

---

# Niruyan Rakulan 214343438

## EECS 4214 Lab 6

### Table of Contents

Problem 1 .....	1
Problem 2 Noise .....	6
40 Times 10 SNR .....	10
40 Times 0 SNR .....	16
40 Times -40 SNR .....	22
40 Times -45 SNR .....	28
40 Times -50 SNR .....	34
40 Times -60 SNR .....	40
Error Freq. ....	46

## Problem 1

```
clc;
clear all;
close all;

%%1.1 Generate Waveform
%given sequence
test=[1 0 1 1 0 0 0 1];

%map to voltages
v_test=test*10-5;

%generate waveform
sample_time=10e-9;
t=0:sample_time:1e-3*length(test)-sample_time;
i=1e-3;
n=1;
for j=1:length(t)
    if t(j)<=n*i
        s(j)=v_test(n);
    else
        n=n+1;
        s(j)=v_test(n);
    end
end

plot(t,s);
title('s(t)');
xlabel('Times(s)');
ylabel('Voltage(V)');
```

```
axis([min(t) max(t) -10 10]);

%%1.2
r=0.8*s;
figure;
plot(t,r);
title('r(t)');
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) -10 10]);

%optimum detection value= (a1+a2)/2=0
y=0;

%Correlator Detector
%times to sample
for j=1:length(test)
    s_time(j)=t((2*j-1)*(length(t)/length(test)/2));
end

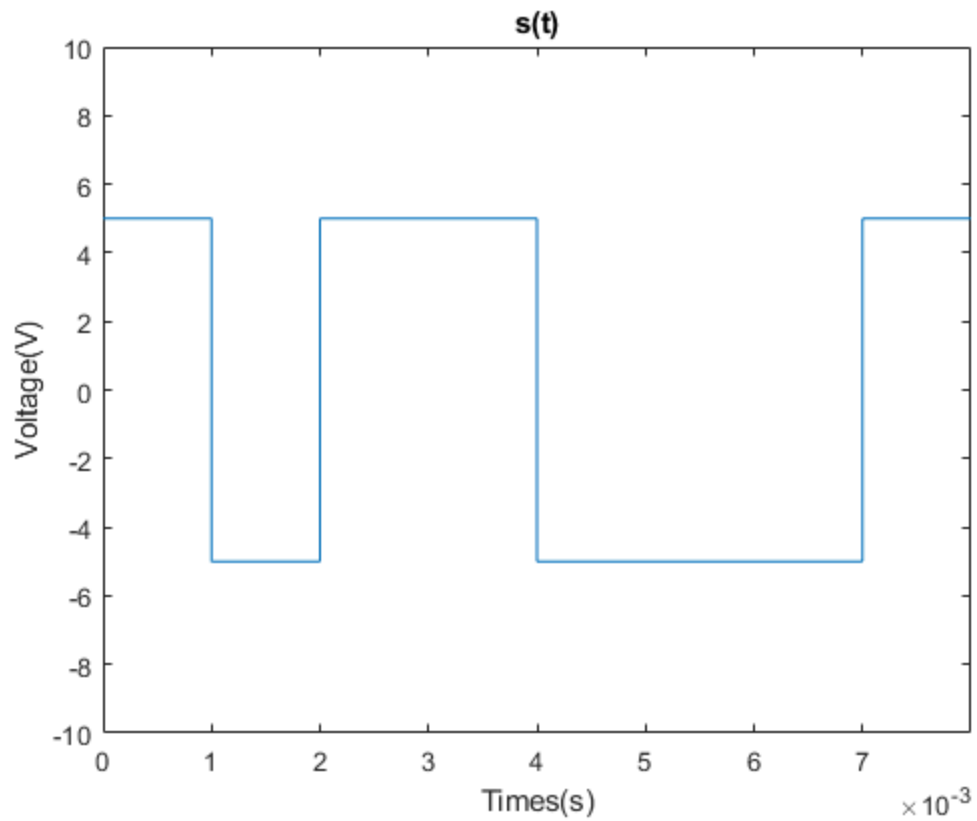
for j=1:length(s_time)
    %integral time
    x=(s_time(j)-s_time(1)):sample_time:s_time(j)+s_time(1);
    %z1
    z1(j)=trapz(x,5.*r((j-1)*length(x)+1:j*length(x)));
    %z2
    z2(j)=trapz(x,-5.*r((j-1)*length(x)+1:j*length(x)));
    %z
    z(j)=z1(j)-z2(j);
    %decision
    if z(j)>y
        sample(j)=1;
    else
        sample(j)=0;
    end
end

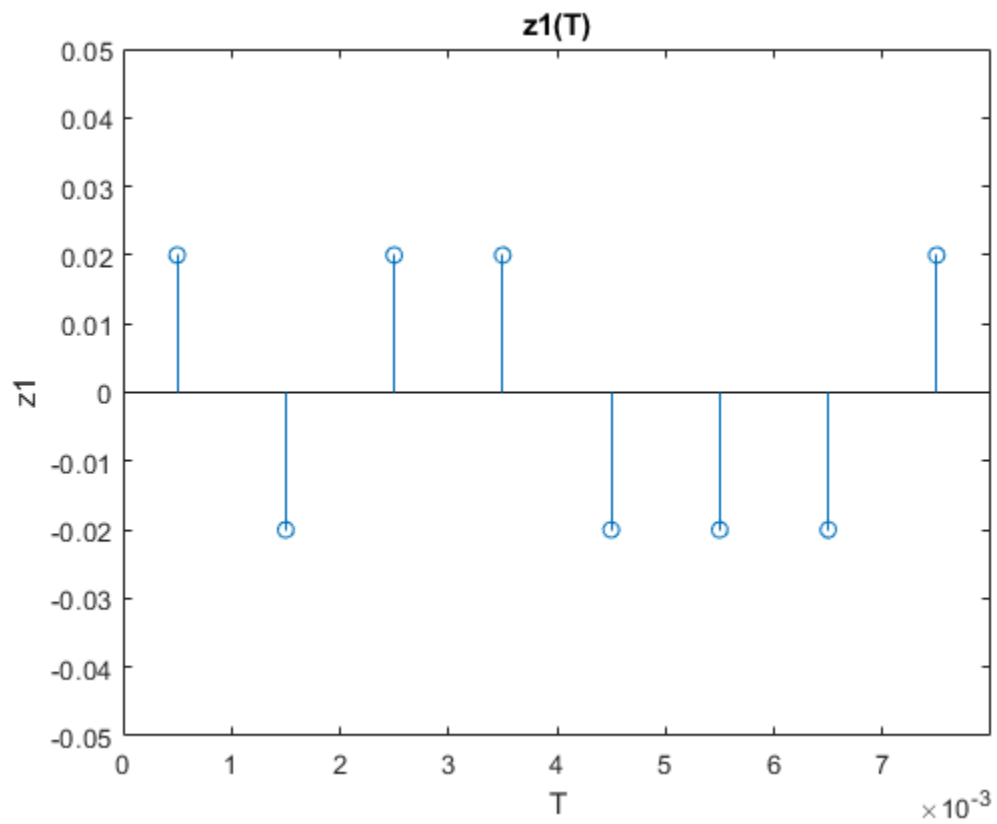
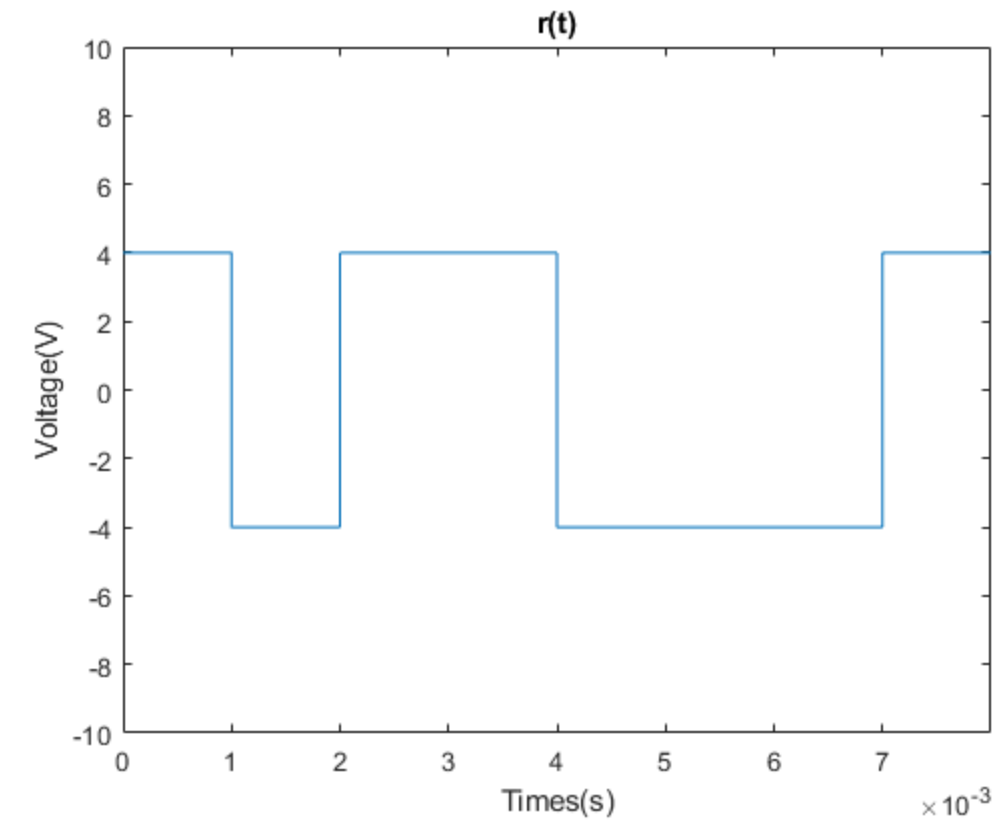
%%1.3
figure;
stem(s_time,z1);
title('z1(T)');
xlabel('T');
ylabel('z1');
axis([min(t) max(t) -0.05 0.05]);

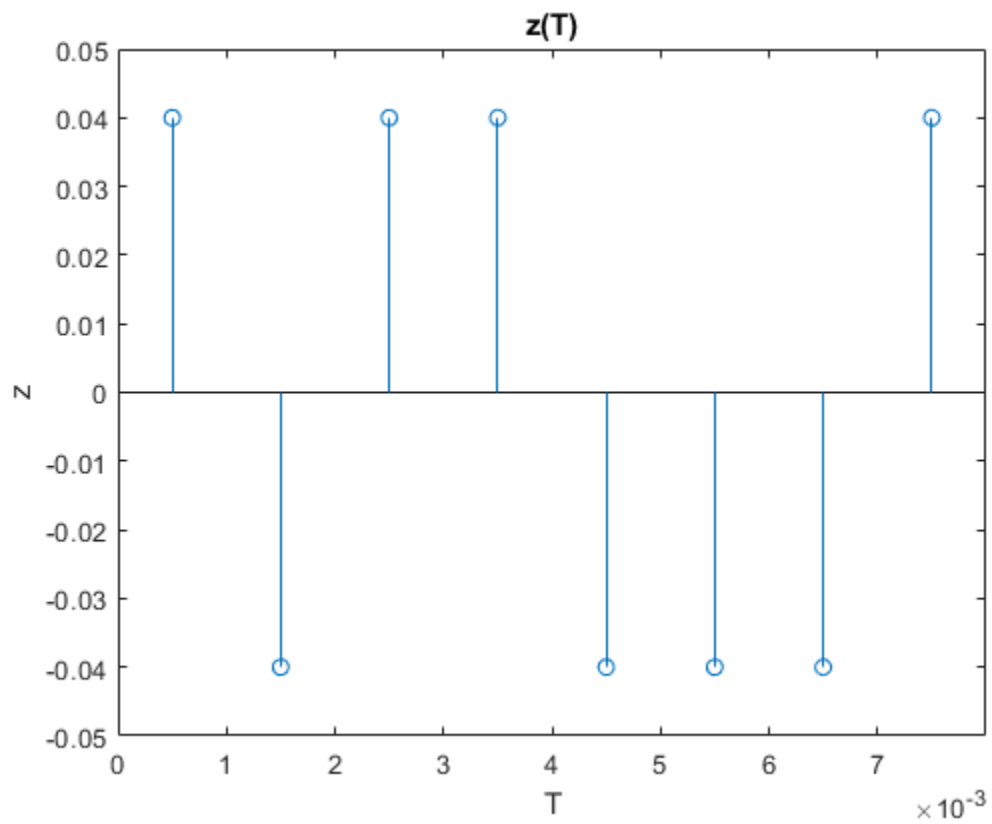
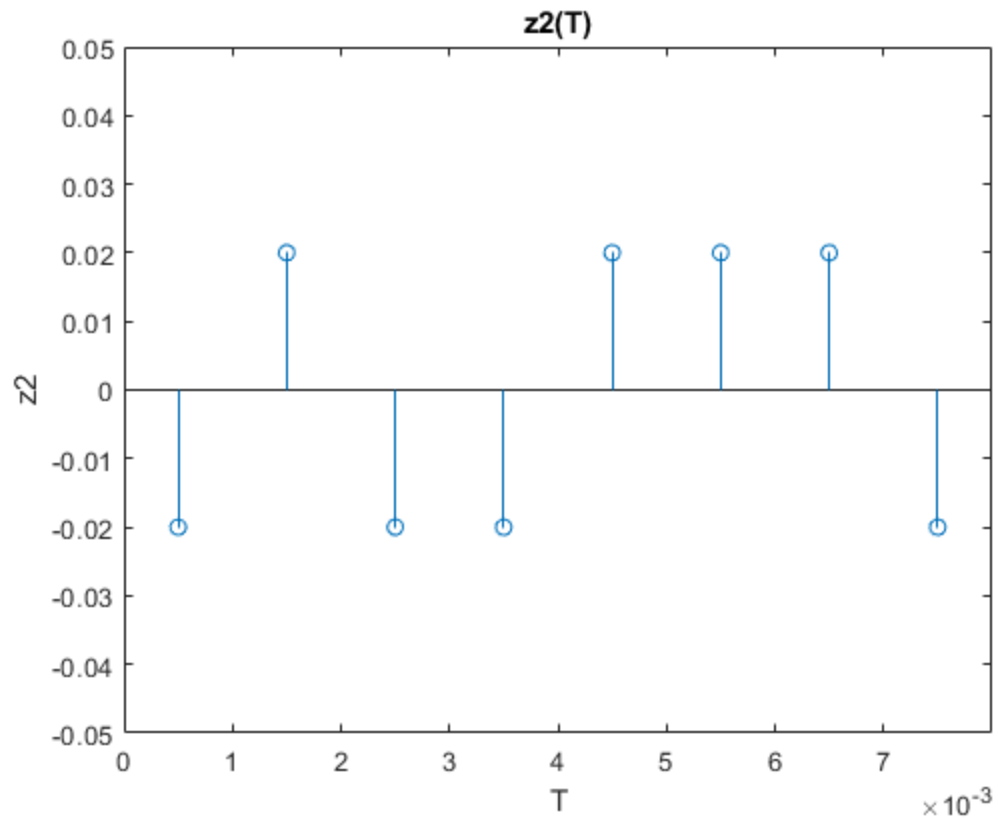
figure;
stem(s_time,z2);
title('z2(T)');
xlabel('T');
ylabel('z2');
axis([min(t) max(t) -0.05 0.05]);

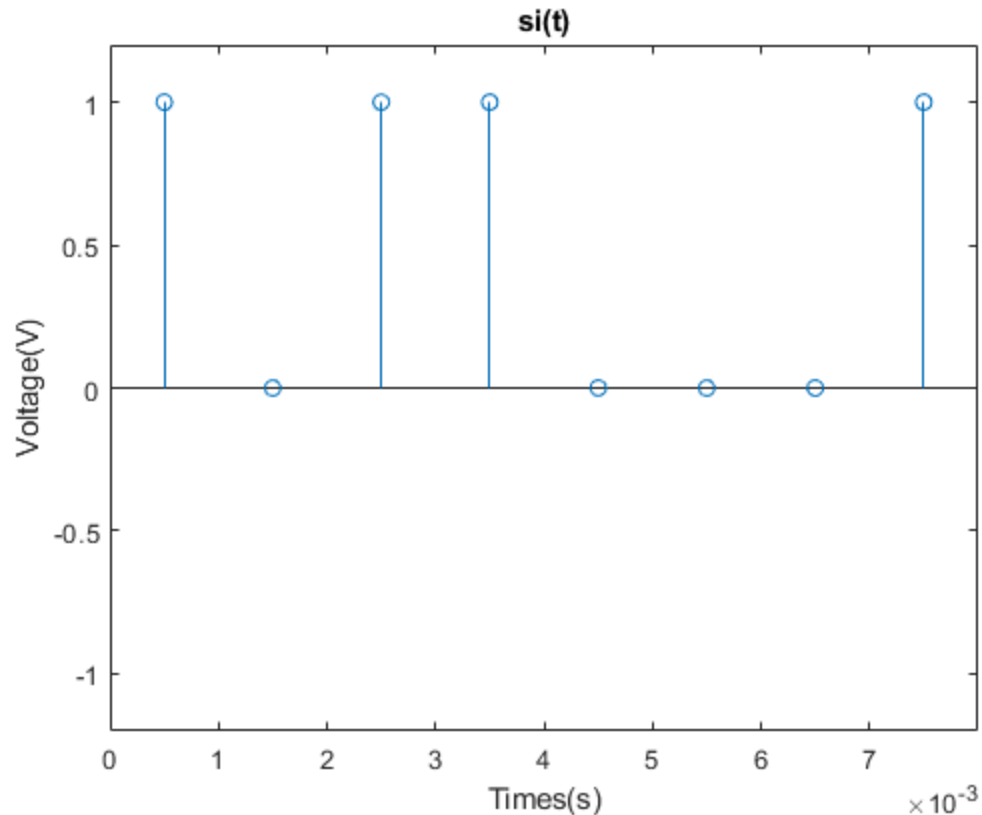
figure;
stem(s_time,z);
```

```
title('z(T)');  
xlabel('T');  
ylabel('z');  
axis([min(t) max(t) -0.05 0.05]);  
  
figure;  
stem(s_time,sample);  
title('si(t)');  
xlabel('Times(s)');  
ylabel('Voltage(V)');  
axis([min(t) max(t) -1.2 1.2]);
```









## Problem 2 Noise

```
%SNR of 10
SNR=10;

%%The power of the signal is found by taking the integral of the
signal
%%squared. The SNR is found by dividing the power of the signal by the
%%power of the noise. It is done on Matlab using 'measured'

r=awgn(0.8*s,SNR,'measured');
plot(t,r);
title('r(t) with Noise');
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) -15 15]);

for j=1:length(s_time)
    %integral time
    x=(s_time(j)-s_time(1)):sample_time:s_time(j)+s_time(1);
    %z1
    z1(j)=trapz(x,5.*r((j-1)*length(x)+1:j*length(x)));
    %z2
    z2(j)=trapz(x,-5.*r((j-1)*length(x)+1:j*length(x)));
end
```

```
%signal in integral time
r((j-1)*length(x)+1:j*length(x));
%z
z(j)=z1(j)-z2(j);
%decision
if z(j)>y
    sample(j)=1;
else
    sample(j)=0;
end
end

figure;
stem(s_time,z1);
title('z1(T)');
xlabel('T');
ylabel('z1');
axis([min(t) max(t) -0.05 0.05]);

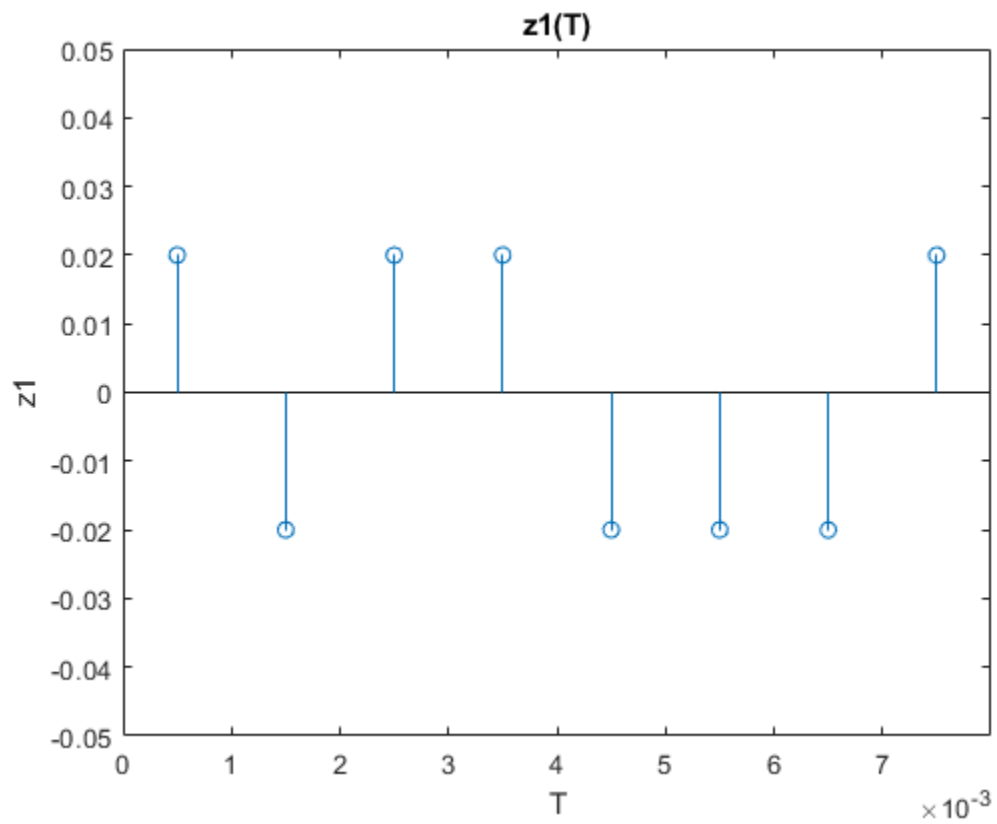
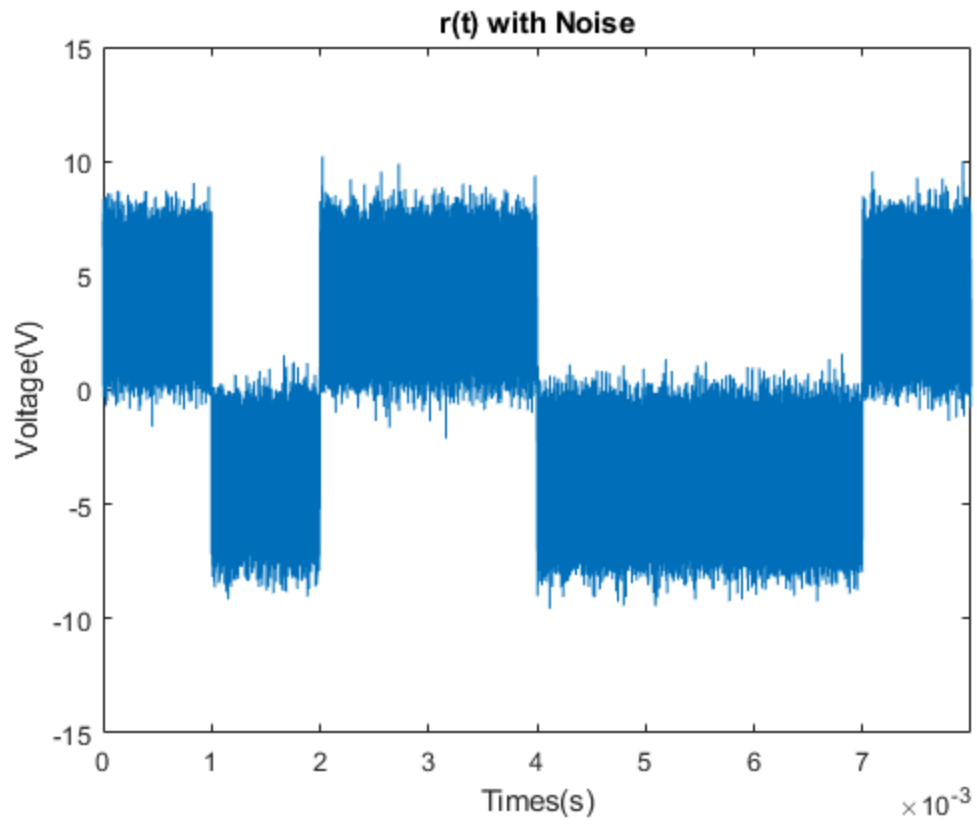
figure;
stem(s_time,z2);
title('z2(T)');
xlabel('T');
ylabel('z2');
axis([min(t) max(t) -0.05 0.05]);

figure;
stem(s_time,z);
title('z(T)');
xlabel('T');
ylabel('z');
axis([min(t) max(t) -0.1 0.1]);

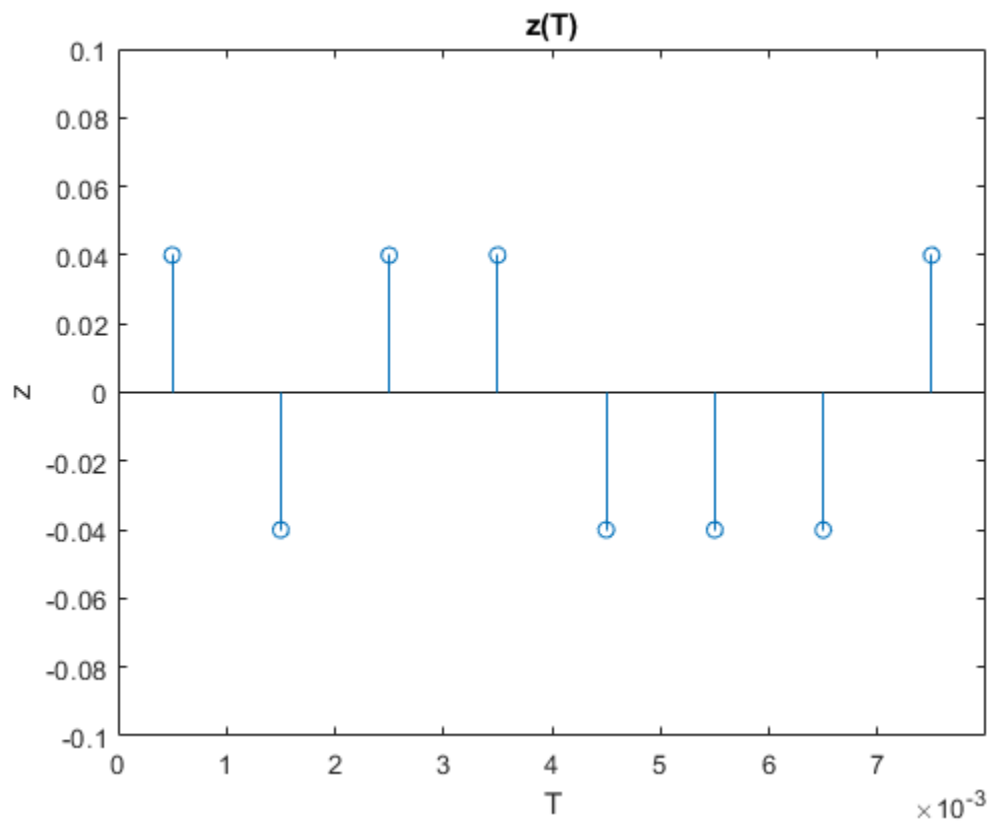
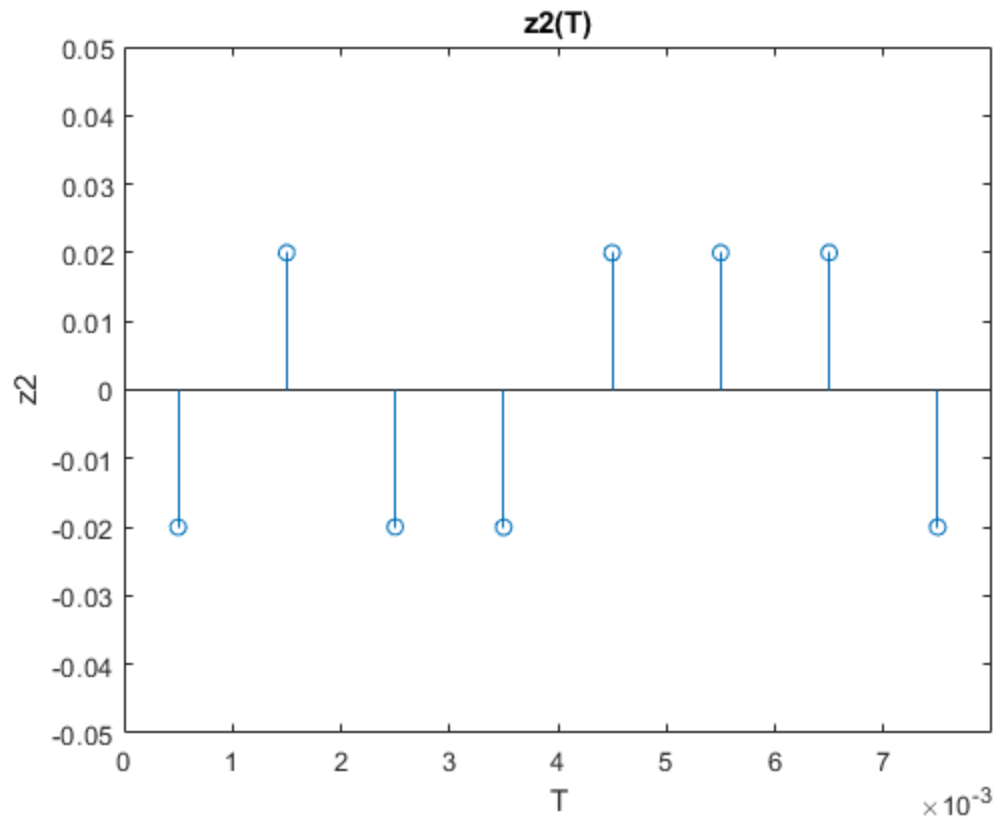
figure;
stem(s_time,sample);
title('si(t)');
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) -1.2 1.2]);

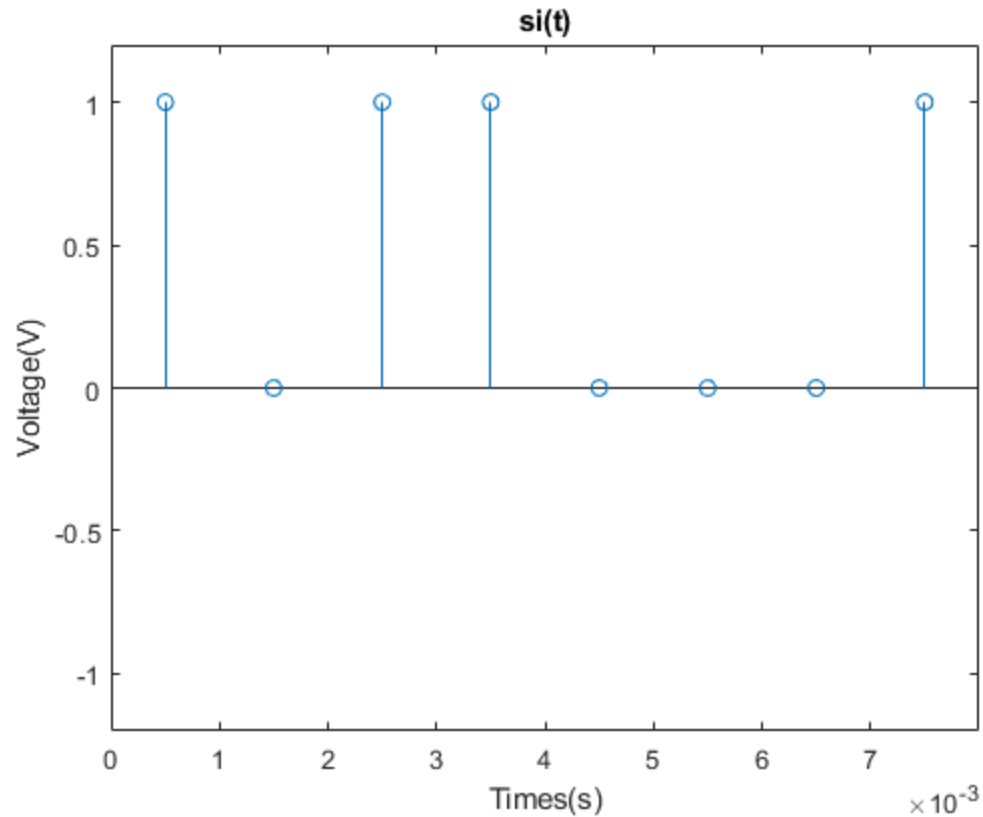
errors=0;
for j=1:length(sample)
    if (sample(j)~=test(j))
        errors=errors+1;
    end
end
fprintf('Nmumber of errors is %i\n',errors);

Nmumber of errors is 0
```









## 40 Times 10 SNR

```
clc;
clear all;
close all;

%given sequence 40 times
test=[1 0 1 1 0 0 0 1];
test= repmat(test,1,10);

%map to voltages
v_test=test*10^-5;

%generate waveform
sample_time=10e-9;
t=0:sample_time:1e-3*length(test)-sample_time;
i=1e-3;
n=1;
for j=1:length(t)
    if t(j)<=n*i
        s(j)=v_test(n);
    else
        n=n+1;
        s(j)=v_test(n);
    end
end
```

```
end

%SNR of 10
SNR=10;
%%The power of the signal is found by taking the integral of the
signal
%%squared. The SNR is found by dividing the power of the signal by the
%%power of the noise. It is done on Matlab using 'measured'

r=awgn(0.8*s,SNR,'measured');
plot(t,r);
title(['r(t) with Noise SNR=',num2str(SNR)]);
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(r)-1 max(r)+1]);

%optimum detection value= (a1+a2)/2=0
y=0;

%Correlator Detector
%times to sample
for j=1:length(test)
    s_time(j)=t((2*j-1)*(length(t)/length(test)/2));
end

for j=1:length(s_time)
    %integral time
    x=(s_time(j)-s_time(1)):sample_time:s_time(j)+s_time(1);
    %z1
    z1(j)=trapz(x,5.*r((j-1)*length(x)+1:j*length(x)));
    %z2
    z2(j)=trapz(x,-5.*r((j-1)*length(x)+1:j*length(x)));
    %signal in integral time
    r((j-1)*length(x)+1:j*length(x));
    %z
    z(j)=z1(j)-z2(j);
    %decision
    if z(j)>y
        sample(j)=1;
    else
        sample(j)=0;
    end
end

figure;
stem(s_time,z1);
title('z1(T)');
xlabel('T');
ylabel('z1');
axis([min(t) max(t) min(z1)-0.05 max(z1)+0.05]);

figure;
stem(s_time,z2);
```

```
title('z2(T)');
xlabel('T');
ylabel('z2');
axis([min(t) max(t) min(z2)-0.05 max(z2)+0.05]);

figure;
stem(s_time,z);
title(['z(T) Mean=', num2str(mean(z)), ' STD=', num2str(std(z))]);
xlabel('T');
ylabel('z');
axis([min(t) max(t) min(z)-0.1 max(z)+0.1]);

figure;
stem(s_time,sample);
title('si(t)');
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(sample)-0.1 max(sample)+0.1 ]);

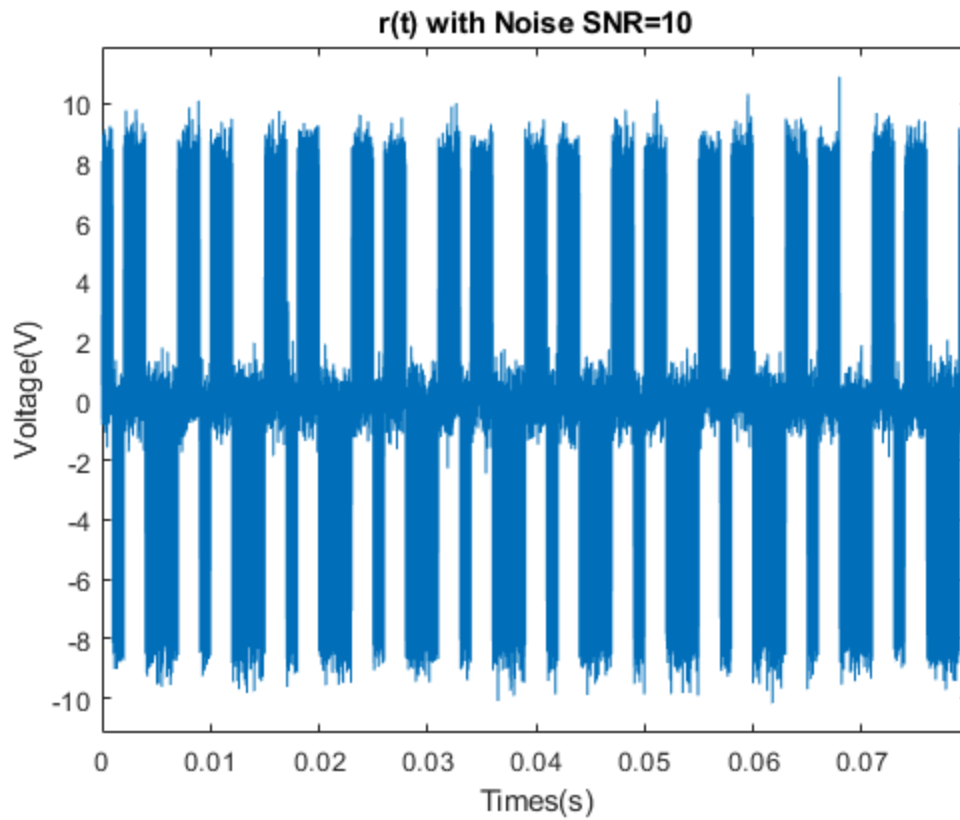
errors=0;
for j=1:length(sample)
    if (sample(j)~=test(j))
        errors=errors+1;
    end
end
fprintf('Number of errors is %i\n',errors);
zs1=0;
zs2=0;
for x=1:length(z)
    if z>0.0
        zs1=zs1+1;
    else
        zs2=zs2+1;
    end
end

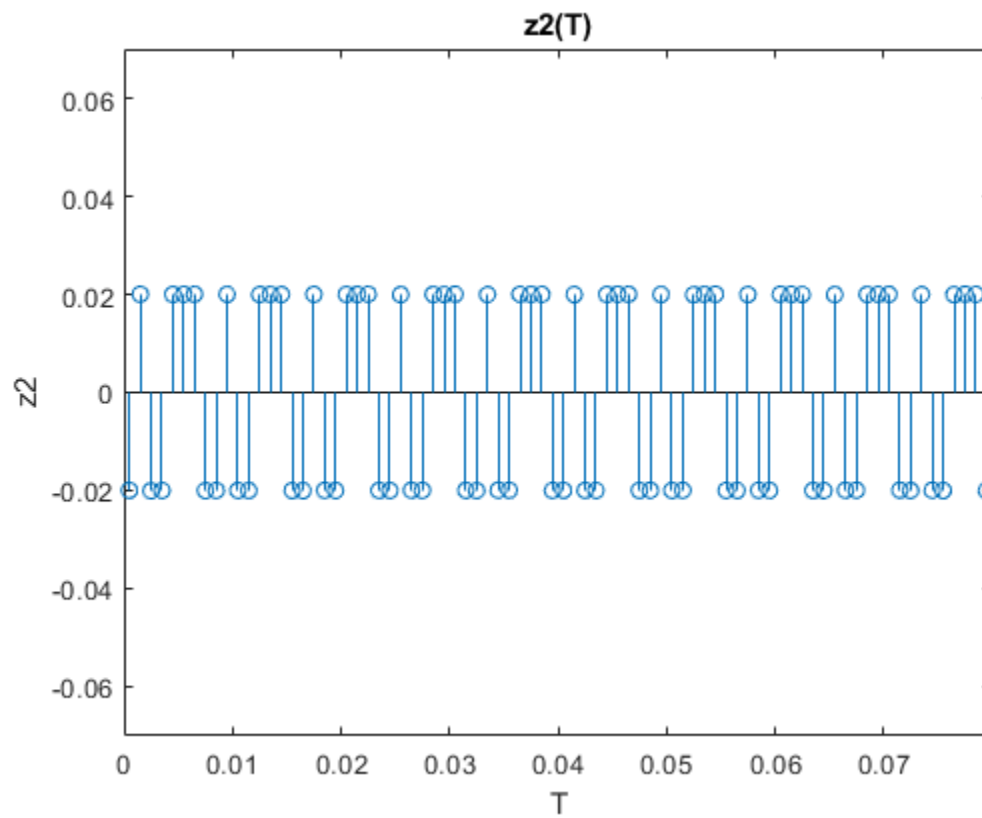
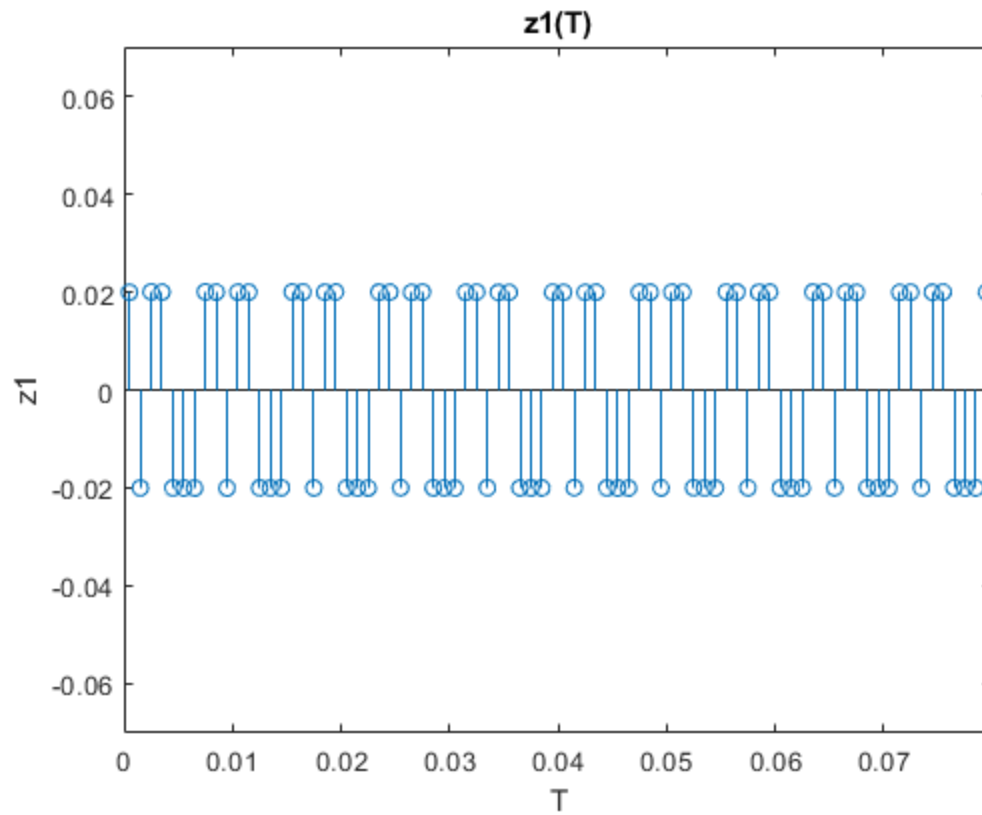
fprintf('Mean is %d\n',mean(z));
fprintf('Standard deviation is %d\n',std(z));

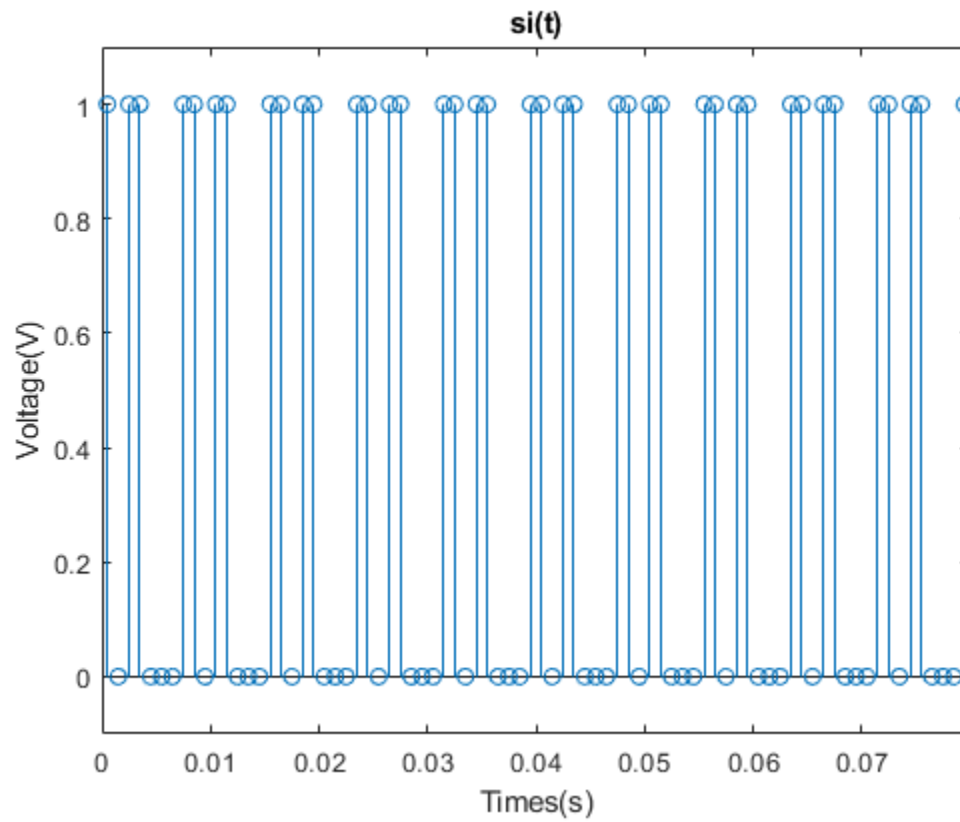
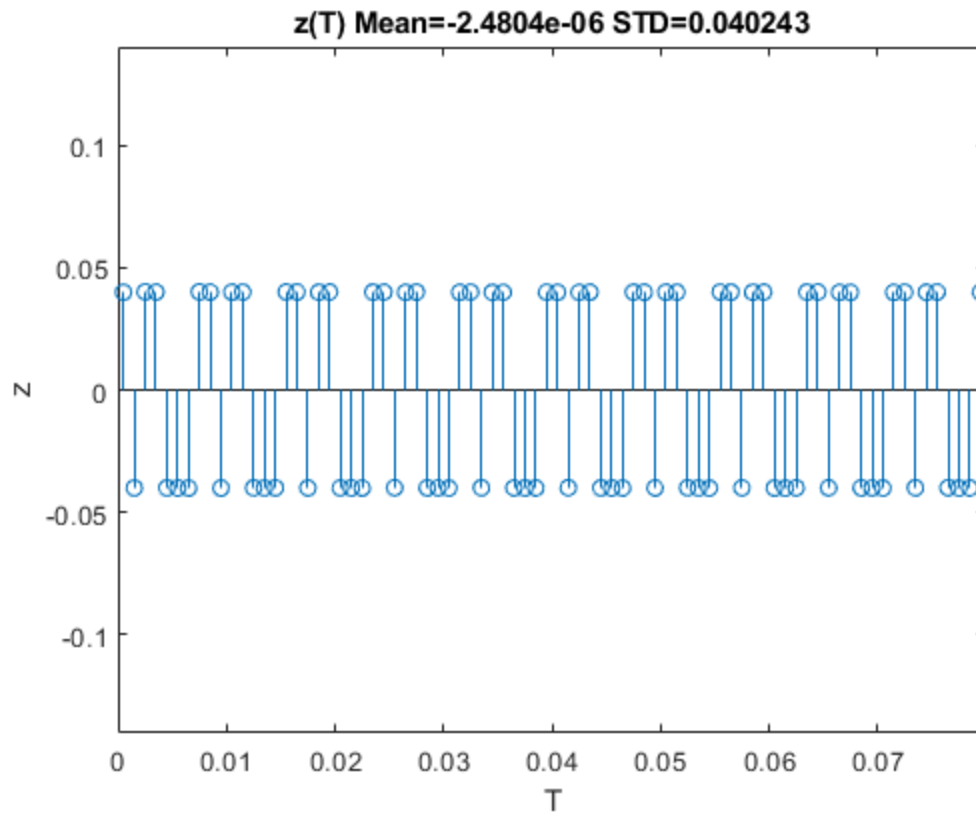
figure;
h=histogram(z);
for n=1:25
    morebins(h);
end
title(['PDF of z(T) with SNR=', num2str(SNR), ' dB Number of Errors=',
    num2str(errors)]);
xlabel('Value');
ylabel('Occurrences');
hold on;

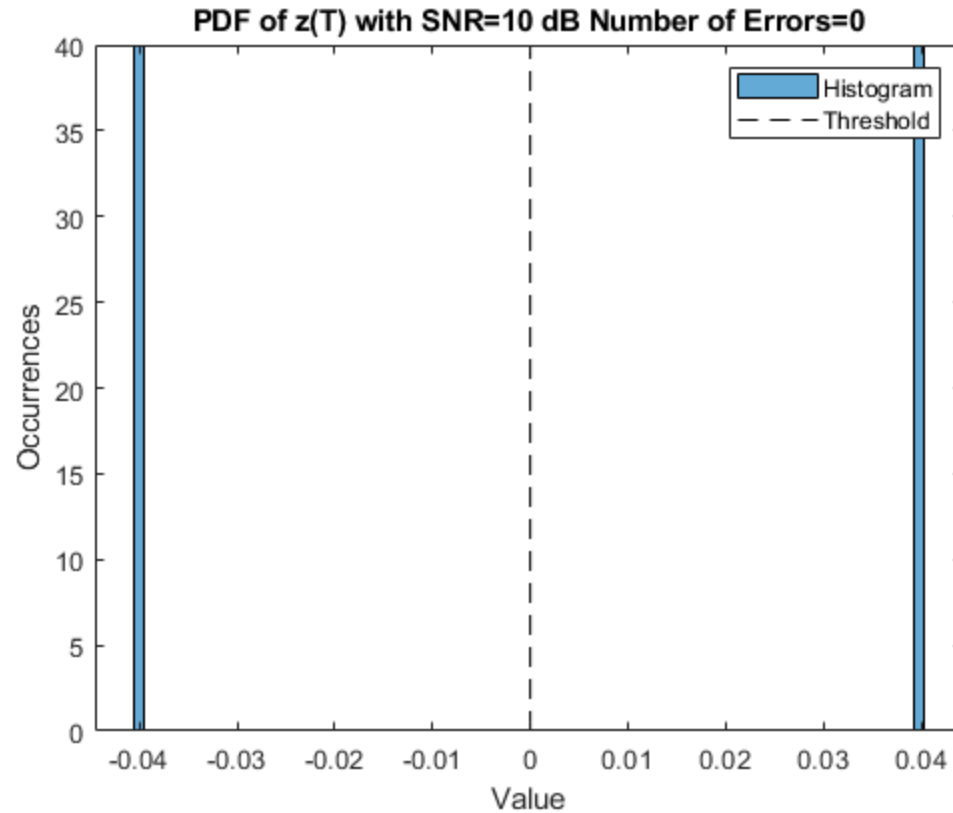
ttime_y=0:1:40;
plot(y*ones(size(ttime_y)),ttime_y,'k --');
```

```
legend('Histogram','Threshold');  
BER=errors/length(test);  
  
fprintf('BER of %f, and Threshold of %f, at SNR of %i dB  
  \n',BER,y,SNR);  
  
Number of errors is 0  
Mean is -2.480350e-06  
Standard deviation is 4.024339e-02  
BER of 0.000000, and Threshold of 0.000000, at SNR of 10 dB
```









## 40 Times 0 SNR

```
clc;
clear all;
close all;

%given sequence 40 times
test=[1 0 1 1 0 0 0 1];
test=repmat(test,1,10);

%map to voltages
v_test=test*10-5;

%generate waveform
sample_time=10e-9;
t=0:sample_time:1e-3*length(test)-sample_time;
i=1e-3;
n=1;
for j=1:length(t)
    if t(j)<=n*i
        s(j)=v_test(n);
    else
        n=n+1;
        s(j)=v_test(n);
    end
end
```



```
end

%SNR of 0
SNR=0;
%%The power of the signal is found by taking the integral of the
signal
%%squared. The SNR is found by dividing the power of the signal by the
%%power of the noise. It is done on Matlab using 'measured'

r=awgn(0.8*s,SNR,'measured');
plot(t,r);
title(['r(t) with Noise SNR=',num2str(SNR)]);
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(r)-1 max(r)+1]);

%optimum detection value= (a1+a2)/2=0
y=0;

%Correlator Detector
%times to sample
for j=1:length(test)
    s_time(j)=t((2*j-1)*(length(t)/length(test)/2));
end

for j=1:length(s_time)
    %integral time
    x=(s_time(j)-s_time(1)):sample_time:s_time(j)+s_time(1);
    %z1
    z1(j)=trapz(x,5.*r((j-1)*length(x)+1:j*length(x)));
    %z2
    z2(j)=trapz(x,-5.*r((j-1)*length(x)+1:j*length(x)));
    %signal in integral time
    r((j-1)*length(x)+1:j*length(x));
    %z
    z(j)=z1(j)-z2(j);
    %decision
    if z(j)>y
        sample(j)=1;
    else
        sample(j)=0;
    end
end

figure;
stem(s_time,z1);
title('z1(T)');
xlabel('T');
ylabel('z1');
axis([min(t) max(t) min(z1)-0.05 max(z1)+0.05]);

figure;
stem(s_time,z2);
```

```
title('z2(T)');
xlabel('T');
ylabel('z2');
axis([min(t) max(t) min(z2)-0.05 max(z2)+0.05]);

figure;
stem(s_time,z);
title(['z(T) Mean=', num2str(mean(z)), ' STD=', num2str(std(z))]);
xlabel('T');
ylabel('z');
axis([min(t) max(t) min(z)-0.1 max(z)+0.1]);

figure;
stem(s_time,sample);
title('si(t)');
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(sample)-0.1 max(sample)+0.1 ]);

errors=0;
for j=1:length(sample)
    if (sample(j)~=test(j))
        errors=errors+1;
    end
end
fprintf('Number of errors is %i\n',errors);
zs1=0;
zs2=0;
for x=1:length(z)
    if z>0.0
        zs1=zs1+1;
    else
        zs2=zs2+1;
    end
end

fprintf('Mean is %d\n',mean(z));
fprintf('Standard deviation is %d\n',std(z));

figure;
h=histogram(z);
for n=1:25
    morebins(h);
end
title(['PDF of z(T) with SNR=',num2str(SNR),' dB Number of Errors=',
    num2str(errors)]);
xlabel('Value');
ylabel('Occurrences');
hold on;

ttime_y=0:1:40;
plot(y*ones(size(ttime_y)),ttime_y,'k --');
legend('Histogram','Threshold');
```

```
BER=errors/length(test);
```

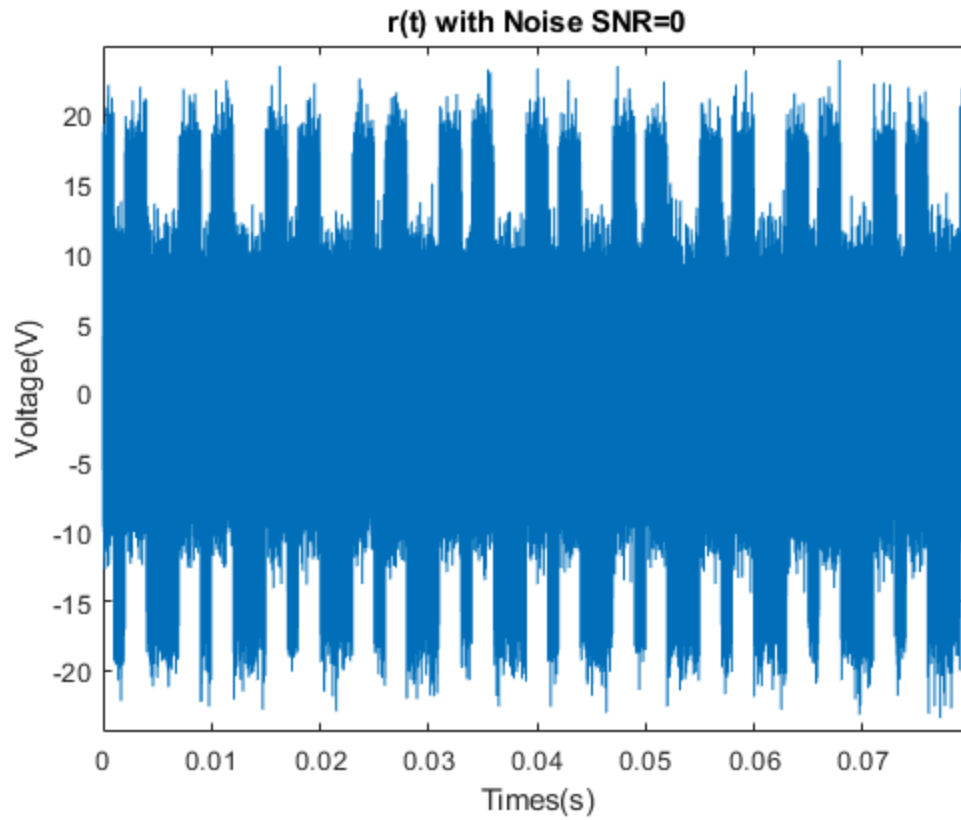
```
fprintf('BER of %f, and Threshold of %f, at SNR of %i dB\n',BER,y,SNR);
```

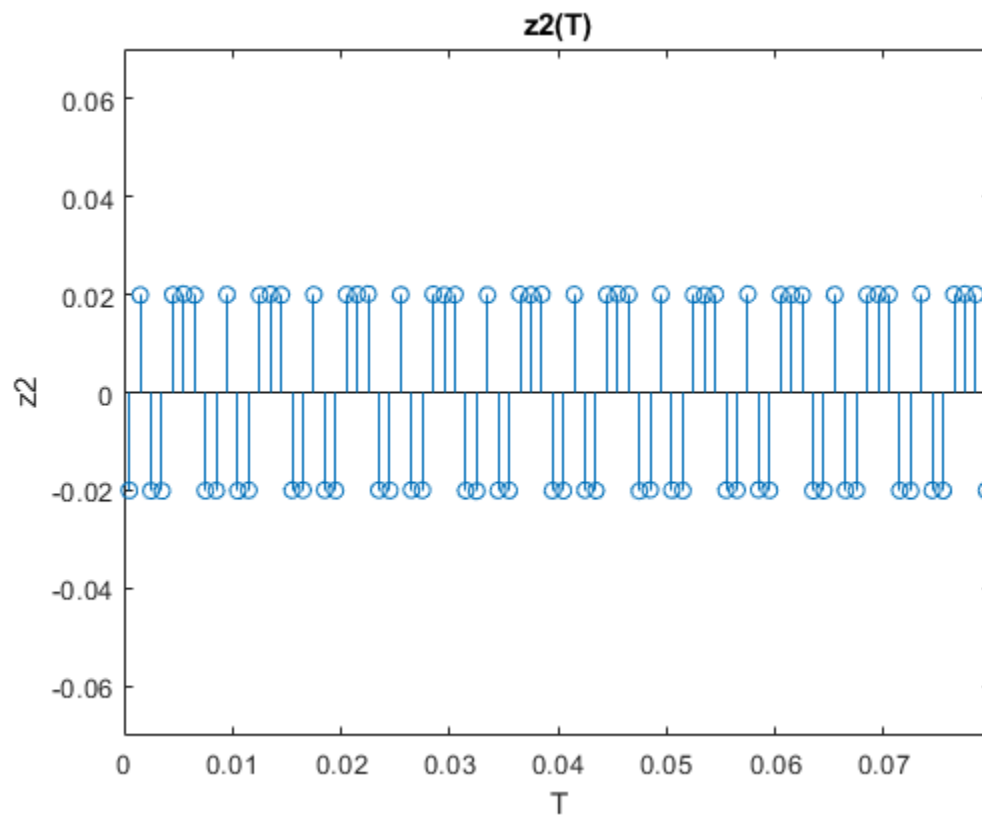
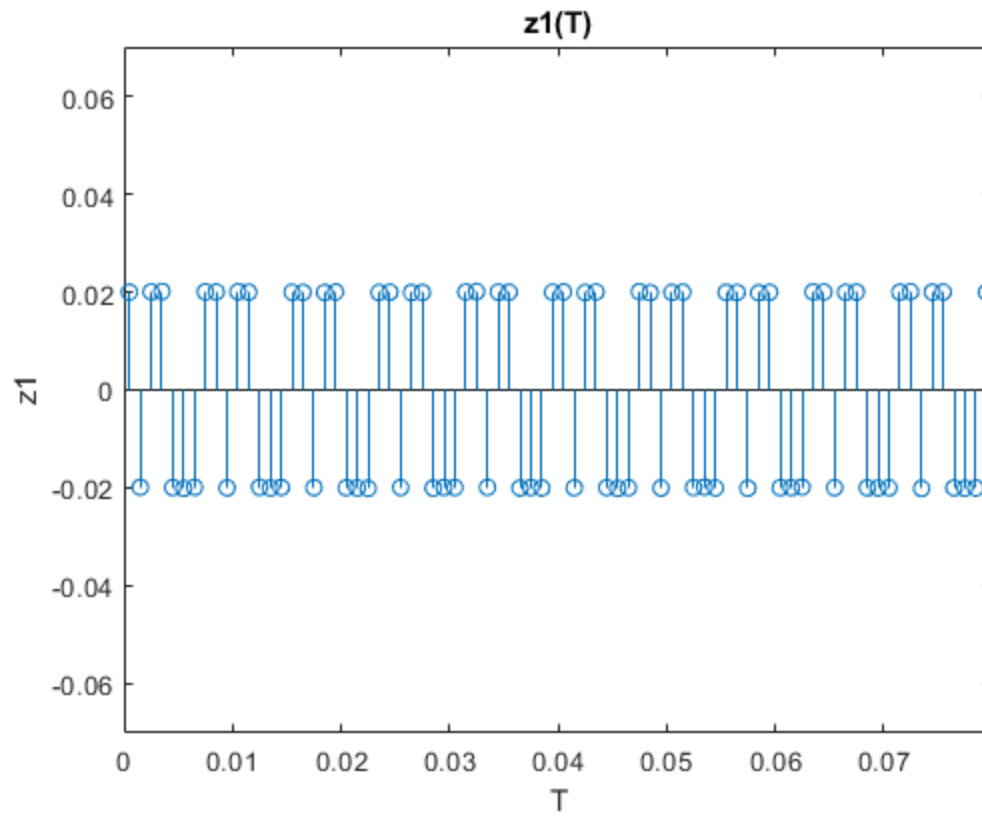
*Number of errors is 0*

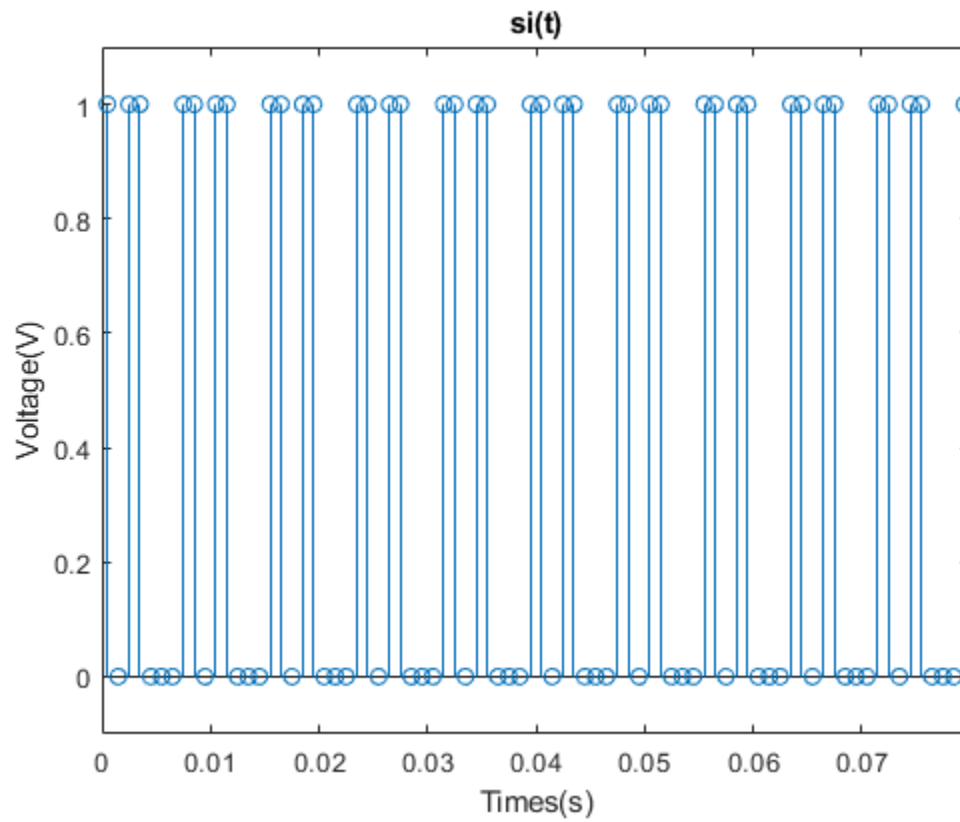
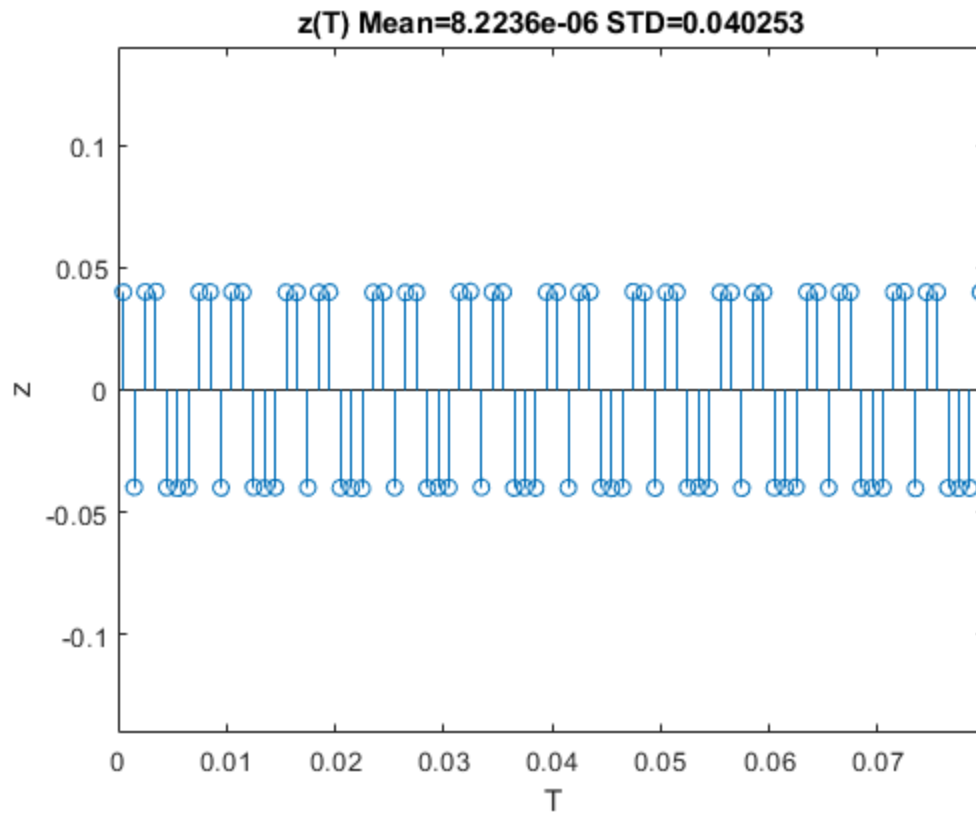
*Mean is 8.223644e-06*

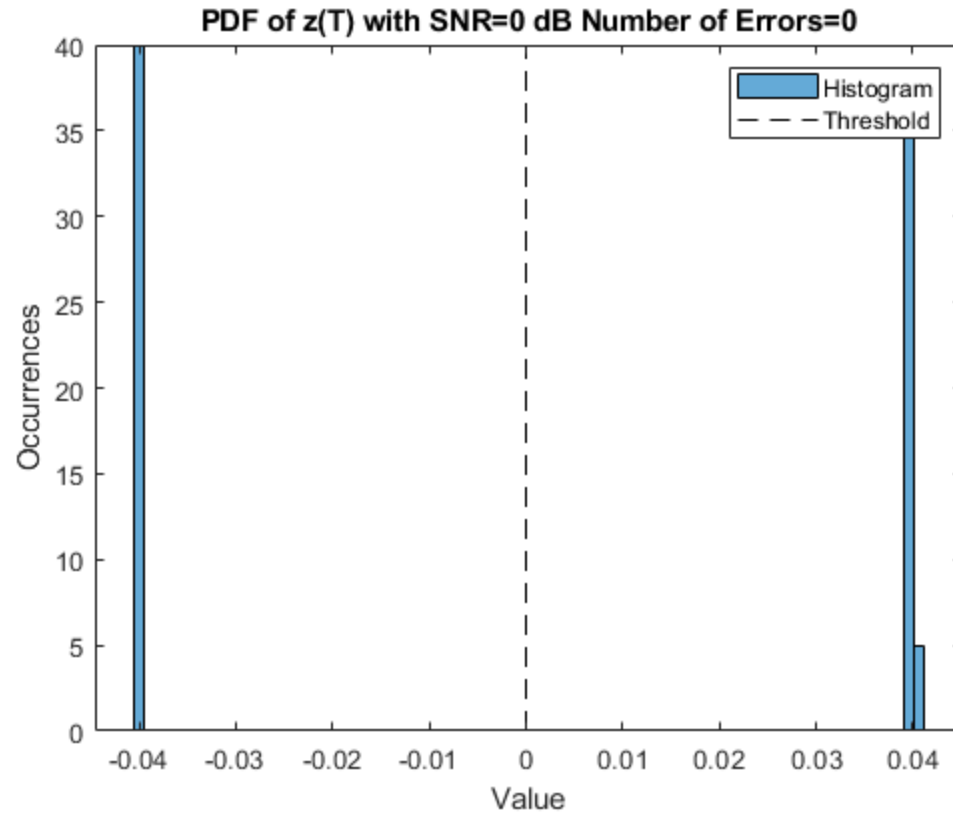
*Standard deviation is 4.025321e-02*

*BER of 0.000000, and Threshold of 0.000000, at SNR of 0 dB*









## 40 Times -40 SNR

```
clc;
clear all;
close all;

%given sequence 40 times
test=[1 0 1 1 0 0 0 1];
test=repmat(test,1,10);

%map to voltages
v_test=test*10-5;

%generate waveform
sample_time=10e-9;
t=0:sample_time:1e-3*length(test)-sample_time;
i=1e-3;
n=1;
for j=1:length(t)
    if t(j)<=n*i
        s(j)=v_test(n);
    else
        n=n+1;
        s(j)=v_test(n);
    end
end
```

```
end

%SNR of -40
SNR=-40;
%%The power of the signal is found by taking the integral of the
signal
%%squared. The SNR is found by dividing the power of the signal by the
%%power of the noise. It is done on Matlab using 'measured'

r=awgn(0.8*s,SNR,'measured');
plot(t,r);
title(['r(t) with Noise SNR=',num2str(SNR)]);
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(r)-1 max(r)+1]);

%optimum detection value= (a1+a2)/2=0
y=0;

%Correlator Detector
%times to sample
for j=1:length(test)
    s_time(j)=t((2*j-1)*(length(t)/length(test)/2));
end

for j=1:length(s_time)
    %integral time
    x=(s_time(j)-s_time(1)):sample_time:s_time(j)+s_time(1);
    %z1
    z1(j)=trapz(x,5.*r((j-1)*length(x)+1:j*length(x)));
    %z2
    z2(j)=trapz(x,-5.*r((j-1)*length(x)+1:j*length(x)));
    %signal in integral time
    r((j-1)*length(x)+1:j*length(x));
    %z
    z(j)=z1(j)-z2(j);
    %decision
    if z(j)>y
        sample(j)=1;
    else
        sample(j)=0;
    end
end

figure;
stem(s_time,z1);
title('z1(T)');
xlabel('T');
ylabel('z1');
axis([min(t) max(t) min(z1)-0.05 max(z1)+0.05]);

figure;
stem(s_time,z2);
```

```
title('z2(T)');
xlabel('T');
ylabel('z2');
axis([min(t) max(t) min(z2)-0.05 max(z2)+0.05]);

figure;
stem(s_time,z);
title(['z(T) Mean=', num2str(mean(z)), ' STD=', num2str(std(z))]);
xlabel('T');
ylabel('z');
axis([min(t) max(t) min(z)-0.1 max(z)+0.1]);

figure;
stem(s_time,sample);
title('si(t)');
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(sample)-0.1 max(sample)+0.1 ]);

errors=0;
for j=1:length(sample)
    if (sample(j)~=test(j))
        errors=errors+1;
    end
end
fprintf('Number of errors is %i\n',errors);
zs1=0;
zs2=0;
for x=1:length(z)
    if z>0.0
        zs1=zs1+1;
    else
        zs2=zs2+1;
    end
end

fprintf('Mean is %d\n',mean(z));
fprintf('Standard deviation is %d\n',std(z));

figure;
h=histogram(z);
for n=1:25
    morebins(h);
end
title(['PDF of z(T) with SNR=', num2str(SNR), ' dB Number of Errors=',
    num2str(errors)]);
xlabel('Value');
ylabel('Occurrences');
hold on;

ttime_y=0:1:40;
plot(y*ones(size(ttime_y)),ttime_y,'k --');
legend('Histogram','Threshold');
```



```
BER=errors/length(test);
```

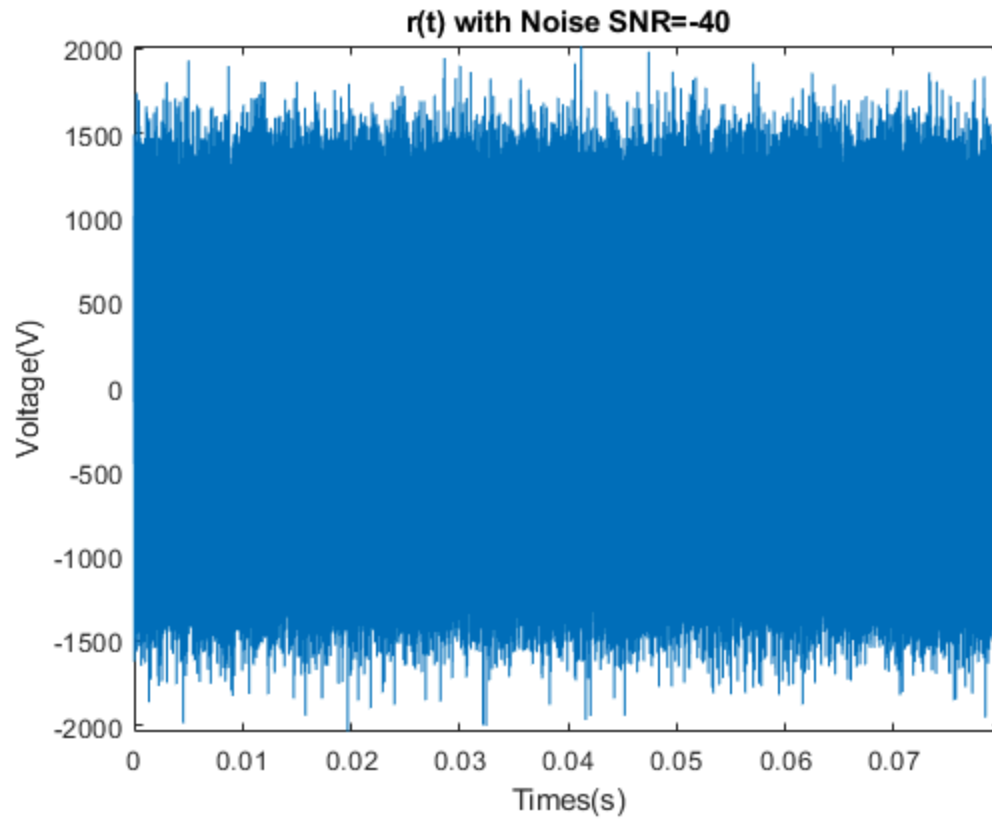
```
fprintf('BER of %f, and Threshold of %f, at SNR of %i dB\n',BER,y,SNR);
```

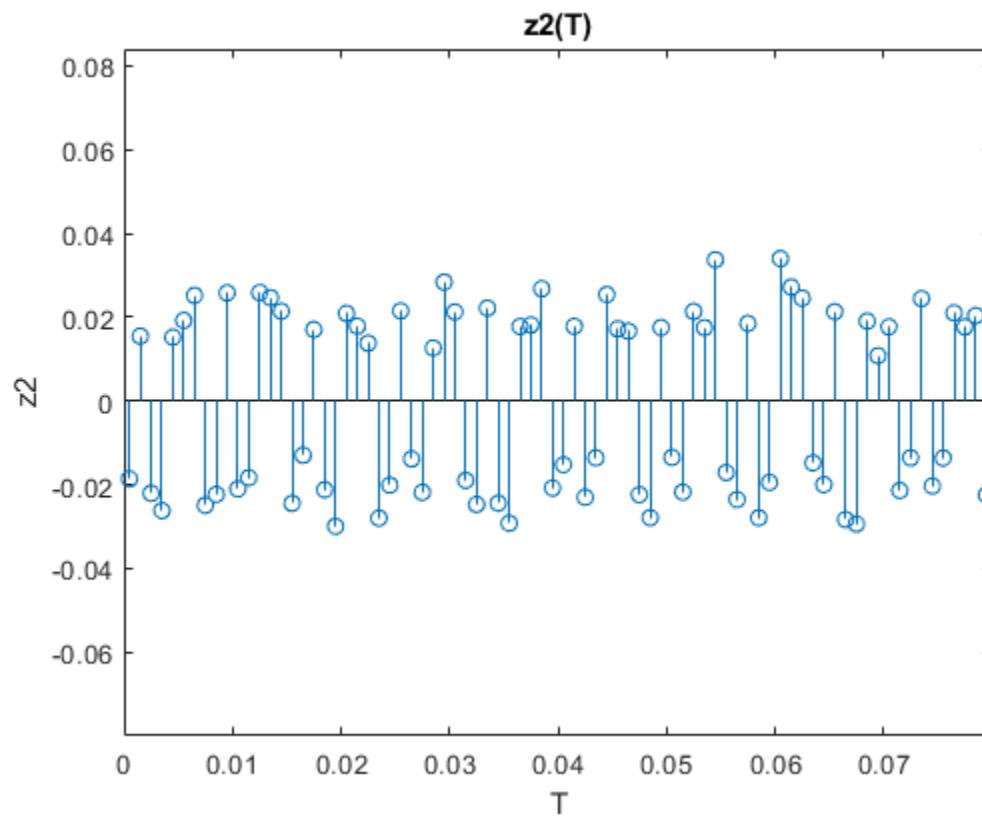
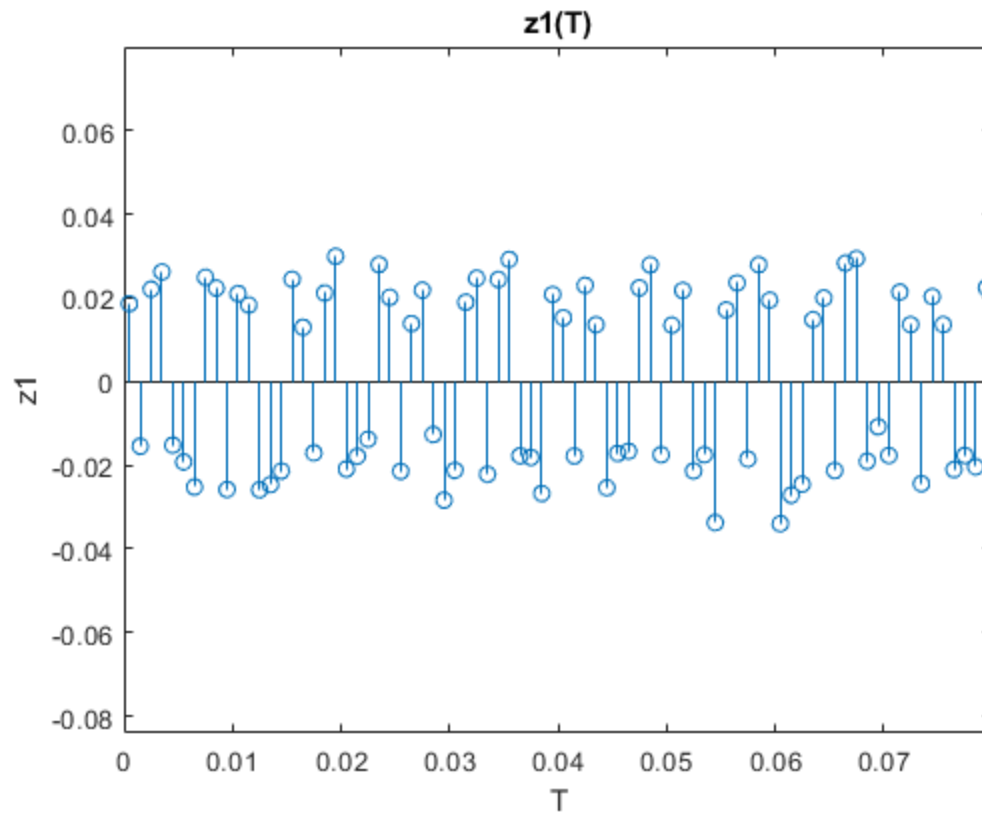
*Number of errors is 0*

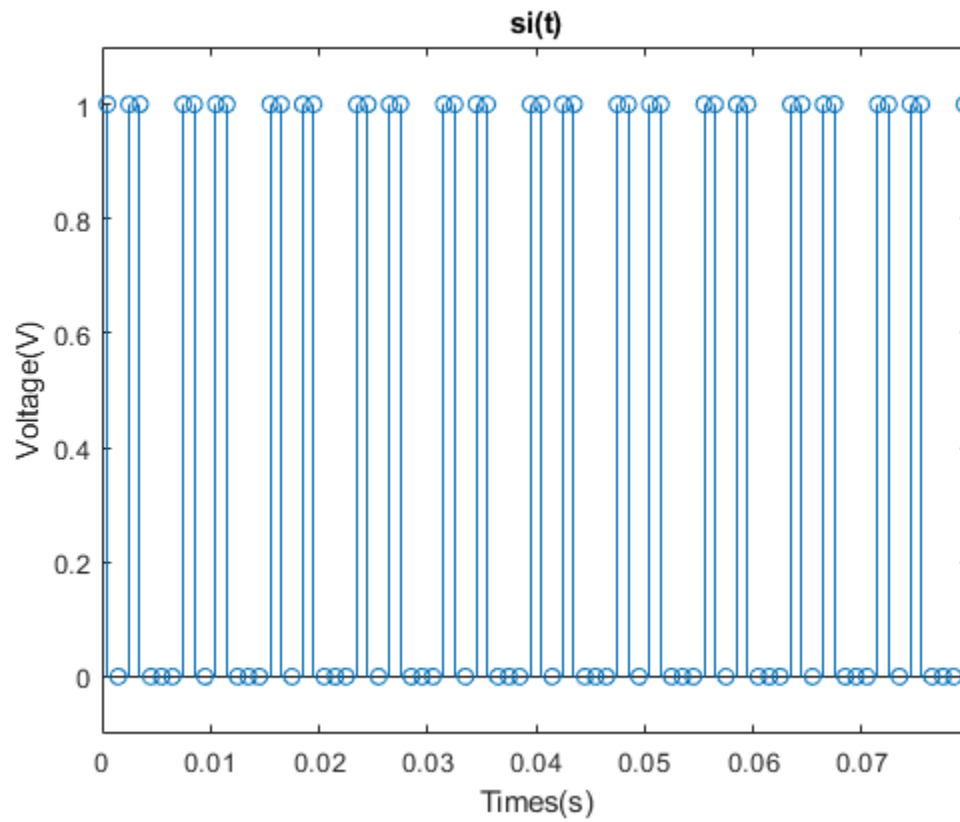
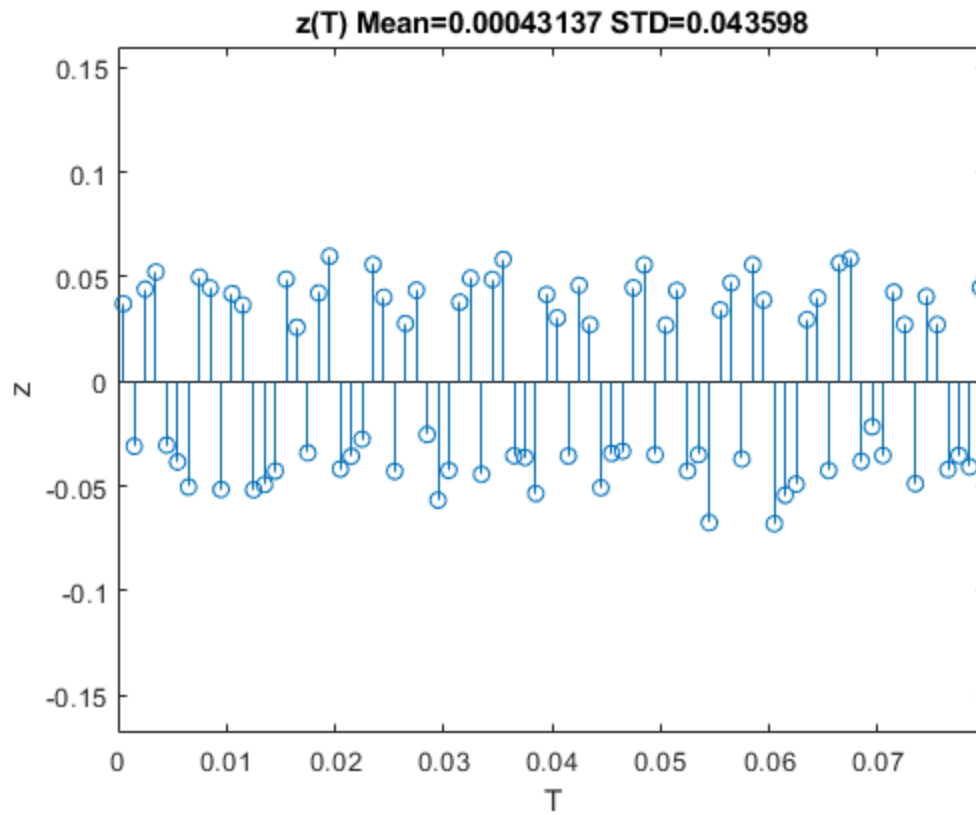
*Mean is 4.313672e-04*

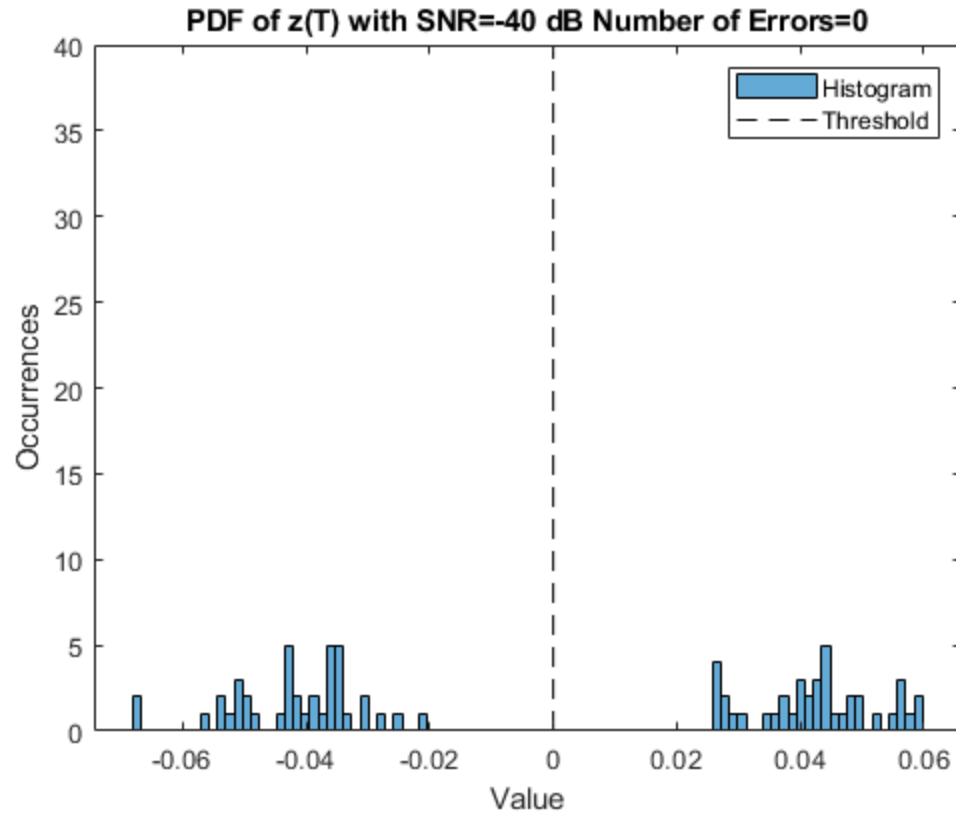
*Standard deviation is 4.359774e-02*

*BER of 0.000000, and Threshold of 0.000000, at SNR of -40 dB*









## 40 Times -45 SNR

```
clc;
clear all;
close all;

%given sequence 40 times
test=[1 0 1 1 0 0 0 1];
test=repmat(test,1,10);

%map to voltages
v_test=test*10-5;

%generate waveform
sample_time=10e-9;
t=0:sample_time:1e-3*length(test)-sample_time;
i=1e-3;
n=1;
for j=1:length(t)
    if t(j)<=n*i
        s(j)=v_test(n);
    else
        n=n+1;
        s(j)=v_test(n);
    end
end
```

```
end

%SNR of -45
SNR=-45;
%%The power of the signal is found by taking the integral of the
signal
%%squared. The SNR is found by dividing the power of the signal by the
%%power of the noise. It is done on Matlab using 'measured'

r=awgn(0.8*s,SNR,'measured');
plot(t,r);
title(['r(t) with Noise SNR=',num2str(SNR)]);
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(r)-1 max(r)+1]);

%optimum detection value= (a1+a2)/2=0
y=0;

%Correlator Detector
%times to sample
for j=1:length(test)
    s_time(j)=t((2*j-1)*(length(t)/length(test)/2));
end

for j=1:length(s_time)
    %integral time
    x=(s_time(j)-s_time(1)):sample_time:s_time(j)+s_time(1);
    %z1
    z1(j)=trapz(x,5.*r((j-1)*length(x)+1:j*length(x)));
    %z2
    z2(j)=trapz(x,-5.*r((j-1)*length(x)+1:j*length(x)));
    %signal in integral time
    r((j-1)*length(x)+1:j*length(x));
    %z
    z(j)=z1(j)-z2(j);
    %decision
    if z(j)>y
        sample(j)=1;
    else
        sample(j)=0;
    end
end

figure;
stem(s_time,z1);
title('z1(T)');
xlabel('T');
ylabel('z1');
axis([min(t) max(t) min(z1)-0.05 max(z1)+0.05]);

figure;
stem(s_time,z2);
```

```
title('z2(T)');
xlabel('T');
ylabel('z2');
axis([min(t) max(t) min(z2)-0.05 max(z2)+0.05]);

figure;
stem(s_time,z);
title(['z(T) Mean=', num2str(mean(z)), ' STD=', num2str(std(z))]);
xlabel('T');
ylabel('z');
axis([min(t) max(t) min(z)-0.1 max(z)+0.1]);

figure;
stem(s_time,sample);
title('si(t)');
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(sample)-0.1 max(sample)+0.1 ]);

errors=0;
for j=1:length(sample)
    if (sample(j)~=test(j))
        errors=errors+1;
    end
end
fprintf('Number of errors is %i\n',errors);
zs1=0;
zs2=0;
for x=1:length(z)
    if z>0.0
        zs1=zs1+1;
    else
        zs2=zs2+1;
    end
end

fprintf('Mean is %d\n',mean(z));
fprintf('Standard deviation is %d\n',std(z));

figure;
h=histogram(z);
for n=1:25
    morebins(h);
end
title(['PDF of z(T) with SNR=', num2str(SNR), ' dB Number of Errors=',
    num2str(errors)]);
xlabel('Value');
ylabel('Occurrences');
hold on;

ttime_y=0:1:40;
plot(y*ones(size(ttime_y)),ttime_y,'k --');
legend('Histogram', 'Threshold');
```

```
BER=errors/length(test);
```

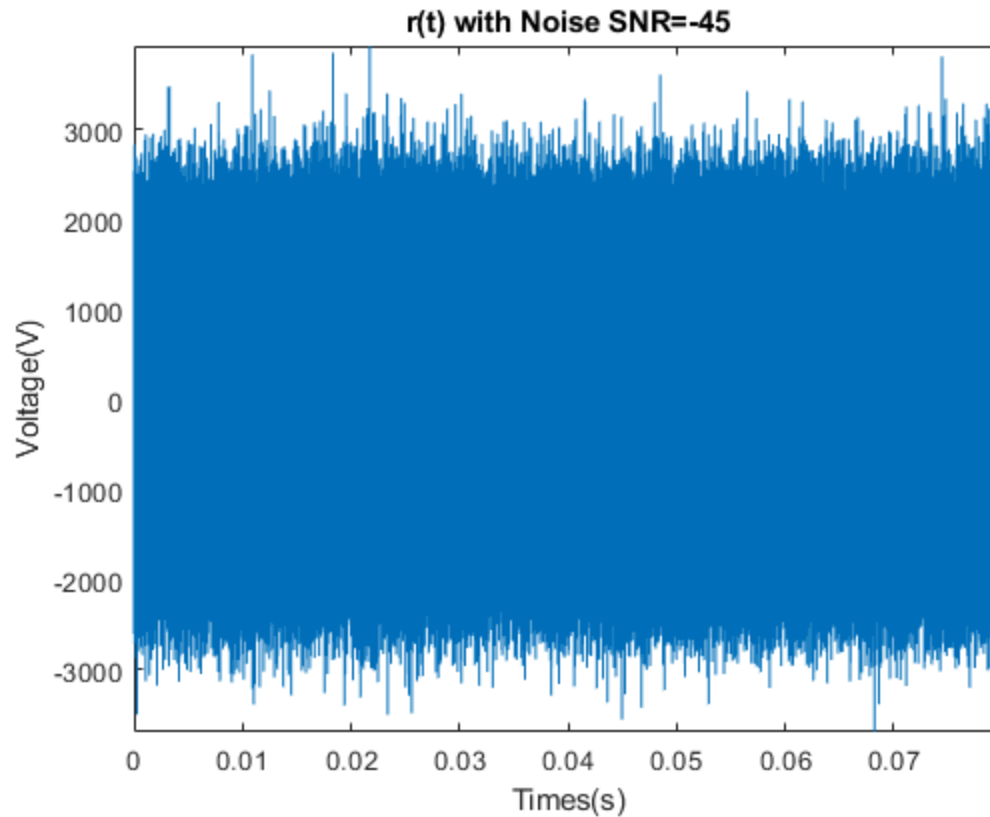
```
fprintf('BER of %f, and Threshold of %f, at SNR of %i dB\n',BER,y,SNR);
```

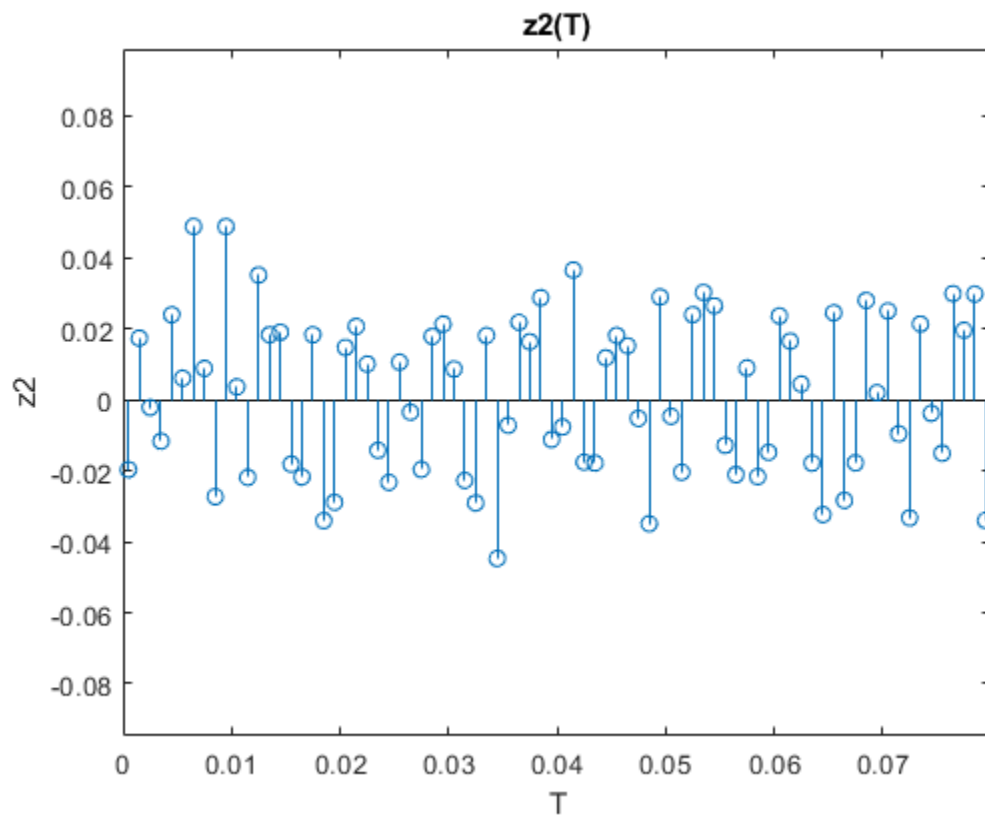
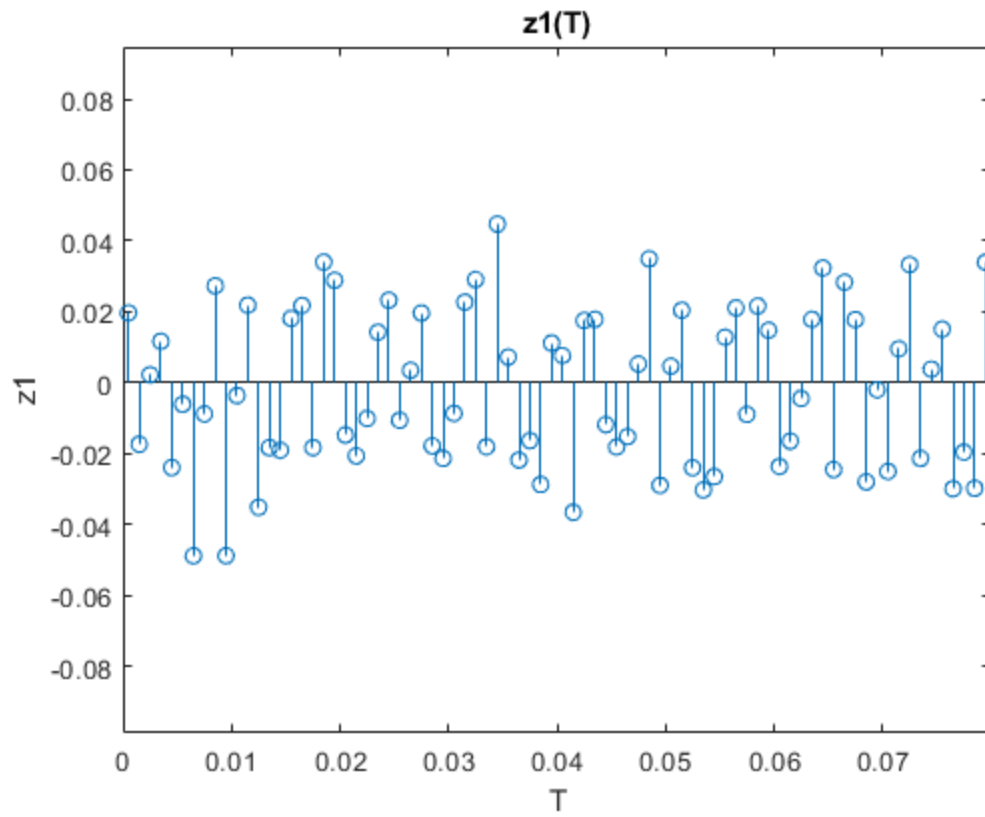
*Number of errors is 2*

*Mean is -3.355104e-03*

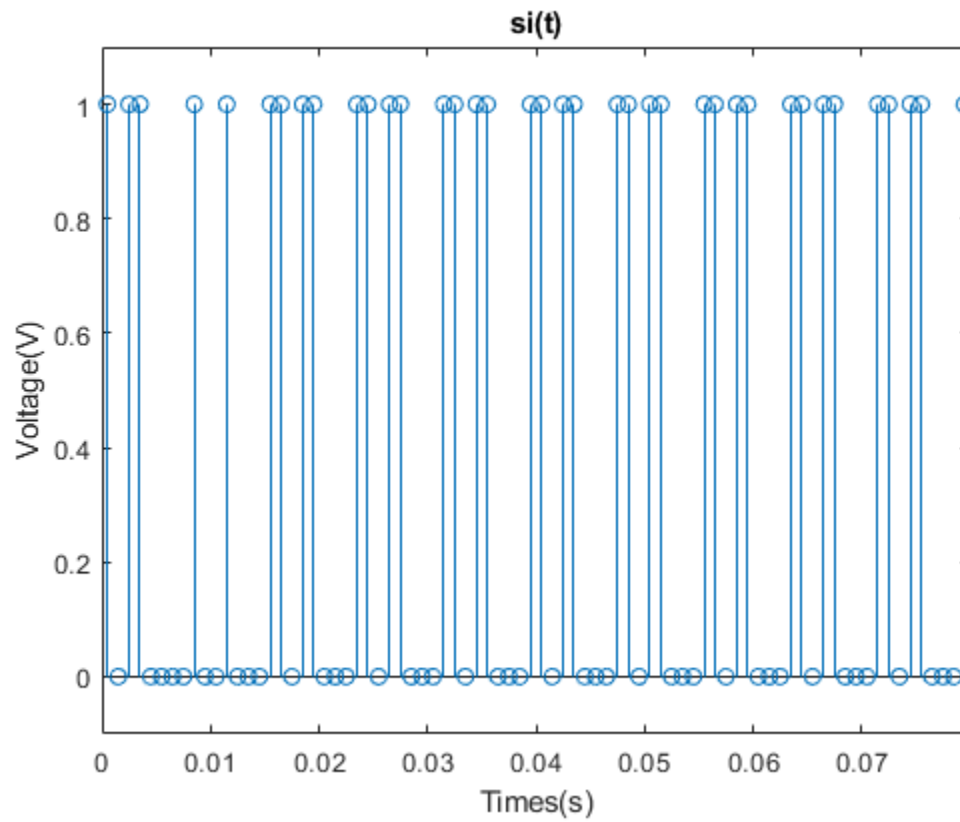
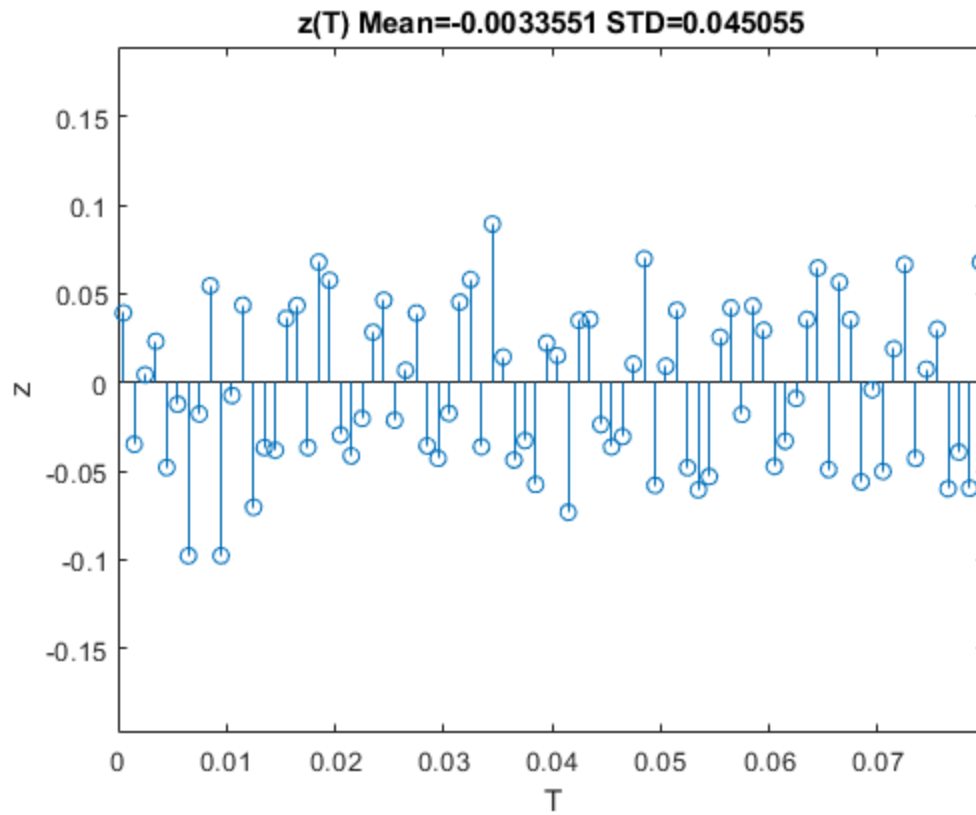
*Standard deviation is 4.505493e-02*

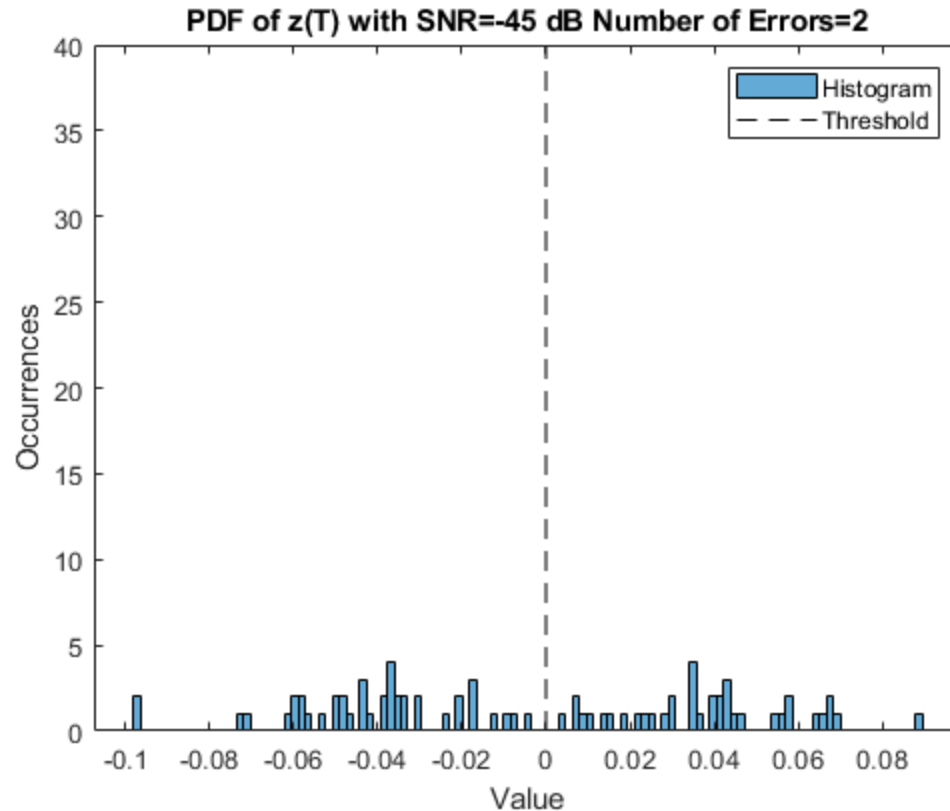
*BER of 0.025000, and Threshold of 0.000000, at SNR of -45 dB*











## 40 Times -50 SNR

```
clc;
clear all;
close all;

%given sequence 40 times
test=[1 0 1 1 0 0 0 1];
test=repmat(test,1,10);

%map to voltages
v_test=test*10-5;

%generate waveform
sample_time=10e-9;
t=0:sample_time:1e-3*length(test)-sample_time;
i=1e-3;
n=1;
for j=1:length(t)
    if t(j)<=n*i
        s(j)=v_test(n);
    else
        n=n+1;
        s(j)=v_test(n);
    end
end
```

```
end

%SNR of -50
SNR=-50;
%%The power of the signal is found by taking the integral of the
signal
%%squared. The SNR is found by dividing the power of the signal by the
%%power of the noise. It is done on Matlab using 'measured'

r=awgn(0.8*s,SNR,'measured');
plot(t,r);
title(['r(t) with Noise SNR=',num2str(SNR)]);
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(r)-1 max(r)+1]);

%optimum detection value= (a1+a2)/2=0
y=0;

%Correlator Detector
%times to sample
for j=1:length(test)
    s_time(j)=t((2*j-1)*(length(t)/length(test)/2));
end

for j=1:length(s_time)
    %integral time
    x=(s_time(j)-s_time(1)):sample_time:s_time(j)+s_time(1);
    %z1
    z1(j)=trapz(x,5.*r((j-1)*length(x)+1:j*length(x)));
    %z2
    z2(j)=trapz(x,-5.*r((j-1)*length(x)+1:j*length(x)));
    %signal in integral time
    r((j-1)*length(x)+1:j*length(x));
    %z
    z(j)=z1(j)-z2(j);
    %decision
    if z(j)>y
        sample(j)=1;
    else
        sample(j)=0;
    end
end

figure;
stem(s_time,z1);
title('z1(T)');
xlabel('T');
ylabel('z1');
axis([min(t) max(t) min(z1)-0.05 max(z1)+0.05]);

figure;
stem(s_time,z2);
```

```
title('z2(T)');
xlabel('T');
ylabel('z2');
axis([min(t) max(t) min(z2)-0.05 max(z2)+0.05]);

figure;
stem(s_time,z);
title(['z(T) Mean=', num2str(mean(z)), ' STD=', num2str(std(z))]);
xlabel('T');
ylabel('z');
axis([min(t) max(t) min(z)-0.1 max(z)+0.1]);

figure;
stem(s_time,sample);
title('si(t)');
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(sample)-0.1 max(sample)+0.1 ]);

errors=0;
for j=1:length(sample)
    if (sample(j)~=test(j))
        errors=errors+1;
    end
end
fprintf('Number of errors is %i\n',errors);
zs1=0;
zs2=0;
for x=1:length(z)
    if z>0.0
        zs1=zs1+1;
    else
        zs2=zs2+1;
    end
end

fprintf('Mean is %d\n',mean(z));
fprintf('Standard deviation is %d\n',std(z));

figure;
h=histogram(z);
for n=1:25
    morebins(h);
end
title(['PDF of z(T) with SNR=',num2str(SNR),' dB Number of Errors=',
    num2str(errors)]);
xlabel('Value');
ylabel('Occurrences');
hold on;

ttime_y=0:1:40;
plot(y*ones(size(ttime_y)),ttime_y,'k --');
legend('Histogram','Threshold');
```

```
BER=errors/length(test);
```

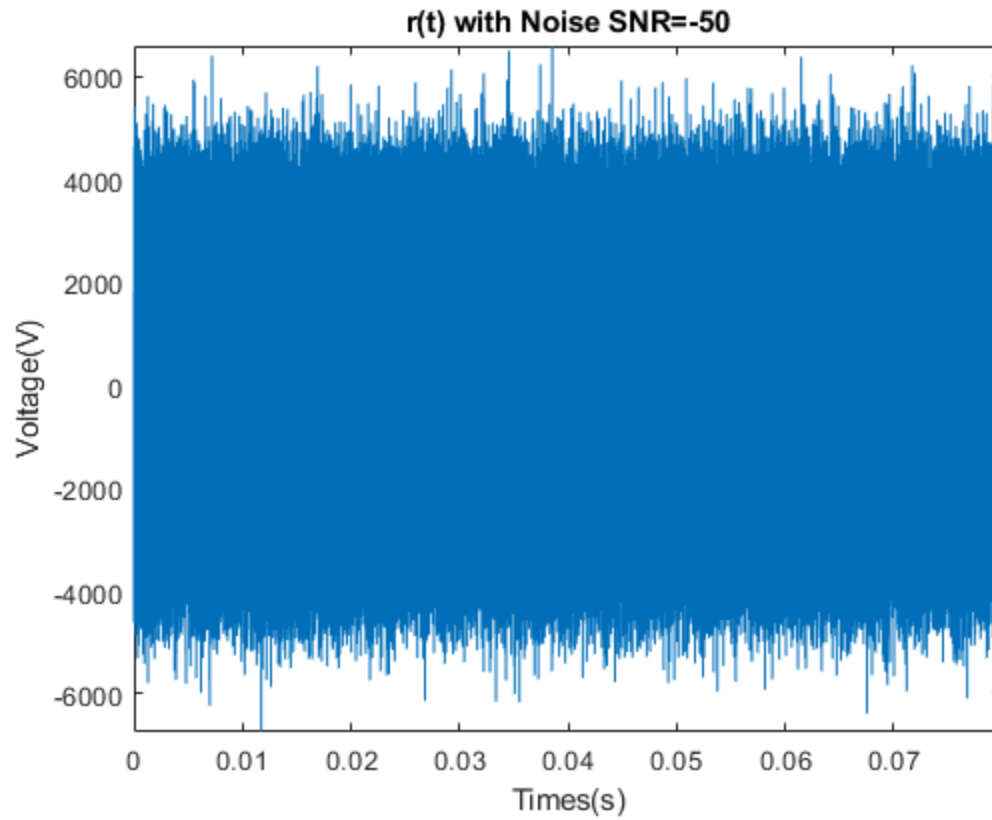
```
fprintf('BER of %f, and Threshold of %f, at SNR of %i dB\n',BER,y,SNR);
```

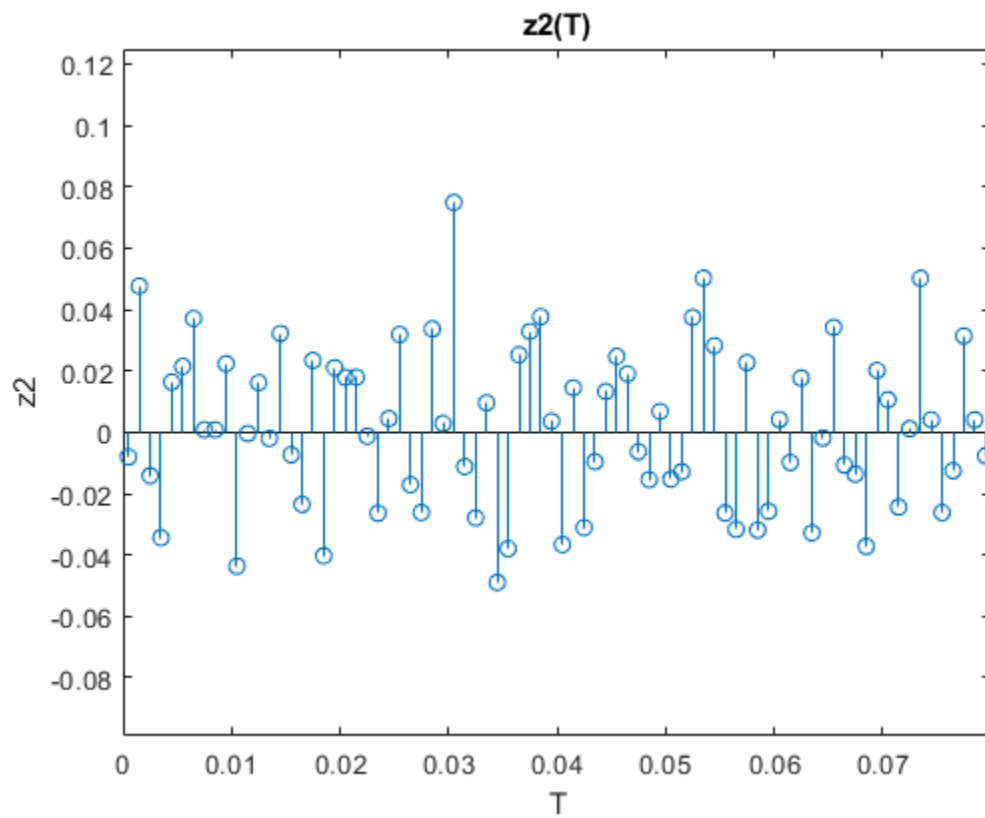
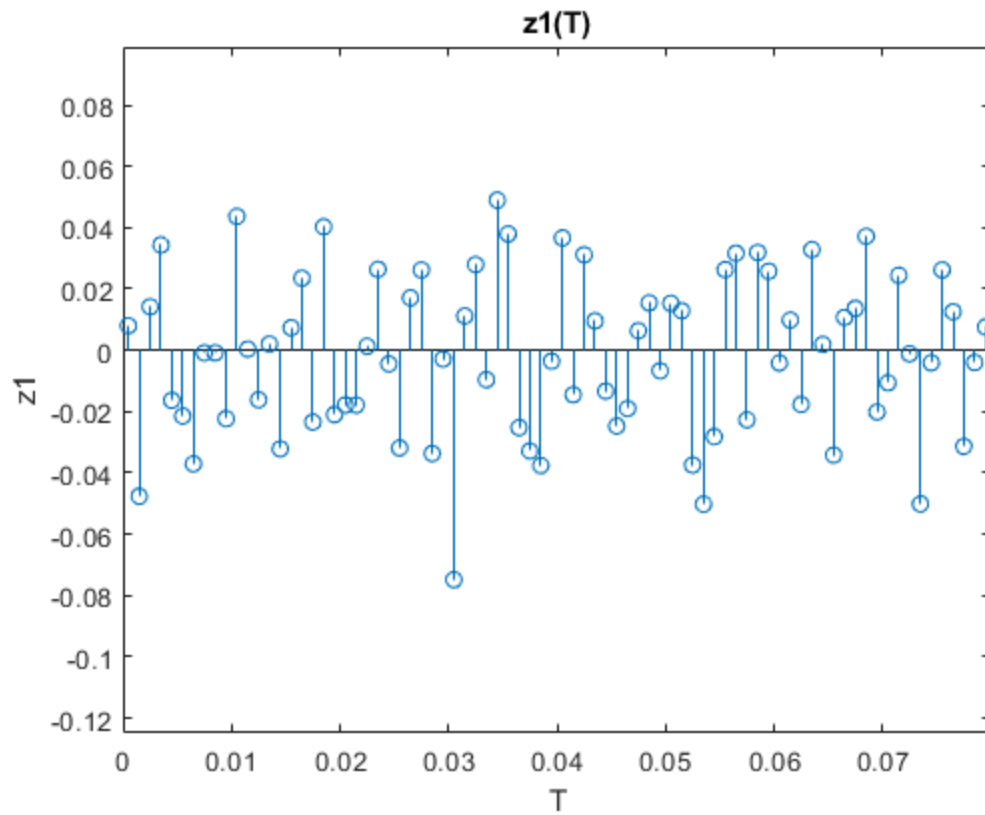
*Number of errors is 12*

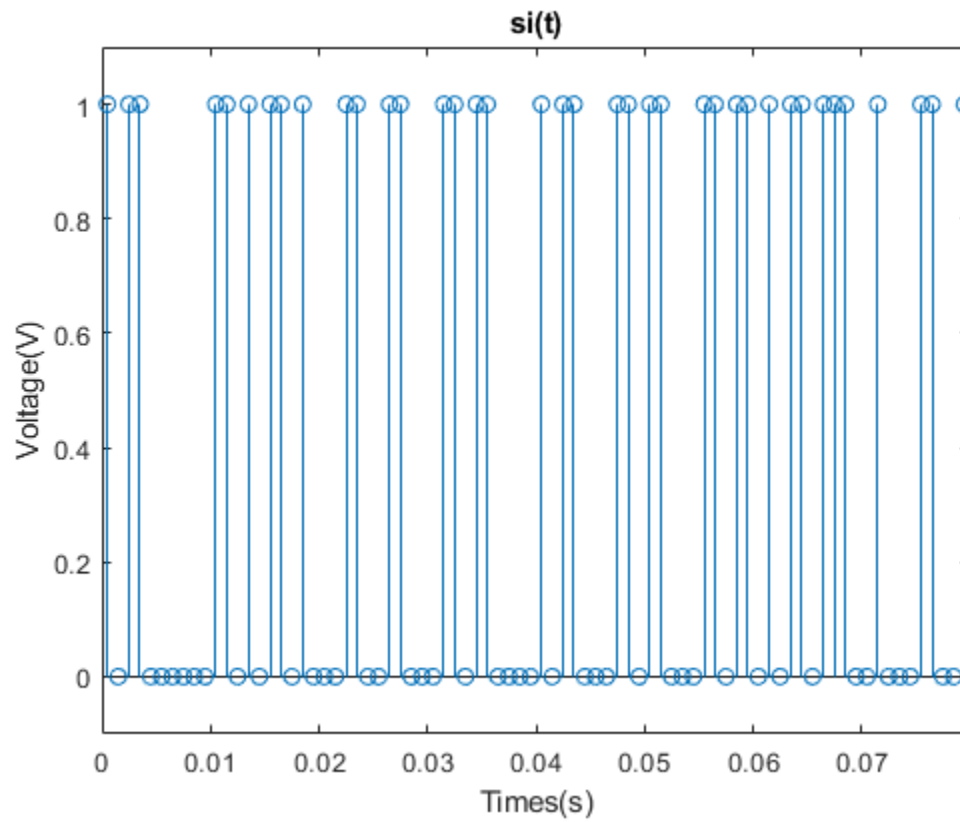
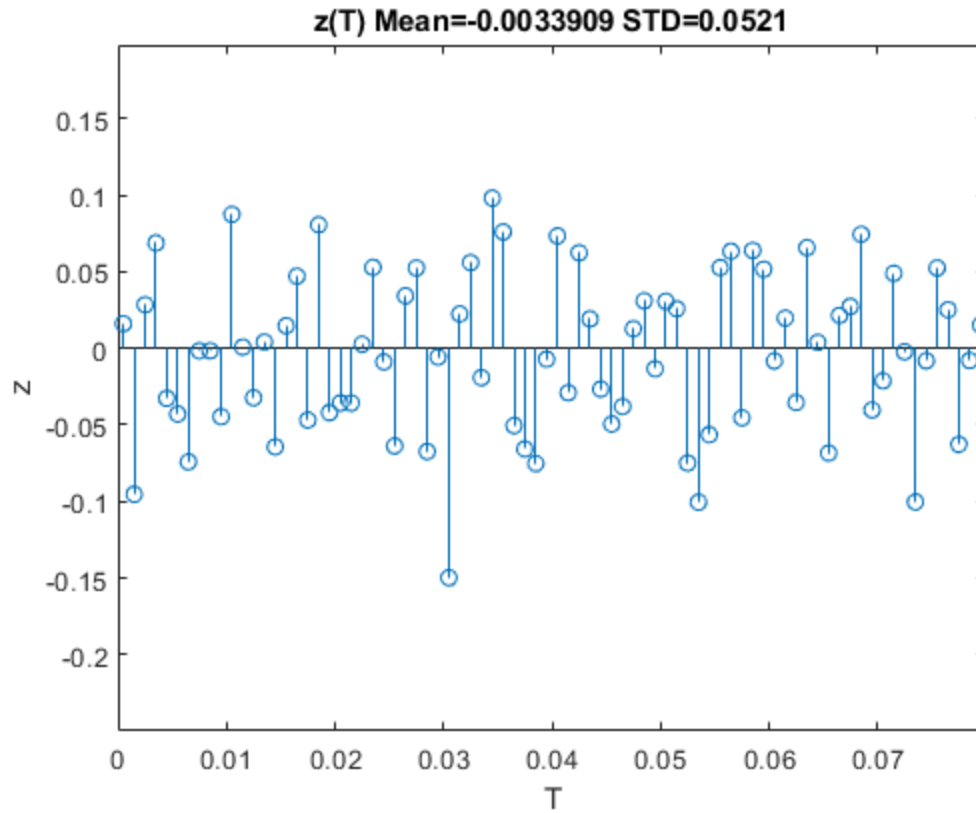
*Mean is -3.390862e-03*

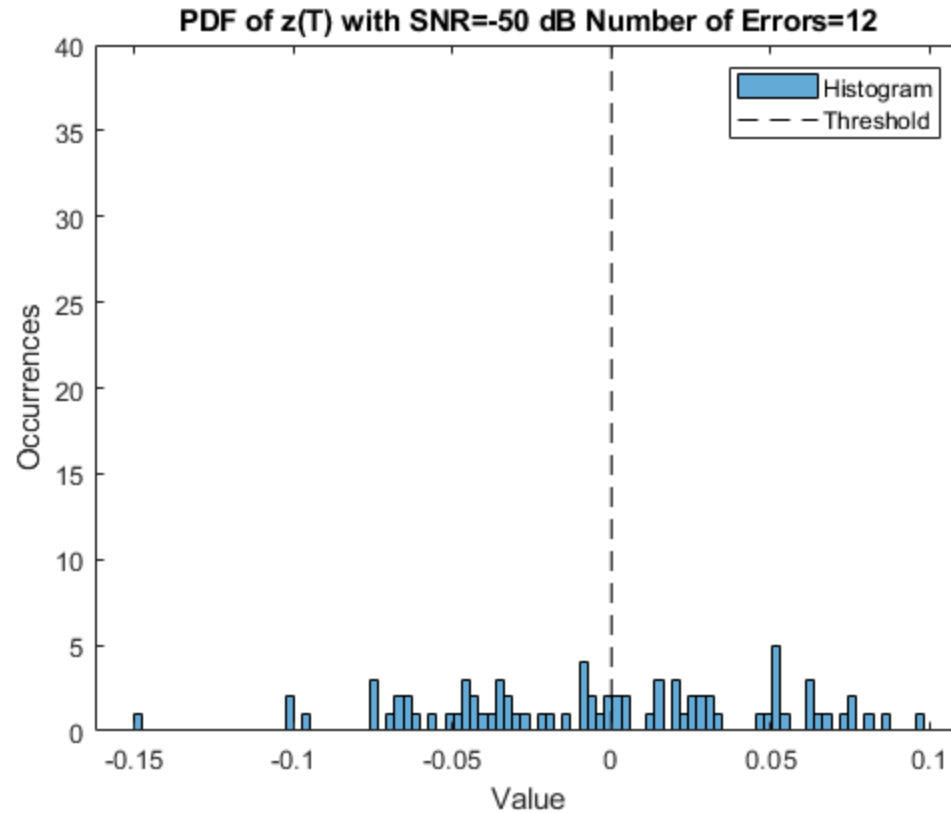
*Standard deviation is 5.210050e-02*

*BER of 0.150000, and Threshold of 0.000000, at SNR of -50 dB*









## 40 Times -60 SNR

```
clc;
clear all;
close all;

%given sequence 40 times
test=[1 0 1 1 0 0 0 1];
test= repmat(test,1,10);

%map to voltages
v_test=test*10-5;

%generate waveform
sample_time=10e-9;
t=0:sample_time:1e-3*length(test)-sample_time;
i=1e-3;
n=1;
for j=1:length(t)
    if t(j)<=n*i
        s(j)=v_test(n);
    else
        n=n+1;
        s(j)=v_test(n);
    end
end
```



```
end

%SNR of -60
SNR=-60;
%%The power of the signal is found by taking the integral of the
signal
%%squared. The SNR is found by dividing the power of the signal by the
%%power of the noise. It is done on Matlab using 'measured'

r=awgn(0.8*s,SNR,'measured');
plot(t,r);
title(['r(t) with Noise SNR=',num2str(SNR)]);
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(r)-1 max(r)+1]);

%optimum detection value= (a1+a2)/2=0
y=0;

%Correlator Detector
%times to sample
for j=1:length(test)
    s_time(j)=t((2*j-1)*(length(t)/length(test)/2));
end

for j=1:length(s_time)
    %integral time
    x=(s_time(j)-s_time(1)):sample_time:s_time(j)+s_time(1);
    %z1
    z1(j)=trapz(x,5.*r((j-1)*length(x)+1:j*length(x)));
    %z2
    z2(j)=trapz(x,-5.*r((j-1)*length(x)+1:j*length(x)));
    %signal in integral time
    r((j-1)*length(x)+1:j*length(x));
    %z
    z(j)=z1(j)-z2(j);
    %decision
    if z(j)>y
        sample(j)=1;
    else
        sample(j)=0;
    end
end

figure;
stem(s_time,z1);
title('z1(T)');
xlabel('T');
ylabel('z1');
axis([min(t) max(t) min(z1)-0.05 max(z1)+0.05]);

figure;
stem(s_time,z2);
```

```
title('z2(T)');
xlabel('T');
ylabel('z2');
axis([min(t) max(t) min(z2)-0.05 max(z2)+0.05]);

figure;
stem(s_time,z);
title(['z(T) Mean=', num2str(mean(z)), ' STD=', num2str(std(z))]);
xlabel('T');
ylabel('z');
axis([min(t) max(t) min(z)-0.1 max(z)+0.1]);

figure;
stem(s_time,sample);
title('si(t)');
xlabel('Times(s)');
ylabel('Voltage(V)');
axis([min(t) max(t) min(sample)-0.1 max(sample)+0.1 ]);

errors=0;
for j=1:length(sample)
    if (sample(j)~=test(j))
        errors=errors+1;
    end
end
fprintf('Number of errors is %i\n',errors);
zs1=0;
zs2=0;
for x=1:length(z)
    if z>0.0
        zs1=zs1+1;
    else
        zs2=zs2+1;
    end
end

fprintf('Mean is %d\n',mean(z));
fprintf('Standard deviation is %d\n',std(z));

figure;
h=histogram(z);
for n=1:25
    morebins(h);
end
title(['PDF of z(T) with SNR=', num2str(SNR), ' dB Number of Errors=',
    num2str(errors)]);
xlabel('Value');
ylabel('Occurrences');
hold on;

ttime_y=0:1:40;
plot(y*ones(size(ttime_y)),ttime_y,'k --');
legend('Histogram', 'Threshold');
```

```
BER=errors/length(test);
```

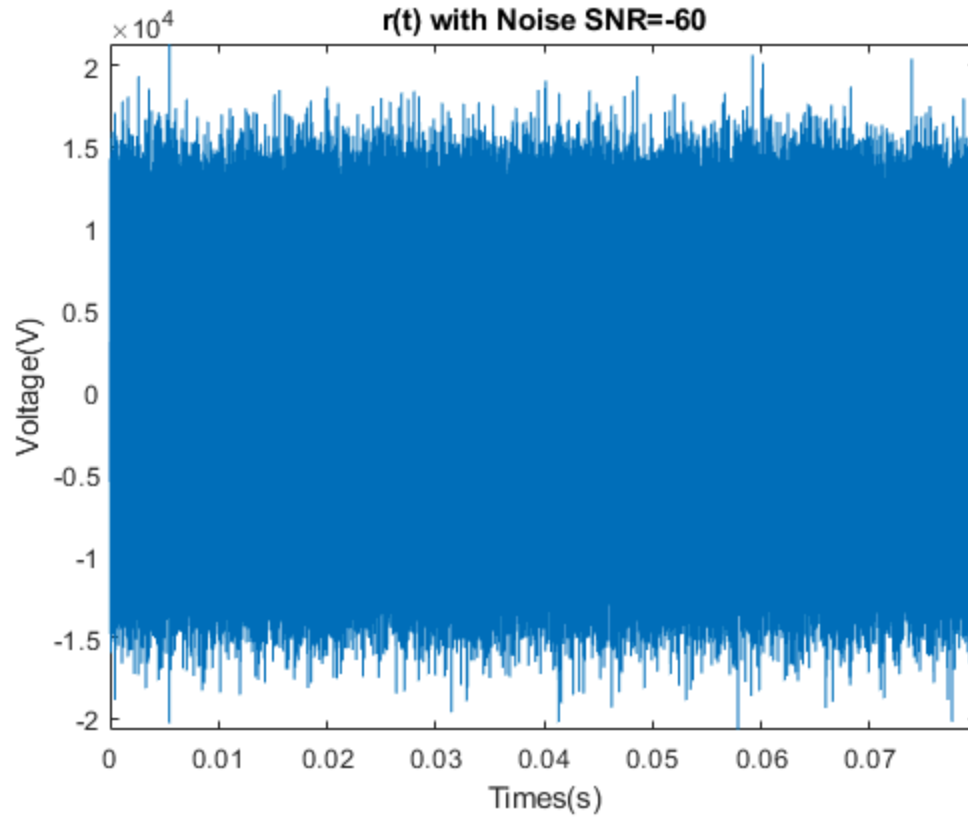
```
fprintf('BER of %f, and Threshold of %f, at SNR of %i dB\n',BER,y,SNR);
```

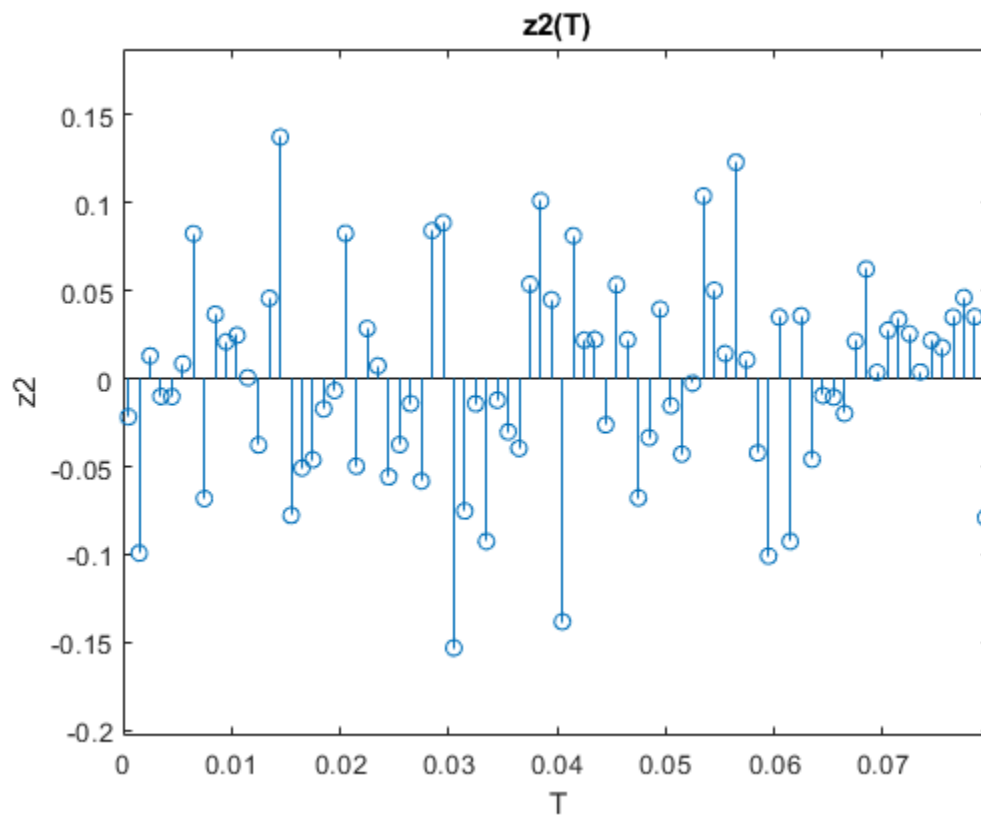
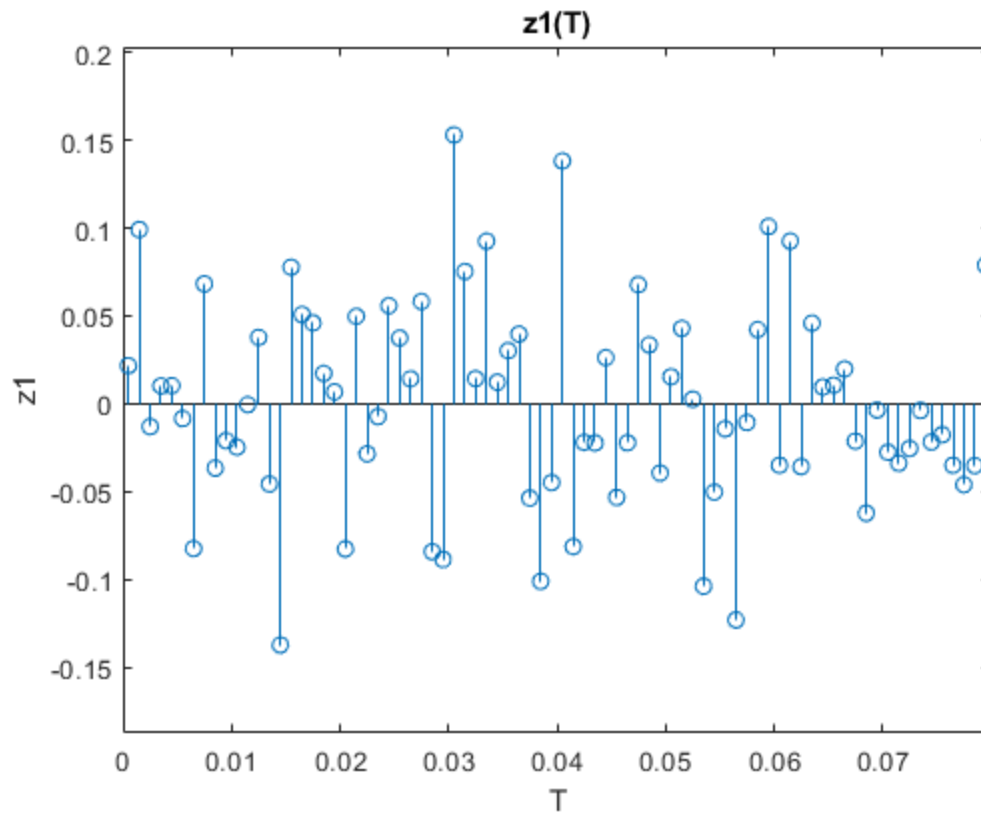
*Number of errors is 28*

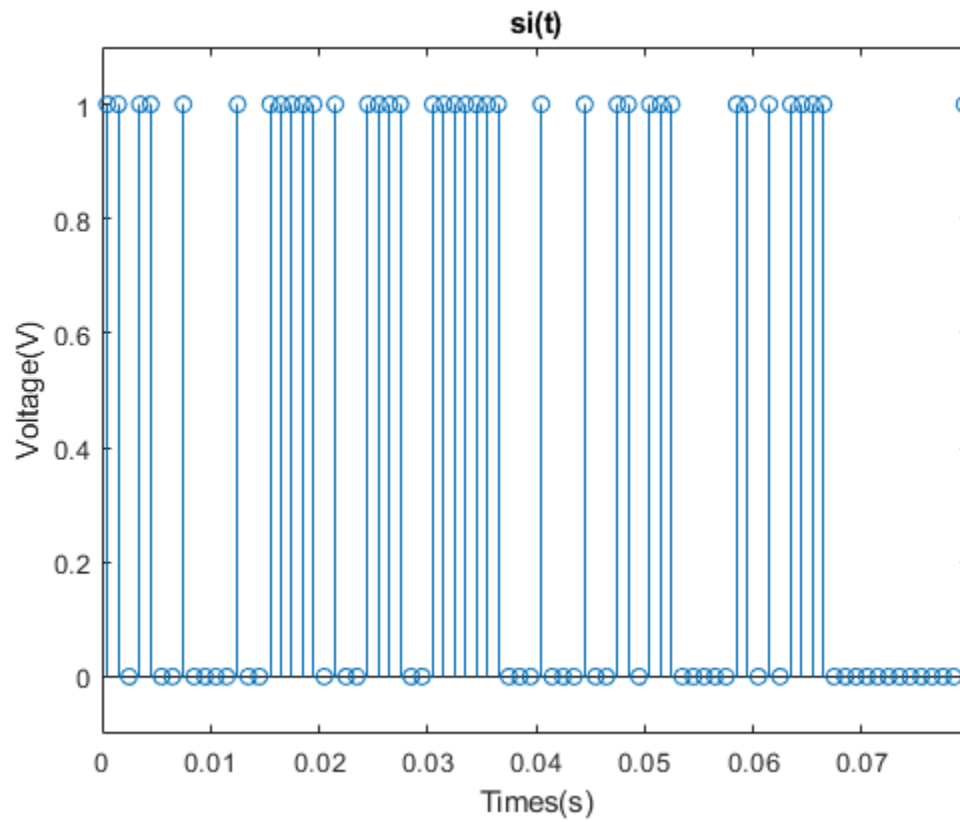
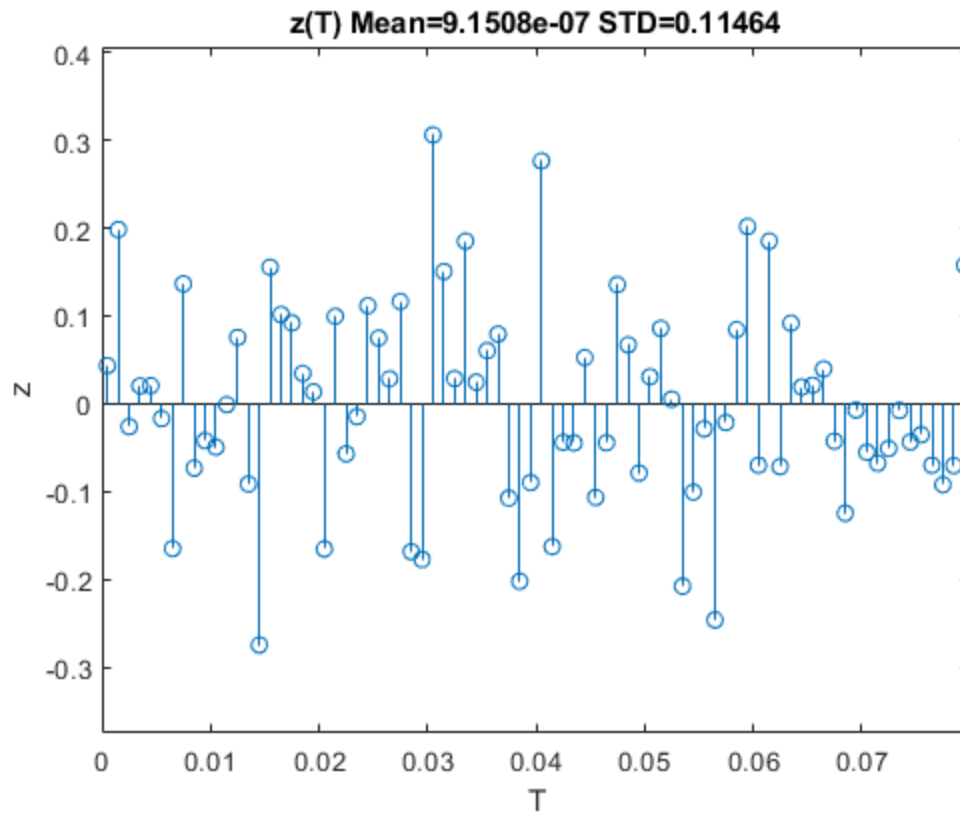
*Mean is 9.150812e-07*

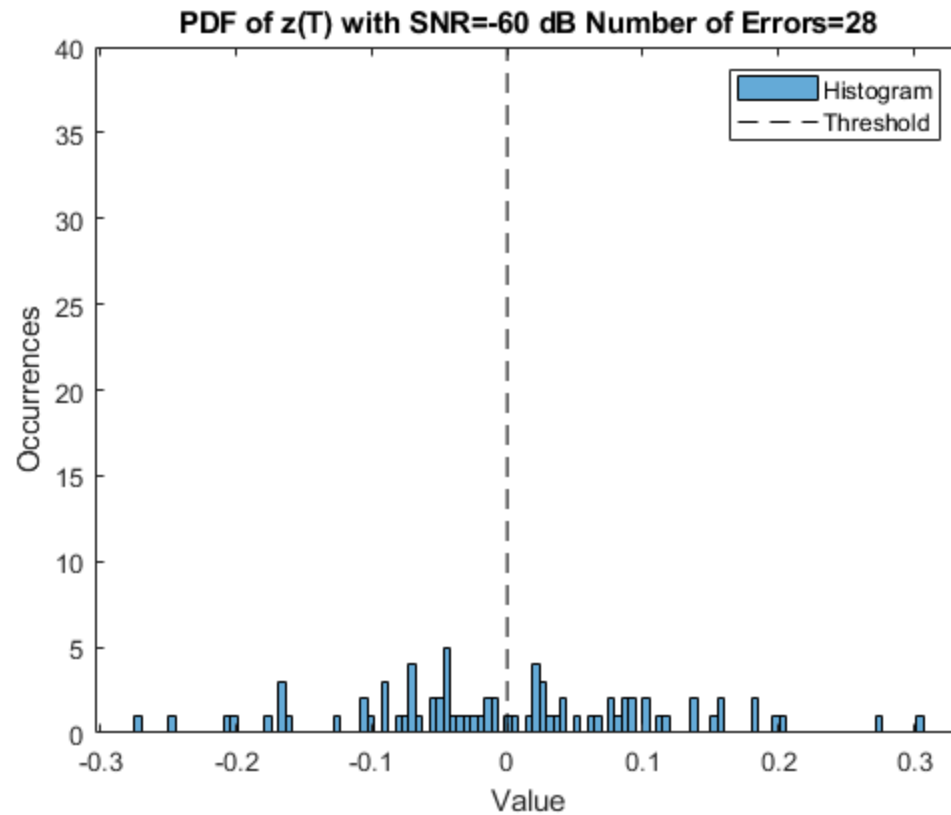
*Standard deviation is 1.146443e-01*

*BER of 0.350000, and Threshold of 0.000000, at SNR of -60 dB*



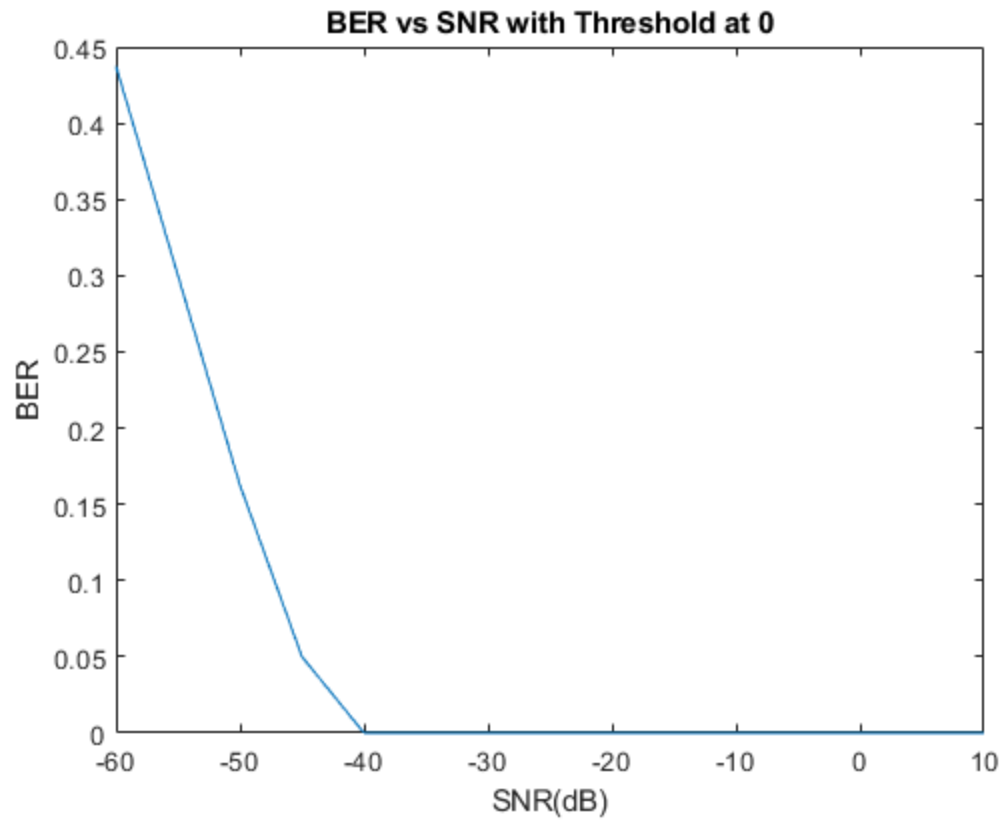






## Error Freq.

```
figure;  
BER_A=[0 0 0 0.05 0.162500 0.437500];  
SNR_A=[10 0 -40 -45 -50 -60];  
plot(SNR_A,BER_A);  
title('BER vs SNR with Threshold at 0');  
xlabel('SNR(dB)');  
ylabel('BER');
```



*Published with MATLAB® R2018b*