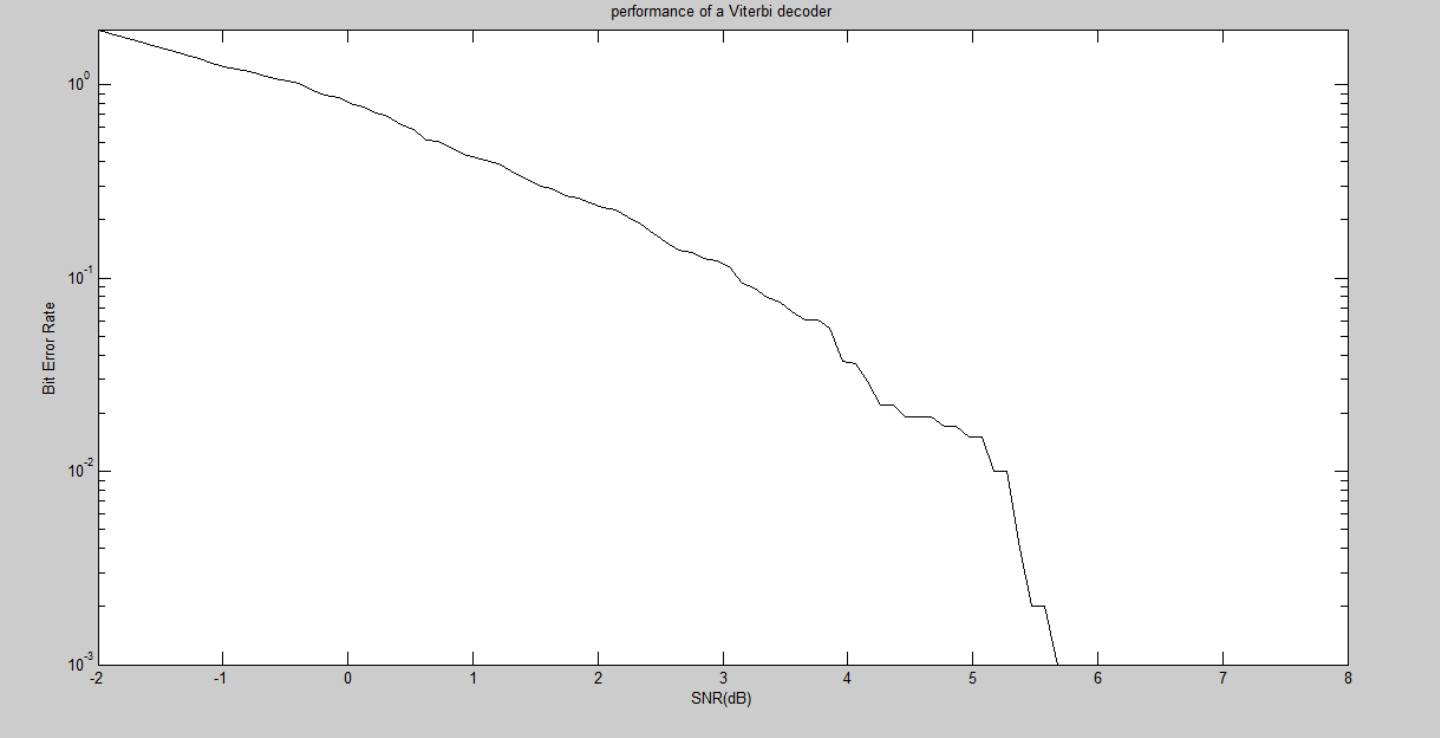
M-File

```
clear;no=1000;l=100;
x=rand(10,no);                          %symbol source assumed to be random
x=round(x);
g0=[1 0 1];                             %impulse response of first output stream
g1=[1 1 1];                            %impulse response of second output stream
for t=1:10
c0=mod(conv(x(t,:),g0),2);     %first output stream
c1=mod(conv(x(t,:),g1),2);     %second output stream
c0_psk=(2*c0)-1;
c1_psk=(2*c1)-1;
snrdB=linspace(-2,6,l);
p=10.^(-snrdB./20);                  %standard deviation of noise
n=p'*randn(1,(no+2));                %normally distributed noise
c0_psk=repmat(c0_psk,l,1);c1_psk=repmat(c1_psk,l,1);
y0=c0_psk+n;
y1=c1_psk+n;
wt_s(1,:)=[0 1000 1000 1000];
out1_s=[-1 -1;-1 1;1 1;1 -1];
out2_s=[1 1;1 -1;-1 -1;-1 1];
for g=1:l
    error=0;
    for i=1:no+2
        r(i,:)=[y0(g,i) y1(g,i)];
        for j=1:4
            a(i,j)=sum((r(i,:)-out1_s(j,:)).*(r(i,:)-out1_s(j,:)));
            b(i,j)=sum((r(i,:)-out2_s(j,:)).*(r(i,:)-out2_s(j,:)));
        end
        wt_s(i+1,1)=min([wt_s(i,1)+a(i,1) wt_s(i,3)+a(i,3)]);
        wt_s(i+1,2)=min([wt_s(i,1)+b(i,1) wt_s(i,3)+b(i,3)]);
        wt_s(i+1,3)=min([wt_s(i,2)+a(i,2) wt_s(i,4)+a(i,4)]);
        wt_s(i+1,4)=min([wt_s(i,2)+b(i,2) wt_s(i,4)+b(i,4)]);
    end
    state=1;d=zeros(2,no+2);
    for i=no+2:-1:1
        switch (state)
            case 1
                if((wt_s(i,1)+a(i,1))<=(wt_s(i,3)+a(i,3)))
                    d(:,i)=[0;0];state=1;
                else
                    d(:,i)=[1;1];state=3;
                end
            case 2
                if((wt_s(i,1)+b(i,1))<=(wt_s(i,3)+b(i,3)))
                    d(:,i)=[1;1];state=1;
                else
                    d(:,i)=[0;0];state=3;
                end
            case 3
                if((wt_s(i,2)+a(i,2))<=(wt_s(i,4)+a(i,4)))
                    d(:,i)=[0;1];state=2;
                else
                    d(:,i)=[1;0];state=4;
                end
            case 4
```

```
                if((wt_s(i,2)+b(i,2))<=(wt_s(i,4)+b(i,4)))
                   d(:,i)=[1;0];state=2;
                else
                    d(:,i)=[0;1];state=4;
                end
        end
    end
    x_d(t,1)=d(1,1);x_d(t,2)=d(1,2);
    for k=3:no
        x_d(t,k)=mod((d(1,k)+x_d(t,k-2)),2);    %demodulated information bits
    end
    error=sum(xor(x(t,:),x_d(t,:)));
    ber(t,g)=error/no;                    %bit error rate
end
end
ber=sum(ber);
snr=10.^(0.1.*snrdB);
ber_a=0;
for j=0:10
    ber_a=ber_a+((2^j)*(j+1)*qfunc(sqrt((j+5)*snr))); %analytical bit error
rate
end
semilogy(snrdB,ber,'k-',snrdB,ber_a,'r-')
xlabel('SNR(dB)'); ylabel('Bit Error Rate');title('performance of a Viterbi
decoder');
axis tight;
```

BER vs SNR for Viterbi Decoder from simulation



BER vs SNR for Viterbi Decoder from simulation and analytical expression