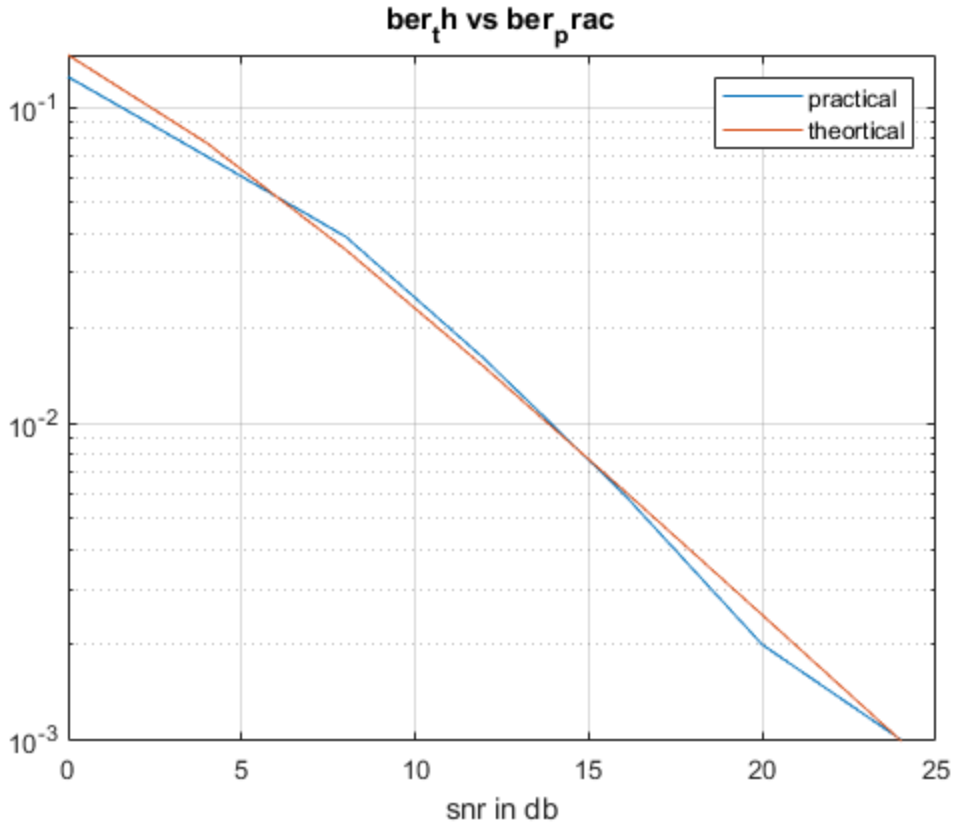
```
clc;
clear all;
close all;

N = 1000;
x = randi([0,1],1,N);

% convert to bpsk
for i = 1:N
    if(x(i)==0)
        t(i) = -1;
    else
        t(i)= x(i);
    end
end

h = (randn(1,N) + 1i*randn(1,N))*sqrt(1/2);  %h complex random ; ray
 light faded channel
n = randn(1,N);                     % noise genrated

snr_db = 0:4:24;
kk =0;



ber_prac = zeros(1, length(snr_db));

for k = 1: length(snr_db)

snr_linear(k) = 10.^(snr_db(k)/10);           % converted snr to
 linear
sigma(k) = 1./(snr_linear(k)).^(1/2);                % find sigma

y = h.*t + sigma(k).*n;                 % find y = channel*bpsk_signal
 + sigma*noise


% convert y sequence into bpsk take threshold value = 0
% z is your constructed signal
% y is output signal with noise
%bpsk wireless communication
z = y./h;


for i = 1:N
    if(z(i)>0)
        r(i)=1;
    else
        r(i)=-1;
    end
end
```

```
ber_th(k) = (1/2)*(1-(snr_linear(k)/(1+snr_linear(k))).^(1/2));

% check bit by bit that r and z is same or not


count_error(k)=0;
for jj=1:N
    if(t(jj)~=r(jj))
        count_error(k)=count_error(k)+1;
%     else
%         count_error;
    end
end
end

ber_prac  = count_error./N;
semilogy(snr_db,ber_prac);
grid on;
hold on;
semilogy(snr_db,ber_th);
title("ber_th vs ber_prac");
xlabel("snr in db");
legend('practical','theortical')
```



Published with MATLAB® R2018b