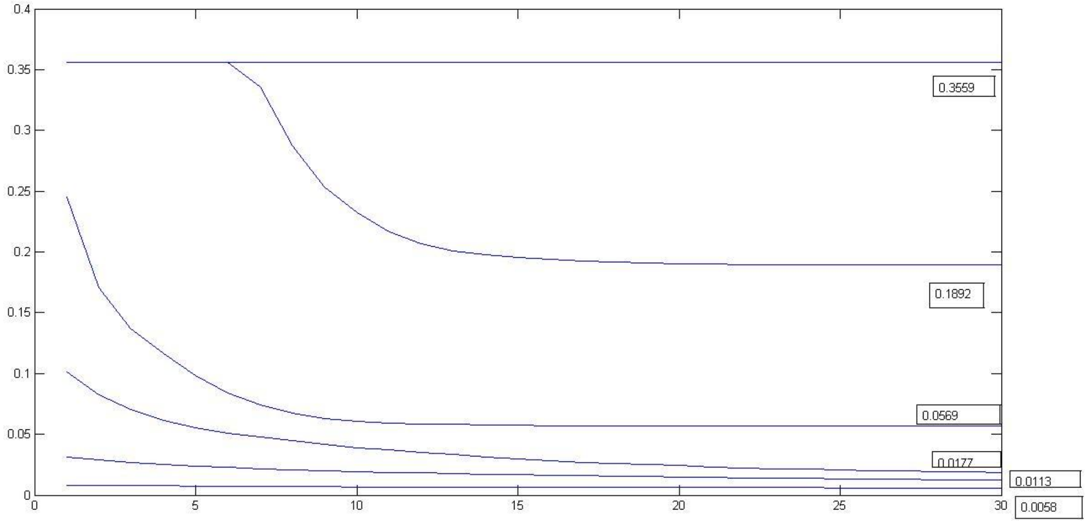
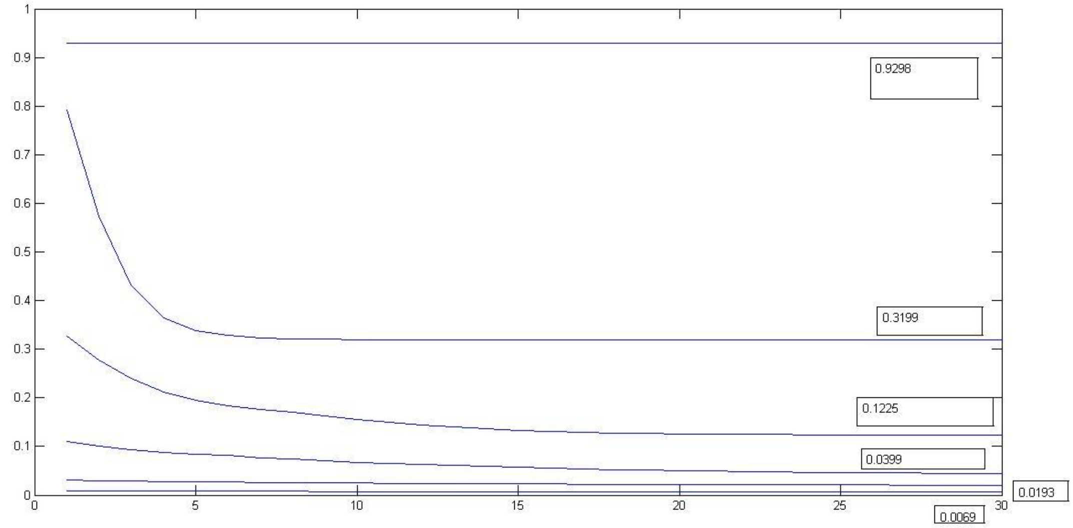
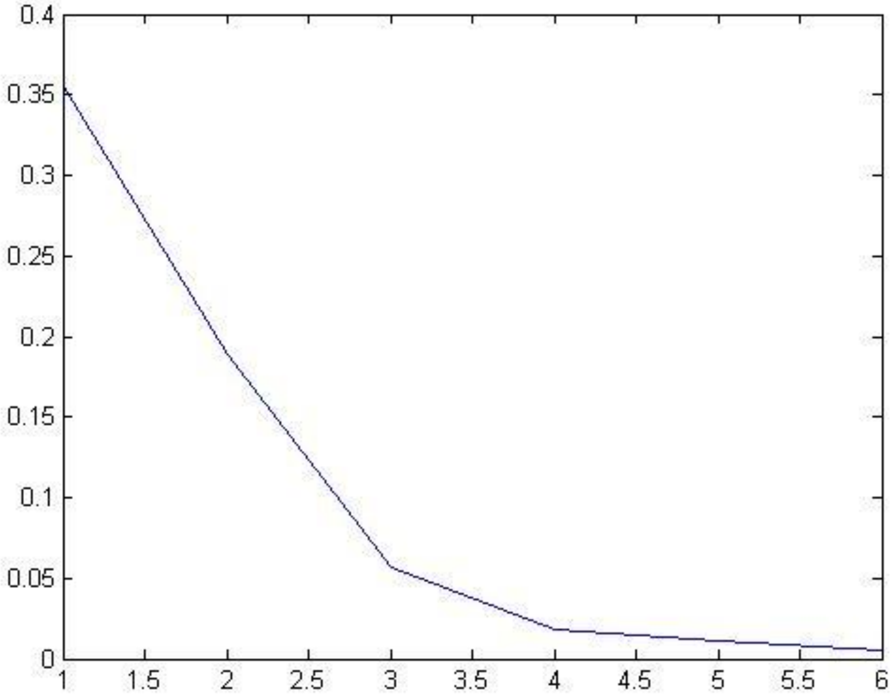
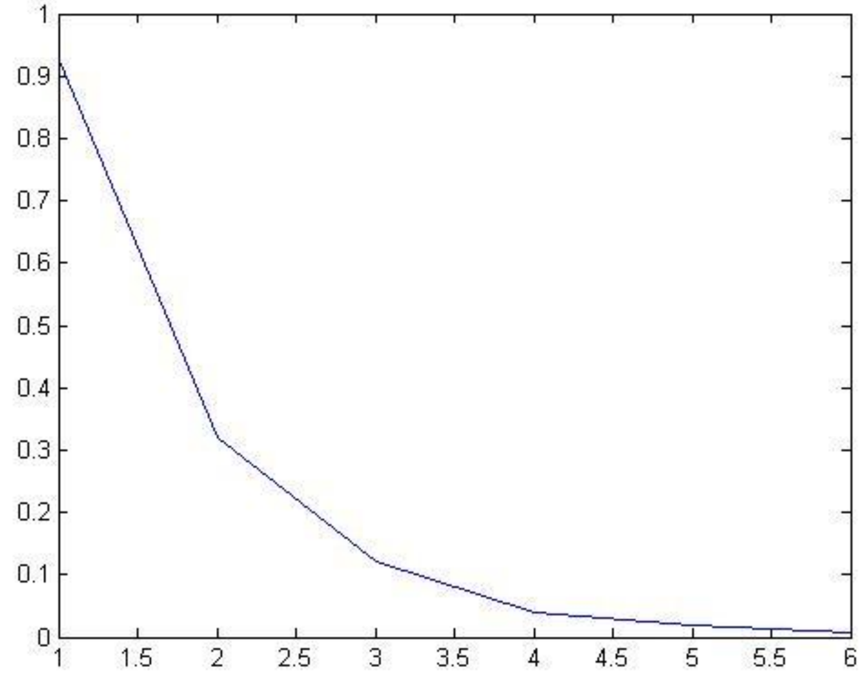
```
%%      Lloyd-Max Algorithm
clear all;
close all;
for q=1:6%%no. of bits
    n_bits = q;
    q_levels = 2^n_bits;%% no of levels
    minv = -10;%% dynamic range
    maxv = 10;
    len=(-1*minv+maxv)/q_levels;%%length of interval
    m=zeros(1,q_levels+1);
    for i=1:q_levels+1
        m(i)=minv+(i-1)*len;%%initialize
    end
    load('Dat_2.mat');
    sig=X;
    sig=sort(sig);
    lu=zeros(1,q_levels);
    sk=zeros(1,q_levels);
    for i=1:100%%iterations
        for k=1:q_levels
            [sk(k),lu(k)]=cent(m(k),m(k+1),sig,k,q_levels);
            new(k)=sk(k)/lu(k);%%centroid
        end
        for k=2:q_levels
            m(k)=(new(k-1)+new(k))/2;%%new intervals
        end
        for h=1:q_levels%%MSE calculation
            for t=1:10000
                if(X(t)<m(h+1) && X(t)>=m(h))
                    Y(t)=new(h);
                end
            end
        end
        a=X-Y;
        b=a.^2;
        mse1(i)=sum(b)/10000;
    end
    plot(mse1)
    hold on
end


function [ sk,lu ] = cent( mint,mfin,sig,k,levels )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
lu=0;
sk=0;
for j=1:10000
    if(sig(j)<mfin && sig(j)>=mint)
        lu=lu+1;
        sk=sk+sig(j);
    end
end
if lu==0
    sk=(mint+mfin)/2;
    lu=1;
```

```
end
% if (lu==0 && k <= levels/2)
%     sk = mint;
%     lu=1;
% elseif (lu == 0 && k > levels/2)
%     sk = mfin;
%     lu=1;
% end

end
```






0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6