ELEN90054 Probability and Random Models

MATLAB Workshop 2 on Discrete Channel Simulation and Repetition Coding
week 4 = week of 19 March 2018

Prepare this workshop individually, in particular **complete** Questions a),b),d),g) and j), and read through all questions before you come to your workshop session.

This workshop is worth 5% of the overall subject assessment and will be done in pairs. Allocation will be random by your demonstrator at the start of your workshop session, therefore you should not choose your project partner yourself. The next workshop will have a new allocation. Be aware that seeking or providing detailed assistance from/to people other than your workshop partner is collusion - see **http://academichonesty.unimelb.edu.au/plagiarism.html**

Each group is expected to upload two files:

1. A pdf (scanned/typed) file containing their worked solutions;
    a. Only **one** member of the group needs to upload the **pdf** file.
    b. The naming convention of the file should include workshop number, day, time and the assigned group number, e.g.
    Workshop_2_Mon09_Gp5.pdf for group 5 in the Monday 0900 hrs workshop slot.
2. A **single** zip file, containing all the required functions and a main script which calls these functions to generate the required outputs as outlined in the workshop questions.
    a. Only **one** member of the group needs to upload the **zip** file.
    b. The following naming convention should **strictly** be followed:
    Workshop_2_Mon12_Gp5_Matlab.zip for group 5 in the Monday 1200 hrs workshop slot.

The workshop times are: Mon09, Mon12, Tue10, Tue17, Wed13, Wed17, Wed18, Wed19, Thu11, Thu19, Fri10, Fri13. The group numbers are randomly assigned by your demonstrator at the start of the workshop.

Both submissions should be made before the start of the next workshop of week 5. **This is a strict deadline**.

In special circumstances you can email the pdf/code to your demonstrator. For demonstrator email address information, see "staff info" on the ELEN90054 LMS site.

Notes:

1) The predefined MATLAB command *rand* generates a random real number U that is *uniformly distributed* in [0,1) – i.e. every value in [0,1) is equally probable.
To simulate an event A with $P[A]=p$, we can use *rand* as follows: construct a suitable sub-interval I in [0,1] of length p, call the *rand* procedure, and declare the event A to have occurred if and only if U is in I (since $P[U \text{ in } I] = p$).

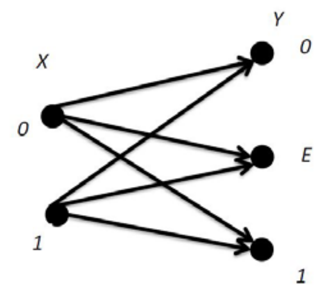
Calling the *rand* function several times in a row (for example via a for-loop) yields mutually independent results (for practical purposes anyway; strictly speaking, they are actually pseudo-random number generators, and cannot yield true unpredictability). For more information, type 'help rand' in the MATLAB command window.

2) In a large number n of independent trials, the **empirical pmf** (probability mass function) of a discrete random variable X is the **proportion** of times that X takes the value x, with x varying over the range of X.

Similarly, its **empirical cdf** (cumulative distribution function) is the proportion of times that X takes values less than or equal to x, with x varying over the range of X.

3) To get accurate results in this workshop, you may have to think carefully about how to avoid multiplying lots of large numbers together, which would propagate numerical rounding errors.

The main purpose of this workshop is to get you to experiment with discrete channel simulation and how to plot channel performance. Again, you'll get a taste of the law of large numbers (to be discussed in more detail in later lectures), as well as repetition coding of digital information.



<u>Discrete Channel Simulation and Repetition Coding</u> (total = 36 marks + 4 on-time attendance marks)

<u>Important:</u> Make sure that all your MATLAB coding has clear comments, so that the marker (and yourself) can understand what is going on.

An imperfect discrete communication channel accepts a transmitted input bit X (with value '0' or '1') and produces a received output symbol Y (with value '0', '1' or 'E') as follows:

- Given an input bit '0' or '1', it is *erased* with probability $\boldsymbol{p_1}$ which means that the output of the channel is regarded as an *erasure symbol E* (this models *receiver erasures*, e.g. due to buffer overflow)
- Given an input bit '0' or '1', it is *swapped* with probability $\boldsymbol{p_2}$, which means that an input '0' yields output '1' and an input '1' yields output '0' (this models *receiver errors*).

Let the probability that the source produces an input bit '0' be $\boldsymbol{q}$ **= 0.5**, unless stated otherwise.

Questions

a) (1 mark) Assume that $\boldsymbol{p_1 = 0.2}$ **and** $\boldsymbol{p_2 = 0.05}$. Copy the above figure (large) and label all edges with probabilities.

b) (3 marks) Assume again that $\boldsymbol{p_1 = 0.2}$ **and** $\boldsymbol{p_2 = 0.05}$. To visualize the computation of the joint probability $P(X=x, Y=y)$, draw the tree diagram that corresponds to one channel use. Also compute $P(Y=1)$.

c) (2 marks) Write a MATLAB function that generates the input bit (see page 1 "Notes") and simulates one channel use; your function should produce the input bit and the received symbol.

d) (5 marks) Assume that $\boldsymbol{p_1 = 0}$ **and** $\boldsymbol{p_2 = 0.05}$. Suppose we try and communicate the value X of one input bit via n=99 independent uses of this channel via **repetition coding**. In other words, thinking of each channel use as a separate time-slot, we transmit the value of the input bit n times. Thus we transmit either 00…0 or 11…1 (strings of length n). Let S_n be the number of occurring bit-swaps. Taking n=99, find an exact expression for its pmf; also Poisson approximate this expression. Plot both the exact and the approximate pmf (use your Poisson expression), using MATLAB; **compare and comment**.

Questions continue on the next page

e) (4 marks) Assume again that $\boldsymbol{p_1 = 0}$ **and** $\boldsymbol{p_2 = 0.05}$. Taking n=99, write a MATLAB function that simulates the output S_n, as defined above, by using your MATLAB code of part (c).
Plot the **empirical pmf's** for S_n by calling this function $\boldsymbol{M}$ **=10,000 times**, and in the same plot the exact pmf's (use your earlier code). **Compare and comment**.

f) (5 marks) Assume that $\boldsymbol{p_1 = 0}$. Use MATLAB and part (e) to plot, for different values of M, the empirical average number of bit-swaps that occur in n=99 channel uses, as a function of the value of p_2, with p_2 ranging from p_2 =0.05 to p_2 =0.8, in steps of 0.05; take M=2, M=5, M=10, M=100, M=1000, M=10,000. **Compare and comment; how do your results relate to the notion of "expected value"**?

g) (5 marks) Assume that n is any positive odd integer > 1, $\boldsymbol{p_1 = 0}$ **and that** $\boldsymbol{p_2}$ **has a value < 0.5.** Again, suppose we try and communicate the value X of an input bit via repetition coding as described in part d). Suppose the receiver acts as a decoder that does not know the value of the input bit but does know q and does know that repetition coding has been used by the transmitter. The receiver observes the n values of Y (here denoted as $y_1, y_2, ..., y_n$) and has the following **Decision Rule**:

'X=1' if and only if less than n/2 of the y_j's are equal to '0'.

Show mathematically that, for any received values $y_1, y_2, ..., y_n$, where less than n/2 of the y_j's are equal to '0', the following inequality holds:

$$P(X=0 \mid y_1, y_2, ..., y_n) < P(X=1 \mid y_1, y_2, ..., y_n).$$

h) (4 marks) Assume that $\boldsymbol{p_1 = 0}$. Taking n=99, use MATLAB and part (g) to plot the empirical average number of decoder errors as a function of the value of p_2, with p_2 ranging from p_2 =0 to p_2 =0.45, in steps of 0.05. Take M=10,000, as in part e).

i) (3 marks) Assume again that $\boldsymbol{p_1 = 0}$. Repeat part h) for n=5. Explain the difference with the plot that you obtained in h) (to see this put the two plots in one figure).

(4 marks) Assume that $\boldsymbol{q = 0.3}$, $\boldsymbol{p_1 = 0.2}$ and $\boldsymbol{p_2 = 0}$. Suppose we try and communicate the value X of an input bit via repetition coding as described in part d). **Formulate** a simple decision rule, similar to the one in g), that decides on the value of X (***hint:*** this is not entirely trivial).

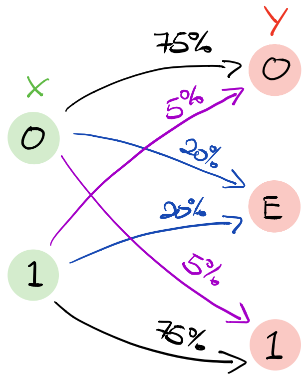
Make sure that with your decision rule, for any received values $y_1, y_2, ..., y_n$, the following inequality holds if and only if your rule says 'X=1' :

$$P(X=0 \mid y_1, y_2, ..., y_n) < P(X=1 \mid y_1, y_2, ..., y_n).$$

Show that your decision rule achieves this.

End of MATLAB Workshop 2 Questions

a) (1 mark) Assume that $p_1 = 0.2$ **and** $p_2 = 0.05$**.** Copy the above figure (large) and label all edges with probabilities.



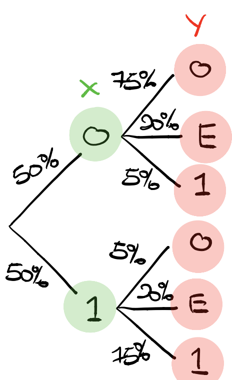
$P(\text{erased}) = 0.2$

$p(\text{swap}) = 0.05$

$p(\text{correct}) = 1 - 0.2 - 0.05$

$= 0.75$

b) (3 marks) Assume again that $p_1 = 0.2$ **and** $p_2 = 0.05$**.** To visualize the computation of the joint probability $P(X=x, Y=y)$, draw the tree diagram that corresponds to one channel use. Also compute $P(Y=1)$.



$P(Y=1)$

$= P(X=0)P(Y=1 \mid X=0) + P(X=1)P(Y=1 \mid X=1)$

$= 0.5\,(0.05) + 0.5\,(0.75)$

$= 0.4$

d) (5 marks) Assume that $p_1 = 0$ **and** $p_2 = 0.05$. Suppose we try and communicate the value X of one input bit via n=99 independent uses of this channel via **repetition coding**. In other words, thinking of each channel use as a separate time-slot, we transmit the value of the input bit n times. Thus we transmit either 00…0 or 11…1 (strings of length n). Let S_n be the number of occurring bit-swaps. Taking n=99, find an exact expression for its pmf; also Poisson approximate this expression. Plot both the exact and the approximate pmf (use your Poisson expression), using MATLAB; **compare and comment**.

$P(Y=1|X=0) = P_2$
$= 0.05$

$p_n(S=s) = C_s^n P_2^s [1-P_2]^{n-s}$

where $n=99$

Poisson approximation,

$P_2 = \frac{\alpha}{n}$

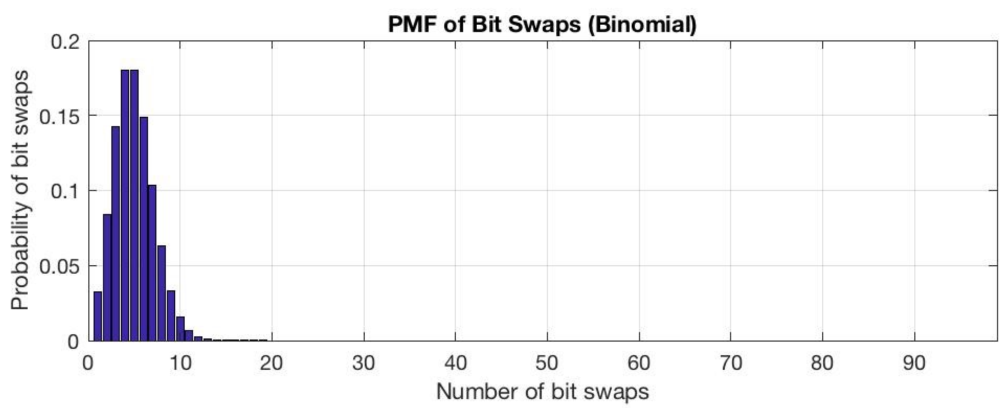
$\alpha = P_2 n$

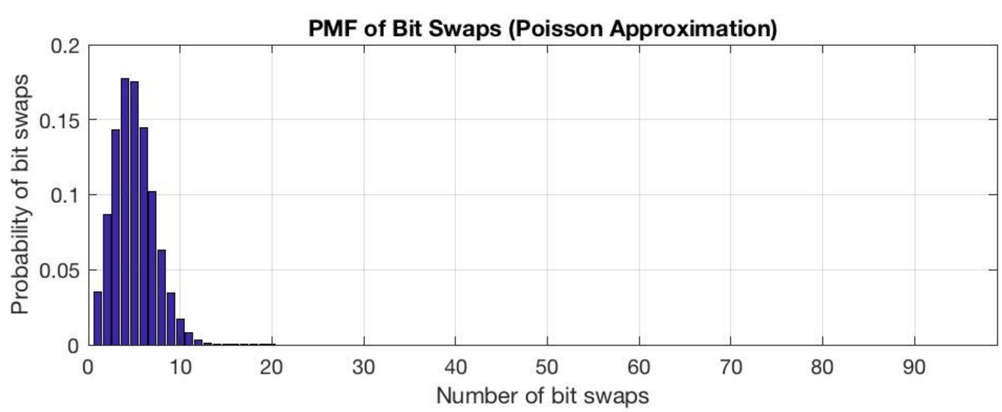
$= 4.95$

$\therefore P_{99}(S=s) \approx \frac{\alpha^s}{s!} e^{-\alpha}$

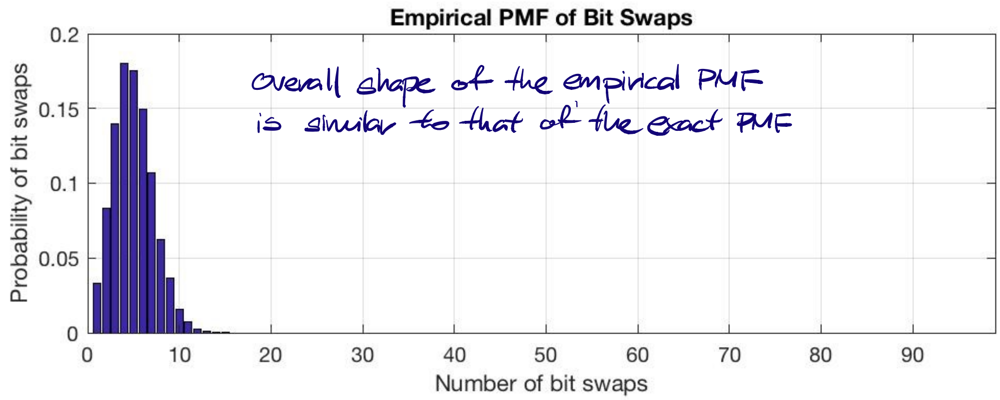
$= \frac{4.95^s}{s!} e^{-4.95}$

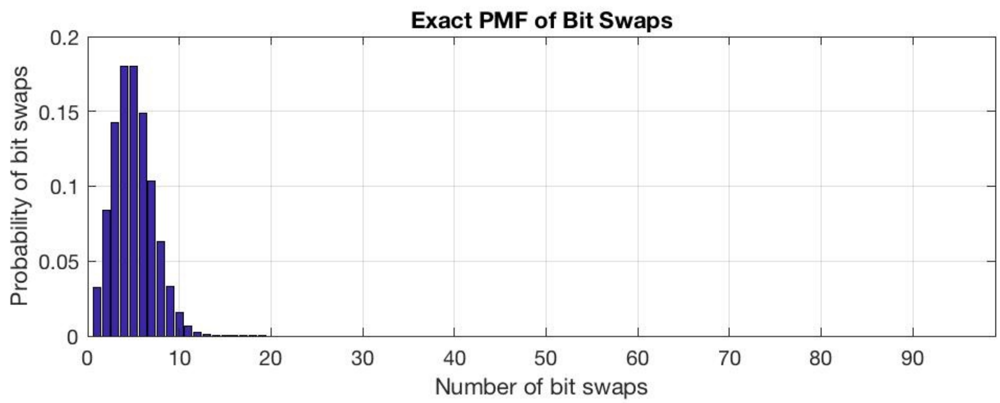
The poisson approximation is close to that of the exact distribution.





e) (4 marks) Assume again that $\boldsymbol{p_1 = 0}$ **and** $\boldsymbol{p_2 = 0.05}$**.** Taking n=99, write a MATLAB function that simulates the output S_n, as defined above, by using your MATLAB code of part (c).
Plot the **empirical pmf's** for S_n by calling this function $\boldsymbol{M}$ **=10,000 times**, and in the same plot the exact pmf's (use your earlier code). **Compare and comment**.

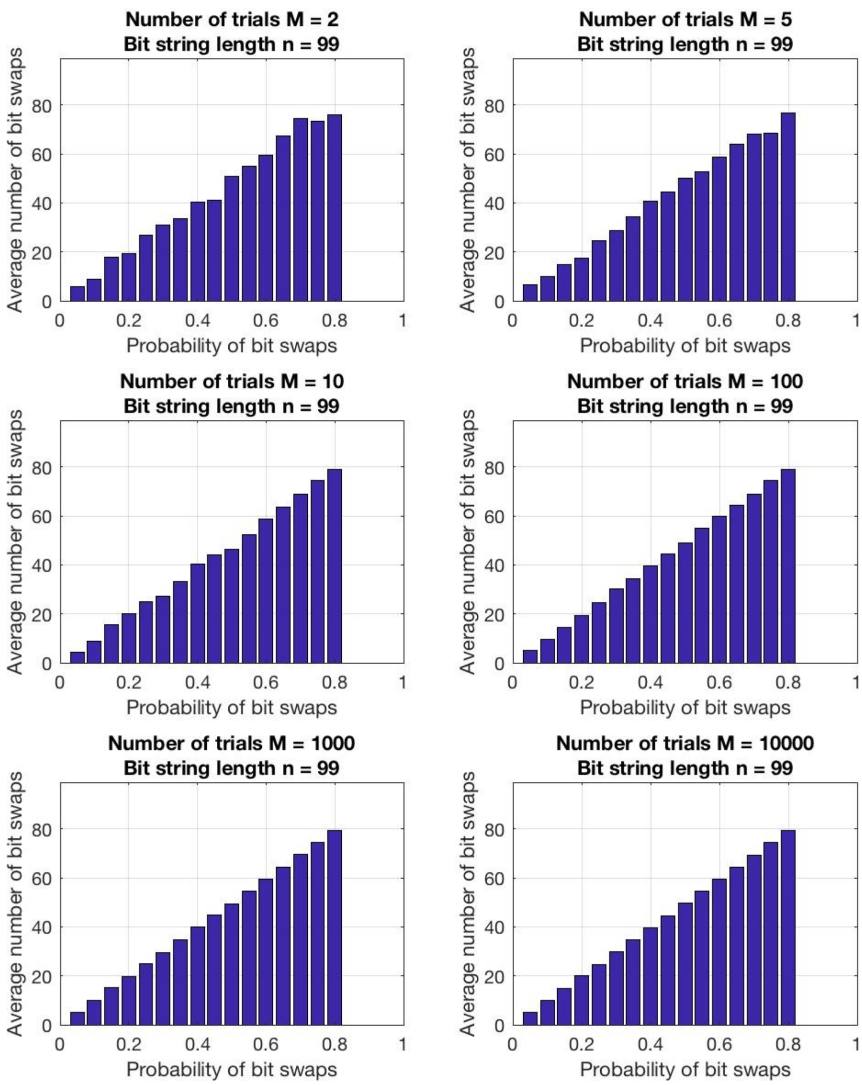




f) (5 marks) Assume that $p_1 = 0$. Use MATLAB and part (e) to plot, for different values of M, the empirical average number of bit-swaps that occur in n=99 channel uses, as a function of the value of p_2, with p_2 ranging from p_2 =0.05 to p_2 =0.8, in steps of 0.05; take M =2, M =5, M =10, M =100, M =1000, M =10,000. **Compare and comment; how do your results relate to the notion of "expected value"?**

↳ The more number of trials conducted, the closer the expected value is to the theoritical expected value calculated with the probability.

↳ The expected values of plot with M = {100, 1000, 10000} are the closest to its respective probability of bit swaps, suggesting that performing 100 trials is sufficient for the empirical average (expected value) to approach the probability of bit swaps.



g) (5 marks) Assume that n is any positive odd integer > 1, $p_1 = 0$ **and that** p_2 **has a value < 0.5.** Again, suppose we try and communicate the value X of an input bit via repetition coding as described in part d). Suppose the receiver acts as a decoder that does not know the value of the input bit but does know q and does know that repetition coding has been used by the transmitter. The receiver observes the n values of Y (here denoted as $y_1, y_2, \ldots, y_n$) and has the following **Decision Rule**:

'X=1' if and only if less than $n/2$ of the y_j's are equal to '0'.

Show mathematically that, for any received values $y_1, y_2, \ldots, y_n$, where less than $n/2$ of the y_j's are equal to '0', the following inequality holds:

$$P(X=0 \mid y_1, y_2, \ldots, y_n) < P(X=1 \mid y_1, y_2, \ldots, y_n).$$

$$P(X=0 \mid y_1, y_2, \ldots, y_n) = \frac{P(y_1, y_2, \ldots, y_n) \mid X=0) \cdot P(X=0)}{P(y_1, y_2, \ldots, y_n)}$$

$$= \frac{P(y_1, y_2, \ldots, y_n) \mid X=0)}{2\, P(y_1, y_2, \ldots, y_n)}$$

$$P(X=0) = q = \frac{1}{2}$$

$$\therefore P(X=1) = 1-q = \frac{1}{2}$$

$$P(X=1 \mid y_1, y_2, \ldots, y_n) = \frac{P(y_1, y_2, \ldots, y_n) \mid X=1) \cdot P(X=1)}{P(y_1, y_2, \ldots, y_n)}$$

$$= \frac{P(y_1, y_2, \ldots, y_n) \mid X=1)}{2\, P(y_1, y_2, \ldots, y_n)}$$

Suppose $Y = \{1, 0, 1\}$, when $X = \{0, 0, 0\}$

X	Y	P(bit swap)
0	1	P_2
0	0	$(1-P_2)$
0	1	P_2

$$P(y_1, y_2, \ldots, y_n \mid X=0) = P_2(1-P_2)P_2$$

For a general case of Y,

$$P(y_1, y_2, \ldots, y_n \mid X=0) = C^n_k P_2^k (1-P_2)^{n-k}$$

Suppose A is $Y = \{1, 0, 1\}$, when $X = \{1, 1, 1\}$

X	Y	P(bit swap)
1	1	$(1-P_2)$
1	0	P_2
1	1	$(1-P_2)$

$$P(y_1, y_2, \ldots, y_n \mid X=1) = (1-P_2)\, P\, (1-P_2)$$

For a general case of Y,

$$P(y_1, y_2, \ldots, y_n \mid X=1) = C^n_k (1-P_2)^k P_2^{n-k}$$

$$\therefore P(x=0 \mid y_1, y_2, \cdots, y_n) = \frac{C_k^n \, P_2^k (1-P_2)^{n-k}}{2P(y_1, y_2, \cdots, y_n)}$$

$$\therefore P(x=1 \mid y_1, y_2, \cdots, y_n) = \frac{C_k^n (1-P_2)^k P_2^{n-k}}{2P(y_1, y_2, \cdots, y_n)}$$

$$\therefore P(x=1 \mid y_1, y_2, \cdots, y_n) - P(x=0 \mid y_1, y_2, \cdots, y_n)$$

$$= \frac{C_k^n (1-P_2)^k P_2^{n-k}}{2P(y_1, y_2, \cdots, y_n)} - \frac{C_k^n \, P_2^k (1-P_2)^{n-k}}{2P(y_1, y_2, \cdots, y_n)}$$

$$= \frac{C_k^n}{2P(A)} \left[(1-P_2)^k P_2^{n-k} - P_2^k (1-P_2)^{n-k} \right]$$

$$= \frac{C_k^n \, P_2^k (1-p)^k}{2P(A)} \left[p^{n-2k} - (1-p)^{n-2k} \right]$$

If $P_2 < 0.5$, then $1-P_2 > P_2$

$$\therefore (1-P_2)^{n-2k} > P_2^{n-2k}$$

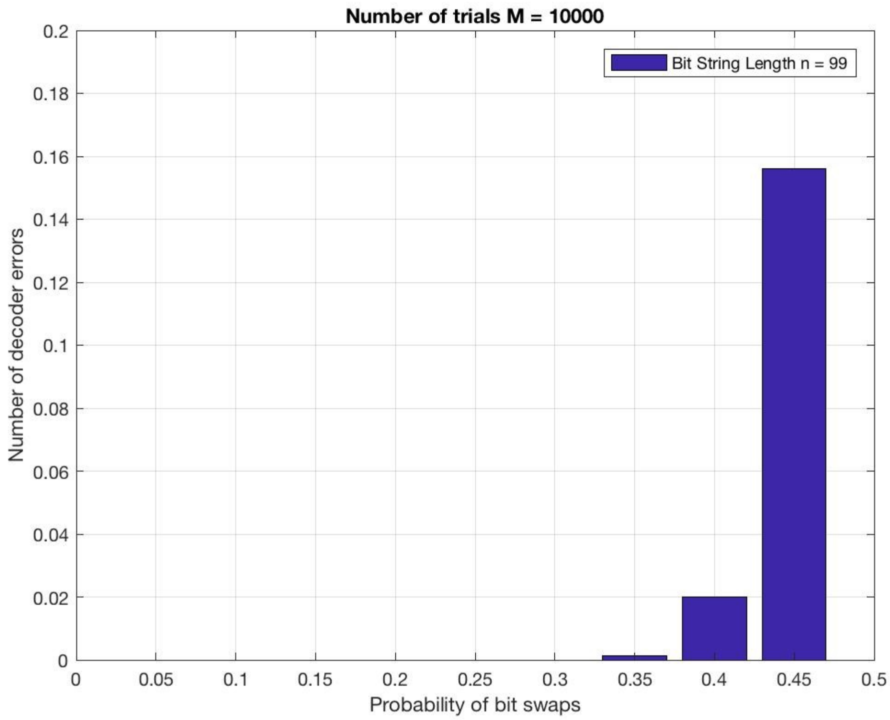
$$(1-P_2)^{n-2k} - P_2^{n-2k} > 0$$

$$\therefore P(x=1 \mid y_1, y_2, \cdots, y_n) - P(x=0 \mid y_1, y_2, \cdots, y_n) = \underbrace{\frac{C_k^n \, P_2^k (1-p)^k}{2P(A)}}_{>0} \underbrace{\left[(1-p)^{n-2k} - p^{n-2k} \right]}_{>0}$$

$$\therefore P(x=1 \mid y_1, y_2, \cdots, y_n) - P(x=0 \mid y_1, y_2, \cdots, y_n) > 0$$

$$P(x=1 \mid y_1, y_2, \cdots, y_n) > P(x=0 \mid y_1, y_2, \cdots, y_n)$$

h) (4 marks) Assume that $\boldsymbol{p_1} = \mathbf{0}$. Taking n=*99*, use MATLAB and part (g) to plot the empirical average number of decoder errors as a function of the value of p_2, with p_2 ranging from p_2 =0 to p_2 =0.45, in steps of 0.05. Take M =10,000, as in part e).

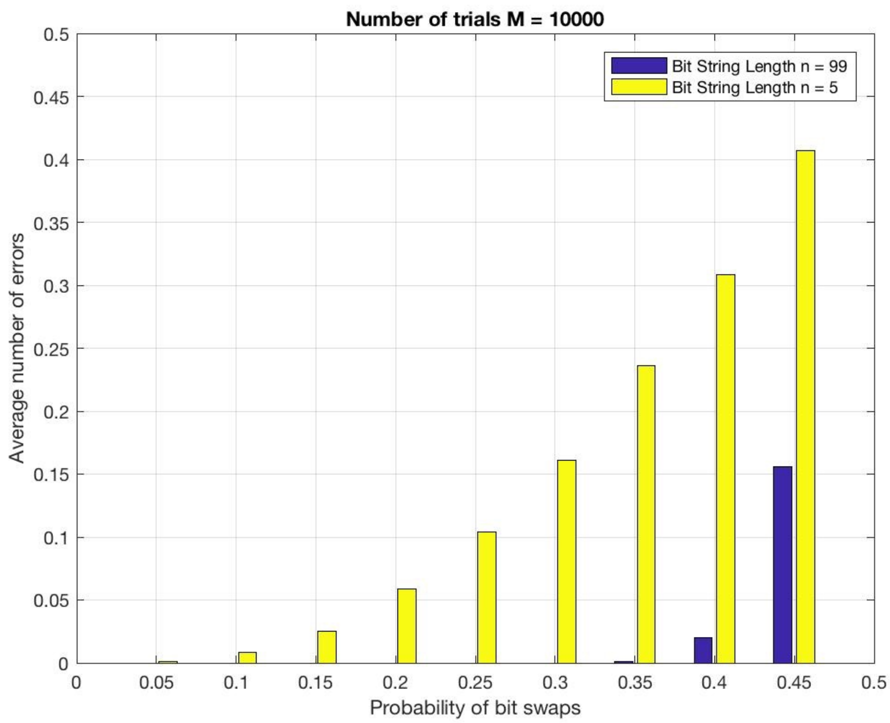


i) (3 marks) Assume again that $\boldsymbol{p_1 = 0}$. Repeat part h) for n=5. Explain the difference with the plot that you obtained in h) (to see this put the two plots in one figure).

(4 marks) Assume that $\boldsymbol{q = 0.3}$, $\boldsymbol{p_1 = 0.2}$ and $\boldsymbol{p_2 = 0}$. Suppose we try and communicate the value X of an input bit via repetition coding as described in part d). **Formulate** a simple decision rule, similar to the one in g), that decides on the value of X (***hint:*** this is not entirely trivial).

Make sure that with your decision rule, for any received values $y_1, y_2, \ldots, y_n$, the following inequality holds if and only if your rule says 'X=1' :

$$P(X=0 \mid y_1, y_2, \ldots, y_n) < P(X=1 \mid y_1, y_2, \ldots, y_n).$$



For a bit string length of n = 99, the bit string is not so prone to errors if the probability of bit swap p_2 is less than 0.4.

For a bit string length of n=5, the bit string is very prone to errors! Given the probability of bit swap p_2 = 0.2, the average number of errors already suppasses 0.05.

Therefore, the longer the bit string length, the less it will be prone to errors.

$P(X=0) = q$ $\quad P(X=1) = 1-q$ $\quad p_1 = 0.2$

$= 0.3$ $\quad = 0.7$ $\quad p_2 = 0$

X and Y are independent

$\therefore P(X=1, Y=0) = P(X=1)P(Y=0)$
$= (1-q)p_2$
$= 0$

$\therefore P(X=0, Y=1) = P(X=0)P(Y=1)$
$= q\,p_2$
$= 0$

Thus if $X=0$, Y cannot be 1 and if $X=1$, Y cannot be 0. Bit swap cannot occur.

We left with 3 cases,

Case 1 : If any bit of Y is 0, then X should be 0.

$$P(X=0 \mid y_1, y_2 \cdots y_n) > P(X=1 \mid y_1, y_2, \cdots y_n)$$

$$P(X=0 \mid y_1, y_2 \cdots y_n) > 0$$

Case 2 : If any bit of Y is 1, then X should be 1.

$$P(X=1 \mid y_1, y_2 \cdots y_n) > P(X=0 \mid y_1, y_2, \cdots y_n)$$

$$P(X=1 \mid y_1, y_2 \cdots y_n) > 0$$

Case 3 : When all bits of Y are 'E', then X should be 1.

Proof

$$\frac{P(X=0 \mid y_1, y_2, \cdots y_n)}{P(X=1 \mid y_1, y_2, \cdots y_n)} = \frac{\dfrac{P(y_1, y_2, \cdots y_n \mid X=0) \cdot P(X=0)}{P(y_1, y_2, \cdots y_n)}}{\dfrac{P(y_1, y_2, \cdots y_n \mid X=1) \cdot P(X=1)}{P(y_1, y_2, \cdots y_n)}}$$

$$= \frac{P(y_1, y_2, \cdots y_n \mid X=0) \cdot P(X=0)}{P(y_1, y_2, \cdots y_n \mid X=1) \cdot P(X=1)}$$

$$= \frac{p_1^n q}{p_1^n (1-q)}$$

$$= \frac{0.2^n (0.3)}{0.2^n (0.7)}$$

$$= \frac{3}{7}$$

Notice that $\frac{3}{7} < 1$

$$\therefore \frac{P(x=0 \mid y_1, y_2, \cdots y_n)}{P(x=1 \mid y_1, y_2, \cdots y_n)} < 1$$

$$P(x=0 \mid y_1, y_2, \cdots y_n) < P(x=1 \mid y_1, y_2, \cdots y_n)$$

The decision rule is that $x=0$ if and only if at least one of y_j is equal to 0.