Solutions to PS # 4.0
Signal Processing Using MATLAB
EECE–495, Spring 2001

Binary Sequence Generator:

```
%**********************************************************
% Binary sequence generator
% that generates equiprobable binary symbols
% USAGE: s_out = bingen(N,opt)
% N : number of samples needed
% opt : Option specified
%     : 1 ---> {1,-1}
%     : 2 ---> {0,1}
% AUTHOR : Balu Santhanam
% DATE   : 02/26/01
%**********************************************************
function s_out = bingen(N,opt)
u = rand(1,N);
switch opt
case 1,
      ind1 = find(u >= 0.5);
      ind2 = find(u < 0.5);
      s_out(ind1) = 1*ones(1,length(ind1));
      s_out(ind2) = -1*ones(1,length(ind2));
otherwise,
      ind1 = find(u >= 0.5);
      ind2 = find(u < 0.5);
      s_out(ind1) = 1*ones(1,length(ind1));
      s_out(ind2) = 0*ones(1,length(ind2));
end
```

AWGN Detector:

```
%  AWGN channel simulator and a detector for equiprobable symbols
%  USAGE:
%  [x_out,p_err] = bindet(x_in,sigma,opt)
%   x_out : Output of decision device
%   x_in: binary sequence of bits
%   sigma : standard deviation of AWGN
%   opt   : Option specifying modulation type
%         : 1 ---> Antipodal Signaling
%         : 2 ---> On--Off Keying
%  AUTHOR: Balu Santhanam
%  DATE : 02/26/2001
%***************************************************
function [x_out,p_err] = bindet(x_in,sigma,opt)
% Error checking
if nargin < 3
   error('Insufficient Input')
elseif isnumeric([x_in,sigma,opt]) ~= 1
   error('Non-numeric input')
elseif length(x_in) == 0
   error('Null Input')
elseif sigma < 0
   error('Standard deviation -ve')
elseif opt ~= 1 & opt~=2
   error('Invalid option')
end
% Corrupt input with zero-mean AWGN with specified variance
xnoisy = awn(x_in,[0,sigma],'AWGN');
% ML sequence detector for WGN channel
x_out = zeros(1,length(x_in));
switch opt
case 1,
   ind1 = find(xnoisy >= 0); ind2 = find(xnoisy < 0);
   x_out(ind1) = 1*ones(1,length(ind1));
   x_out(ind2) = -1*ones(1,length(ind2));
otherwise
   ind1 = find(xnoisy >= 0.5); ind2 = find(xnoisy < 0.5);
   x_out(ind1) = 1*ones(1,length(ind1));
   x_out(ind2) = 0*ones(1,length(ind2));
end
p_err = length(find(x_out ~= x_in))/length(x_in);
return
```

Script for Plots:

```
% Script file for generating the plots
% for PS # 4.0
sig = 10.^[-2.0:0.1:2.0];
N = 5000;
opt1 = 1; opt2 = 2;
s1 = bingen(N,opt1);
s2 = bingen(N,opt2);
for i = 1:1:length(sig)
   [s1,p_err1] = bindet(s1,sig(i),opt1);
   [s2,p_err2] = bindet(s2,sig(i),opt2);
    P1(i) = p_err1; P2(i) = p_err2;
    P1_th(i) = 1 - normcdf(1./sig(i),0,1);
    P2_th(i) = 1 - normcdf(0.5/sig(i),0,1);
end
semilogx(sig,P1)
hold on
semilogx(sig,P1_th,’r*--’)
semilogx(sig,P2,’m’)
semilogx(sig,P2_th,’x--’)
xlabel(’\sigma’)
ylabel(’{\rm Pr}(\epsilon)’)
title(’N = 5000, p = 0.5, binary AWGN channel’)
legend(gca,’BAS-EST’,’BAS-ACT’,’OOK-EST’,’OOK-ACT’,-1)
hold off
orient landscape
print -deps perr.eps
```

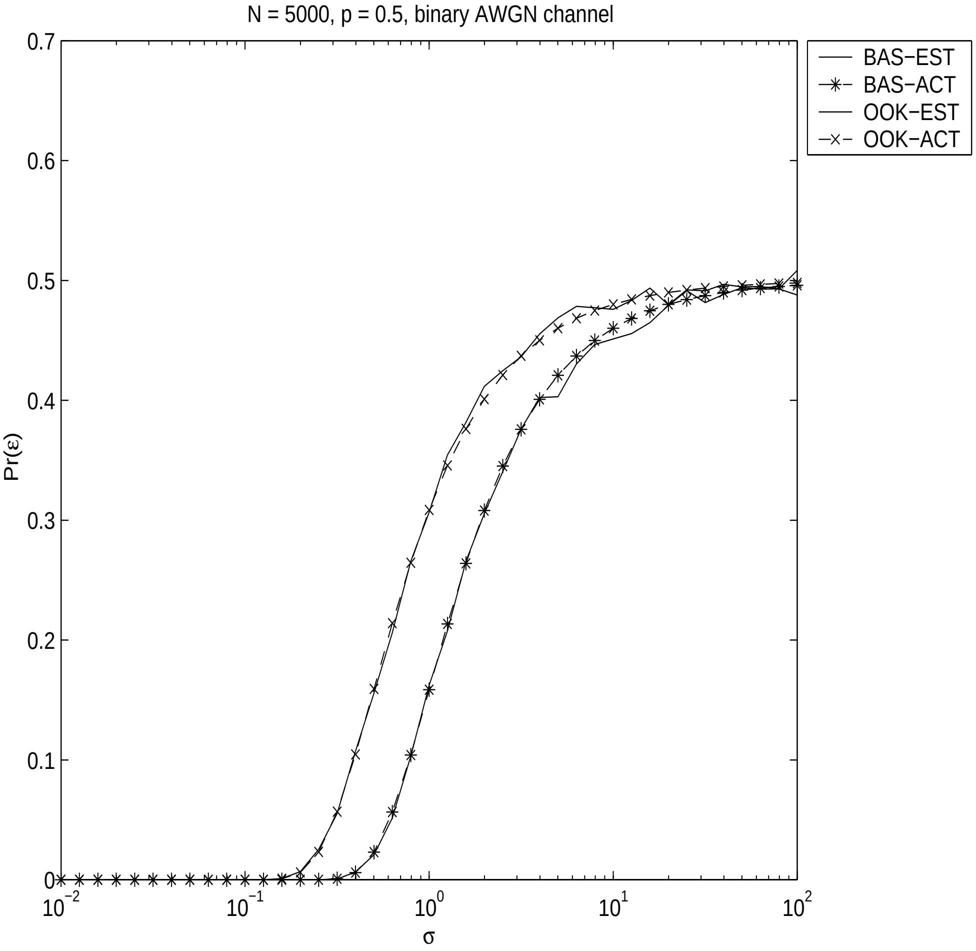

Figure 1: Error curves: estimated and theoretical $\Pr(\epsilon)$ vs σ for BAS and OOK signaling.