Digital Communication Systems ECS 452

Asst. Prof. Dr. Prapun Suksompong

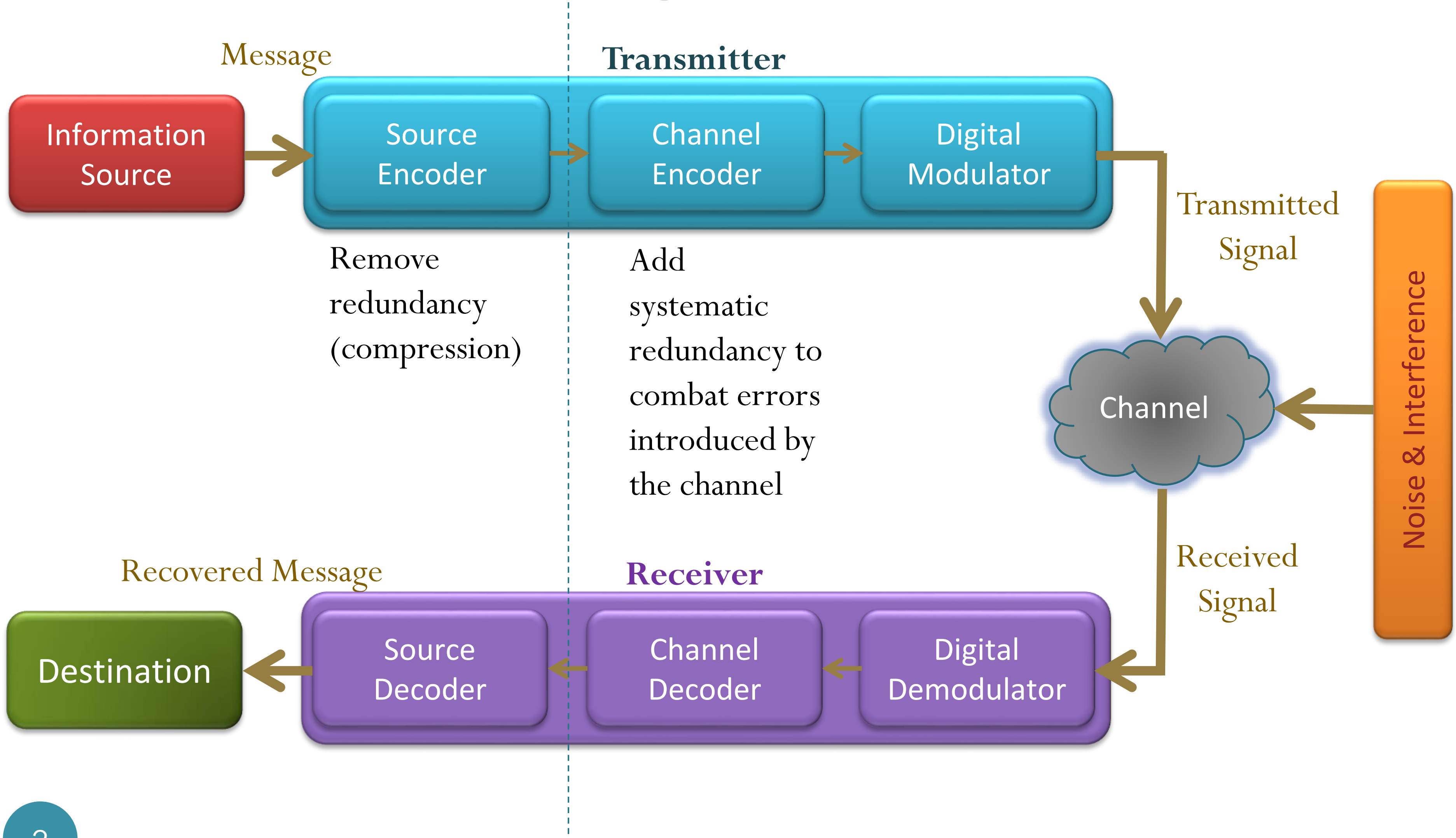
prapun@siit.tu.ac.th

Discrete Channel

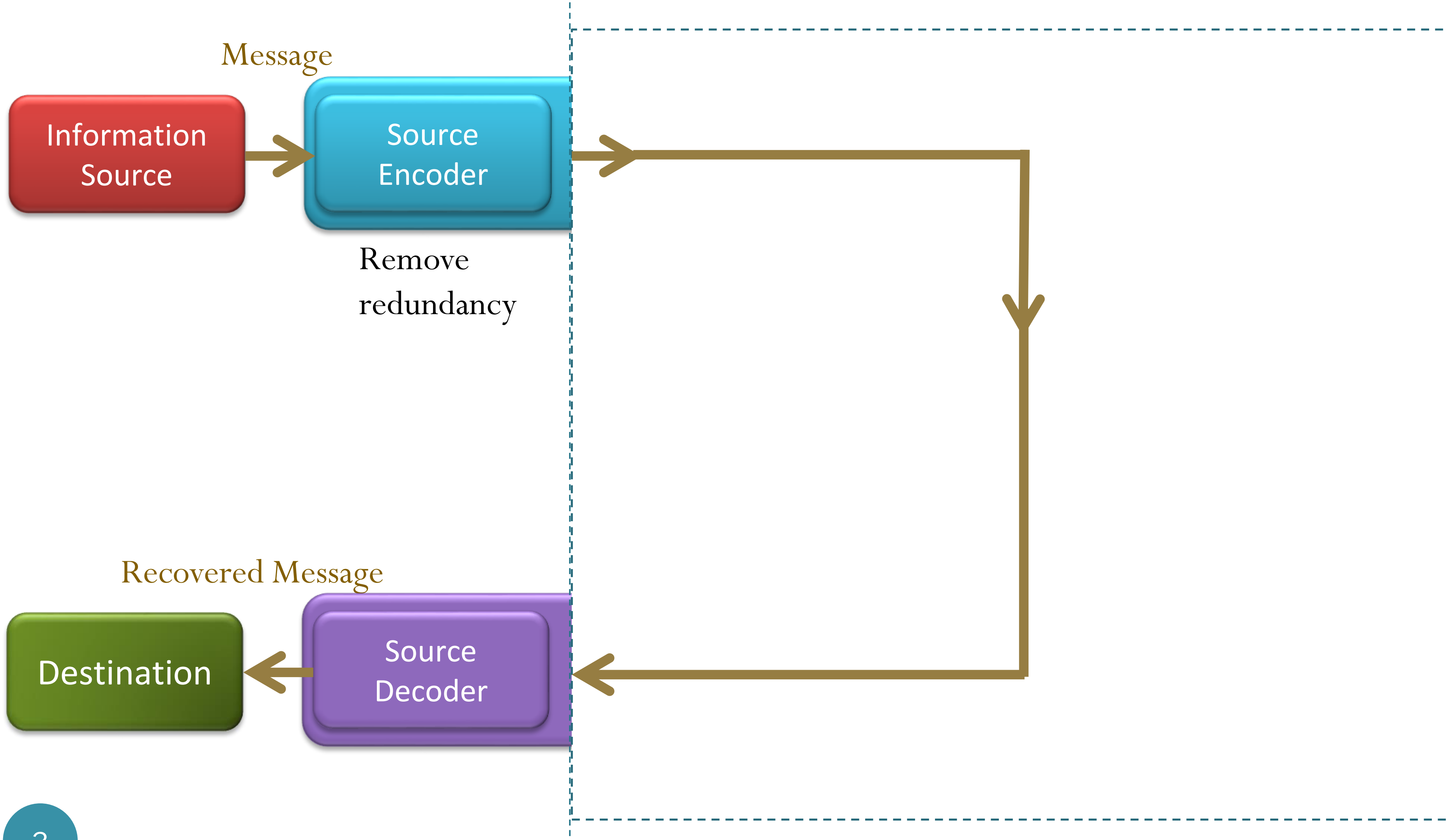
Office Hours:
BKD 3601-7

Monday	14:00-16:00
Wednesday	14:40-16:00

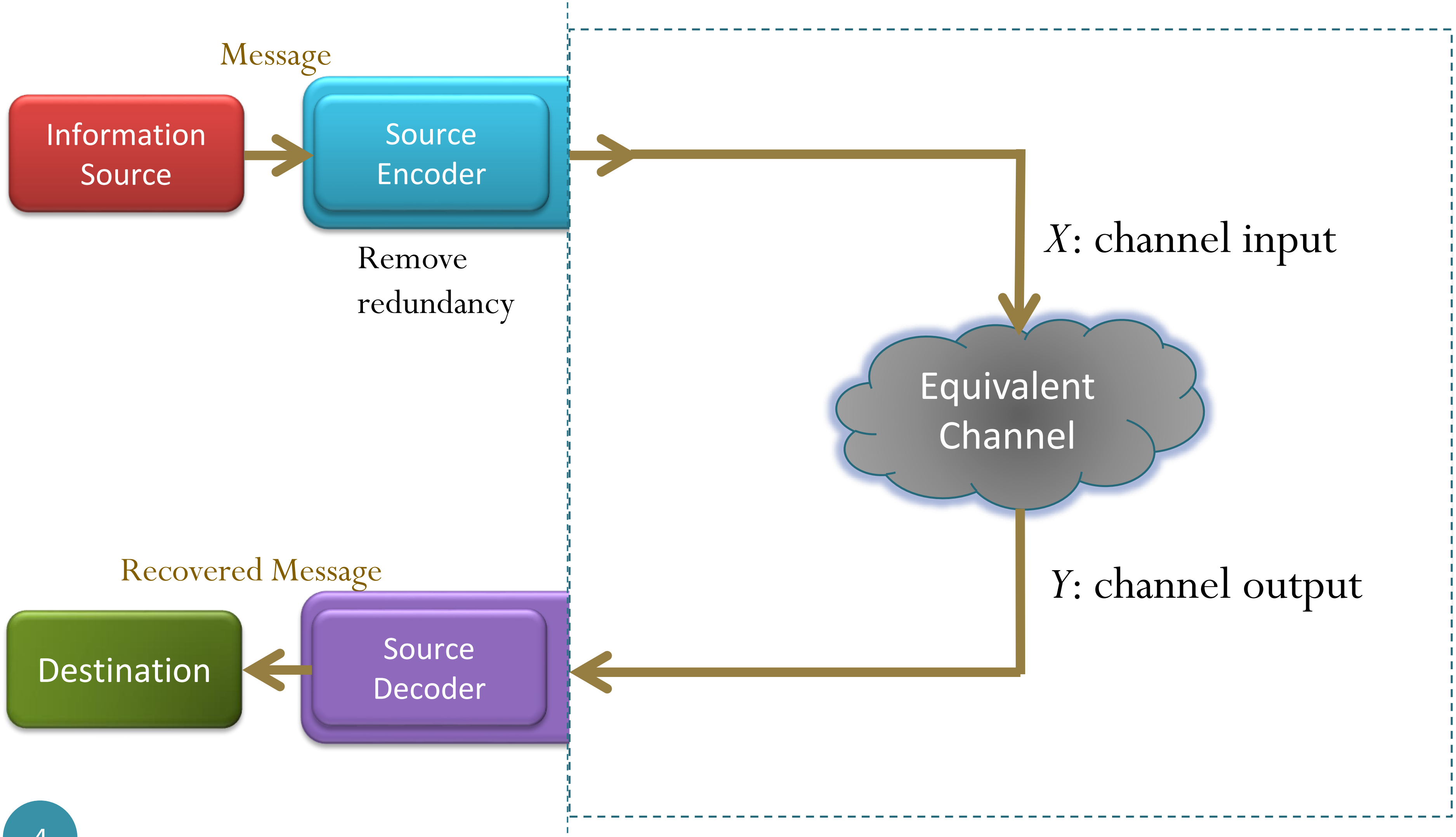
Elements of digital commu. sys.



System considered previously



System considered in this section



Ex: BSC

```
%% Simulation parameters
% The number of symbols to be transmitted
n = 20;
% Channel Input
S_X = [0 1]; S_Y = [0 1];
p_X = [0.3 0.7];
% Channel Characteristics
p = 0.1; Q = [1-p p; p 1-p];
```

```
>> BSC_demo
ans =
1  0  1  1  1  1  1  1  1  1  1  1  0  1  0  1  1  1  1  1
ans =
1  1  1  1  1  1  1  1  1  0  1  1  0  1  0  1  1  1  1  1

p_X =
   0.3000    0.7000
p_X_sim =
   0.1500    0.8500
q =
   0.3400    0.6600
q_sim =
   0.1500    0.8500

Q =
   0.9000    0.1000
   0.1000    0.9000
Q_sim =
   0.6667    0.3333
   0.0588    0.9412
PE_sim =
   0.1000
PE_theretical =
   0.1000

Elapsed time is 0.134992 seconds.
```

[BSC_demo.m]

MATLAB

```matlab
%% Generating the channel input x
x = randsrc(1,n,[S_X;p_X]); % channel input

%% Applying the effect of the channel to create the channel output y
y = DMC_Channel_sim(x,S_X,S_Y,Q); % channel output
```

```matlab
function y = DMC_Channel_sim(x,S_X,S_Y,Q)
%% Applying the effect of the channel to create the channel output y
y = zeros(size(x)); % preallocation
for k = 1:length(x)
    % Look at the channel input one by one. Choose the corresponding row
    % from the Q matrix to generate the channel output.
    y(k) = randsrc(1,1,[S_Y;Q(find(S_X == x(k)),:)]);
end
```

[DMC_Channel_sim.m]

[DMC_Analysis_demo.m]

Rel. freq. from the simulation

```
%% Statistical Analysis
% The probability values for the channel inputs
p_X                         % Theoretical probability
p_X_sim = hist(x,S_X)/n % Relative frequencies from the simulation
% The probability values for the channel outputs
q = p_X*Q                   % Theoretical probability
q_sim = hist(y,S_Y)/n       % Relative frequencies from the simulation
% The channel transition probabilities from the simulation
Q_sim = [];
for k = 1:length(S_X)
    I = find(x==S_X(k)); LI = length(I);
    rel_freq_Xk = LI/n;
    yc = y(I);
    cond_rel_freq = hist(yc,S_Y)/LI; Q_sim = [Q_sim; cond_rel_freq];
end
Q                           % Theoretical probability
Q_sim                       % Relative frequencies from the simulation
```


[DMC_Analysis_demo.m]

MATLAB

```
%% Naive Decoder
x_hat = y;
```

```
%% Error Probability
PE_sim = 1-sum(x==x_hat)/n % Error probability from the simulation
```

```
% Calculation of the theoretical error probability
PC = 0;
for k = 1:length(S_X)
    t = S_X(k);
    i = find(S_Y == t);
    if length(i) == 1
        PC = PC+ p_X(k)*Q(k,i);
    end
end
PE_theretical = 1-PC
```


[DMC_Analysis_demo.m]

Ex: BSC

```
%% Simulation parameters
% The number of symbols to be transmitted
n = 1e4;
% Channel Input
S_X = [0 1]; S_Y = [0 1];
p_X = [0.3 0.7];
% Channel Characteristics
p = 0.1; Q = [1-p p; p 1-p];
```

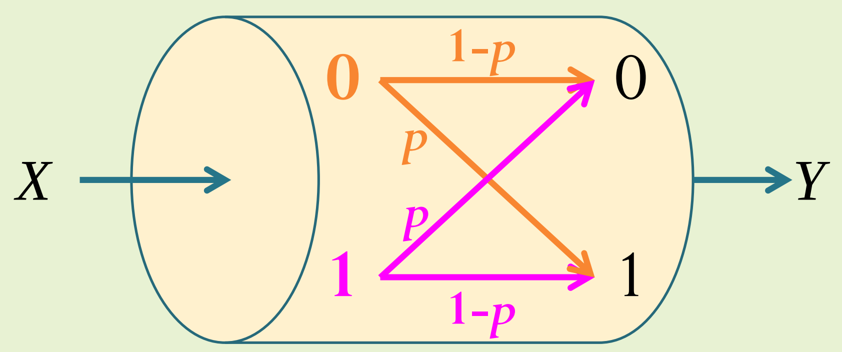
```
>> BSC_demo

p_X =
    0.3000    0.7000
p_X_sim =
    0.3037    0.6963
q =
    0.3400    0.6600
q_sim =
    0.3407    0.6593

Q =
    0.9000    0.1000
    0.1000    0.9000
Q_sim =
    0.9078    0.0922
    0.0934    0.9066
PE_sim =
    0.0930
PE_theretical =
    0.1000

Elapsed time is 0.922728 seconds.
```


[BSC_demo.m]

Binary Symmetric Channel (BSC)

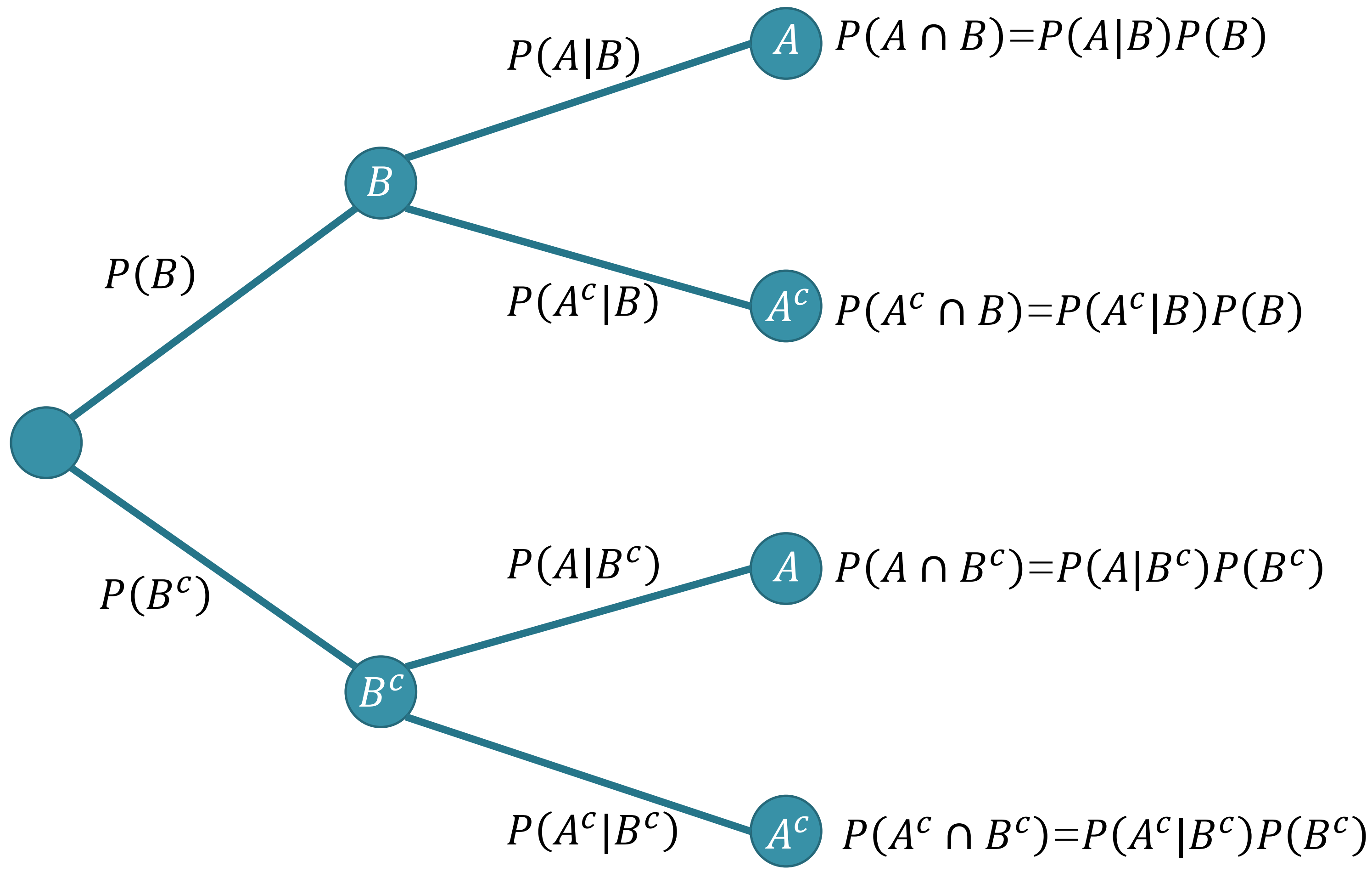
Three equivalent descriptions of BSC:

$$P\left[Y=0 \mid X=0\right]=Q(0|0)=1-p$$
$$P\left[Y=1 \mid X=0\right]=Q(1|0)=p$$
$$P\left[Y=0 \mid X=1\right]=Q(0|1)=p$$
$$P\left[Y=1 \mid X=1\right]=Q(1|1)=1-p$$

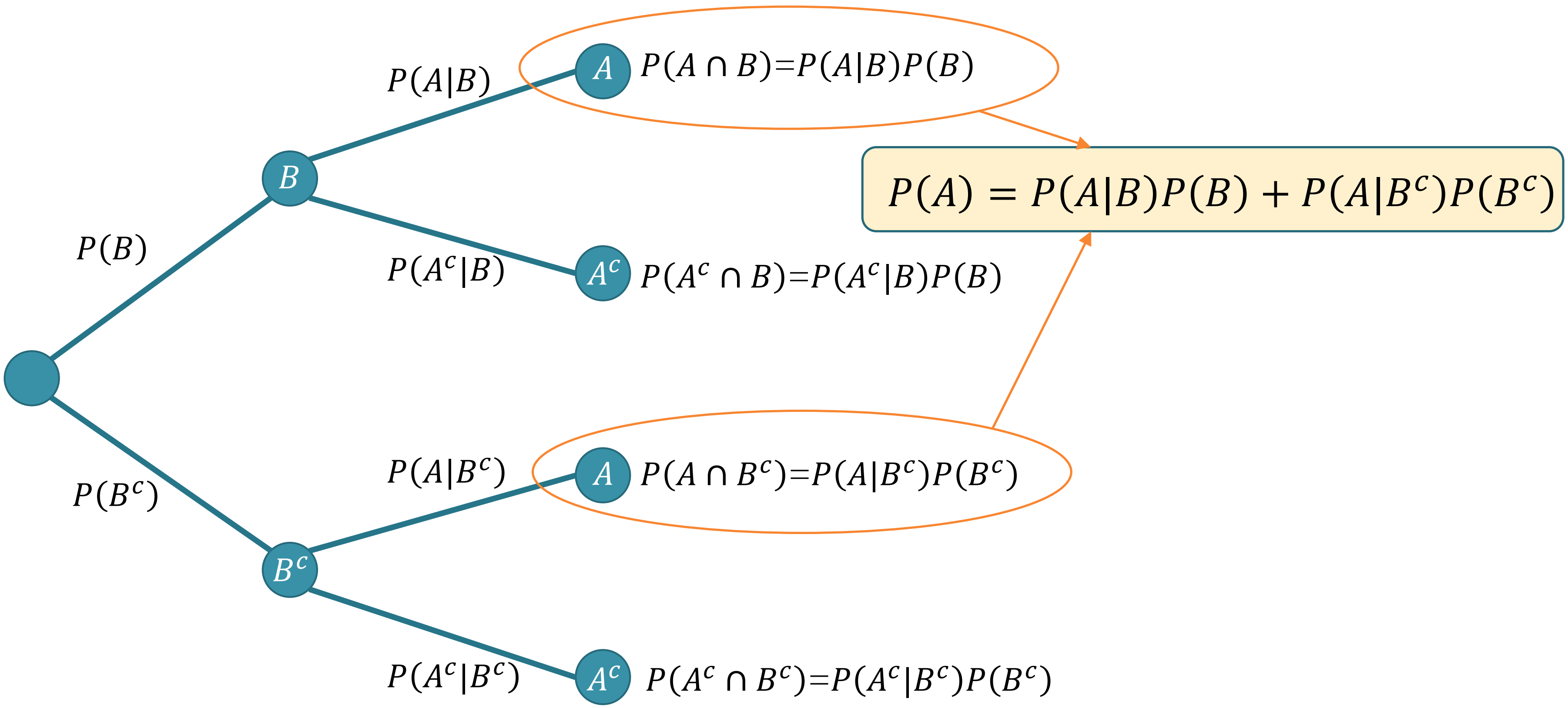


$$Q=\begin{array}{c|cc} x \backslash y & 0 & 1 \\ \hline 0 & 1-p & p \\ 1 & p & 1-p \end{array}$$

Tree Diagram and Conditional Probability



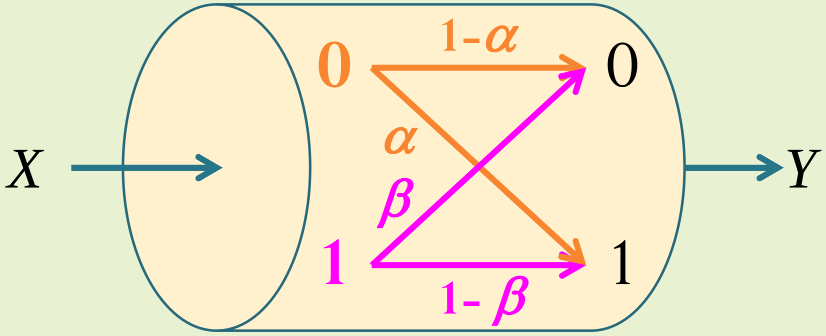
Tree Diagram and Total Probability Theorem



Binary Asymmetric Channel (BAC)

Three equivalent descriptions of BAC:

$$P\left[Y=0\,|\,X=0\right]=Q\left(0|0\right)=1-\alpha$$
$$P\left[Y=1\,|\,X=0\right]=Q\left(1|0\right)=\alpha$$
$$P\left[Y=0\,|\,X=1\right]=Q\left(0|1\right)=\beta$$
$$P\left[Y=1\,|\,X=1\right]=Q\left(1|1\right)=1-\beta$$

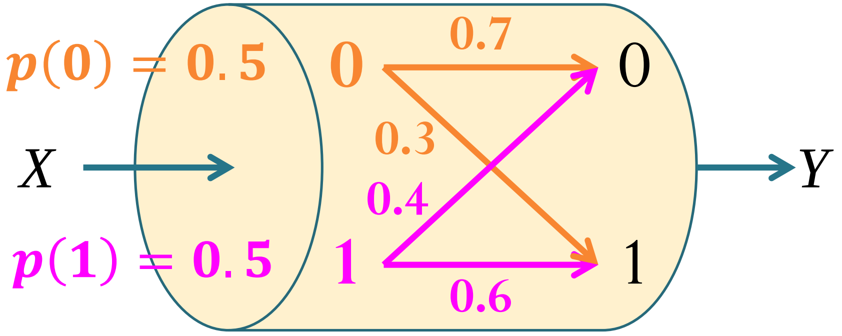


$$Q=\begin{array}{c|cc} x \backslash y & 0 & 1 \\ \hline 0 & 1-\alpha & \alpha \\ 1 & \beta & 1-\beta \end{array}$$

Ex: BAC

```
%% Simulation parameters
% The number of symbols to be transmitted
n = 20;
% Binary Assymmetric Channel (BAC)
% Ex 3.8 in lecture note (11.3 in [Z&T, 2010])
% Channel Input
S_X = [0 1]; S_Y = [0 1];
p_X = [0.5 0.5];
% Channel Characteristics
Q = [0.7 0.3; 0.4 0.6];
```

```
>> BAC_demo
ans =
x:  0 0 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0
ans =
y:  0 0 1 1 0 0 0 1 1 1 0 0 1 0 0 0 0 0 1 0
```




```
p_X =
   0.5000   0.5000
p_X_sim =
   0.7000   0.3000
q =
   0.5500   0.4500
q_sim =
   0.6500   0.3500
Q =
   0.7000   0.3000
   0.4000   0.6000
Q_sim =
   0.7143   0.2857
   0.5000   0.5000
PE_sim =
   0.3500
PE_theretical =
   0.3500
```

[BAC_demo.m]

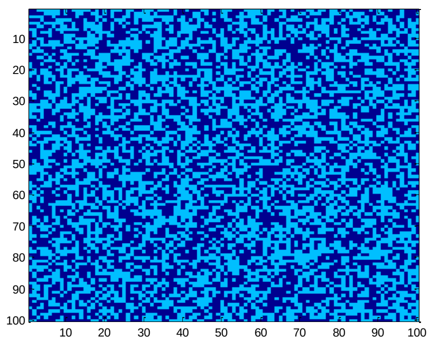
MATLAB

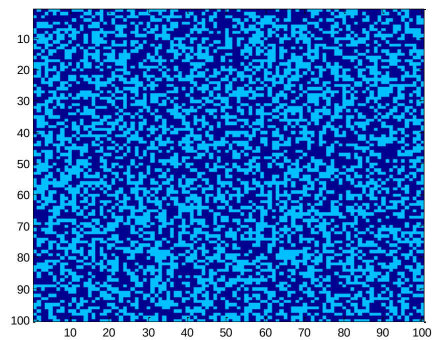
```
%% Naive Decoder
x_hat = y;

%% Error Probability
PE_sim = 1-sum(x==x_hat)/n % Error probability from the simulation
% Calculation of the theoretical error probability
PC = 0;
for k = 1:length(S_X)
    t = S_X(k);
    i = find(S_Y == t);
    if length(i) == 1
        PC = PC+ p_X(k)*Q(k,i);
    end
end
PE_theretical = 1-PC
```

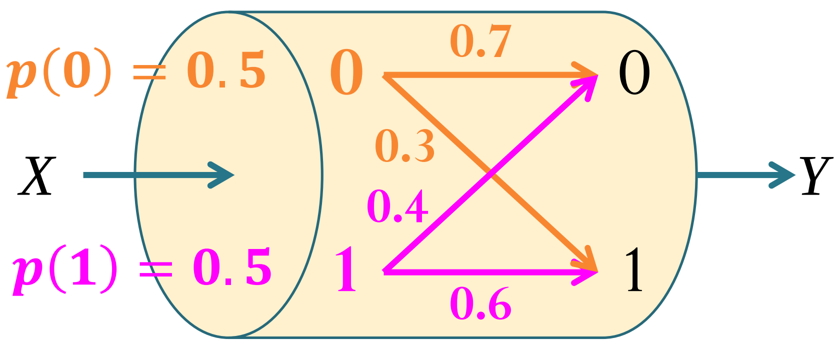

[DMC_Analysis_demo.m]

Ex: BAC

```matlab
%% Simulation parameters
% The number of symbols to be transmitted
n = 1e4;
% Binary Assymmetric Channel (BAC)
% Ex 3.8 in lecture note (11.3 in [Z&T, 2010])
% Channel Input
S_X = [0 1]; S_Y = [0 1];
p_X = [0.5 0.5];
% Channel Characteristics
Q = [0.7 0.3; 0.4 0.6];
```







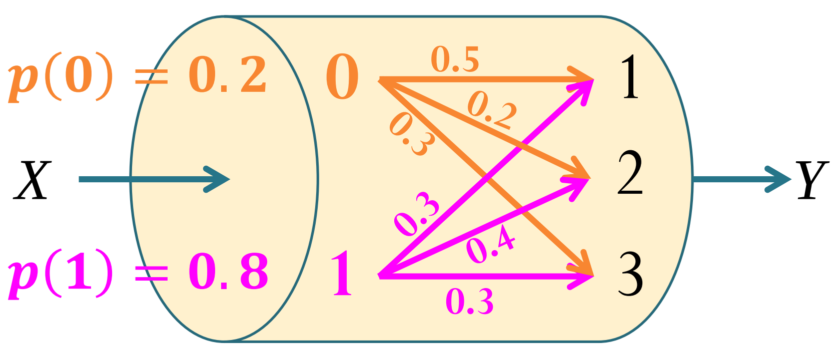

```
p_X =
    0.5000    0.5000
p_X_sim =
    0.5043    0.4957
q =
    0.5500    0.4500
q_sim =
    0.5532    0.4468
Q =
    0.7000    0.3000
    0.4000    0.6000
Q_sim =
    0.7109    0.2891
    0.3928    0.6072
PE_sim =
    0.3405
PE_theretical =
    0.3500
```

[BAC_demo.m]

Ex: DMC

```
%% Simulation parameters
% The number of symbols to be transmitted
n = 20;
% General DMC
% Ex. 3.16 in lecture note
% Channel Input
S_X = [0 1]; S_Y = [1 2 3];
p_X = [0.2 0.8];
% Channel Characteristics
Q = [0.5 0.2 0.3; 0.3 0.4 0.3];
```

```
>> DMC_demo
ans =
x:  1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 0 1
ans =
y:  1 3 2 2 1 2 1 2 2 3 1 1 1 3 1 3 2 3 1 2
```

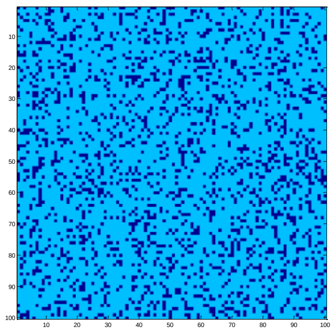


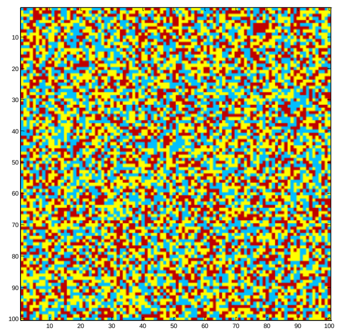

```
p_X =
    0.2000    0.8000
p_X_sim =
    0.2000    0.8000
q =
    0.3400    0.3600    0.3000
q_sim =
    0.4000    0.3500    0.2500
Q =
    0.5000    0.2000    0.3000
    0.3000    0.4000    0.3000
Q_sim =
    0.7500         0    0.2500
    0.3125    0.4375    0.2500
PE_sim =
    0.7500
PE_theretical =
    0.7600
```

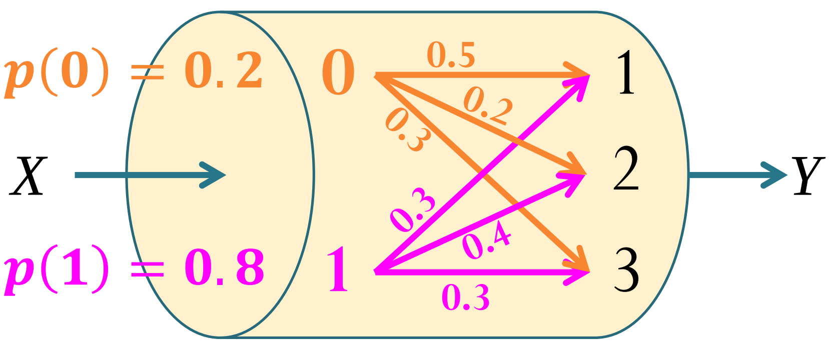
[DMC_demo.m]

Ex: DMC

```
%% Simulation parameters
% The number of symbols to be transmitted
n = 1e4;
% General DMC
% Ex. 3.16 in lecture note
% Channel Input
S_X = [0 1]; S_Y = [1 2 3];
p_X = [0.2 0.8];
% Channel Characteristics
Q = [0.5 0.2 0.3; 0.3 0.4 0.3];
```








```
p_X =
    0.2000    0.8000
p_X_sim =
    0.2011    0.7989
q =
    0.3400    0.3600    0.3000
q_sim =
    0.3387    0.3607    0.3006
Q =
    0.5000    0.2000    0.3000
    0.3000    0.4000    0.3000
Q_sim =
    0.4943    0.1914    0.3143
    0.2995    0.4033    0.2972
PE_sim =
    0.7607
PE_theretical =
    0.7600
```

[DMC_demo.m]

Evaluation of Probability

- Consider two random variables X and Y.
- Suppose their **joint pmf matrix** is

$P_{X,Y}$

x \ y	2	3	4	5	6
1	0.1	0.1	0	0	0
3	0.1	0	0	0.1	0
4	0	0.1	0.2	0	0
6	0	0	0	0	0.3

- Find $P[X + Y < 7]$

Step 1: Find the pairs (x,y) that satisfy the condition "x+y < 7"

One way to do this is to first construct the matrix of x+y.

$x + y$

x \ y	2	3	4	5	6
1	3	4	5	6	7
3	5	6	7	8	9
4	6	7	8	9	10
6	8	9	10	11	12

Evaluation of Probability

- Consider two random variables X and Y.
- Suppose their **joint pmf matrix** is

$P_{X,Y}$

x \ y	2	3	4	5	6
1	0.1	0.1	0	0	0
3	0.1	0	0	0.1	0
4	0	0.1	0.2	0	0
6	0	0	0	0	0.3

- Find $P[X+Y<7]$

Step 2: Add the corresponding probabilities from the joint pmf (matrix)

$x+y$

x \ y	2	3	4	5	6
1	3	4	5	6	7
3	5	6	7	8	9
4	6	7	8	9	10
6	8	9	10	11	12

$$P[X+Y<7] = 0.1+0.1+0.1 = 0.3$$

Evaluation of Probability

- Consider two random variables X and Y.
- Suppose their **joint pmf matrix** is

$$P_{X,Y} \quad \begin{array}{c|ccccc} x \backslash y & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0.1 & 0.1 & 0 & 0 & 0 \\ 3 & 0.1 & 0 & 0 & 0.1 & 0 \\ 4 & 0 & 0.1 & 0.2 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0.3 \end{array}$$

- Find $P[X = Y]$

$$P[X = Y] = 0 + 0.2 + 0.3 = 0.5$$

Sum of two discrete RVs

- Consider two random variables X and Y.
- Suppose their **joint pmf matrix** is

$P_{X,Y}$

x \ y	2	3	4	5	6
1	0.1	0.1	0	0	0
3	0.1	0	0	0.1	0
4	0	0.1	0.2	0	0
6	0	0	0	0	0.3

- Find $P[X+Y=7]$

$P[X+Y=7] = 0.1$

$x+y$

x \ y	2	3	4	5	6
1	3	4	5	6	7
3	5	6	7	8	9
4	6	7	8	9	10
6	8	9	10	11	12

Optimal Decoder for BSC

```
>> BSC_decoder_ALL_demo
Decoder_Table_ALL =
     0     1
     1     0
     1     1
     0     0
ans =
         0    1.0000    0.1000
    1.0000         0    0.9000
    1.0000    1.0000    0.8000
         0         0    0.2000
Optimal_Detector =
     0     1
Min_PE =
    0.1000
Elapsed time is 0.008709 seconds.
```

Column labels on the output: $\hat{x}(0)$, $\hat{x}(1)$, $P(\mathcal{E})$

```
close all; clear all;
tic

%% Simulation parameters
% Channel Input
S_X = [0 1]; S_Y = [0 1];
p0 = 0.8; p1 = 1-p0; p_X = [p0 p1];
% Channel Characteristics
p = 0.1; Q = [1-p p; p 1-p];

%% All possible "reasonable" decoders
% X_hat = Y; X_hat = 1-Y; X_hat = 1; X_hat = 0
Decoder_Table_ALL = [0 1; 1 0; 1 1; 0 0]

%% Calculate the error probability for each of the decoder
PE_ALL = [];
for k = 1:size(Decoder_Table_ALL,1)
    Decoder_Table = Decoder_Table_ALL(k,:);
    PC = 0;
    for k = 1:length(S_X)
        I = (Decoder_Table == S_X(k));
        Q_row = Q(k,:);
        PC = PC + p_X(k)*sum(Q_row(I));
    end
    PE_theretical = 1-PC;
    PE_ALL = [PE_ALL; PE_theretical];
end

%% Display the results
[Decoder_Table_ALL PE_ALL]

%% Find the optimal detectors
[V I] = min(PE_ALL);
Optimal_Detector = Decoder_Table_ALL(I,:)
Min_PE = V

toc
```


[BSC_decoder_ALL_demo.m]

DIY Decoder

```
>> DMC_decoder_DIY_demo
ans =
1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 0 0 1 0 1
ans =
2 1 1 3 3 1 2 2 1 2 1 2 3 1 1 3 1 3 1 1
ans =
1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0
PE_sim =
   0.5500
PE_theretical =
   0.5200
Elapsed time is 0.081161 seconds.
```

```
%% Simulation parameters
% The number of symbols to be transmitted
n = 20;
% General DMC
% Ex. 3.16 in lecture note
% Channel Input
S_X = [0 1]; S_Y = [1 2 3];
p_X = [0.2 0.8];
% Channel Characteristics
Q = [0.5 0.2 0.3; 0.3 0.4 0.3];
```

```
%% DIY Decoder
Decoder_Table = [0 1 0]; % The decoded
values corresponding to the received Y
```


[DMC_decoder_DIY_demo.m]

DIY Decoder

```
%% DIY Decoder
Decoder_Table = [0 1 0]; % The decoded values corresponding to the received Y
```

```
% Decode according to the decoder table
x_hat = y; % preallocation
for k = 1:length(S_Y)
    I = (y==S_Y(k));
    x_hat(I) = Decoder_Table(k);
end

PE_sim = 1-sum(x==x_hat)/n % Error probability from the simulation
```

```
% Calculation of the theoretical error probability
PC = 0;
for k = 1:length(S_X)
    I = (Decoder_Table == S_X(k));
    q = Q(k,:);
    PC = PC+ p_X(k)*sum(q(I));
end
PE_theretical = 1-PC
```

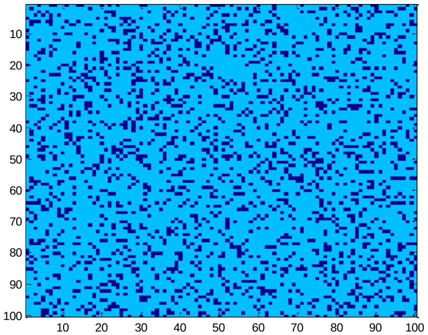

[DMC_decoder_DIY_demo.m]

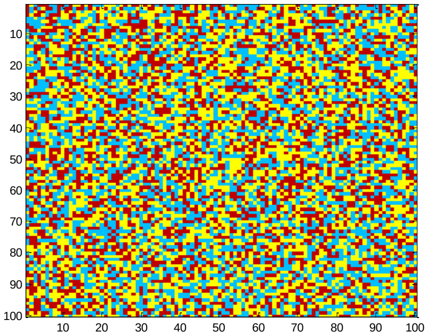
DIY Decoder

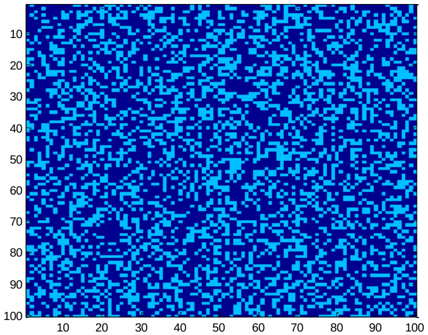
```
%% Simulation parameters
% The number of symbols to be transmitted
n = 1e4;
% General DMC
% Ex. 3.16 in lecture note
% Channel Input
S_X = [0 1]; S_Y = [1 2 3];
p_X = [0.2 0.8];
% Channel Characteristics
Q = [0.5 0.2 0.3; 0.3 0.4 0.3];
```

```
%% DIY Decoder
Decoder_Table = [0 1 0]; % The decoded
values corresponding to the received Y
```

```
>> DMC_decoder_DIY_demo
PE_sim =
   0.5213
PE_therethical =
   0.5200
Elapsed time is 2.154024 seconds.
```







[DMC_decoder_DIY_demo.m]

Searching for the Optimal Detector

```
>> DMC_decoder_ALL_demo
ans =   x̂(1)       x̂(2)       x̂(3)       P(ℰ)
             0          0          0     0.8000
             0          0     1.0000     0.6200
             0     1.0000          0     0.5200
             0     1.0000     1.0000     0.3400
        1.0000          0          0     0.6600
        1.0000          0     1.0000     0.4800
        1.0000     1.0000          0     0.3800
        1.0000     1.0000     1.0000     0.2000
Min_PE =
        0.2000
Optimal_Detector =
          1     1     1
Elapsed time is 0.003351 seconds.
```

Guessing Game

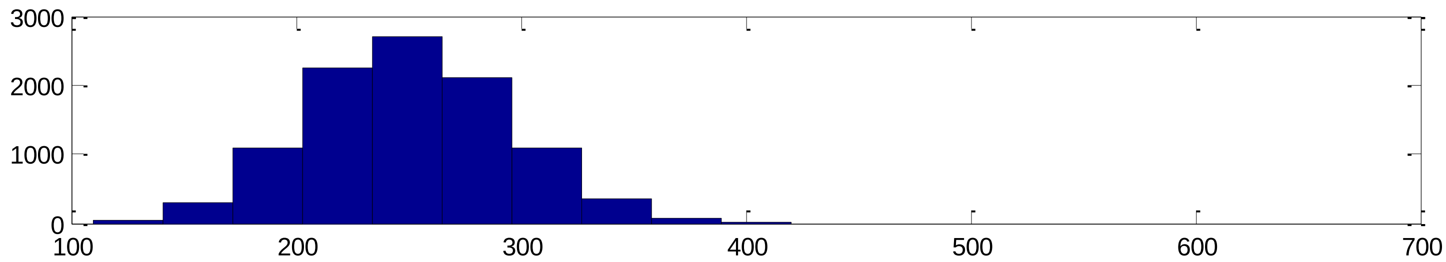
- Consider a random variable X whose pmf is

$$p_X(x) = \begin{cases} 1/4, & x = 1, 2, \\ 1/2, & x = 3, \\ 0, & \text{otherwise.} \end{cases}$$

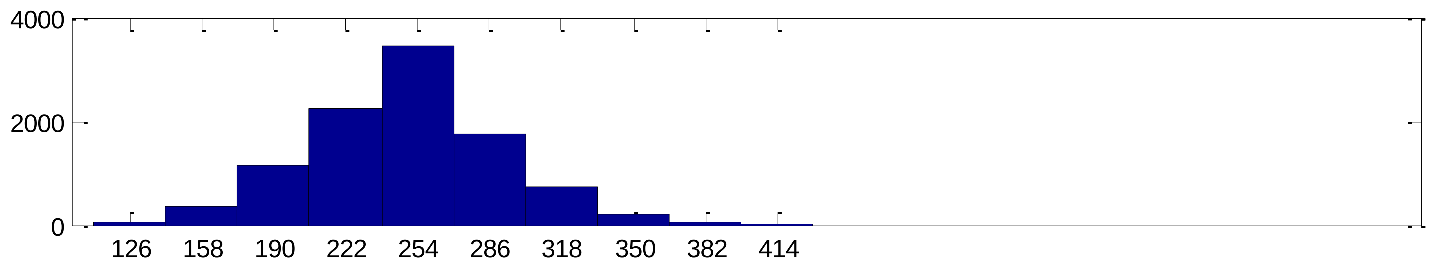
- You have to guess the value of X.
- This game is repeated many times. (Say, 100 times)
- The "goodness" of your guess is determined by how often you guess correctly.
  - For example, get ฿10 each time that you guess correctly.
- How should you guess?

Guessing Game

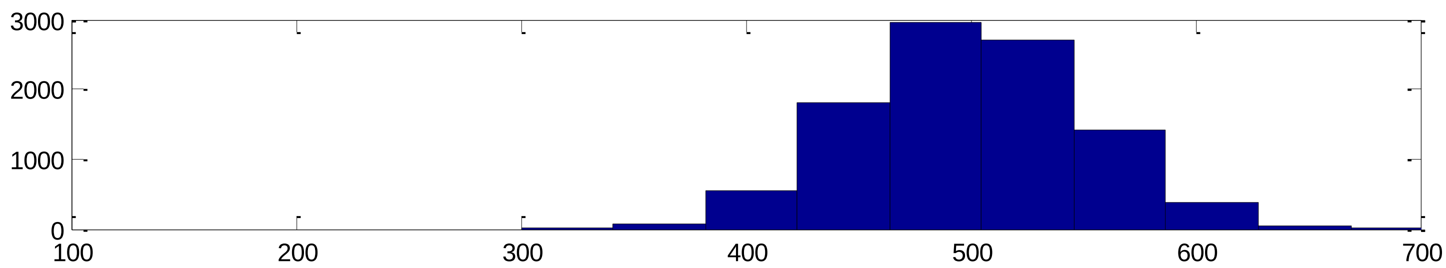
$$\hat{x} = 1$$



$$\hat{x} = 2$$



$$\hat{x} = 3$$



Guessing Game: Version 2

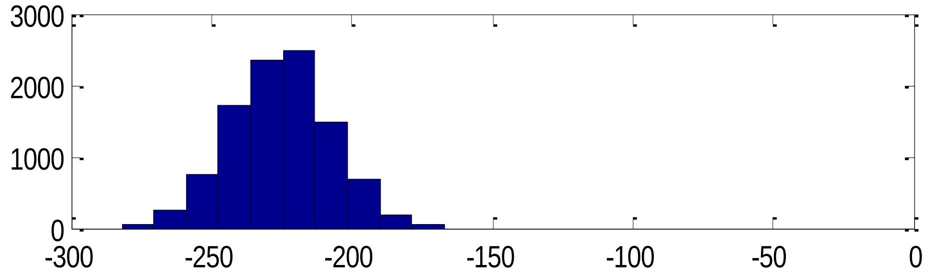
- Consider a random variable X whose pmf is

$$p_X(x) = \begin{cases} 1/4, & x = 1, 2, \\ 1/2, & x = 3, \\ 0, & \text{otherwise.} \end{cases}$$

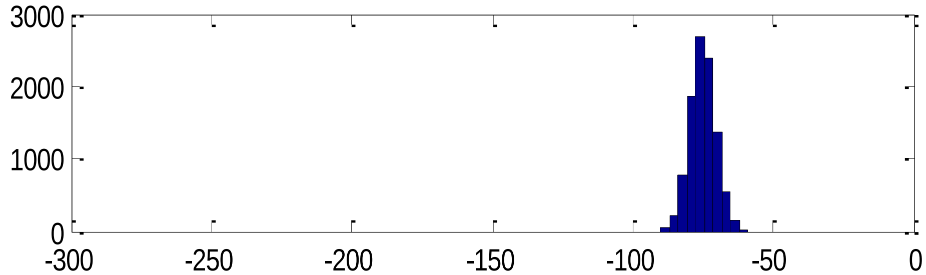
- You have to guess the value of X.
- This game is repeated many times. (Say, 100 times)
- The "goodness" of your guess is determined by how close you are to the value of X.
  - For example, you pay nothing if your guess is correct but you have to pay $(X - \hat{x})^2$ if your guess is wrong.
- How should you guess?

Guessing Game: Version 2

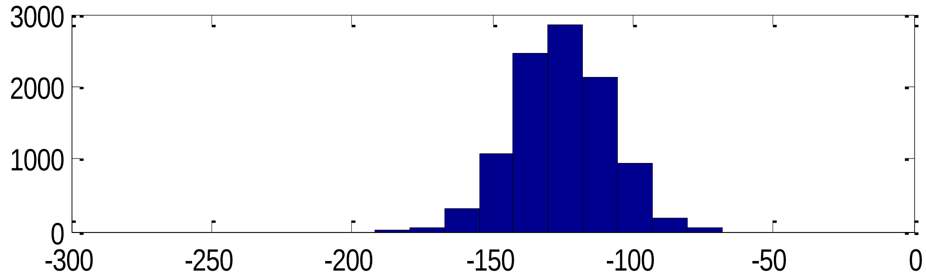
$\hat{x} = 1$



$\hat{x} = 2$

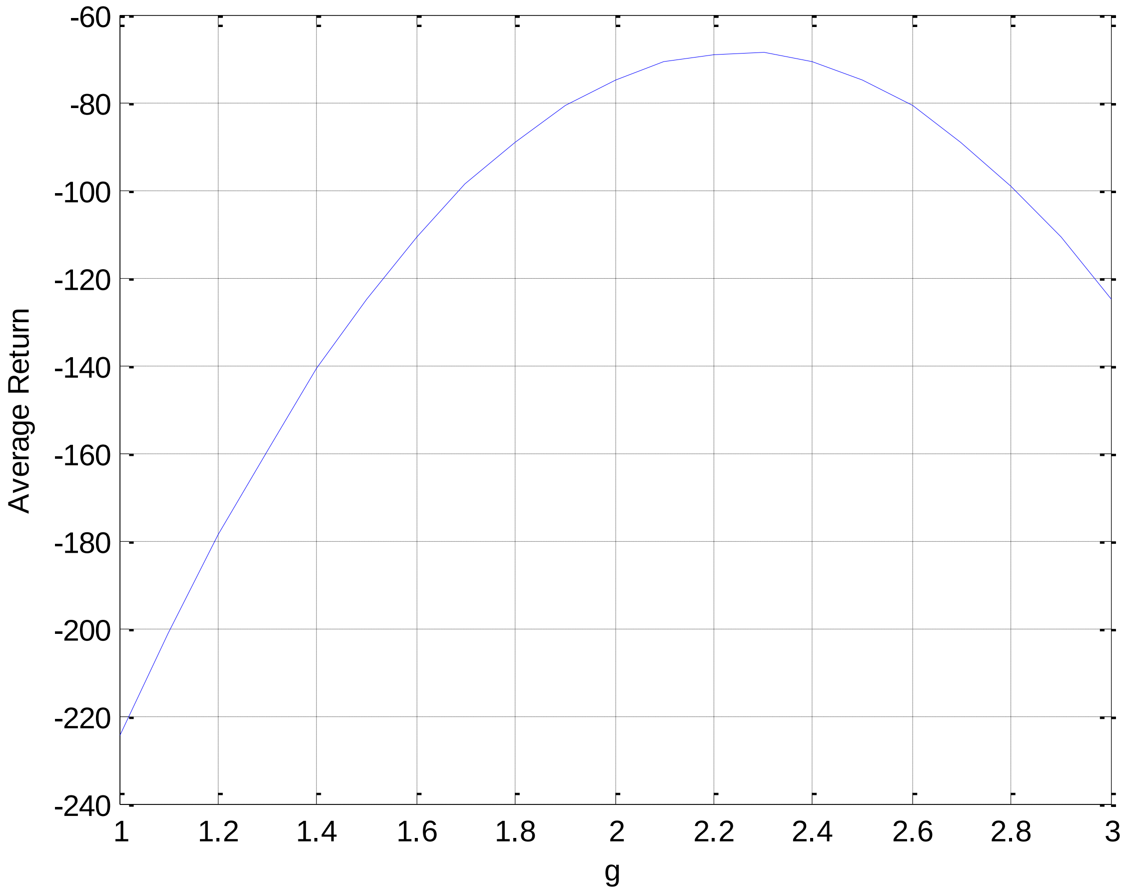


$\hat{x} = 3$



[GuessingGame_2.m]

Guessing Game: Version 2



[GuessingGame_2_2.m]

MAP Decoder

```
%% MAP Decoder
P = diag(p_X)*Q; % Weight the channel transition probability by the
                 % corresponding prior probability.
[V I] = max(P); % For I, the default MATLAB behavior is that when there are
                % multiple max, the index of the first one is returned.
Decoder_Table = S_X(I) % The decoded values corresponding to the received Y
```

```
%% Decode according to the decoder table
x_hat = y; % preallocation
for k = 1:length(S_Y)
    I = (y==S_Y(k));
    x_hat(I) = Decoder_Table(k);
end

PE_sim = 1-sum(x==x_hat)/n % Error probability from the simulation
```

```
%% Calculation of the theoretical error probability
PC = 0;
for k = 1:length(S_X)
    I = (Decoder_Table == S_X(k));
    Q_row = Q(k,:);
    PC = PC+ p_X(k)*sum(Q_row(I));
end
PE_theretical = 1-PC
```


[DMC_decoder_MAP_demo.m]

ML Decoder

```
%% ML Decoder
[V I] = max(Q); % For I, the default MATLAB behavior is that when there are
                % multiple max, the index of the first one is returned.
Decoder_Table = S_X(I) % The decoded values corresponding to the received Y
```

```
%% Decode according to the decoder table
x_hat = y; % preallocation
for k = 1:length(S_Y)
    I = (y==S_Y(k));
    x_hat(I) = Decoder_Table(k);
end

PE_sim = 1-sum(x==x_hat)/n % Error probability from the simulation
```

```
%% Calculation of the theoretical error probability
PC = 0;
for k = 1:length(S_X)
    I = (Decoder_Table == S_X(k));
    Q_row = Q(k,:);
    PC = PC+ p_X(k)*sum(Q_row(I));
end
PE_theretical = 1-PC
```


[DMC_decoder_ML_demo.m]