EE 5356 Digital Image Processing

LAB Assignment 1 A

1. Lloyd Max Quantiser (with S.D=2)
   1. Matlab program -

clc;

clear all;

%Set desired L value eg:2,4,8,16,32,64,128

L=8

error=1;

rk=zeros(1,L);

%Generate a sample sequence for tk

tk(1)=0;

for i=2:L

tk(i)=tk(i-1)+(10/(L-1));

end

tk(L+1)=10000;

F1 = @(x) (x./1.5^2).\*exp(-power(x,2)./(2\*(1.5^2)));

F2 = @(x) x.\*(x./1.5^2).\*exp(-power(x,2)./(2\*(1.5^2)));

%Minimize error

while (error>0.001)

error=0;

entropy=0;

%Using tk, calculate rk

for k=1:1:L

num=quad(F2,tk(k),tk(k+1));

den=quad(F1,tk(k),tk(k+1));

r(k)=num/den;

entropy=entropy-(den\*log2(den));

end

%Determine error of received samples

for i=1:L

error=error+abs((r(i)-rk(i)));

end

rk=r;

%Calculate new tk using rk values

for k=2:L

tk(k)=(r(k)+r(k-1))/2;

end

end

%Calculate MSE

MSE=0;

for k=1:L

F=@(u) (((u-r(k)).^2).\*(u./1.5^2).\*exp(-(u.^2)./(2\*(1.5^2))));

MSE=MSE+quad(F,tk(k),tk(k+1));

end

%Calculate SNR

SNR=-10\*log10(MSE);

tk=tk';

rk=r';

len = length(tk);

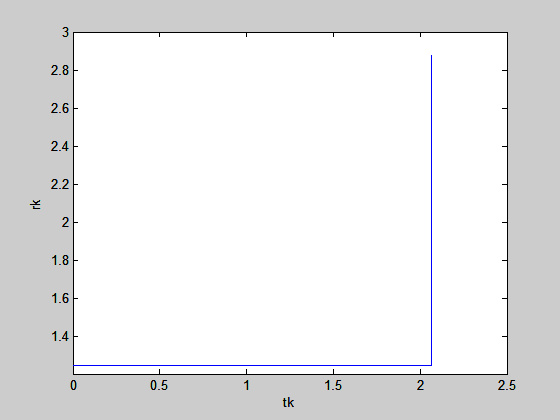
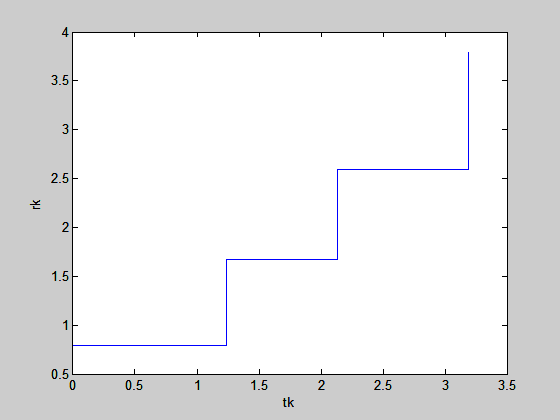
tk1 = tk(1:len-1);

stairs(tk1,rk);

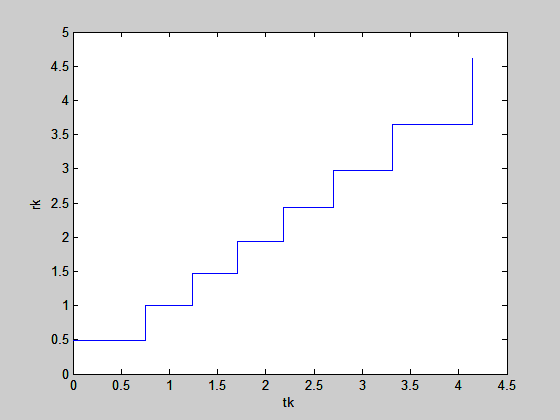
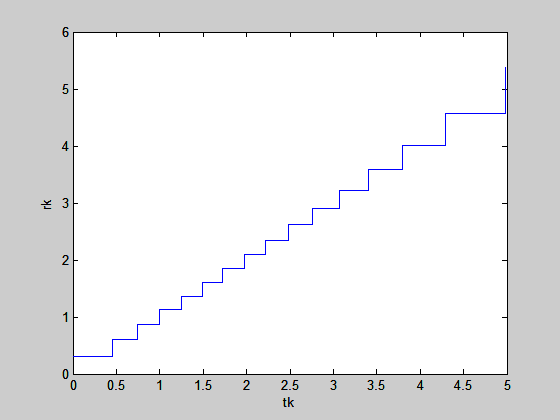
xlabel('tk')

ylabel('rk')

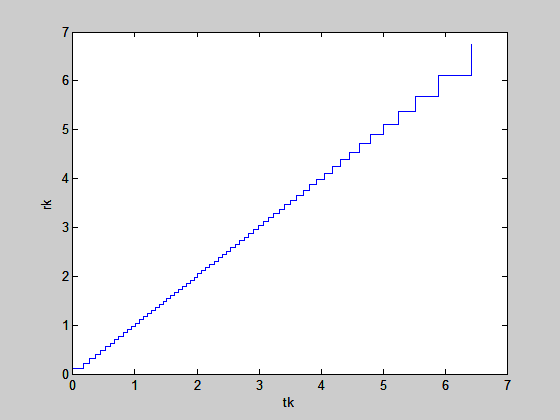
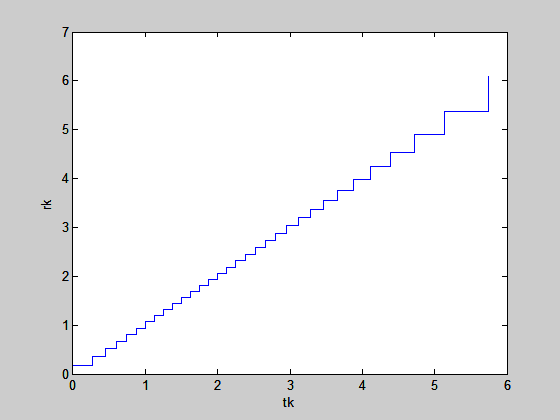
* 1. plots:

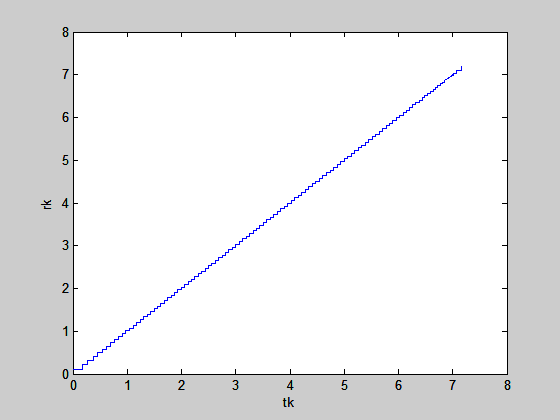
L=2 L=4

L=8 L=16



L=32 L=64



L=128

1. Uniform Mean Square Quantiser
2. Matlab program -

clc;

clear all;

%Set desired L value eg: 2,4,8,16,32,64,128

L=8

%Initialize rk and tk t zero

rk=zeros(1,L);

tk=zeros(1,L);

%Range of tk=10

tk(L+1)=10;

A=tk(L+1)-tk(1);

%Calculate the tk values from the initial zero sequence and range A

for k=1:L

F=@(x) power((x/1.5^2).\*exp(-(x.^2)/(2\*(1.5^2))),(-1/3));

z=(k/L)\*A+tk(1);

a=quad(F,tk(1),z+tk(1));

num=A\*a;

den=quad(F,tk(1),tk(L+1));

tk(k+1)=(num/den)+tk(1);

end

%Calculate corresponding rk values

for k=1:L

rk(k)=(tk(k)+tk(k+1))/2;

end

%Calculate MSE

F1=@(x) power((x/1.5^2).\*exp(-(x.^2)/(2\*(1.5^2))),(1/3));

q=quad(F1,tk(1),tk(L+1));

MSE=(1/(12\*(L^2))\*(q^3));

%Calculate SNR

SNR=-10\*log10(MSE);

%Calculate entropy

entropy=0;

F2=@(x) (x/1.5^2).\*exp(-(x.^2)/(2\*(1.5^2)));

for k=1:L

a=quad(F2,tk(k),tk(k+1));

entropy=entropy-(a\*log2(a));

end

tk=tk';

rk=rk';

len = length(tk);

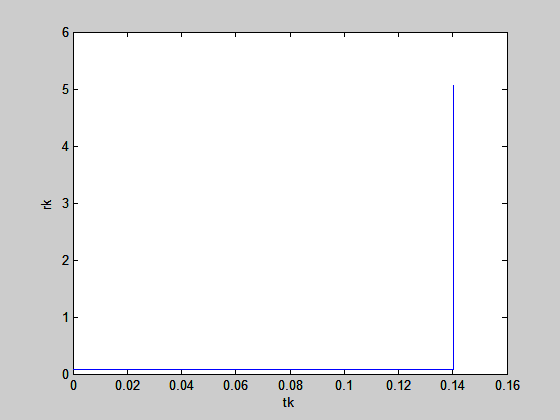
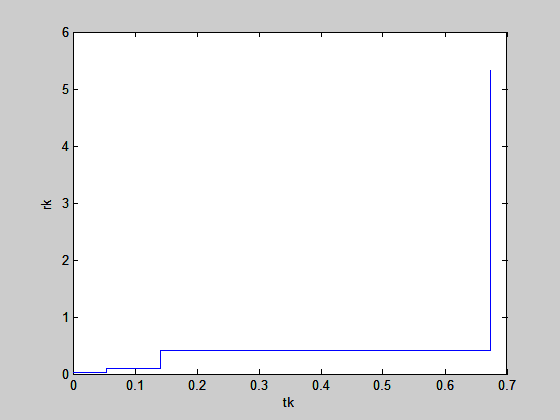
tk1 = tk(1:len-1);

stairs(tk1,rk);

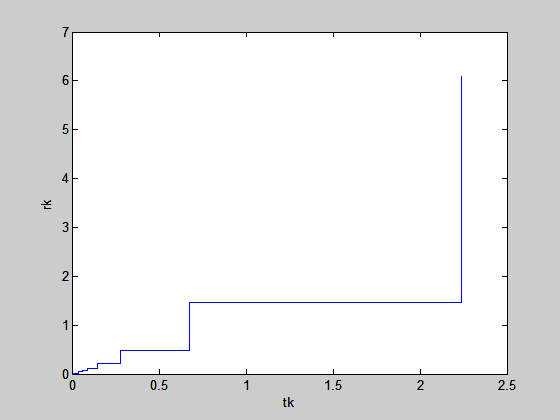
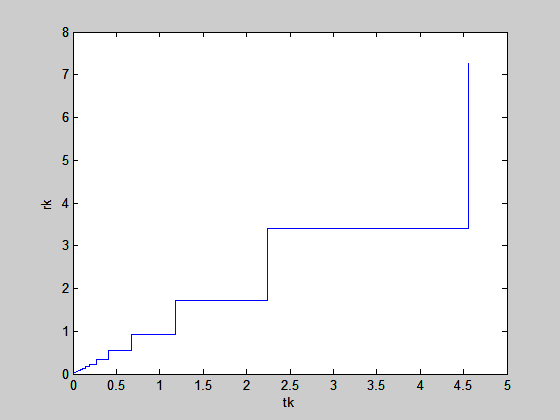
xlabel('tk')

ylabel('rk')

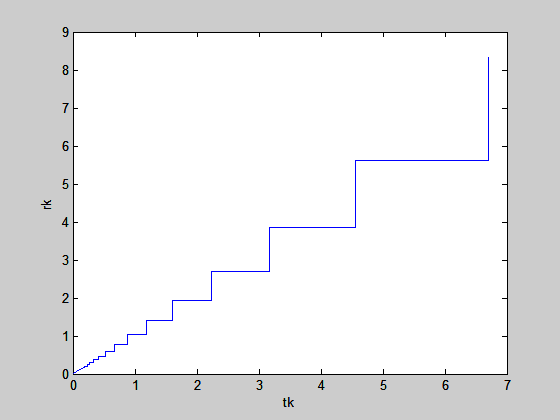
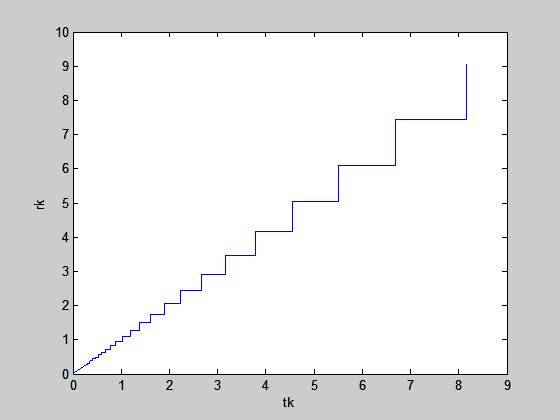
1. Plots –

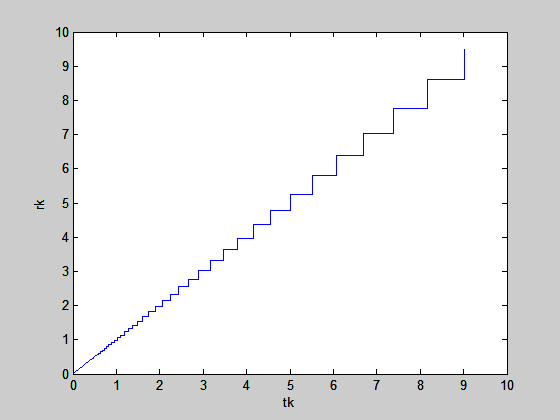
L=2 L=4

L=8 L=16

L=32 L=64



L=128

1. Optimum Mean Square Uniform Quantiser
   1. Matlab code –

clc;

clear all;

sigma=1;

sd=1;

L=2;

g2=0;

g=0;

B=0.1;

q=1;

for i = 1:80

g1=0;

for j=1:(L/2)-1

I = @(u)((u-((2.\*j-1).\*q)/2).\*(u.\*exp(-1.\*(u.^2)/2)));

g1 = g1-2.\*((2.\*j-1).\*quad(I,(j-1).\*q,j.\*q));

end;

I1= @(u)((u-((L-1).\*q)/2).\*(u.\*exp(-1.\*(u.^2)/2)));

g2=g2-2.\*(L-1).\*quad(I1,(L/2-1).\*q,10);

g=g1+g2;

q = q-B.\*g;

end;

tk=zeros(1,L+1);

rk=zeros(1,L+1);

for i=1:L+1

tk(i)=q.\*(i-1);

end;

for i=1:L

rk(i)=(tk(i)+tk(i+1))/2;

end;

rk(L+1) = rk(L);

a = L\*q/2;

MSE=0;

for i=1:L

I2=@(u)((u-rk(i)).^2.\*u.\*exp(-1\*(u.^2)/2));

MSE=MSE+quad(I2,tk(i),tk(i+1));

end;

SNR=10\*log10(1/MSE);

I3= @(u)(u.\*exp(-1.\*(u.^2)/2));

F=@(u)((u/sigma).\*exp(-u.^2/(2\*sigma.^2)));

entropy=0;

for i=1:1:L

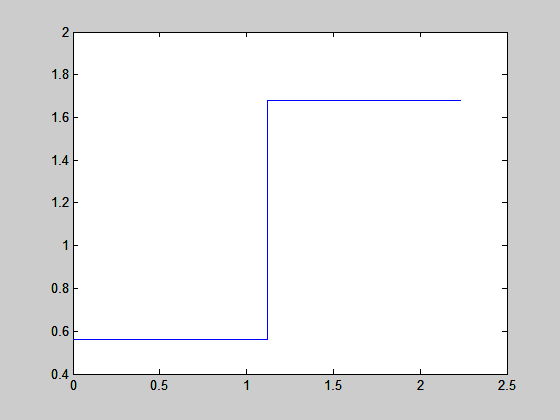
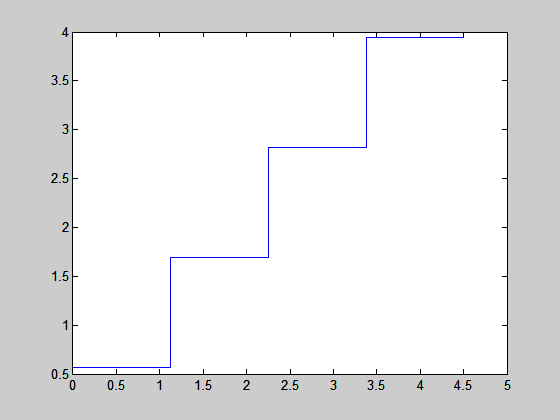
a2(i)=quad(F,tk(i),tk(i+1));

entropy=entropy-(a2(i)\*log2(a2(i)));

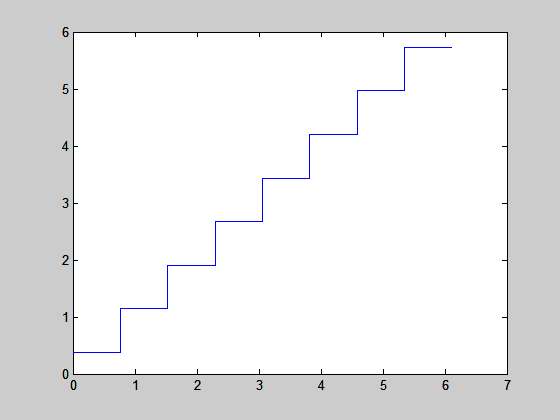
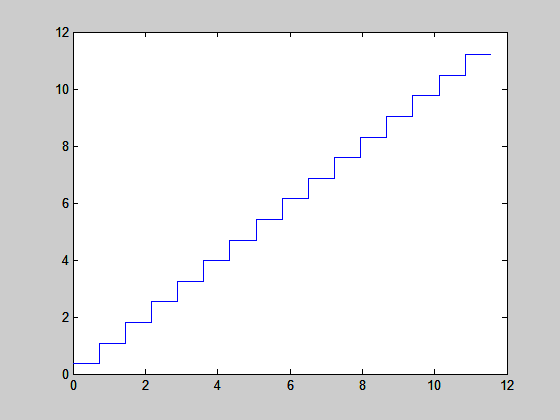
end

stairs(tk,rk);

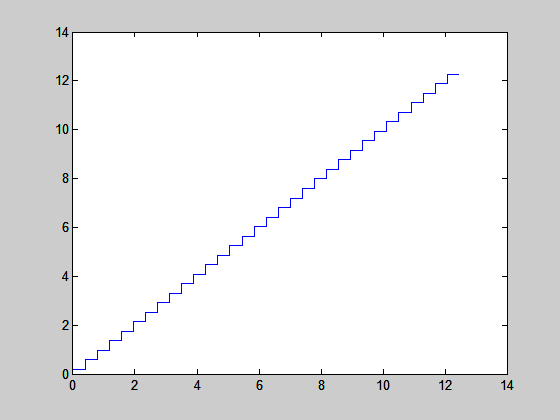
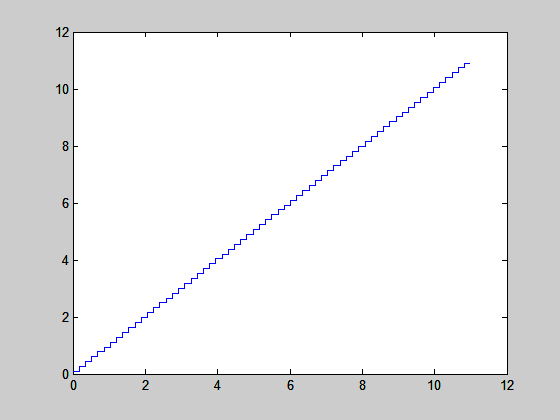
* 1. Plots -

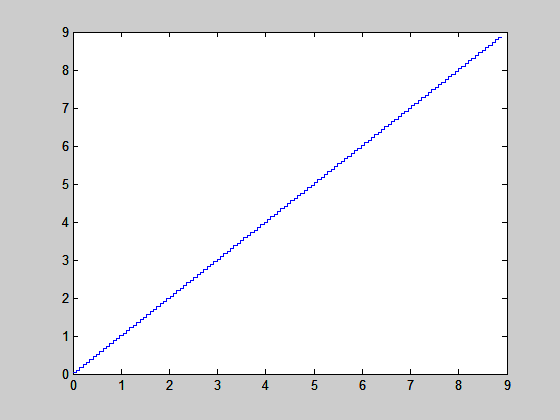
L=2 L=4

L=8 L=16

L=32 L=64



L=128