# EE 114 - Coding Assignment I

This exercise will focus on understanding the role of probability in communication channels. Alice wants to send Bob a message $M$ composed of N=100 bits. This message is a length N sequence of 0's and 1's. The challenge is that the communication channel between Alice and Bob can introduce errors by flipping 0 to 1 or vice versa. We are given that the channel flip probability is p=0.1. In other words,

$$\text{if Alice transmits } x \in \{0,1\}, \text{ Bob receives } \begin{cases} x \text{ with probability } 1-p \\ 1-x \text{ with probability } p \end{cases}.$$

**Step 1 encoding:** To safely transfer the message, Alice applies encoding by repeating the bits $R$ times and transmits the encoded message $M_{enc}$. For instance, for $N = 3$ and $R = 3$, an example message is

$$M = 0\ 1\ 0 \rightarrow \text{encoding} \rightarrow M_{enc} = 000\ 111\ 000$$

**Step 2 transmission:** After transmission, Bob receives a message $M_{rec}$ which is a corrupted version of $M_{enc}$ due to channel flips. For instance, we might have

$$M_{enc} = 000\ 111\ 000 \rightarrow \text{channel} \rightarrow M_{rec} = 001\ 101\ 110$$

**Step 3 decoding:** Now, Bob needs to decode $M_{rec}$ to find $\hat{M}$.

**His goal is ensuring $\hat{M} = M$**

Let us use "majority rule" for decoding: Bob assigns each R-bit chunk to 0 or 1 by counting which one is more. For instance, we assign $001 \rightarrow 0$ whereas $111 \rightarrow 1$. For the example above, this means

$$M_{rec} = 001\ 101\ 110 \rightarrow \text{decoding} \rightarrow \hat{M} = 0\ 1\ 1$$

Unfortunately, for this example $\hat{M} \neq M$ at the end of the **encoding, transmission, and decoding process**. This HW aims to understand the probability of successful communication as a function of $R$, $N$ and $p$. Our goal is understanding what makes the communication process safe.

You are expected to answer following questions by doing numerical simulations. You should turn in your report explaining how you arrived at your answers and also turn in your code.

1. **(3 pts)** Write a code for generating the procedure described above. Your code should allow for any $p, N,$ and $R$ values. You can assume $R$ is an odd integer. Your code should be able to carry out one communication experiment and output whether all bits of Alice was correctly transmitted to Bob.
2. **(1 pts)** Set $N = 10, R = 9, p = 0.1$ and run your experiment for 1000 times. Record the resulting probability of successful decoding $p_{success} = P(M = \hat{M})$.
3. **(2 pts)** Now try $N = 30, 100, 300, 1000$. Record the resulting probability of successful decoding $p_{success}$. Plot $p_{success}$ as a function of $N$ and comment on how it changes.
4. **(1 pts)** Set $p = 0.2$ and $N = 100$. Suppose we wish to ensure $p_{success} \geq 0.9$. What is the minimum $R$ we need to choose? You should find your $R$ choice by running sufficiently many communication experiments.
5. **(2 pts)** How about for ensuring $p_{success} \geq 0.99$ and $p_{success} \geq 0.999$?
6. **(1 pts)** How does $p_{success}$ change as a function of $N$, $R$, $p$? You should explain your answer (why it is increasing or decreasing).