Digital Communication Systems
ECS 452

Asst. Prof. Dr. Prapun Suksompong

prapun@siit.tu.ac.th

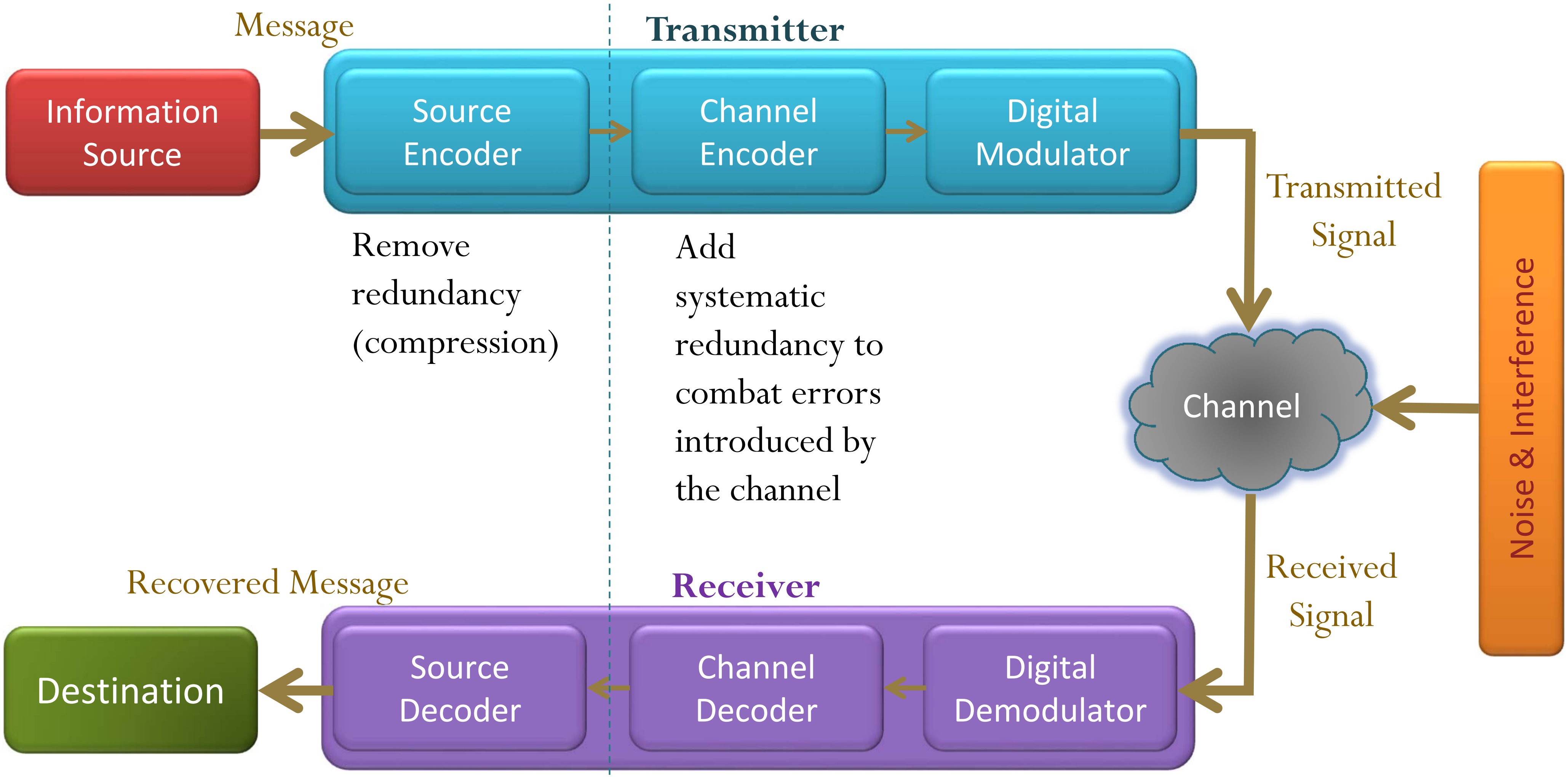
3 Discrete Memoryless Channel (DMC)

Office Hours:

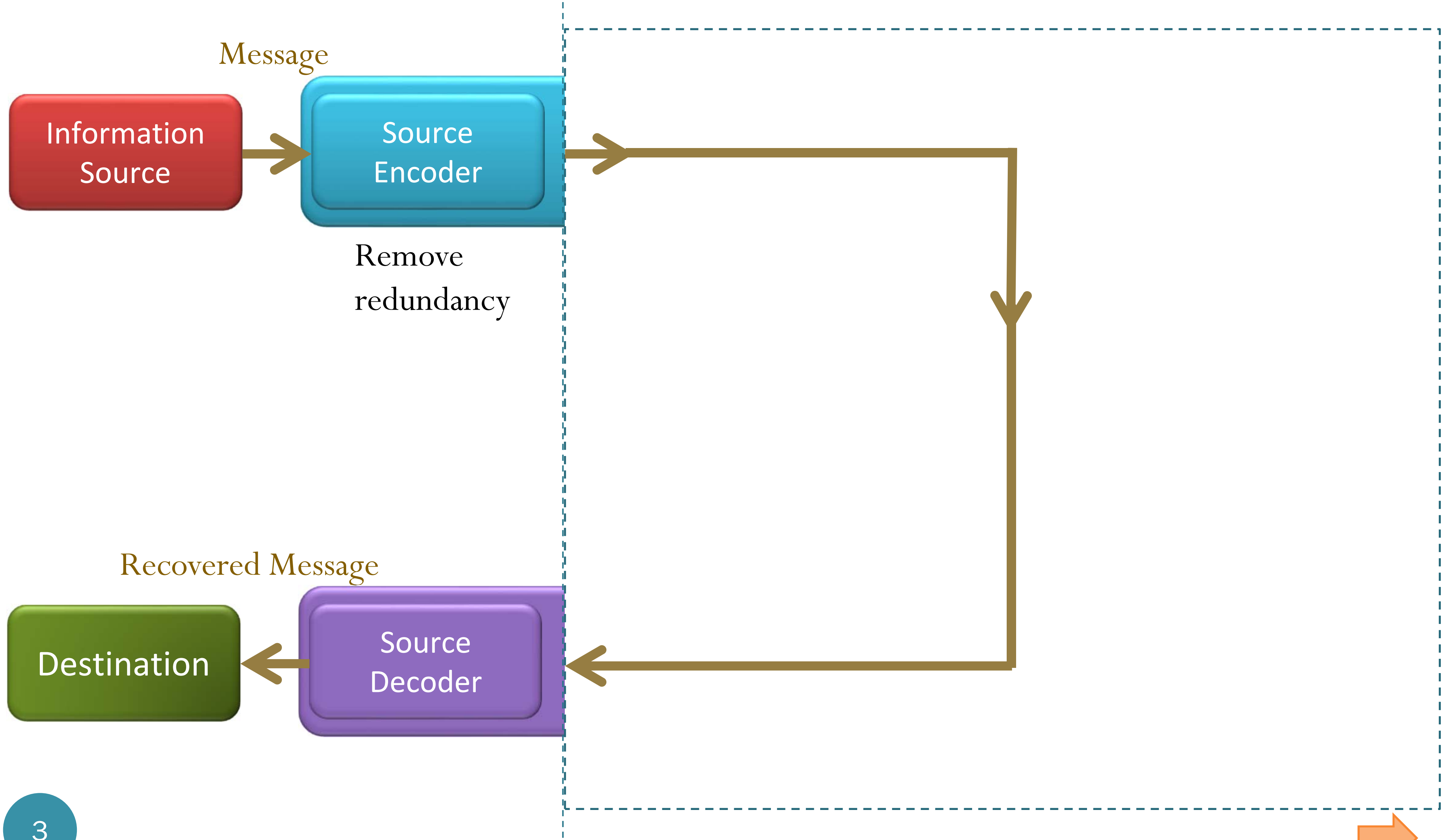
BKD, 6th floor of Sirindhralai building

Tuesday	14:20-15:20
Wednesday	14:20-15:20
Friday	9:15-10:15

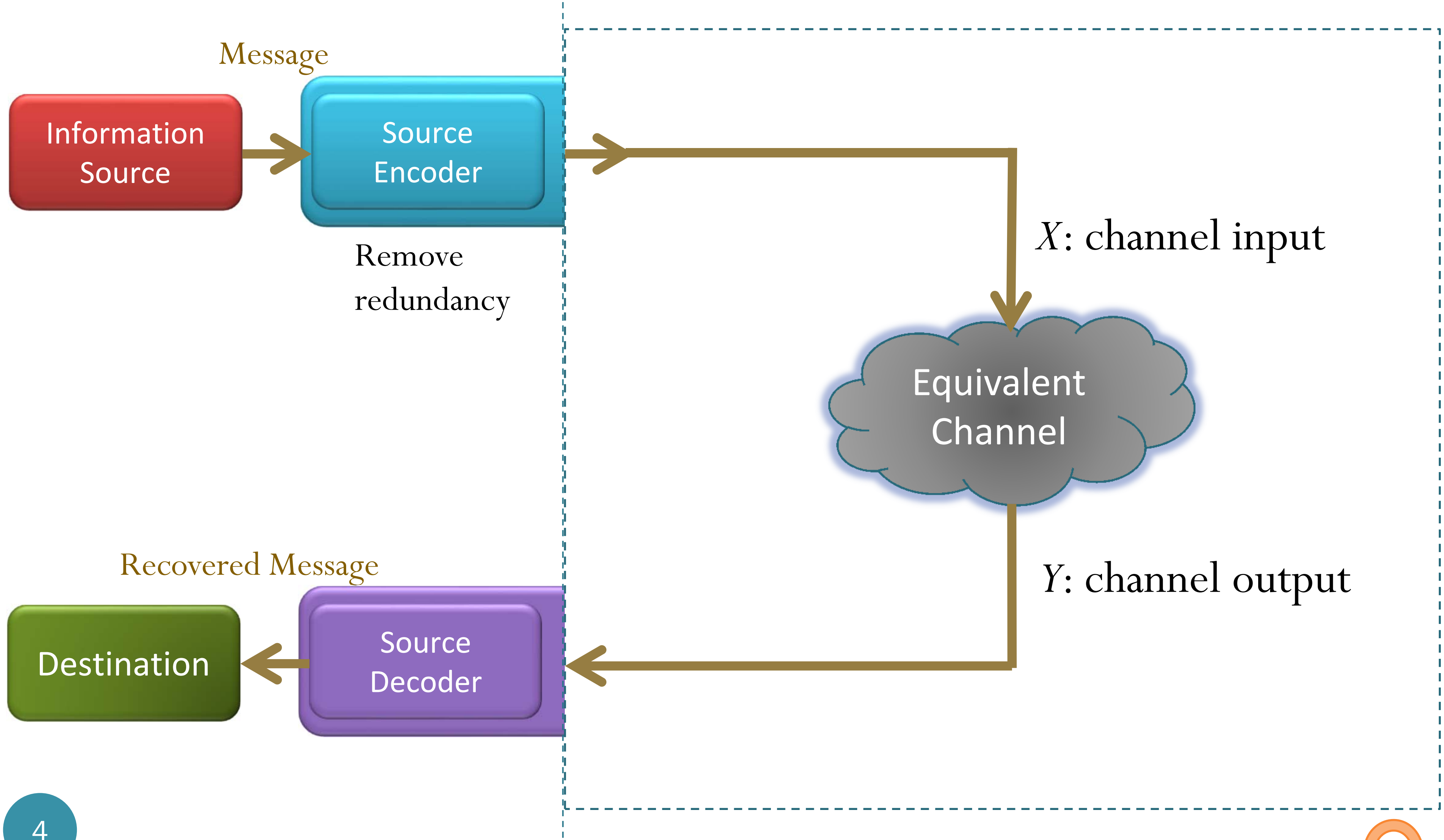
Elements of digital commu. sys.



System considered previously



System considered in this section



Digital Communication Systems
ECS 452

Asst. Prof. Dr. Prapun Suksompong
prapun@siit.tu.ac.th

3.1 DMC Models

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
```

```
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]

[Example 3.2]

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

Q =
    0.9000    0.1000
    0.1000    0.9000

q =
    0.3400    0.6600

Elapsed time is 0.134992 seconds.
```

[BSC_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]

[Example 3.2]

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

Elapsed time is 0.134992 seconds.
```

Because there are only 20 samples, we can't expect the relative freq. from the simulation to match the specified or calculated probabilities.

[BSC_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];
```

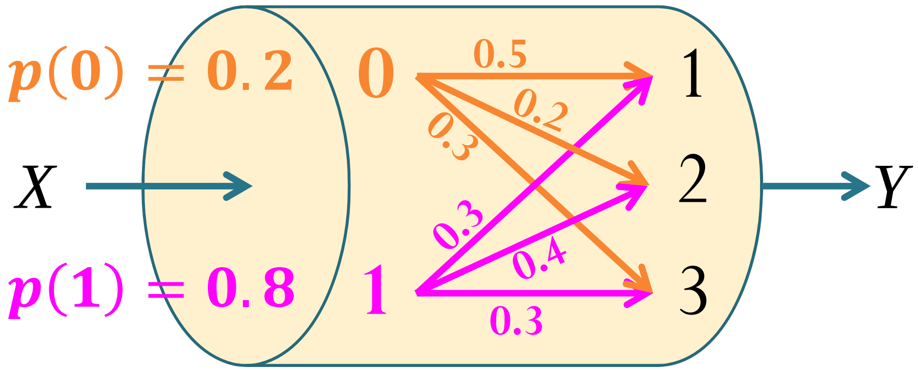
```
>> BSC_demo

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

Elapsed time is 0.922728 seconds.
```

[BSC_demo.m]

Ex: DMC

```
%% Simulation parameters
% The number of symbols to be transmitted
n = 20;
% General DMC
% Ex. 3.6 amd 3.12 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
>> sym(Q_sim)
ans =
[  3/4,    0, 1/4]
[ 5/16, 7/16, 1/4]
```

[DMC_demo.m]

Ex: DMC

```
>> p = [0.2 0.8]
p =
    0.2000    0.8000
>> p = [0.2 0.8];
>> Q = [0.75 0 0.25; 0.3125 0.4375 0.25];
>> p*Q
ans =
    0.4000    0.3500    0.2500
```

Block Matrix Multiplications

$$\left(\underbrace{\begin{matrix}10 & 6\\ 9 & 7\end{matrix}}_{A}\ \underbrace{\begin{matrix}6 & 4 & 3\\ 3 & 5 & 9\end{matrix}}_{B}\right)\times\left(\begin{matrix}\overbrace{\begin{matrix}2 & 2 & 5\\ 3 & 3 & 4\end{matrix}}^{C} & \overbrace{\begin{matrix}10 & 2 & 10 & 2 & 5\\ 5 & 10 & 5 & 3 & 6\end{matrix}}^{D}\\ \underbrace{\begin{matrix}3 & 3 & 4\\ 7 & 2 & 5\\ 8 & 3 & 6\end{matrix}}_{E} & \underbrace{\begin{matrix}1 & 1 & 5 & 5 & 6\\ 3 & 10 & 6 & 10 & 3\\ 9 & 8 & 3 & 6 & 5\end{matrix}}_{F}\end{matrix}\right)$$

$$=\left(\underbrace{\begin{matrix}108 & 73 & 136\\ 155 & 85 & 164\end{matrix}}_{AC+BE}\ \underbrace{\begin{matrix}175 & 150 & 193 & 126 & 149\\ 224 & 213 & 197 & 158 & 165\end{matrix}}_{AD+BF}\right)$$

$$\left(\underbrace{\begin{matrix}10 & 6 & 6 & 4 & 3\\ 9 & 7 & 3 & 5 & 9\end{matrix}}_{X}\right)\times\left(\underbrace{\begin{matrix}2 & 2 & 5 & 10\\ 3 & 3 & 4 & 5\\ 3 & 3 & 4 & 1\\ 7 & 2 & 5 & 3\\ 8 & 3 & 6 & 9\end{matrix}}_{G}\ \underbrace{\begin{matrix}2 & 10 & 2 & 5\\ 10 & 5 & 3 & 6\\ 1 & 5 & 5 & 6\\ 10 & 6 & 10 & 3\\ 8 & 3 & 6 & 5\end{matrix}}_{H}\right)$$

$$=\left(\underbrace{\begin{matrix}108 & 73 & 136 & 175\\ 155 & 85 & 164 & 224\end{matrix}}_{XG}\ \underbrace{\begin{matrix}150 & 193 & 126 & 149\\ 213 & 197 & 158 & 165\end{matrix}}_{XH}\right)$$

Review: Evaluation of Probability from the Joint PMF Matrix

- 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

Review: Evaluation of Probability from the Joint PMF Matrix

- 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)

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

$x + y$

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

Review: Evaluation of Probability from the Joint PMF Matrix

- 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 $P[X = Y]$

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

Review: 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

Ex: DMC

```
>> p = [0.2 0.8];
>> Q = [0.5 0.2 0.3; 0.3 0.4 0.3];
>> p*Q
ans =
    0.3400    0.3600    0.3000
>> P = (diag(p))*Q
P =
    0.1000    0.0400    0.0600
    0.2400    0.3200    0.2400
>> sum(P)
ans =
    0.3400    0.3600    0.3000
```

Digital Communication Systems
ECS 452

Asst. Prof. Dr. Prapun Suksompong

prapun@siit.tu.ac.th

3.2 Decoder and $P(\mathcal{E})$

$P(\mathcal{E})$ for Naïve Decoder

```
%% 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
```

Formula derived in 3.19 of lecture notes

[DMC_Analysis_demo.m]

Ex: Naïve Decoder and BAC [Ex. 3.18]

```
%% 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];
```

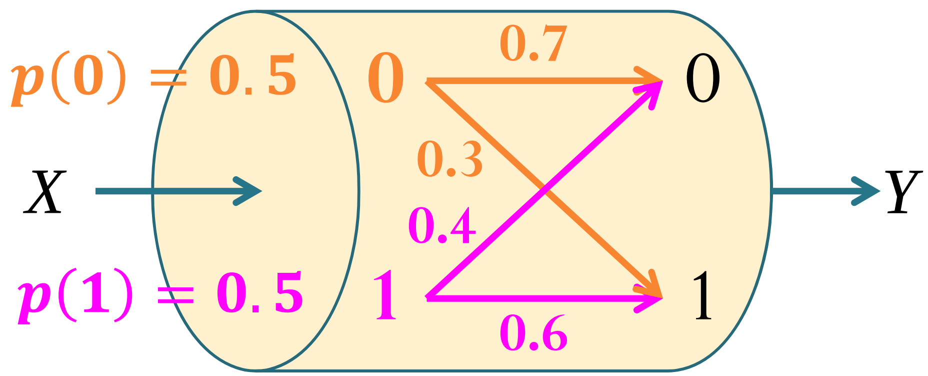
>> BAC_demo

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

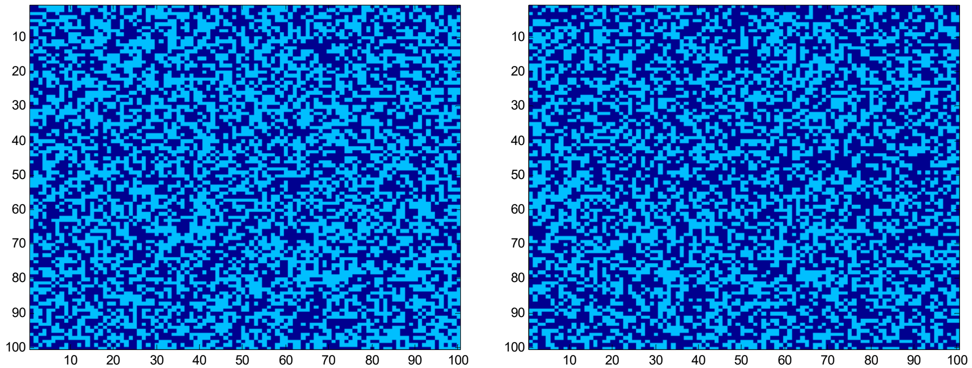
$\frac{7}{20}$ →

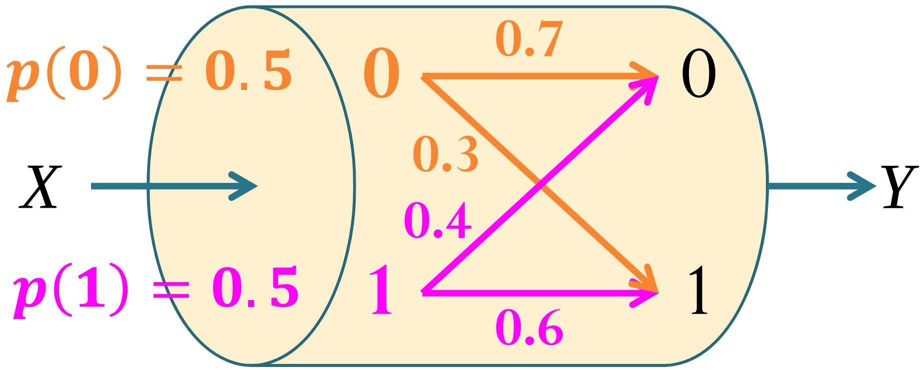
PE_sim =
0.3500
PE_theretical =
0.3500

[BAC_demo.m]

Ex: Naïve Decoder and BAC [Ex. 3.18]

```
%% 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: Naïve Decoder and DMC [Ex. 3.21]

```
%% 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];
```

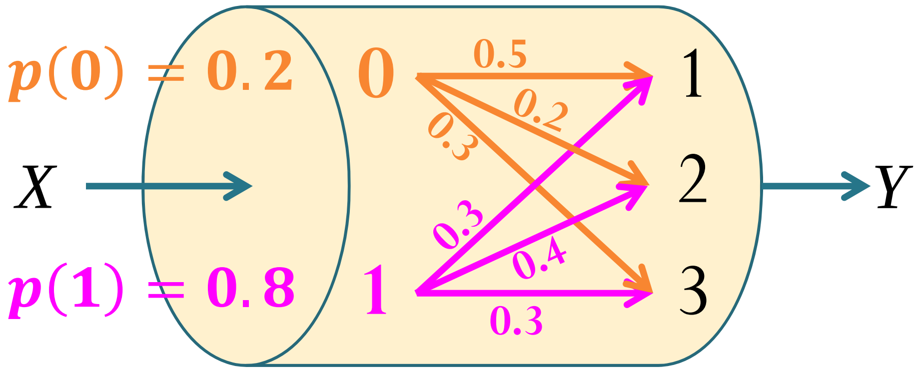
>> DMC_demo [Same samples as in Ex. 3.6]

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



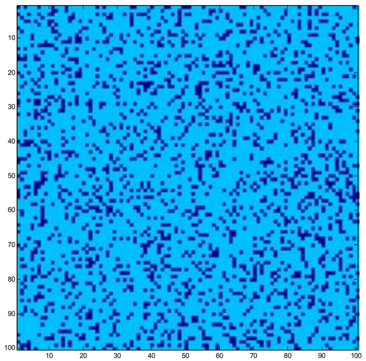
$$\frac{20-4}{20}$$

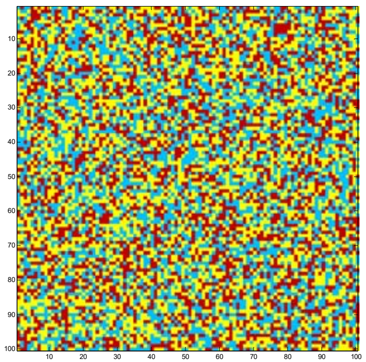
```
p_X =
   0.2000   0.8000
p_X_sim =
   0.2000   0.8000
q =
   0.3400   0.3600   0.3000
q_sim =
   0.4000   0.3500   0.2500
Q =
   0.5000   0.2000   0.3000
   0.3000   0.4000   0.3000
Q_sim =
   0.7500        0   0.2500
   0.3125   0.4375   0.2500
PE_sim =
   0.7500
PE_theretical =
   0.7600
```

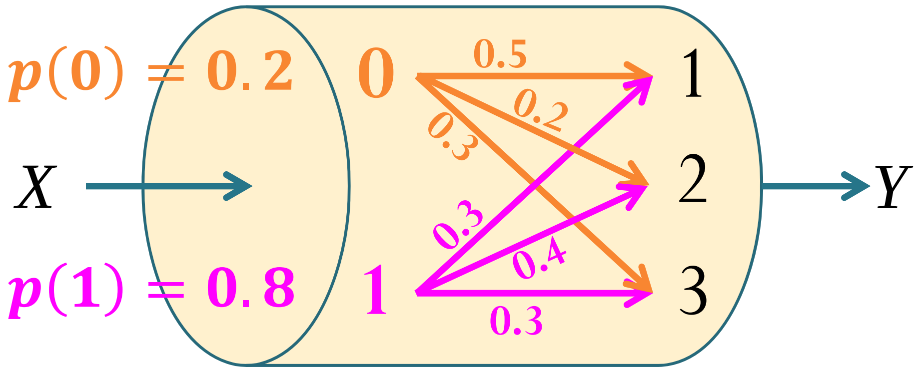
[DMC_demo.m]

Ex: Naïve Decoder and DMC [Ex. 3.21]

```
%% 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]

DIY Decoder [Ex. 3.22]

>> DMC_decoder_DIY_demo

ans =

X 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 0 0 1 0 1

ans =

Y 2 1 1 3 3 1 2 2 1 2 1 2 3 1 1 3 1 3 1 1

ans =

$\hat{X}$ 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 [Ex. 3.22]

```
%% 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 [Ex. 3.22]

```
%% 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

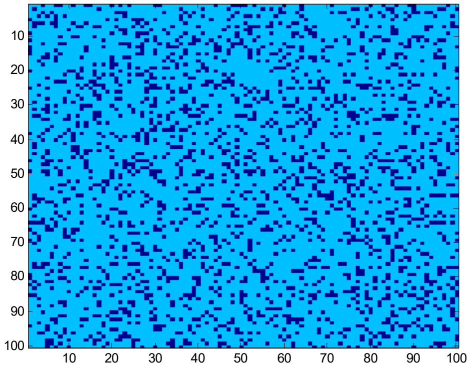
PE_sim =

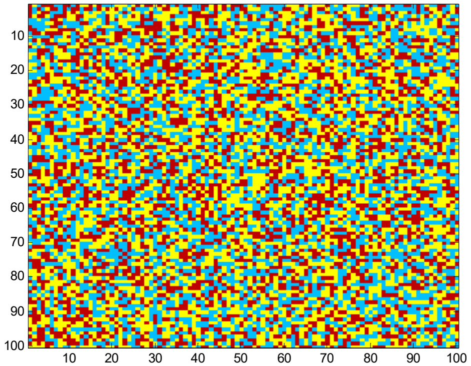
0.5213

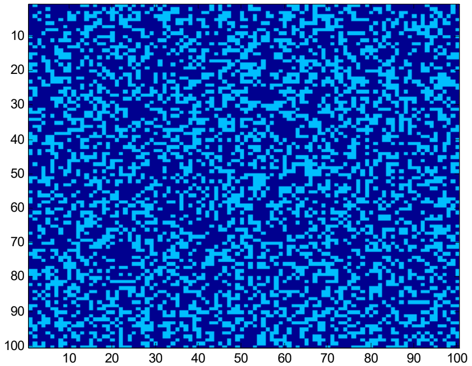
PE_theretical =

0.5200

Elapsed time is 2.154024 seconds.







[DMC_decoder_DIY_demo.m]

Digital Communication Systems
ECS 452

Asst. Prof. Dr. Prapun Suksompong

prapun@siit.tu.ac.th

3.3 Optimal Decoder

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    Ex. 3.22
            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    Ex. 3.23
       1.0000        1.0000        1.0000         0.2000
Min_PE =
       0.2000
Optimal_Detector =
        1      1      1
Elapsed time is 0.003351 seconds.
```

Review: ECS315 (2016)

6.4. ***Interpretation***: It is sometimes useful to interpret $P(A)$ as our knowledge of the occurrence of event A *before* the experiment takes place. Conditional probability[24] $P(A|B)$ is the ***updated probability*** of the event A given that we now know that B occurred (but we still do not know which particular outcome in the set B did occur).

Definition 6.5. Sometimes, we refer to $P(A)$ as

- **a priori probability**, or
- the **prior probability** of A, or
- the **unconditional probability** of A.

in which case, we refer to $P(A|B)$ as

- a posteriori probability
- the posterior probability
- conditional probability

Guessing Game 1

- There are 15 cards.
  - Each have a number on it.
  - Here are the 15 cards:

1 2 2 3 3 3 4 4 4 4 5 5 5 5 5

- One card is randomly selected from the 15 cards.
- You need to guess the number on the card.
- Have to pay 1 Baht for incorrect guess.

- The game is to be repeated $n = 10{,}000$ times.
- What should be your guess value?

```
close all; clear all;

n = 5; % number of time to play this game

D = [1 2 2 3 3 3 4 4 4 4 5 5 5 5 5];
X = D(randi(length(D),1,n));

if n <= 10
    X
end

g = 1
cost = sum(X ~= g)

if n > 1
averageCostPerGame = cost/n
end
```

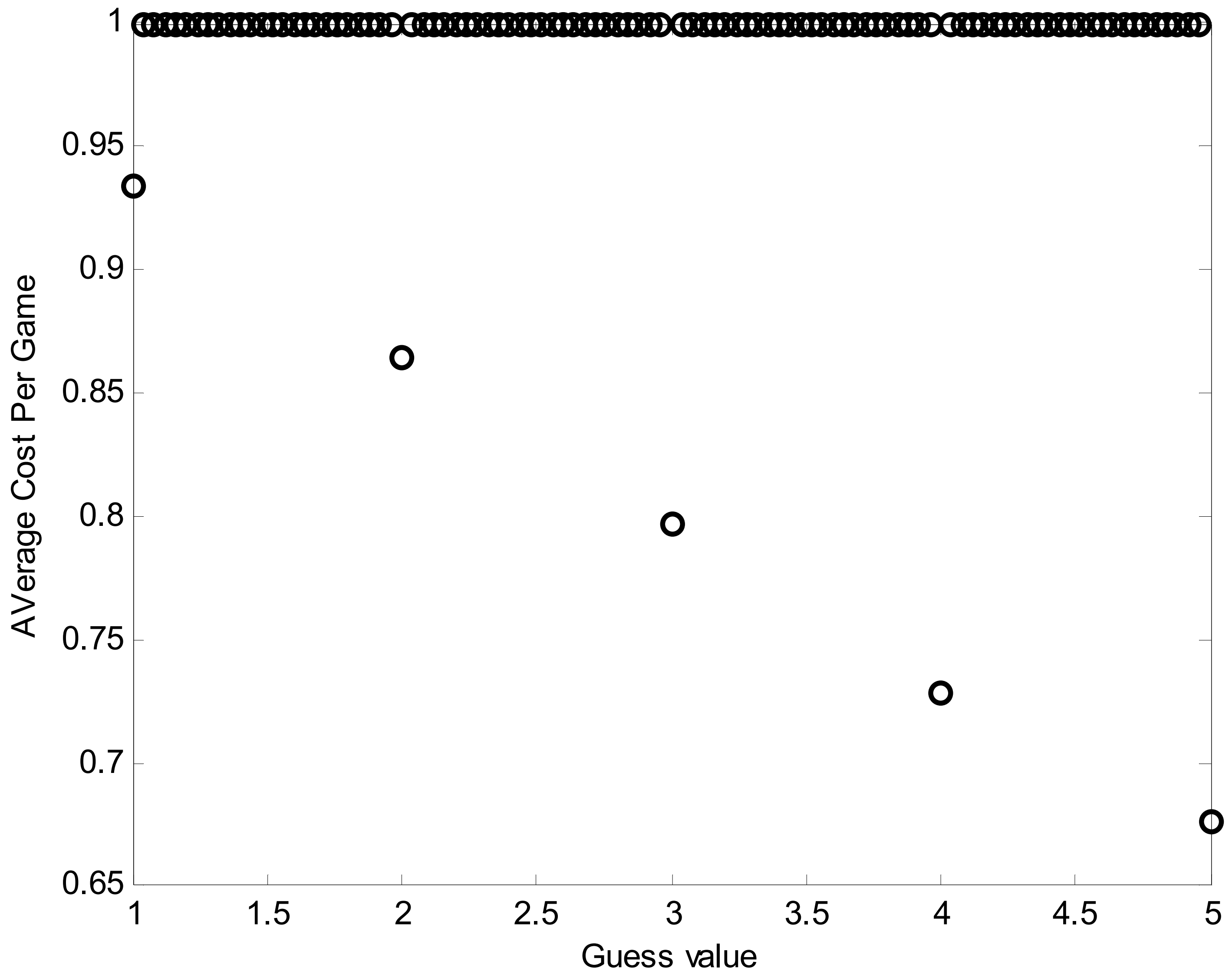
```
>> GuessingGame_4_1_1
X =
     3     5     1     2     5
g =
     1
cost =
     4
averageCostPerGame =
    0.8000
```

```
close all; clear all;

n = 5; % number of time to play this game

D = [1 2 2 3 3 3 4 4 4 4 5 5 5 5 5];
X = D(randi(length(D),1,n));

if n <= 10
    X
end

g = 3.3
cost = sum(X ~= g)

if n > 1
averageCostPerGame = cost/n
end
```

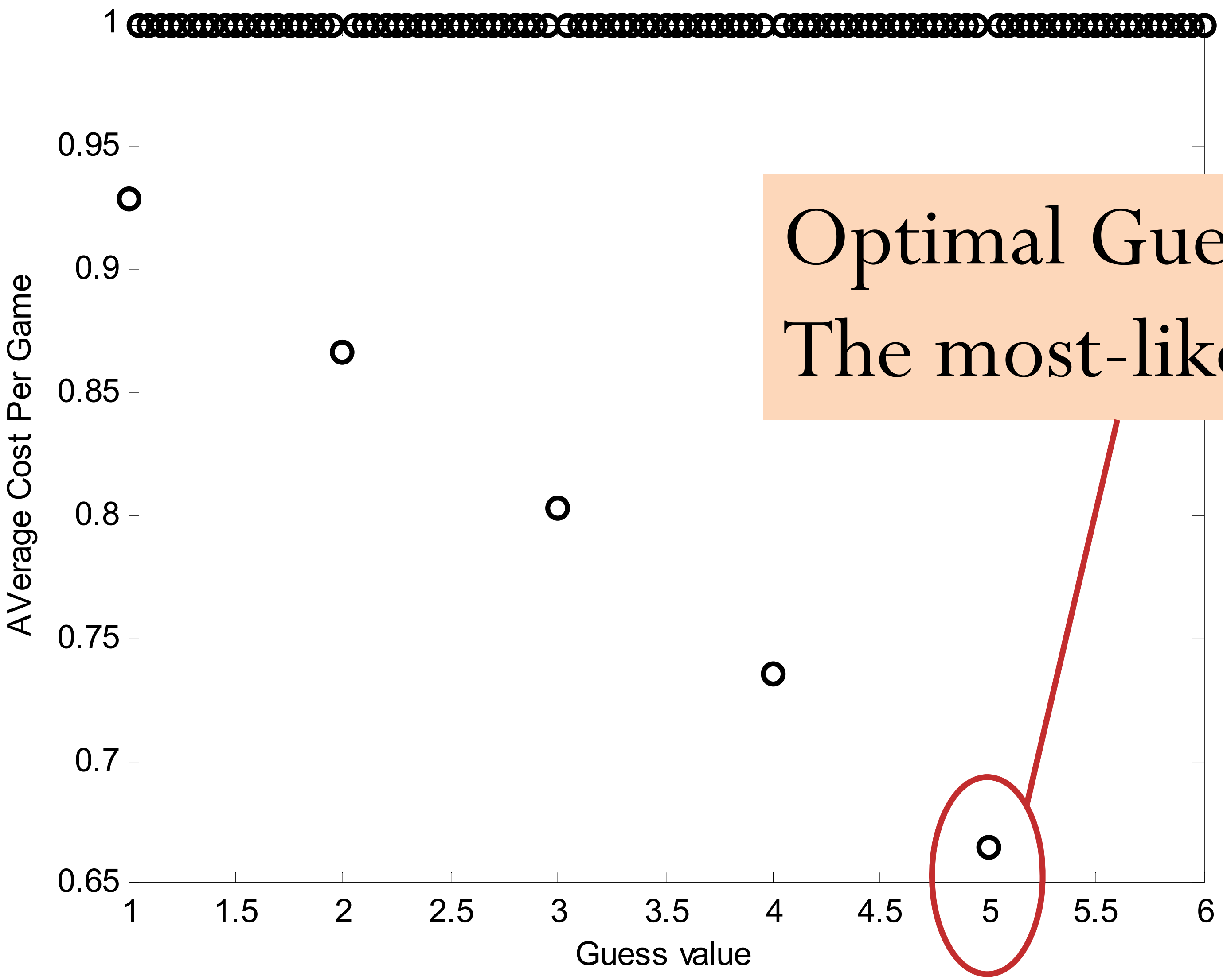
```
>> GuessingGame_4_1_1
X =
     5     3     2     4     1
g =
    3.3000
cost =
     5
averageCostPerGame =
     1
```

```
close all; clear all;

n = 1e4; % number of time to play this game

D = [1 2 2 3 3 3 4 4 4 4 5 5 5 5 5];
X = D(randi(length(D),1,n));

if n <= 10
    X
end

g = ?
cost = sum(X ~= g)

if n > 1
averageCostPerGame = cost/n
end
```

Guessing Game 1



Guessing Game 1



Optimal Guess:
The most-likely value

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]