Principles of Communication HW 4 Code

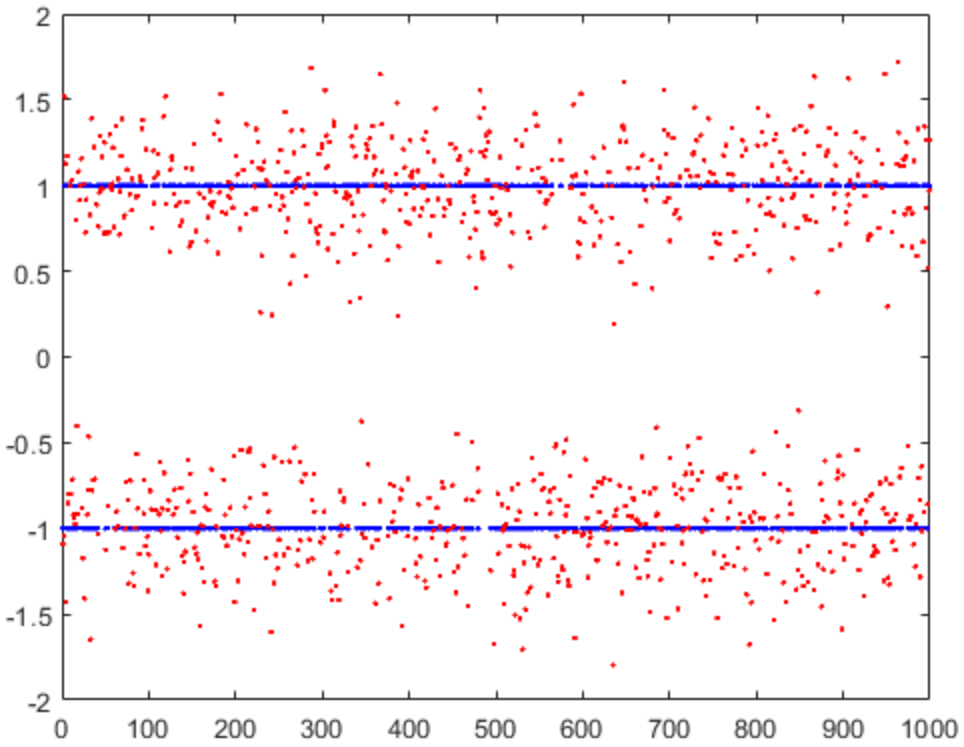
Table of Contents

Preface

This is the code for Exercises 14.13 and 14.14. Exercises 14.2 and 14.16 can be found in the PDF file named PoC_HW_4, while this file shall be named PoC_HW_4_Code.

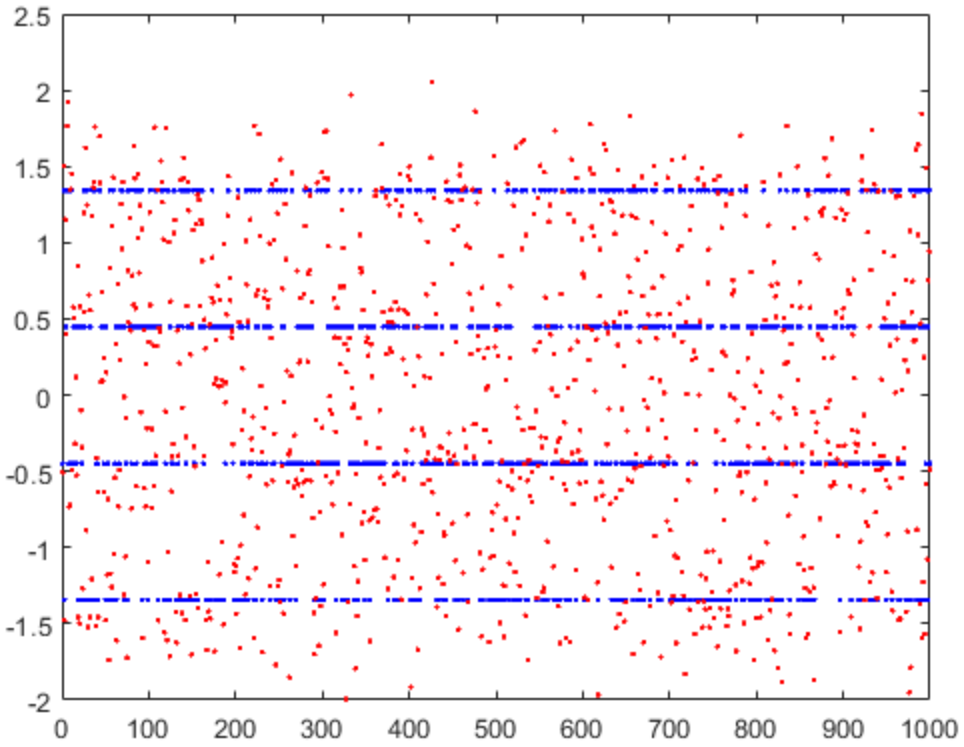
Exercise 14.13a.1 2-PAM

```
m=1000;                          % length of data sequence
p=1/15; s=1.0;                   % power of noise and signal
x=pam(m,2,s);                    % 4-PAM input with power 1...
L=sqrt(1/5);                     % ...with amp levels L
n=sqrt(p)*randn(1,m);            % noise with power p
n_max=sqrt(p)*m;
y=x+n;                           % output adds noise to data
qy=quantalph(y,[-3*L,-L,L,3*L]); % quantize to [-3*L,-L,L,3*L]
err=sum(abs(sign(qy'-x)))/m;     % percent transmission errors
plot(1:1000,x,'b', 1:1000,y, 'r', 'Marker','.', 'LineStyle', 'none')
% This is the plot for the 2 level system
```



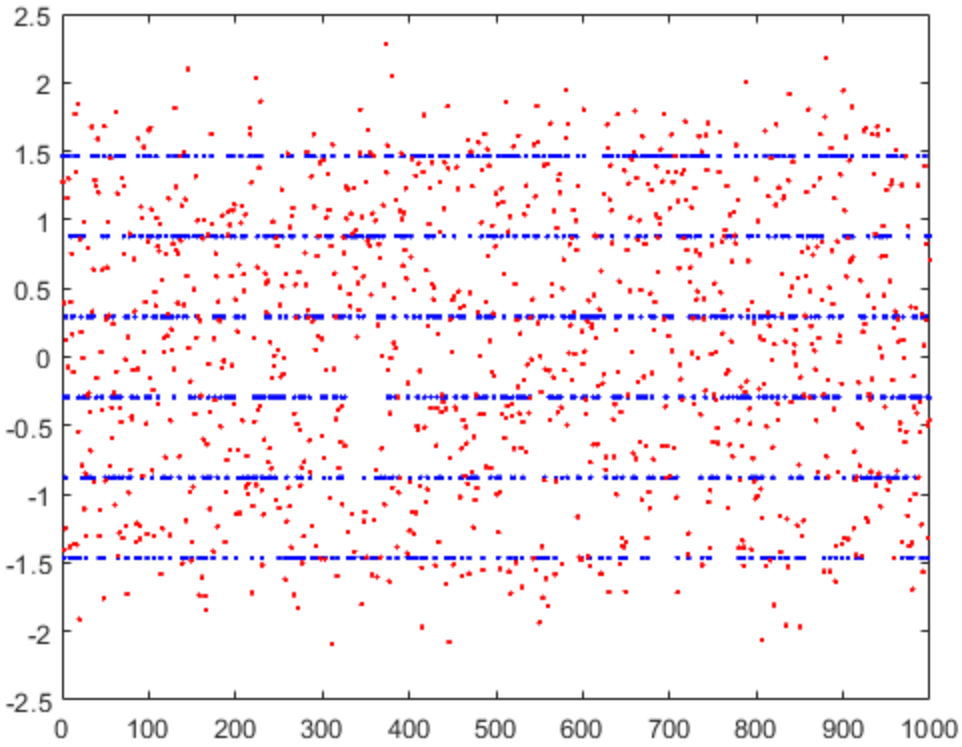
Exercise 14.13a.2 4-PAM

```
m=1000;                              % length of data sequence
p=1/15; s=1.0;                       % power of noise and signal
x=pam(m,4,s);                        % 4-PAM input with power 1...
L=sqrt(1/5);                         % ...with amp levels L
n=sqrt(p)*randn(1,m);                % noise with power p
y=x+n;                               % output adds noise to data
qy=quantalph(y,[-3*L,-L,L,3*L]); % quantize to [-3*L,-L,L,3*L]
err=sum(abs(sign(qy'-x)))/m;     % percent transmission errors
plot(1:1000,x,'b', 1:1000,y, 'r', 'Marker','.', 'LineStyle', 'none')
% This is the plot for the 4 level system, or the original code
```



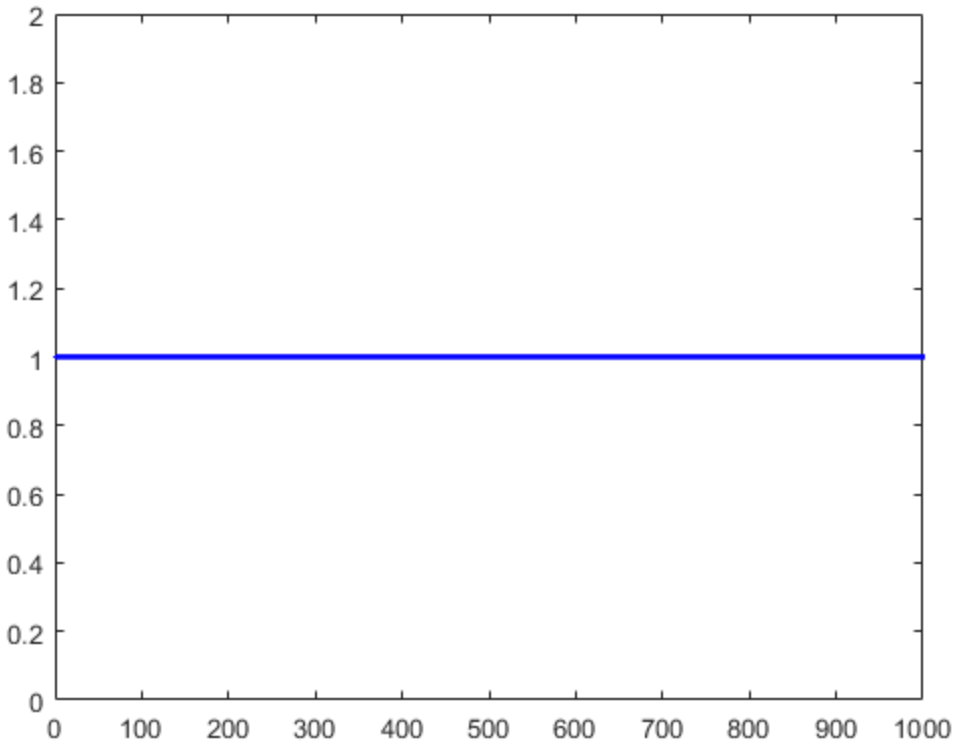
Exercise 14.13a.3: 6-PAM

```
m=1000;                              % length of data sequence
p=1/15; s=1.0;                       % power of noise and signal
x=pam(m,6,s);                        % 4-PAM input with power 1...
L=sqrt(1/5);                         % ...with amp levels L
n=sqrt(p)*randn(1,m);                % noise with power p
y=x+n;                               % output adds noise to data
qy=quantalph(y,[-3*L,-L,L,3*L]); % quantize to [-3*L,-L,L,3*L]
err=sum(abs(sign(qy'-x)))/m;     % percent transmission errors
plot(1:1000,x,'b', 1:1000,y, 'r', 'Marker','.', 'LineStyle', 'none')
% This is the plot for the 6 level system
```



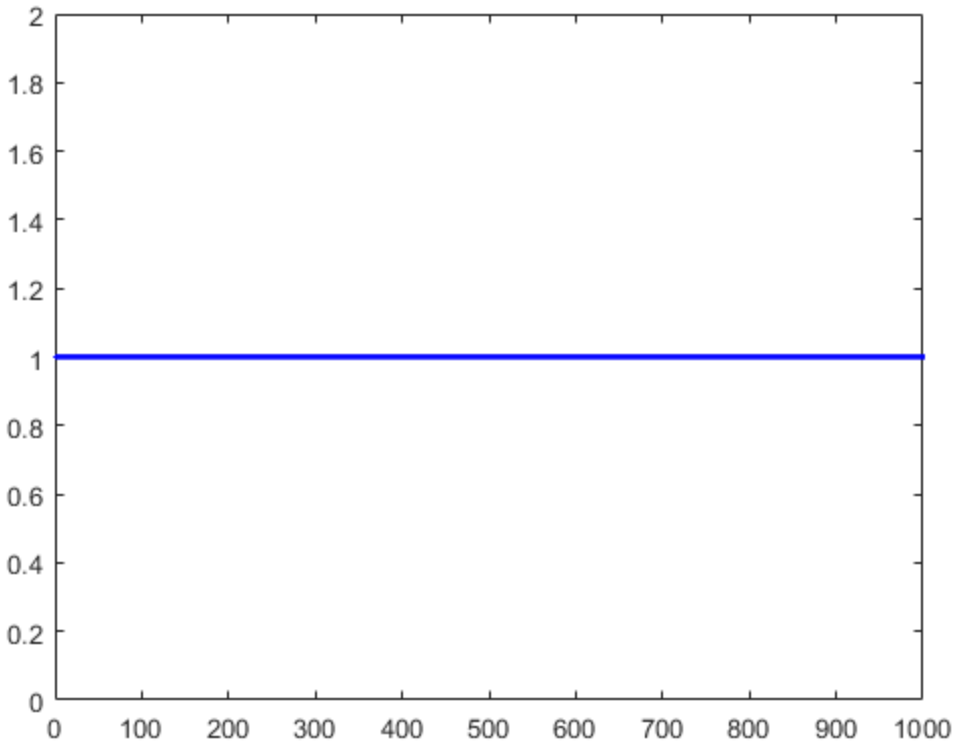
Exercise 14.13b.1

```
m=1000;                              % length of data sequence
p=1/15; s=1.0;                       % power of noise and signal
x=pam(m,2,s);                        % 4-PAM input with power 1...
L=sqrt(1/5);                         % ...with amp levels L
n=sqrt(p)*randn(1,m);                % noise with power p
y=x+n;                               % output adds noise to data
qy=quantalph(y,[-3*L,-L,L,3*L]); % quantize to [-3*L,-L,L,3*L]
err=sum(abs(sign(qy'-x)))/m;     % percent transmission errors
plot(1:1000,err,'b', 'Marker','.', 'LineStyle', 'none')
% Noise power vs error plot for the 2 level system
```



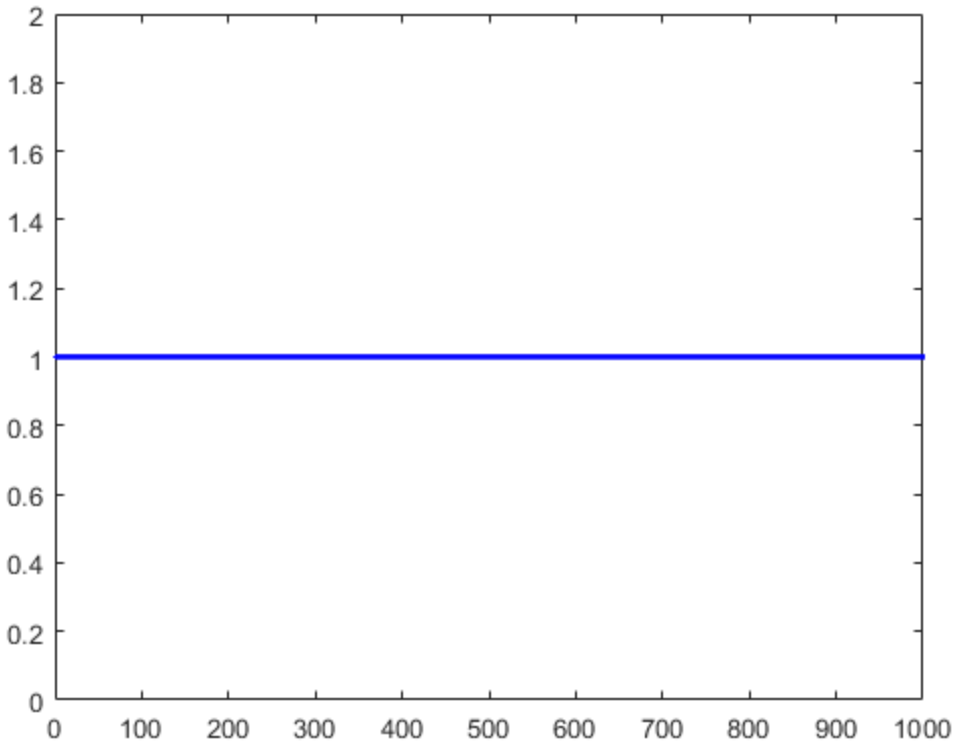
Exercise 14.13b.2

```
m=1000;                              % length of data sequence
p=0.01; s=1.8;                       % power of noise and signal
x=pam(m,4,s);                        % 4-PAM input with power 1...
L=sqrt(1/5);                         % ...with amp levels L
n=sqrt(p)*randn(1,m);                % noise with power p
y=x+n;                               % output adds noise to data
qy=quantalph(y,[-3*L,-L,L,3*L]); % quantize to [-3*L,-L,L,3*L]
err=sum(abs(sign(qy'-x)))/m;     % percent transmission errors
plot(1:1000,err,'b', 'Marker','.', 'LineStyle', 'none')
% Noise power vs error plot for the 4 level system
```



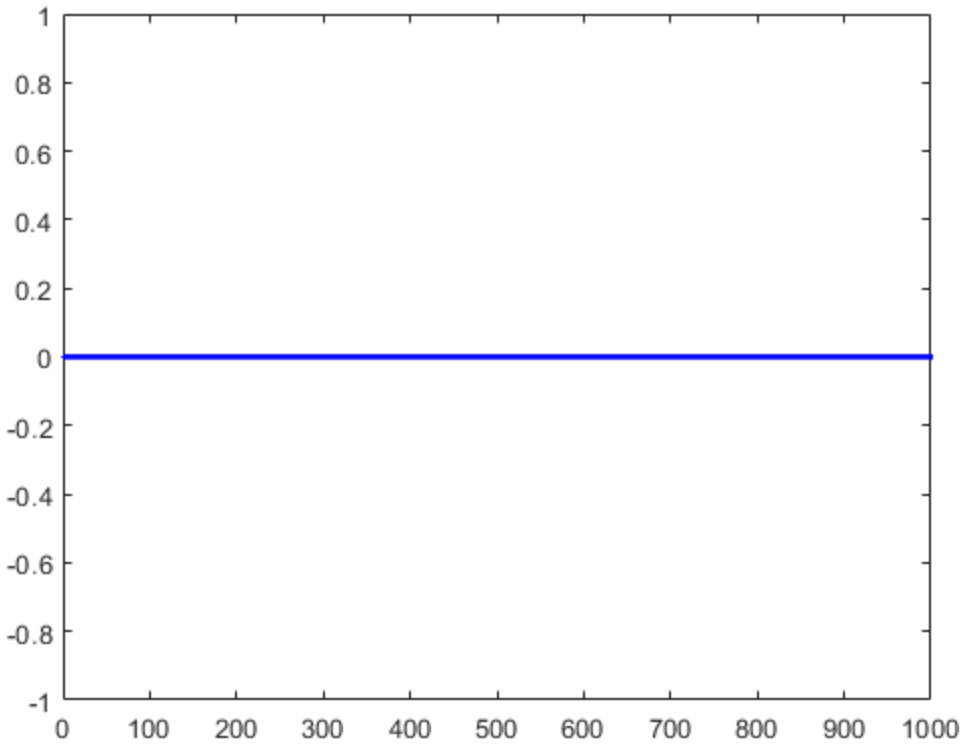
Exercise 14.13b.3

```
m=1000;                              % length of data sequence
p=1/15; s=1.0;                       % power of noise and signal
x=pam(m,6,s);                        % 4-PAM input with power 1...
L=sqrt(1/5);                         % ...with amp levels L
n=sqrt(p)*randn(1,m);                % noise with power p
y=x+n;                               % output adds noise to data
qy=quantalph(y,[-3*L,-L,L,3*L]); % quantize to [-3*L,-L,L,3*L]
err=sum(abs(sign(qy'-x)))/m;     % percent transmission errors
plot(1:1000,err,'b', 'Marker','.', 'LineStyle', 'none')
% Noise power vs error plot for the 6 level system
```



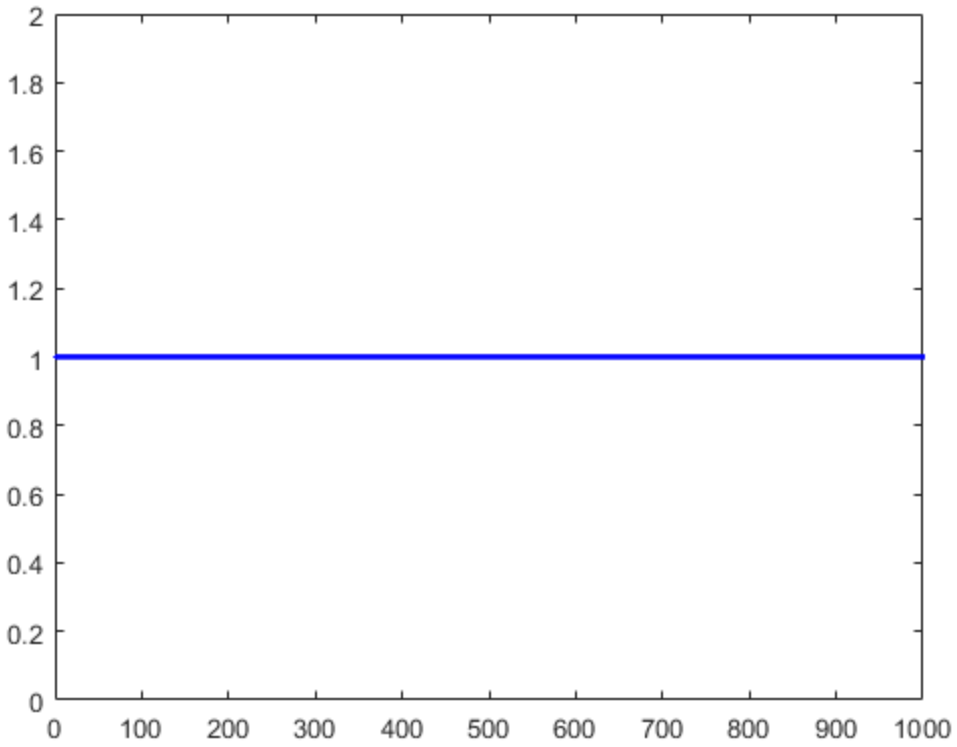
Exercise 14.14.1

```
m=1000;                           % length of data sequence
p=0.01; s=1.8;                    % power of noise and signal
x=pam(m,2,s);                     % 4-PAM input with power 1...
L=sqrt(1/5);                      % ...with amp levels L
n=sqrt(p)*randn(1,m);             % noise with power p
y=x+n;                            % output adds noise to data
qy=quantalph(y,[-3*L,-L,L,3*L]); % quantize to [-3*L,-L,L,3*L]
err=sum(abs(sign(qy'-x)))/m;      % percent transmission errors
plot(1:1000,err,'b', 'Marker','.', 'LineStyle', 'none')
% The power level S would have to be at a value of 1.8 if the noise
 power
% level P was at 0.01, which would allow the 2 level system to have
 the
% same amount of error as the 4 level system.
```



Exercise 14.14.2

```
m=1000;                          % length of data sequence
p=0.01; s=1.0;                   % power of noise and signal
x=pam(m,6,s);                    % 4-PAM input with power 1...
L=sqrt(1/5);                     % ...with amp levels L
n=sqrt(p)*randn(1,m);            % noise with power p
y=x+n;                           % output adds noise to data
qy=quantalph(y,[-3*L,-L,L,3*L]); % quantize to [-3*L,-L,L,3*L]
err=sum(abs(sign(qy'-x)))/m;     % percent transmission errors
plot(1:1000,err,'b', 'Marker','.', 'LineStyle', 'none')
% There was not a feasible positive power level S to allow for an
 error
% value equivalent to the four level system with a noise power level P
 at
% 0.01. In order to calculate the value of P that would result in an
 error
% value of zero at a noise power level of 0.01, if the value of
 capacity
% and bandwidth are known, the equation for capacity given in the
 textbook
% can be modified in order to produce an S value that would result in
% minimal error, if the capacity is greater than the rate at which
% information is being presented to this system.
```



Published with MATLAB® R2020b