Project Member Application

SPARC

2025-26

Mathematics Club

Centre for Innovation
IIT Madras

v2.3

Instructions

General Instructions

- Mention the following details at the start of your application.

Name:	Insti Nickname:
Roll No:	CGPA:
Phone (WA):	Email:

- Join the Aspiring Team Members WhatsApp group for further updates: Whatsapp Group
- The recommended font is a standard font size 11-13.
- The applications have to be submitted in PDF format, named as:
  `<First_name>_<Roll_Number>_Mathematics_Club_PM_SPARC.pdf`

  For example, `Vignesh_EE23B087_Mathematics_Club_PM_SPARC.pdf`.
- You can upload the finished applications in this Google Form
- You may submit the completed application on or before **11:59 PM, 30/05/2025.**

Project Specific Instructions:

- It is fine even if you don't answer all the questions.
- Focus on the compulsory questions before attempting the bonus questions.
- The section Section 1: Probability is optional and non-evaluative. However, we encourage you to attempt the section to get your basics right.
- If you choose to attempt the bonus section, Section 6: Let's Simulate LDPC!, we encourage you to attempt all the questions in this section and not just a few of them.
- If you have any queries, you can reach out to the Project Leads anytime you want:
  - Achintya Raghavan (EE23B189): 9606852240
  - Vignesh Natarajkumar (EE23B087): 6364229598

Contents

HR Questions

Yep. We know its boring, which is why we wanted to get it out of the way immediately.

Questions

0. Tell us a bit about yourself. Assume you are a master of flexing and proceed.
1. Why do you want to join this project? What skills or qualities do you possess that make you suitable for the work involved?
2. Mention all PoRs/activities you are planning to take this year. Weekly, how much time do you think you will be able to commit to this project? How much time will you commit to other PoRs and academics?
3. What would keep you motivated throughout the period of two semesters? How would you ensure that you are consistent in contributing to the project throughout the tenure?
4. **(10x Weightage)** What does the project name stand for?

Good luck in the technical section!

ALL YOU NEED IS MOTIVATION
FALSE: YOU NEED FEAR AND AN APPROACHING DEADLINE
memegenerator.net

Figure 1: Dwight speaking facts

§1 (Optional) Probability

Hello there, my old friend

What is probability really? Well, it turns out that mathematically, probability is just a set of axioms! We won't go into the details now (we have more interesting stuff for you) however, these axioms are all you need to build everything you know about probability.

§1.1 The Randomness in Variables

We start off with something fundamental to the study of information theory: **Random Variables**. Random Variables, simply put, are just objects which output real numbers whilst following a certain probability constraint. This constraint is usually expressed in the form of a distribution, i.e. a function which returns the probability of every number the RV can take occurring.
Let us solve a few problems to make this idea clear:

Questions

- Consider a coin which is tossed 10 times. Let the number of heads obtained be termed as the random variable X. What is $\mathbf{P}(X = 5)$ i.e. the probability that the 5 heads are obtained?
  Find the distribution of the RV X i.e. probability of $X = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6$.
- Suppose an RV has its distribution as $\mathbf{P}(X = n) = \dfrac{c}{10^n}$ where $n \in \mathbb{N}$ and is 0 everywhere else. Find the value of c.
  (Hint: Remember the fundamental rules of probability.)
- Suppose that two balanced dice are rolled, and let X denote the absolute value of the difference between the two numbers that appear. Determine and sketch the distribution of X.

§1.2 Let's make it continuous

In the above section, we looked at cases where the distribution only takes on certain **discrete** values. Such distribution functions are called **Probability Mass Functions**. But real life so rarely offers us such distributions. Consider voltages in a circuit. Naturally, they make take on any real value. Thus, we define the continuous analogue of PMFs called the **Probability Distribution Functions** or PDFs as follows:

$$f_X(x)\ dx := \mathbf{P}\left(x - \frac{dx}{2} \leq X \leq x + \frac{dx}{2}\right)$$

Intuitively, what this says is that the area under the graph of the PDF for a range is the probability that the random variable can take on that range.
It naturally follows that:

$$\mathbf{P}(a \leq X \leq b) = \int_a^b f_X(x)\ dx$$

Can you prove this?

Questions

- Consider an RV X which has the following properties:
  - It takes on values only between a and b.
  - $\mathbf{P}(X \leq c)$ is proportional to $c - a \; \forall \; c \in [a, b]$

  Find the PDF of X.
- Do you think that RVs have a unique PDF i.e. that there is only one $f_X(x)$ such that the above properties are satisfied? Can you provide a counterexample if that isn't the case?

§1.3 Let's Add S**t

Let us consider two RVs, X and Y. One might wonder what happens when we add them i.e. look at the output when the two values X and Y return are added. This new RV $Z := X + Y$ will have its own distribution which somehow depends on the distributions of X and Y.

Questions

- Can you develop a method to calculate the PMF/PDF of Z, given the PMFs/PDFs of X and Y given that X and Y are independent of each other?
- Can you do this in general for $Z := aX + bY$?

§1.4 Da Important Distributions

There are certain types of PDFs which come up naturally in many applications, namely, the uniform and gaussian distributions. This section aims to familiarize you with aforementioned distributions.

- $X \sim \text{Uniform}(a, b) \implies$ the PDF of X is constant in the range $[a, b]$ and is zero everywhere else (Look familiar to a previous problem?).
- $X \sim N(\mu, \sigma^2) \implies f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$ and is called a Normal/Gaussian distribution.

No problems for you here. Read on for continued torture.

§1.5 What did you expect?

The notion of weighted average is probably already familiar to you. When it comes to Random Variables, one might ask what value is the RV *expected* to take i.e. can we find a value around which the other values are centered so to speak? Thus we define **expectation**:

$$E[X] := \sum_{\forall \; x} x \; \mathbf{P}(X = x) \qquad \text{or} \qquad E[X] := \int_{-\infty}^{\infty} x f_X(x) \; dx$$

depending on the type of distribution.
Similarly we define the **variance** σ^2 as:

$$\sigma^2[X] := E[(x - E[X])^2]$$

Questions

- Find the expectation and variance of the Normal and Uniform distributions.
- Prove that $E[aX + bY + c] = aE[X] + bE[Y] + c$. (Hint: Find $E[X + Y]$ and $E[aX + b]$ separately.)
- In real applications, a transmitted signal is usually affected by Gaussian/Uniform Noise. How would you model this effect using random variables?

§1.6 A few conditions?

So far, we have seen PMFs of single random variables. However, this notion can be extended to a pair of RVs to what we call a joint PMF. Here, the probability of each value that an RV takes is somehow correlated with the value of the other RV. Thus the PMF becomes something of the form $\mathbf{P}(X = x, Y = y)$ for the RVs X, Y. Obviously, when X and Y have no relation i.e. they are independent, we have

$$\mathbf{P}(X = x, Y = y) = \mathbf{P}(X = x)\mathbf{P}(Y = y)$$

We can then ask ourselves if it is possible to express any two RVs in a form like this. Thus, we define the conditional probability $\mathbf{P}(Y = y \mid X = x)$ as:

$$\mathbf{P}(Y = y, X = x) = P(Y = y \mid X = x)P(X = x)$$

Intuitively, the conditional probability of X is how its probability function looks given that we know the value for Y i.e we know some information about X.
To make this concept clear, let us solve a few problems:

Questions

- Consider the RVs (X, Y) with the following distribution:
  $\mathbf{P}(X = 1, Y = 1) = 0.5$, $\mathbf{P}(X = 1, Y = 2) = 0.2$, $\mathbf{P}(X = 2, Y = 1) = 0.2$, $\mathbf{P}(X = 2, Y = 3) = 0.05$, $\mathbf{P}(X = 3, Y = 2) = 0.05$

  Find $\mathbf{P}(X = x)$, $\mathbf{P}(Y = y)$, $\mathbf{P}(X = x \mid Y = 2)$.
  (Hint: Think fundamentals.)
- Find expressions for $\mathbf{P}(X = x)$ and $\mathbf{P}(Y = y)$ given $\mathbf{P}(X = x, Y = y)$. Is the other way round

possible?

- Can you relate $\mathbf{P}(Y = y|X = x)$ and $P(Y = y)$ if X and Y are independent?
- Can you set up a matrix to represent $\mathbf{P}(Y = y\ |X = x)$. Given X, how would you recover information about Y from this matrix?

Right. Too much math? Read on for some interesting bit manipulation.

§2 Bits...And More Bits

§2.1 Irritating Errors and Where to Find Them

It is highly convenient for us to represent information using bits - streams of 0s and 1s. When we wish to transmit said information, however, we would like to use a symbolic representation for these bits that satisfies 2 criteria:

- The symbols can be represented by electrical pulses, so pulses like +1 and -1 should work.
- A single symbol should represent as many bits as possible, so that we can send as few symbols as possible.

With these two motivating factors in mind, read up on Binary Phase-Shift Keying (BPSK) and higher-order Phase-Shift Keying systems.

There will inevitably be certain disturbances to the symbols we transmit owing to various reasons. We quantify these disturbances as **noise**, which we model mathematically as an additive term w that corrupts each transmitted symbol x. When this noise is assumed to be Gaussian—meaning its instantaneous values follow the familiar bell-curve distribution—and to affect all frequencies equally, we call it Additive White Gaussian Noise (AWGN). Concretely, if x is our transmitted value, the receiver observes

$$y = x + w,$$

where $w \sim \mathcal{N}(0, N_0/2)$ is a random variable with zero mean and variance $N_0/2$.

To understand noise better, we use a metric called the Power Spectral Density (PSD) of the noise. Imagine you have a noisy recording and you look at its power at every frequency - that gives us an idea of the Power Spectral Density.

Questions

1. Plot out the PSD for AWGN noise and explain why the nature of the curve makes sense. You may use any standard Python functions for the same.

To measure how strong our desired signal is relative to the noise, we use the Signal-to-Noise Ratio (SNR) metric, commonly expressed as

$$\frac{E_b}{N_0},$$

where E_b denotes the average energy used to transmit one bit and N_0 is the noise's PSD, since it tells us the energy of the noise in a frequency range. A larger E_b/N_0 means the signal stands out more clearly above the noise floor, and hence we can transmit our bits more reliably.
The linear SNR is also equivalently written as:

$$\text{SNR} = \frac{P_{\text{sig}}}{P_{\text{noise}}}.$$

To convert this ratio into decibels, we take ten times the base-10 logarithm:

$$\mathrm{SNR}_{\mathrm{dB}} \;=\; 10\,\log_{10}\!\left(\frac{P_{\mathrm{sig}}}{P_{\mathrm{noise}}}\right).$$

In reality, an SNR of 10 dB is very good, and is a convenient standard used in simulations.

We capture the performance of a transmission by its **Bit Error Rate (BER)**, defined as fraction of bits decoded incorrectly:

$$\mathrm{BER} \;=\; \frac{\text{Number of bits decoded incorrectly}}{\text{Total number of bits sent}}.$$

Note that BER is calculated over the number of bits, not symbols, that are transmitted.

Once the received message vector is available to us, we require methods of **demodulating** the corrupted symbols back to bits or sequences of bits. One simple method is described below, and is a good starting point for you to think about more complex methods of message demodulation.

Simple Sign Demodulation:

$$\hat{x}[i] = \begin{cases} +1, & y[i] > 0, \\ -1, & y[i] \leq 0. \end{cases}$$

Map the decoded symbols back to bits to obtain a predicted bit vector from y.
No questions here as well. Move on, aspiring PM!

§3 Let's study.... information?

Enough of bits for now. We shall return to some math fundamentals regarding information theory.

§3.1 Its quite random!

One of the fundamental concepts in information theory is the entropy of an RV:

$$\mathbf{H}(X) := -\sum_{\forall\ x} p_X(x) \log_2(p_X(x))$$

where $p_X(x) := \mathbf{P}(X = x)$ is just the PMF and the unit of entropy being `bits`.
(For all further discussions, it will be implicitly assumed that log is to the base 2.)

What's the intuition you ask?

Questions

2. Prove that entropy is always non-negative.

3. Consider discrete RV $X \sim Uniform(0, 4)$. Find the entropy. Do it for $X \sim Uniform(0, 2^n)$. What do you notice? How does the value change with the amount of "randomness" in the RV? Can you see that entropy somehow represents the amount of disorder?

4. Can you provide some sort of intuitive understanding behind the value of entropy? Why is it that the unit of entropy is bits?

5. **Simulation Task:** Look up Poisson Distributions. Plot the PDF as well as the entropy on Desmos and vary the parameter λ. Can you relate the entropy to the notion of "randomness"?

Remember when we extended distributions to multiple RVs? It's time to do it again. The joint entropy is defined as:

$$\mathbf{H}(X, Y) = -\sum_{\forall\ (x,y)} p_{XY}(x, y) \log_2\left(p_{XY}(x, y)\right)$$

Let's look at some properties:

Questions

6. Extending the above to conditional probability, what would $\mathbf{H}(Y|X = x)$ be?
(Hint: Don't over-complicate. Think of $Y|X = x$ as a new RV.)

7. Define $\mathbf{H}(Y|X) := \sum_{\forall x} p_X(x)\mathbf{H}(Y|X = x)$, a sort of average entropy over the RV X.
Prove that:
$$\mathbf{H}(X, Y) = \mathbf{H}(X) + \mathbf{H}(Y|X)$$
What this says is that the randomness in two RVs is the addition of the randomness in the first and

the randomness in the second assuming we know the first!

§3.2 Mutual ~~interaction~~ information

As we will see in the next section, transmission through a medium can be modeled as function operating on an RV. Thus, to study properties of the medium, we must somehow quantify it's ability to transmit information. Thus we define a quantity mutual information for two RVs X and Y:

$$I(X;Y) := \sum_{\forall\ (x,y)} p_{XY}(x,y) \log_2 \left(\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)} \right)$$

How is this even remotely related to the ability to transmit information you ask? Let's find out.

Questions

8. What is the relation between $I(X;Y)$ and $I(Y;X)$
9. Prove that $I(X;Y) = H(Y) - H(Y|X)$.
   What happens to $I(X;Y)$ when X and Y are completely independent? What if Y was completely determined by X (eg. $Y = aX$)?
10. Suppose X was your input to your medium and Y was what your recovered after X passed through. What do you think $I(X;Y)$ would represent. What does a high/low value of $I(X;Y)$ mean intuitively?

§4 Channels, Capacities and Codes

Now that we know a bit of information theory, let's move on to the concepts you will actually be dealing with.

§4.1 Coding again??

No. Not that sort of coding! Mathematically, a code C s just a mapping from the range of outputs of an RV to a string of symbols from an alphabet D. Eg. $X = 1 \implies C(X) = 01$, $X = 2 \implies C(X) = 10$ etc. (binary mapping with alphabet $0, 1$). We define the expected length $L(C)$ of a code C as:

$$L(C) := \sum_{\forall x} p_X(x)\ l(x)$$

where $l(x)$ is the length of $C(X)$ for $X = x$.

Questions

11. Consider the distribution for the RV X: $p_X(1) = 0.6$, $p_X(2) = 0.25$, $p_X(3) = 0.15$. Find its entropy. Now try and find a code which produces the least expected length (obviously we would try and mminimise the amount of redundant information in practical applications so as to be as efficient as possible).
Do you notice anything?

12. Repeat the above for $p_X(1) = 0.5$, $p_X(2) = 0.25$, $p_X(3) = 0.25$. What about now? What can you postulate something about entropy and expected length? Can you estimate how "good" a code is based on the entropy?

13. Look up Morse Code. Assuming a binary representation, find both the entropy as well as the expected length. Is it a good code? You may look up the statistical distribution of alphabets in typical text online. (You are allowed to use code for this arguably tedious task :))

§4.2 Channeling your inner methie

At last we come to the focal point of the project. In information theory, channels are modeled basically as black boxes which take in a random variable and Spit out another random variable. Actually, a general channel takes in a string/block of alphabets of a particular code and returns a sequence of RV outputs. However, this is beyond the scope of this application (although you will see some aspects of this in subsequent sections).

We define the the capacity of a channel as:

$$\max_{\text{all } X} I(X;\ Y)$$

where X is your input random variable, Y is your output random variable and I is the mutual information. This makes sense intuitively. Suppose that $H(Y|X) = 0$ i.e. Y is completely determined by X. Then $I(X;Y) = H(X)$ which is just the amount of information contained in input random variable itself (i.e. all available information can be transmitted). Similarly $H(Y|X) = H(Y)$ implies that $I(X;Y) = 0$ i.e. nothing can be transmitted (since Y is now just completely random!).

Questions

14. Consider a channel which sees only a binary input. It flips the bits with a probability p. What is the capacity of the channel?
    Once we have an output, we use some sort of guessing function to guess the input. What would be your guessing function in this case?

In the previous section, we tried to minimize the code such that it contained approximately the same information as the RV. However, now we find ourselves doing the opposite. In order to reduce the loss in information, we will need to add redundant data.
If the number of symbols in our alphabet D is M and the expected length of our code is n, then we define the rate of the code $(M,\ n)$ as:

$$R := \frac{\log_2 M}{n} \quad \text{bits per transmission}$$

What does this expression attempt to convey? Since M is the number of symbols, $log_2 M$ is maximum number of bits required to represent it (Recall Section 3.1). Thus the amount of information each bit in the code carries is R, which agrees with our intuitive understanding of rate.

Questions

15. For the codes in the above section, find the rate. Do you think these are good codes to send through our binary scrambling channel?

16. Read up on Hamming Codes. Derive an expression for the rate of Hamming codes.

17. Simulate a binary scrambling channel. Pass a hamming code encoded message into it and try and guess the output. Observe what happens to the error when the number of bits are increased. Is hamming code a good code to achieve capacity of a channel?

Consider a rate R. We call R achievable if we can construct a sequence of codes $C_n : (2^{nR},\ n)$ such that the $\max_{allX} \mathbf{P}(g(Y) \neq x\ |X = x) = 0$ as $n \to \infty$ i.e. the input becomes completely determined by the output as the code length becomes very large. Achievable implies that the channel supports the rate of a select subset of codes.

Questions

18. Consider a channel such that every alternate input bit has a 0.5 probability for being flipped. Can you prove that a rate of 0.5 is achievable.

19. Consider a channel which takes in a binary string x and outputs another binary string such that the output $y = x \pm 6$. Give an achievable rate for this channel and prove it.

§5 Queues and Erasure Queue Channels

§5.1 Waiting in Line... :(

The M/M/1 queue is a very simple model for a line (or queue) where things arrive randomly and are served one at a time. In this case, the items arriving and leaving are symbols in a channel. The longer a symbol waits in the queue, the more the channel can affect it.

The M/M/1 queue describes 3 aspects of waiting in the queue:

- The first "M" stands for **Markovian arrivals**. This means that the time between symbol arrivals is random and follows an exponential distribution. On average, symbols arrive at a rate λ. The key feature is that short gaps happen more often than long ones, and the probability of a gap of length t seconds is

  $$P(\text{gap is } t) = \lambda e^{-\lambda t}, \quad t \geq 0,$$

  This kind of arrival process is called a **Poisson process** (You have already seen the Poisson distribution for a Random Variable.)

- The second "M" means the **service times** (how long it takes to transmit or process a symbol) are also random and exponential. The average service rate is μ.

- The "1" means there is only **one server** — only one symbol can be processed and sent to the receiver at a time.

Imagine each symbol arrives at a transmitter that can send only one symbol at a time. If a symbol arrives while the previous symbol is still being sent, it must **wait in the queue**. The time a symbol spends in the channel is called its **sojourn time**, written W. It includes both the time waiting in line and the time being served. The average sojourn time is

$$\mathbb{E}[W] = \frac{1}{\mu - \lambda}.$$

Questions

20. Given this model of a queue, can you think of a situation, or a combination of values of μ and λ, that would lead to an unstable, or an infinitely growing, queue?

Interestingly, the waiting time W also follows an exponential distribution (when the system is in steady state). The probability of a wait time of length t is

$$f_W(t) = (\mu - \lambda)e^{-(\mu-\lambda)t}, \quad t \geq 0, \ \mu > \lambda$$

§5.2 Erasure

Now, let us look at Erasure Channel Models. In such a channel, each transmitted bit either arrives correctly or is "erased" (lost), and the receiver knows exactly which bits were lost. In the simplest **Memoryless** Erasure

Channel, each bit is independently erased with probability ε.

The more realistic **erasure-queue channel (EQC)** captures correlated erasures: bits that wait longer in the queue face higher erasure probability $p(W) = 1 - e^{-0.1W}$, where W is the sojourn time in an M/M/1 queue with arrival rate λ and service rate μ.

Below are two simple **Simulation Tasks** to build your intuition:

Questions

21. Generate a random bit vector **b** of length 100 (each entry 0 or 1 with equal probability). Choose an erasure probability ε (e.g. 0.2), then simulate the memoryless erasure channel by independently "dropping" each bit with probability ε. At the receiver, mark erased positions (for example with a "?") and report the fraction of erased bits. Plot the original and received vectors (using e.g. 0,1 and "?"). Finally, suggest two simple coding or interleaving strategies that could help recover from such random erasures.

22. Now simulate the EQC described above.
    a) Set the arrival rate $\lambda = 0.8$ and the service rate $\mu = 1.0$.
    b) For all bits, sample a random number from an exponential distribution with mean $1/\lambda$. This tells you how long after the previous bit this one arrives. Plot the distribution of arrival times.
    c) For all bits, sample a random number from an exponential distribution with mean $1/\mu$. This tells you how long the server will take to transmit this bit. Plot the distribution of service times.
    d) For every bit, compute the sojourn time W and track when this bit will be transmitted. Set a reference time and compute sojourn times for each bit.
    e) Use the formula
    $$p(W) = 1 - e^{-0.1\,W}$$
    to compute the erasure probability based on its sojourn time. With this probability, randomly decide whether the bit is erased.
    f) After all 100 bits have passed through the system, mark erased positions (for example with a "?") and report the fraction of erased bits. Plot the original and received vectors (using e.g. 0,1 and "?"). Plot the histogram of the sojourn times as well.

§6 (Bonus) Let's Simulate LDPC!

Armed with the necessary basics of Probability, Information Theory, Forward Error Correction and the basics of simulating Information Transmission, let us work through a Forward Error Correction problem - **Transmission of bits through an AWGN channel** using Low Density Parity Check (LDPC) codes for **Forward Error Correction (FEC)**.

Consider a BPSK-modulated system over a real AWGN channel. Bits are mapped to symbols ($0 \to +1$, $1 \to -1$), corrupted by Gaussian noise, and then decoded. Your task is to demonstrate the improvement in Bit-Error Rate (BER) due to LDPC coding.

Before starting the simulation, understand the different parameters that must be specified in order to define and use a particular LDPC code. Some keywords include the codeword length and rate of coding, which are ideas you have been introduced to in previous sections. This video would be a good starting point for this purpose.

Another important consideration - we have used the SNR to highlight the energy of the bit vector as compared to the energy of the noise in earlier transmissions. But what happens when we use symbols instead of bits, and what happens when multiple bits map onto the same symbol (such as in M-PSK mapping)? What happens when we add additional redundant bits (as in the case of Error-Correcting Codes)? Highlight how you would quantify noise levels in such cases (Hint - it is merely a scaling factor that must be accounted for in such cases).

Questions

- Plot out the capacity of the channel model for SNR values in the range of 2 dB to 10 dB. Plot it as a BER vs SNR curve. This is the theoretical best performance that can be achieved!
- Perform a simulation of the channel by taking 10^5 independent bits, encoding them via LDPC, corrupting them by adding Gaussian Noise (i.e. $y_n = x_n + N$ where y_n is the output, x_n is the input and N is a random variable following a normal distribution) and decoding them back. You may use simple sign analysis as the guessing function for the output. Run it for the above range of SNR values using a rate 0.5 LDPC code, and a codeword length of 1000. Read up ways of using this code in Python (you could try writing the encoding and decoding functions by yourself in Python!).
- The coding gain is the improvement in E_b/N_0 (in dB) needed to achieve the same BER when using LDPC coding. For example, find the BER $= 10^{-3}$ point on both curves and compute:

  $$\text{Coding Gain (dB)} = E_b/N_0|_{\text{uncoded}} - E_b/N_0|_{\text{LDPC}}.$$

  Report this gain and briefly discuss how close your LDPC scheme gets to the Shannon capacity empirically.

§7 Resources

Here are a set of resources (apart from the ones linked throughout the app) for you to get a good grasp on the concepts discussed:

1. DeGroot, M. H., & Schervish, M. J. (2012). *Probability and Statistics* (4th ed.). Pearson Education.
2. NVIDIA Corporation. (n.d.). *Sionna: A Library for GPU-Accelerated Link-Level Simulations*. Retrieved from https://github.com/NVIDIA/sionna
3. Mandalapu, J., Jagannathan, K., Chatterjee, A., & Thangaraj, A. (2023). *Capacity Achieving Codes for an Erasure Queue-Channel*. arXiv preprint arXiv:2305.04155. Department of Electrical Engineering, IIT Madras.
4. Chatterjee, A., Seo, D., & Varshney, L. R. (2017). Capacity of systems with queue-length dependent service quality. *IEEE Transactions on Information Theory*, 63(6), 3950–3963.
5. *An Introduction to LDPC Codes*. Retrieved from https://cmrr-star.ucsd.edu/static/presentations/ldpc_tutorial.pdf
6. Thangaraj, A. (n.d.). *NOC: LDPC and Polar Codes in 5G Standard*. NPTEL, IIT Madras. Retrieved from https://archive.nptel.ac.in/courses/117/106/108106137/
7. MacKay, D. J. C. (n.d.). *Error-Correcting Codes Resources*. Inference Group, University of Cambridge. Retrieved from http://www.inference.org.uk/mackay/CodesFiles.html
8. Weissman, T. (2025). *EE 276: Information Theory (Winter Quarter 2024–25)*. Stanford University. Retrieved from https://web.stanford.edu/class/ee276/outline.html