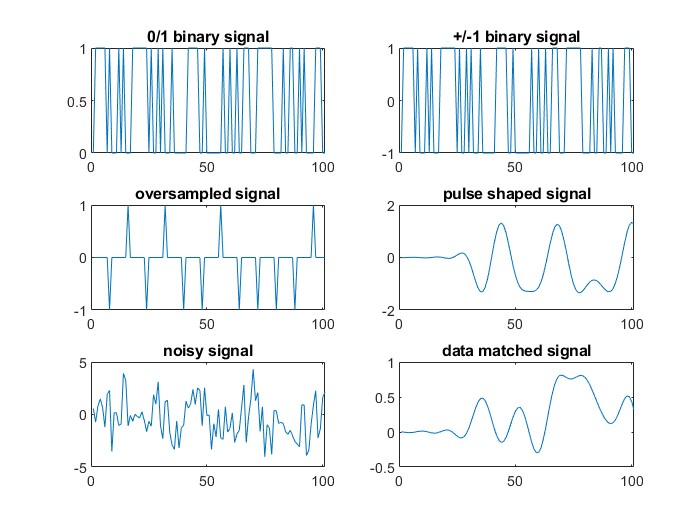
Part 1 (BER Graph)

Chart, line chart

Description automatically generated

Part 2 (Samples of all data)



Part 3 (Sampled Points 50-150)

Oversampled

A picture containing diagram

Description automatically generated

Pulse Shaped

Chart

Description automatically generated

Noisy Signal

Graphical user interface

Description automatically generated

Data after Receive

Chart, line chart

Description automatically generated

Code (W/ Changes)

% Program 3-1

% bpsk.m

%

% Simulation program to realize BPSK transmission system

%

% Programmed by H.Harada and T.Yamamura,

%

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Preparation part \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sr=256000.0; % Symbol rate

ml=1; % Number of modulation levels

br=sr.\*ml; % Bit rate (=symbol rate in this case)

nd = 1000; % Number of symbols that simulates in each loop

IPOINT=8; % Number of oversamples

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Filter initialization \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

irfn=21; % Number of filter taps

alfs=0.5; % Rolloff factor

[xh] = hrollfcoef(irfn,IPOINT,sr,alfs,1); %Transmitter filter coefficients

[xh2] = hrollfcoef(irfn,IPOINT,sr,alfs,0); %Receiver filter coefficients

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* START CALCULATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

nloop=100; % Number of simulation loops

noe = 0; % Number of error data

nod = 0; % Number of transmitted data

for ebn0=0:10

for iii=1:nloop

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Data generation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

data=rand(1,nd)>0.5; % rand: built in function

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BPSK Modulation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

data1=data.\*2-1;

[data2] = oversamp( data1, nd , IPOINT) ;

data3 = conv(data2,xh); % conv: built in function

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Attenuation Calculation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

spow=sum(data3.\*data3)/nd;

attn=0.5\*spow\*sr/br\*10.^(-ebn0/10);

attn=sqrt(attn);

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Fading channel \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% Generated data are fed into a fading simulator

% In the case of BPSK, only Ich data are fed into fading counter

% [ifade,qfade]=sefade(data3,zeros(1,length(data3)),itau,dlvl,th1,n0,itnd1,now1,length(data3),tstp,fd,flat);

% Updata fading counter

%itnd1 = itnd1+ itnd0;

%\*\*\*\*\*\*\*\*\*\*\*\* Add White Gaussian Noise (AWGN) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

inoise=randn(1,length(data3)).\*attn; % randn: built in function

data4=data3+inoise;

data5=conv(data4,xh2); % conv: built in function

sampl=irfn\*IPOINT+1;

data6 = data5(sampl:8:8\*nd+sampl-1);

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BPSK Demodulation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

demodata=data6 > 0;

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Bit Error Rate (BER) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

noe2=sum(abs(data-demodata)); % sum: built in function

nod2=length(data); % length: built in function

noe=noe+noe2;

nod=nod+nod2;

fprintf('%d\t%e\n',iii,noe2/nod2);

end % for iii=1:nloop

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Plot data (Part 2) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

figure

subplot(3, 2, 1)

plot(data(50:150))

title('0/1 binary signal')

subplot(3, 2, 2)

plot(data1(50:150))

title('+/-1 binary signal')

subplot(3, 2, 3)

plot(data2(50:150))

title('oversampled signal')

subplot(3, 2, 4)

plot(data3(50:150))

title('pulse shaped signal')

subplot(3, 2, 5)

plot(data4(50:150))

title('noisy signal')

subplot(3, 2, 6)

plot(data5(50:150))

title('data matched signal')

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Plot data (Part 3) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

figure

plotspec(data2(50:150))

figure

plotspec(data3(50:150))

figure

plotspec(data4(50:150))

figure

plotspec(data5(50:150))

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Output result \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ber(ebn0+1) = noe/nod;

fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod);

fid = fopen('BERbpsk.dat','a');

fprintf(fid,'%d\t%e\t%f\t%f\t\n',ebn0,noe/nod,noe,nod);

fclose(fid);

end

figure

semilogy(0:10,ber)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* end of file \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*