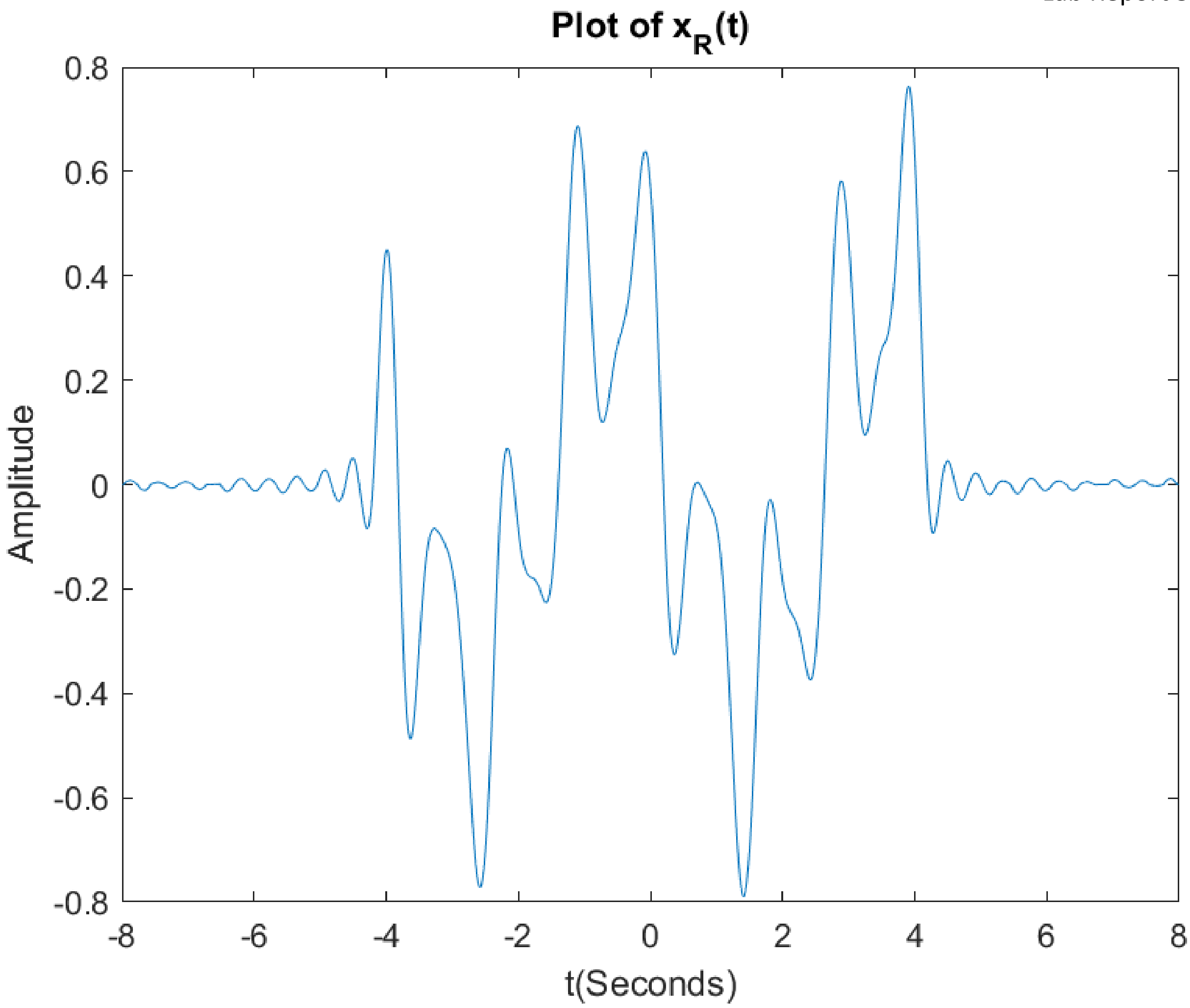


Figure 15: Plot of $x_R(t)$ After Using Ideal Band when Ts = 0.2+0.01*D6.

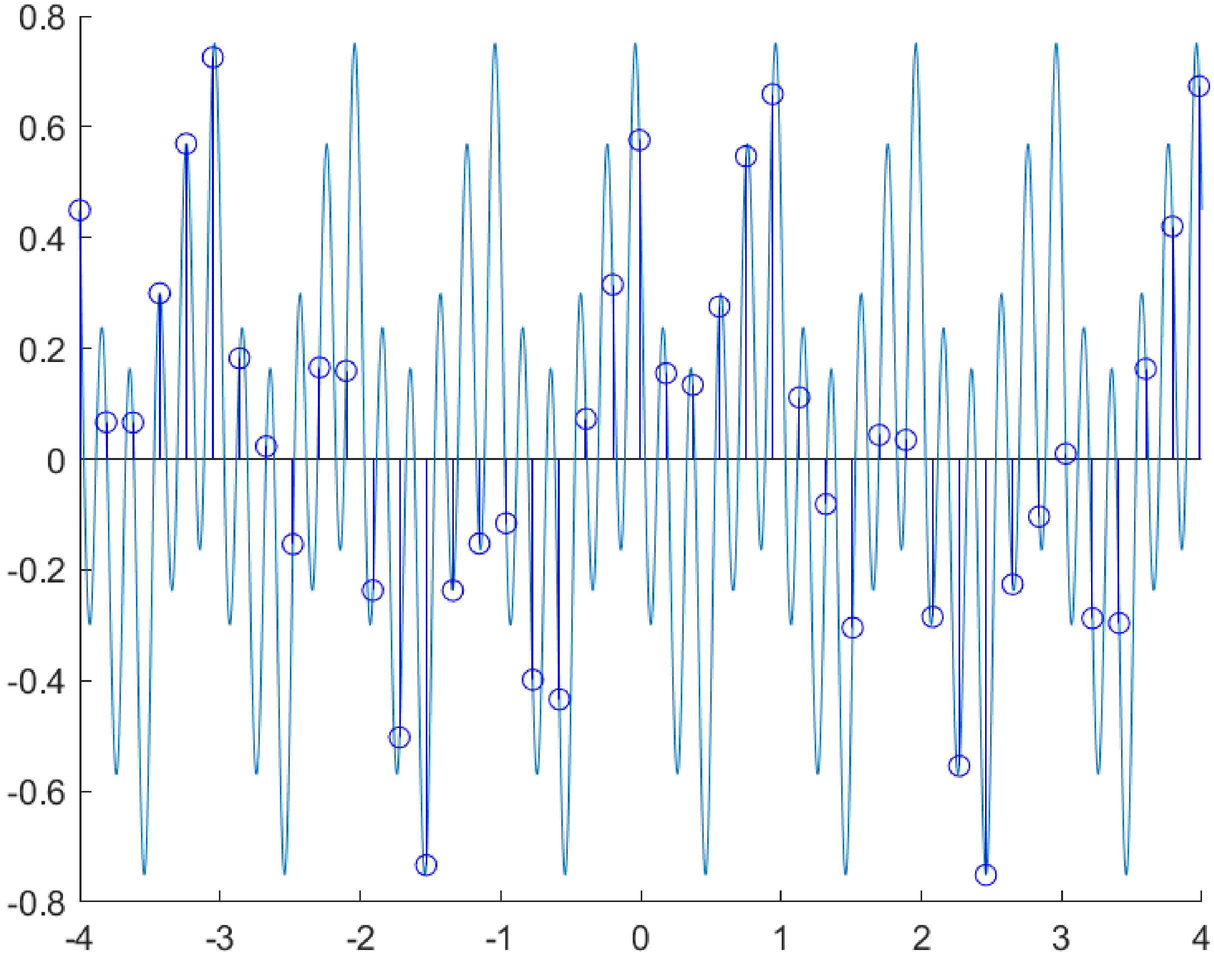


Figure 16: Plot of x(t) and $x_{sampled}(t)$ in The Same Graph when Ts = 0.18+0.005*(D6 +1).

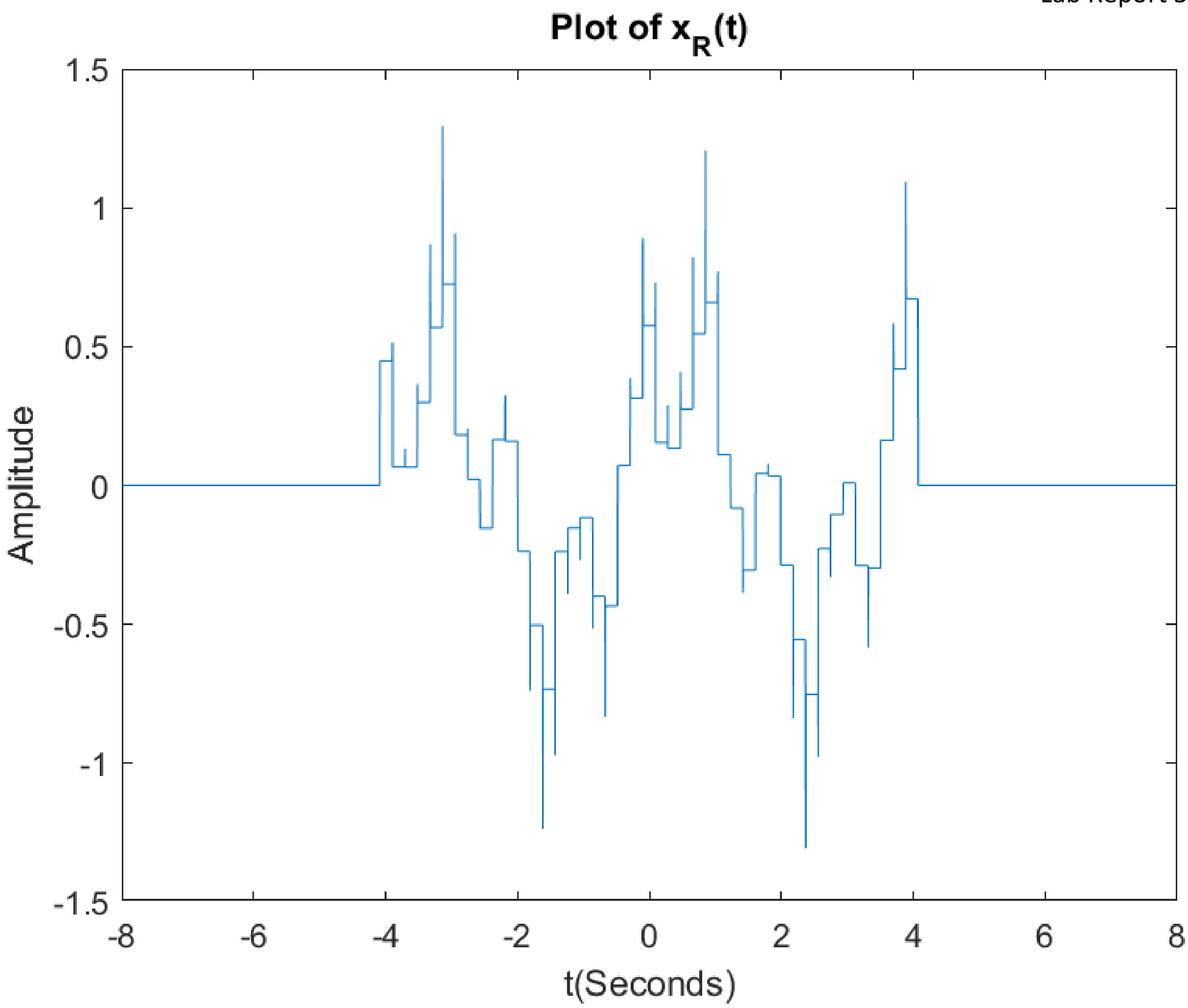


Figure 17: Plot of $x_R(t)$ After Using Zero - Order Hold Interpolation when Ts = 0.18+0.005*(D6 +1).

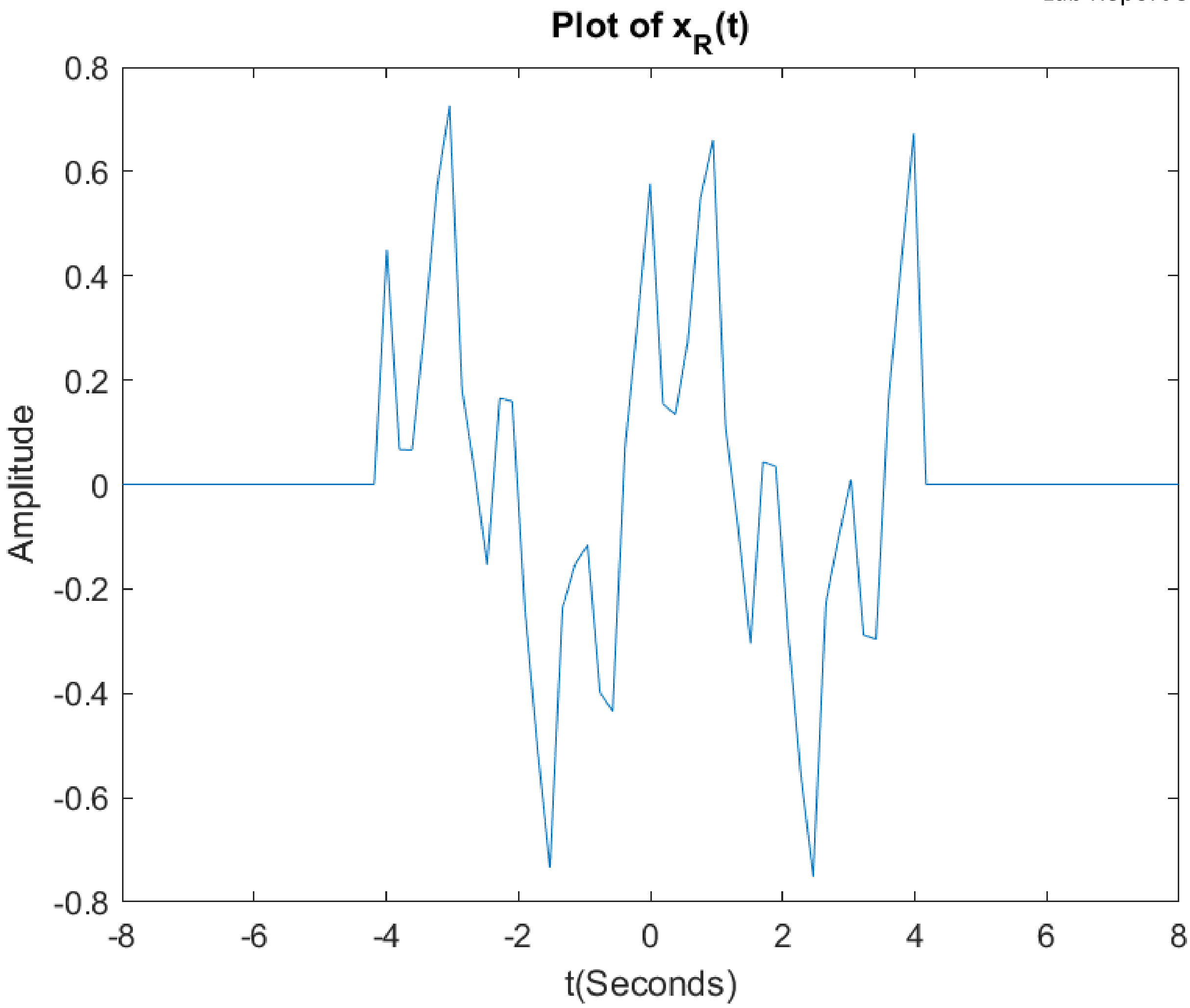


Figure 18: Plot of $x_R(t)$ After Using Linear Interpolation when Ts = 0.18+0.005*(D6 +1).

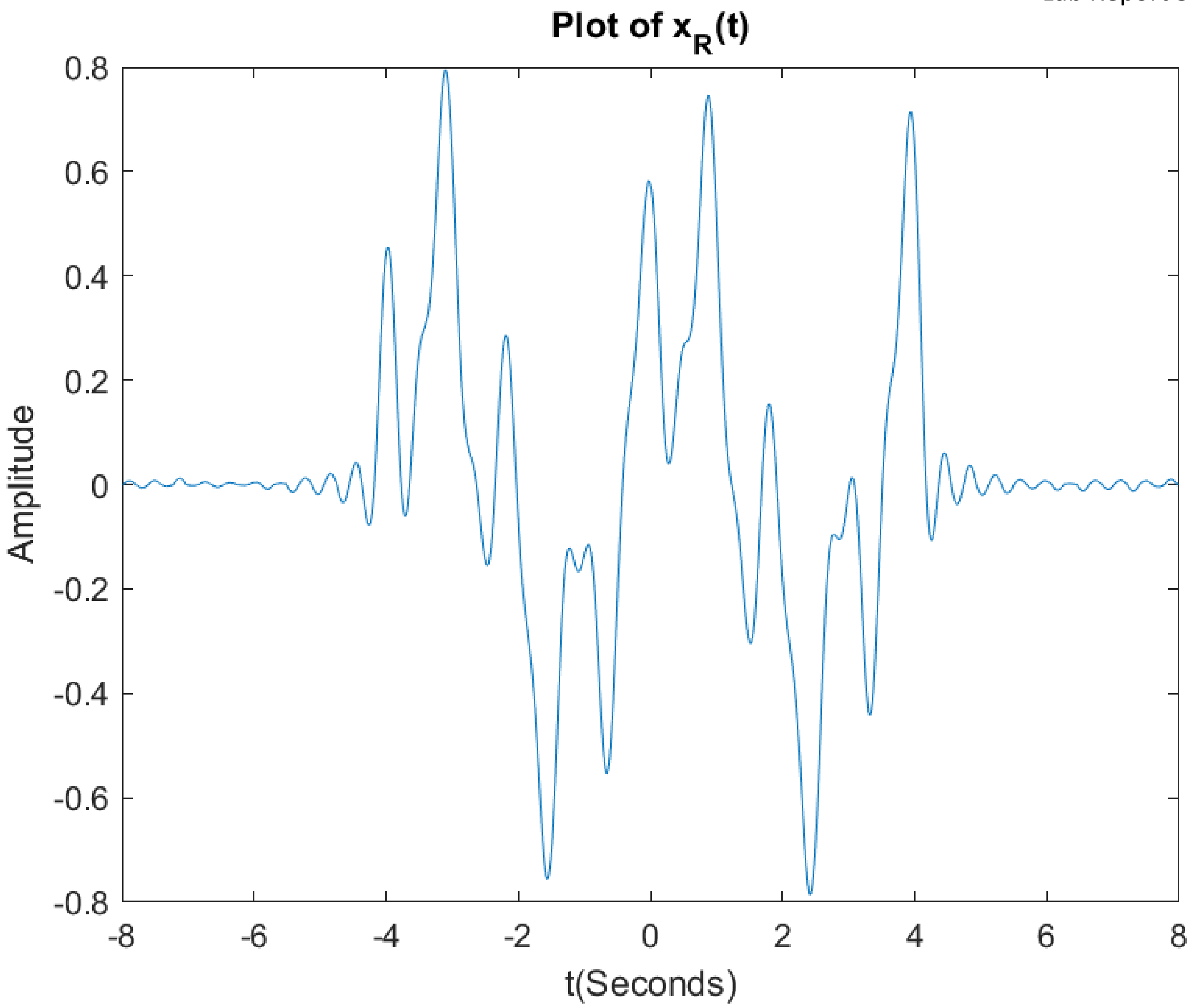


Figure 19: Plot of $x_R(t)$ After Using Ideal Band - Limited Interpolation when Ts = 0.18+0.005*(D6 +1).

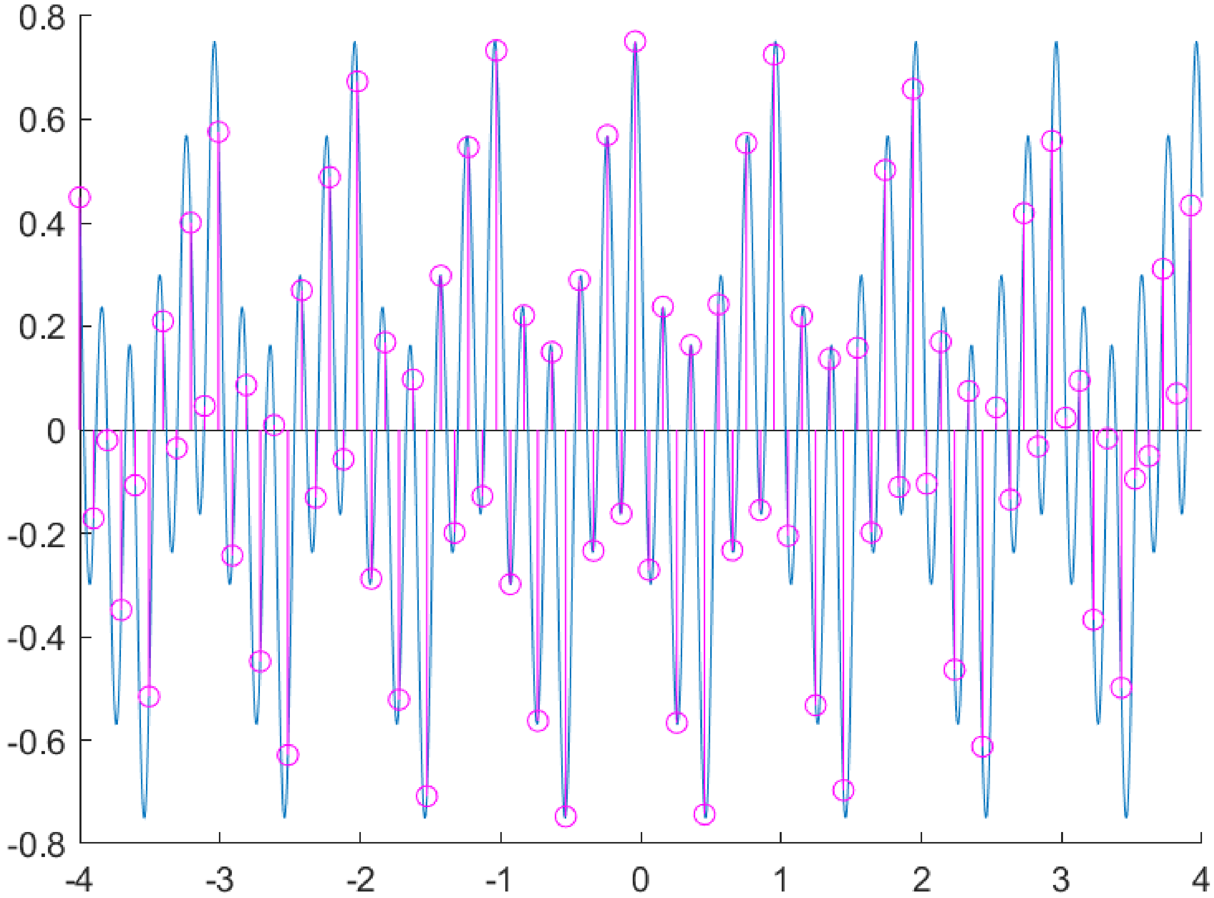


Figure 20: Plot of x(t) and $x_{sampled}(t)$ in The Same Graph when Ts = 0.099.

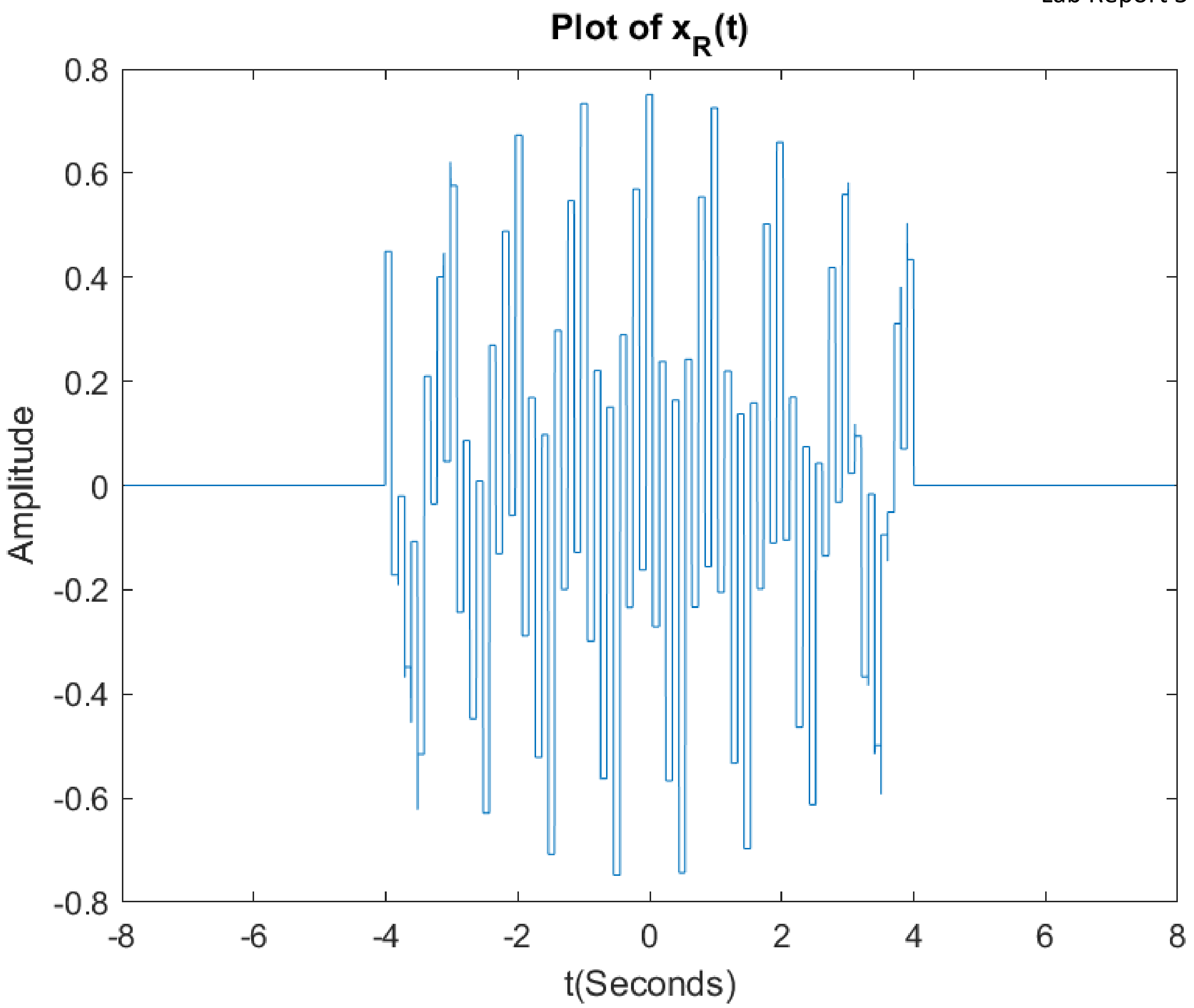


Figure 21: Plot of $x_R(t)$ After Using Zero - Order Hold Interpolation when Ts = 0.099.

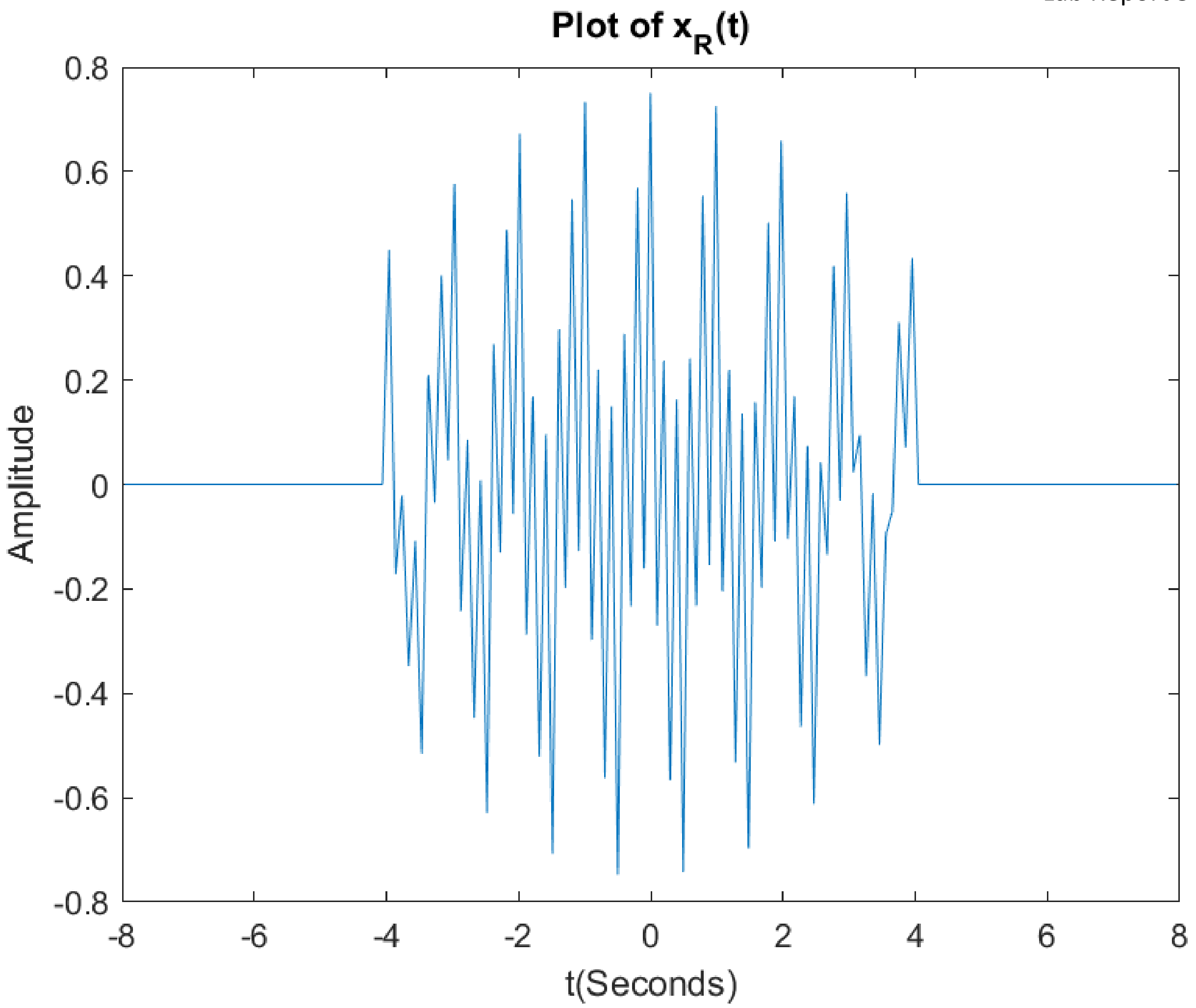


Figure 22: Plot of $x_R(t)$ After Using Linear Interpolation when Ts = 0.099.

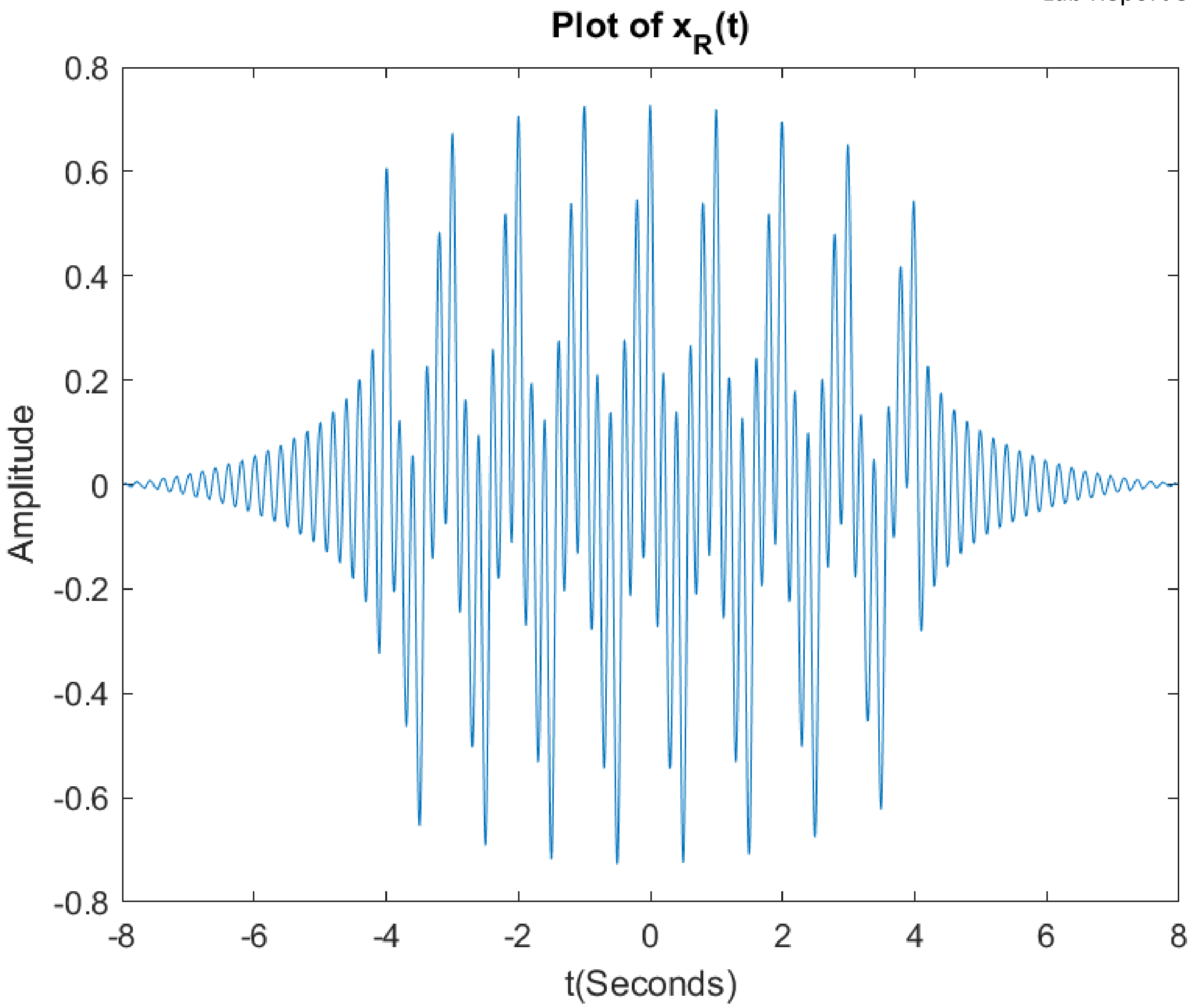


Figure 23: Plot of $x_R(t)$ After Using Ideal Band - Limited Interpolation when Ts = 0.099.

Atahan İyiekici
21801985
EEE321-001
Lab Report 5

Matlab Codes

```
%% Part 3 %%
dur = mod(21801985,9);
p_rect = generateInterp(0, dur/5, dur);
%figure; plot(t, p); xlabel('Time (Seconds)'); ylabel('Magnitude');

p_tri = generateInterp(1, dur/5, dur);
%figure; plot(t, p); xlabel('Time (Seconds)'); ylabel('Magnitude');

p_sinc = generateInterp(2, dur/5, dur);
%figure; plot(t, p); xlabel('Time (Seconds)'); ylabel('Magnitude');

%% Part 5 %%
Ts = 1 / (15 * (3 + (8 - 3) .* rand(1, 1)));

t_sampling = -4 : Ts : 4;

g = zeros(1, length(t_sampling));
g(find(t_sampling == -1) : find(t_sampling == 0) - 1) = 1;
g(find(t_sampling == 0) + 1 : find(t_sampling == 1)) = -2;
%stem(t_sampling, g);

g2 = zeros(1, length(t_sampling));
g2(t_sampling > -1 & t_sampling < 0) = -1;
g2(t_sampling > 0 & t_sampling < 1) = 1;
%stem(t_sampling, g2);

%gR_w_rect = DtoA(0, Ts, 8, g2); xlabel('Time (Seconds)'); ylabel('g_R(t)');
title('Zero - Order Hold Interpolation');
%gR_w_tri = DtoA(1, Ts, 8, g2); plot(linspace(-4, 4, length(gR_w_tri)), gR_w_tri);
xlabel('Time (Seconds)'); ylabel('g_R(t)'); title('Linear Interpolation');
%gR_w_sinc = DtoA(2, Ts, 8, g2); xlabel('Time (Seconds)'); ylabel('g_R(t)');
title('Ideal Band - Limited Interpolation');

%% Part 6 %%
D6 = rem(21801985,6);
% Ts = 0.005 * (D6 + 1);
%Ts = 0.2 + 0.01 * D6;
%Ts = 0.18 + 0.005 * (D6 + 1);
%Ts = 0.099;
t = -4 : Ts/1000 : 4; sam_t = -4 : Ts : 4;
x_t = (0.3) * cos(2 * pi * t + pi / 4) + (0.1) * cos(6 * pi * t + pi / 8) + (0.4)
* cos(10 * pi * t + 1.2);
x_sam = (0.3) * cos(2 * pi * sam_t + pi / 4) + (0.1) * cos(6 * pi * sam_t + pi /8)
+ (0.4) * cos(10 * pi * sam_t + 1.2);
hold on; plot(t, x_t); stem(sam_t, x_sam, 'Blue'); hold off;
figure; DtoA(0, Ts, 8, x_sam); title('Plot of x_R(t)'); xlabel('t(Seconds)');
ylabel('Amplitude');
figure; DtoA(1, Ts, 8, x_sam); title('Plot of x_R(t)'); xlabel('t(Seconds)');
ylabel('Amplitude');
figure; DtoA(2, Ts, 8, x_sam); title('Plot of x_R(t)'); xlabel('t(Seconds)');
ylabel('Amplitude');
```

Atahan İyiekici
21801985
EEE321-001
Lab Report 5

```
%% Functions %%

function xR = DtoA(type, Ts, dur, Xn)
     p = generateInterp(type, Ts, dur);
    len_p = length(p); len_x = length(Xn); len_r = floor(len_p * (1 + (len_x -1) *
Ts/dur));
    tot = floor(len_r / (len_x -1 + dur/Ts));
    xR = zeros(1, len_r);
    t = linspace(-(dur + (len_x - 1) * Ts) / 2, (dur + (len_x - 1) * Ts) /
2,len_r);
 for i = 0 : len_x - 1
 xR(tot * i + 1 : tot * i + len_p) = xR(tot * i + 1 : tot * i + len_p) + Xn(i + 1)
* p;
 end
 plot(t, xR);
end
function p = generateInterp(type,Ts,dur)
 t = (-dur/2 : Ts/1000 : dur/2 -Ts/1000);

 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;
 end
end
```