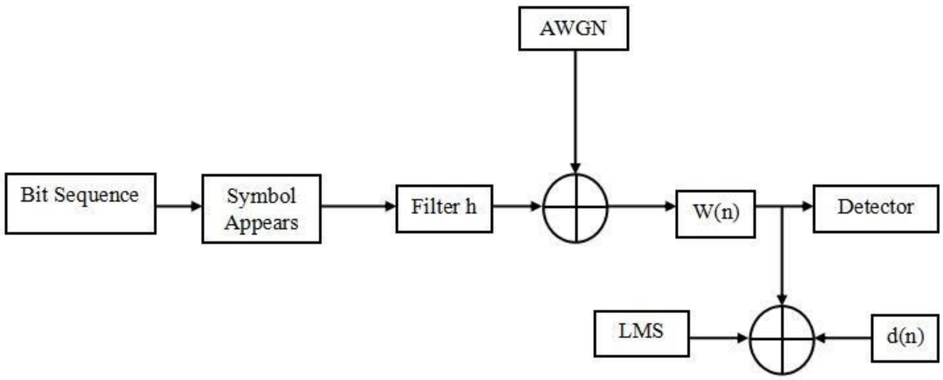
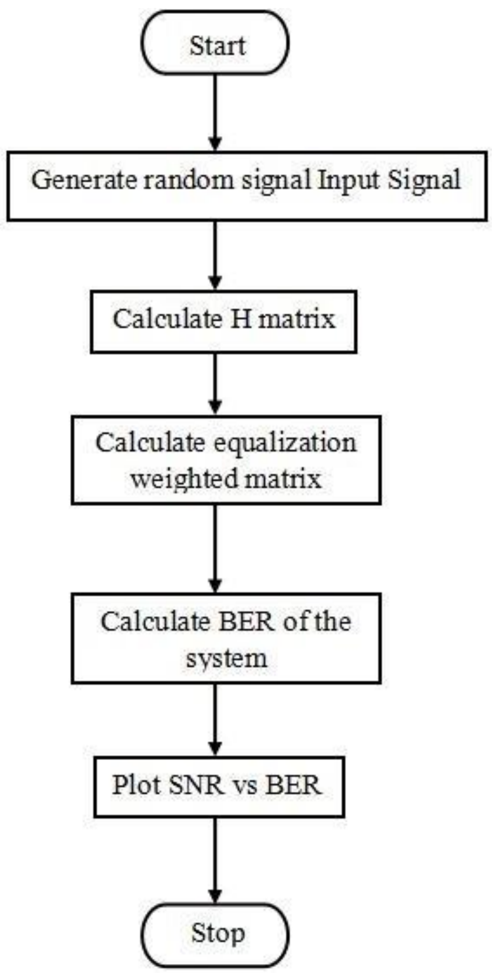
BLOCK DIAGRAM:



FLOW CHART: (Zero forcing Equalizer)



EXP NO: DATE:	**Performance of Equalizers**

AIM:

To implement equalization techniques for wireless channels using Least Mean Squares (LMS) and Zero Forcing algorithms.

SOFTWARE REQUIRED:

MATLAB R2022b

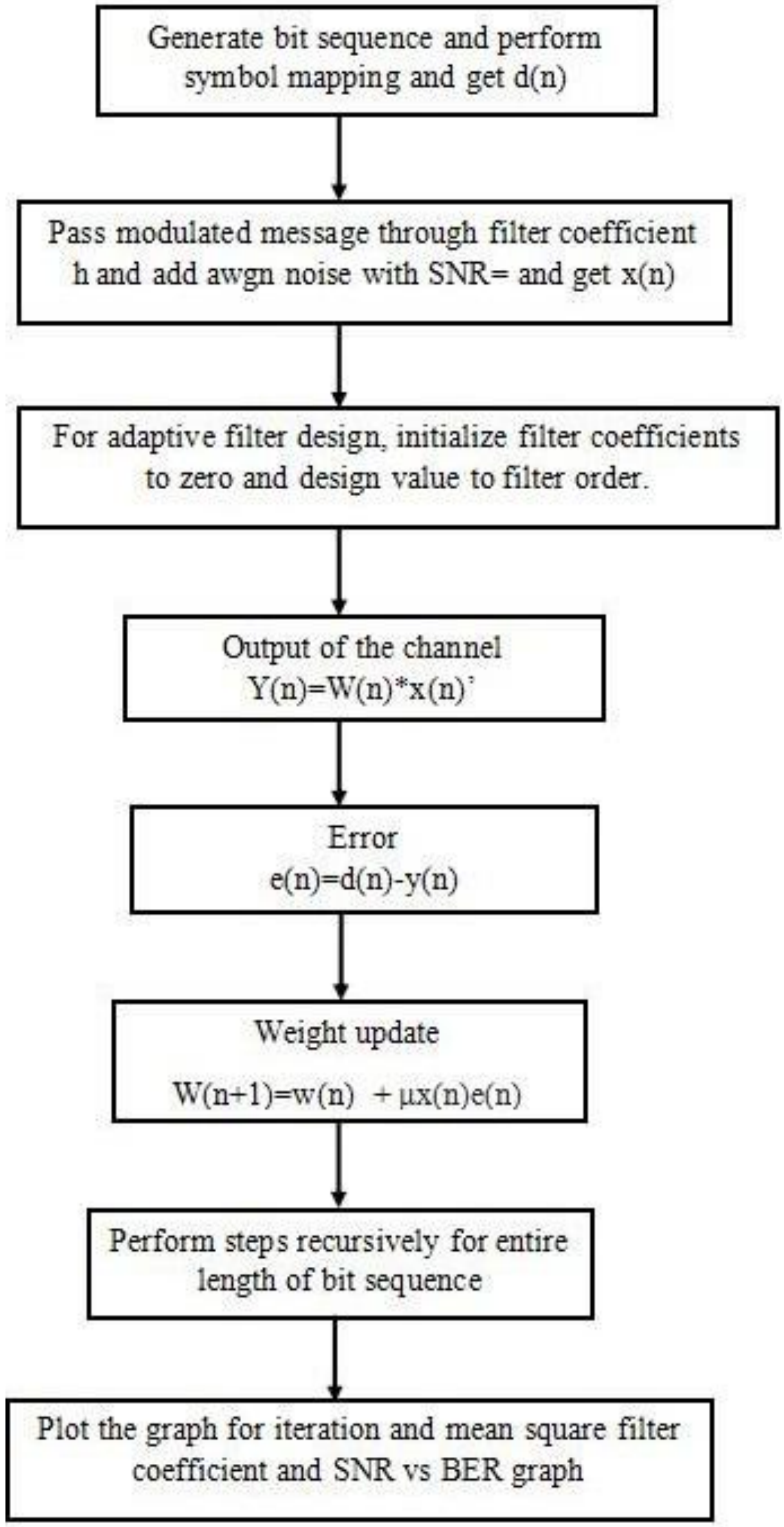
THEORY:

A channel equalizer is an important component of a communication system and is used to mitigate the ISI (inter symbol interference) introduced by the channel. The equalizer depends upon the channel characteristics. These are usually employed to reduce the depth and duration of the fades experienced by a receiver in a local area which are due to motion. An equalizer within a receiver compensates for the average range of expected channel amplitude and delay characteristics. Equalizers must be adaptive since the channel is generally unknown and varies with time.

Zero Forcing Equalizers is a linear equalizer which uses unit impulse response to compensate for channel effect. An overall impulse response is equal to one for the defected symbol and zero for all other received symbols. The noise may be increased in this process.

Least Mean Squares (LMS) algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean square of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time.

FLOW CHART: (LMS)



ALGORITHM:

Zero Forcing:

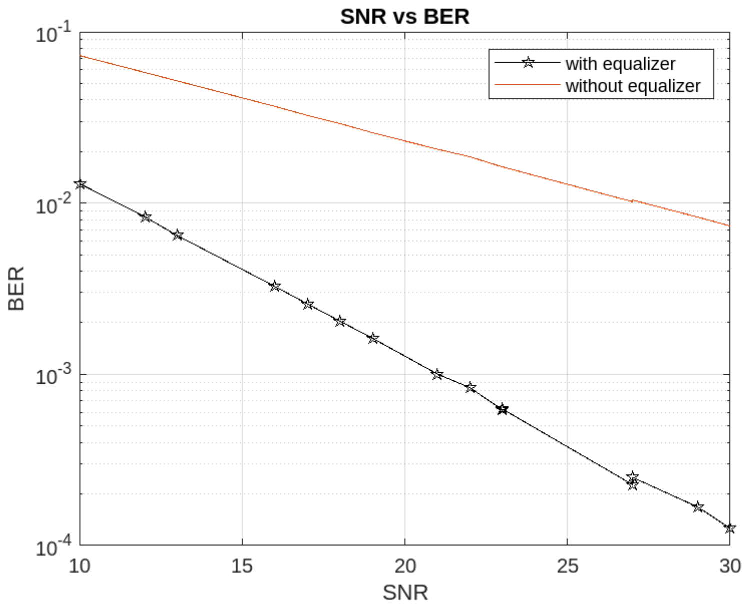
- Generate input bits & channel matrix H.
- Modulate input signal.
- Add AWGN noise.
- Perform equalization to multiply with H^{-1}.
- Demodulate the bits.
- Plot BER VS SNR.

Least Mean Square:

- Generate N Random bits.
- Fix step size (μ)and filter co-efficients.
- Modulate input to get d(n).
- Fix cut-off frequency.
- Filter d(n) to get x(n).
- for i = 1:N
    - $y(n) = w_n^T x(n)$
    - $e(n) = d(n)-y(n)$
    - $w(n+1) = w_n + \mu\, e(n)\, x^*(n)$
- Plot filter co-efficients vs number of iterations and Mean square error vs number of iterations.

MATLAB OUTPUT:

1. Zeroforcing



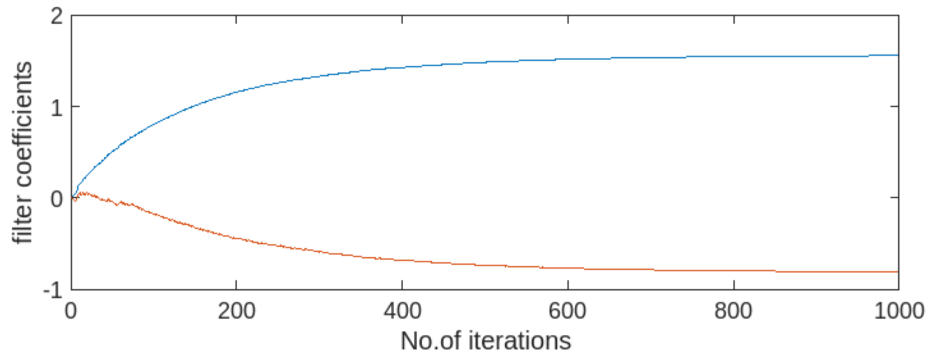
MATLAB CODE:

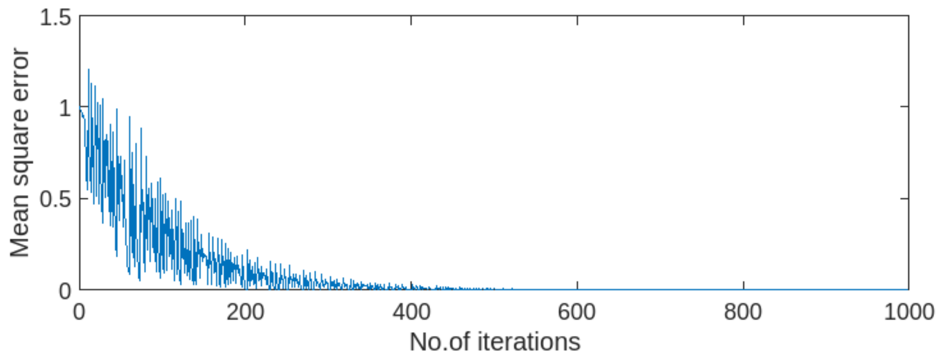
```
clc; clear; close all;
x=randi([0,1],1,1000000);
mod=pskmod(x,2);
h=rand(1,length(x))+1j*rand(1,length(x));
hz=1./h;
snr=randi([10,30],1,15);
snr=sort(snr);
be1=[];be2=[];

%with equalizer(Zero Forcing)
for i=1:length(snr)
n1=h.*mod;
q1=awgn(n1,snr(i),'measured');
p1=q1.*hz;
demod=pskdemod(p1,2);
[u1 e1]=biterr(x,demod);
be1=[be1 e1];
end
% without equalizer
for k=1:length(snr)
n2=h.*mod;
q2=awgn(n2,snr(k),'measured');
demod2=pskdemod(q2,2);
[u2 e2]=biterr(x,demod2);
be2=[be2 e2];
end

semilogy(snr,be1,'Pentagram-C','Color','Black');
hold on
semilogy(snr,be2);
grid on
xlabel("SNR");ylabel("BER");
title("SNR vs BER"),
legend ("with equalizer","without equalizer")
```

2. Least Mean Squares





```matlab
% Least Mean Square

clc; clear; close all;
N=1000;
x=randi([0,1],1,N);
dn=pskmod(x,2);
fc=0.5;p=2;
u=0.02;
a=[1.56,-0.81];
b=1;
xn=filter(b,a,dn);
w=zeros(p,1);
xb=zeros(p,1);
ens=[];

for n=1:N
xb(2:end)=xb(1:end-1);
xb(1)=xn(n);
y(n)=w(:,n)'*xb;
e(n)=dn(n)-y(n);
ens=[ens e(n)];
w=[w w(:,n)+(u.*e(n).*xb)];
end

subplot(211);
plot(0:N,w);
xlabel("No.of iterations");
ylabel("filter coefficients");
subplot(212);
plot(0:N-1,(ens.^2));
xlabel("No.of iterations");
ylabel("Mean square error");
```

CALCULATION:

$$y_k = h_0 x_k + h_1 x_{k-1} + n_0$$

$$y_{k+2} = h_0 x_{k+2} + h_1 x_{k+1} + n_2$$

$$y_{k+1} = h_0 x_{k+1} + h_1 x_k + n_1$$

$$\begin{bmatrix} y_{k+2} \\ y_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & 0 & 0 \\ 0 & h_0 & h_1 & 0 \\ 0 & 0 & h_0 & h_1 \end{bmatrix} . \begin{bmatrix} x_{k+2} \\ x_{k+1} \\ x_k \\ x_{k-1} \end{bmatrix} + \begin{bmatrix} n_2 \\ n_1 \\ n_0 \end{bmatrix}$$

$$\overline{1_2} = C^T H = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

$$C = (HH^T)^{-1} . \overline{1_2}$$

Where,

- y_k → Output sequence at instance k.
- x_k → Input sequence at instant k.
- n_k → Noise sequence at instant k.
- h_o and h_1 → Equalizing filter co-efficients.
- C → Zero Forcing Constant.

INFERENCE:

- ✓ In the Least Mean Squares (LMS) algorithm of equalization techniques of wireless channels, when the epoch iterations increase, error decreases thereby decreasing the Mean Squared Error (MSE).

- ✓ In the case of Zero Forcing Equalizer, it can be observed that it nullifies inter-symbol interference (ISI) in the modulated signals. But noise obtained in the process increases accordingly.

- ✓ Hence, comparing both the algorithms, we can observe that Least Mean Squares (LMS) algorithm is efficient than that of Zero Forcing Equalization technique in terms of noise performance.

RESULT:

Hence, channel equalization using the Least Mean Squares (LMS) algorithm and Zero Forcing Equalizer are successfully performed using MATLAB and the cost function/BER-SNR graphs for the same are plotted.