|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Information Theory and Coding Techniques** | **Semester:** | **V** |
| **Date of Performance:** | **23 / 07 / 2024** | **Batch No:** | **B - 1** |
| **Faculty Name:** | **Prof. Makarand Kulkarni** | **Roll No:** | **16014022050** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | **\_\_\_ / 25** |

**Experiment No.: 1**

**Title:** **To implement Channel Capacity Analysis with different communication channels and to plot variation in Entropy vs probability, using MATLAB/Python/C++**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| * To evaluate Channel Capacity of various communication channels (known bandwidth). * To calculate the Entropy and Maximum Channel Capacity for probabilistic statistics of messages is given. * To Plot Entropy vs probability curve. |

|  |
| --- |
| **COs to be achieved:** |
| **CO1:** Apply concepts of Information Theory. |

|  |
| --- |
| **Theory:** |
| The information theory is used to quantify the amount of information contained in a message signal, as well as the capacity of a channel or communication medium for sending information. It is used by engineers to design and analyze the communication systems—telephone networks, modems, radio communication, etc.  Channel Capacity (measured in bps) is defined as the rate at which information can be transmitted over a communication channel reliably. Channel capacity, as a fundamental concept in information theory, plays a significant role in the design and optimization of modern communication systems. From system design to network bandwidth management and mobile and wireless communication, understanding and applying the concept of channel capacity is key to achieving efficient and reliable communication. Shannon’s Theorem provides the formula for calculating channel capacity in noisy communication channels:  where, C represents the channel capacity (in bits per second), B is the bandwidth of the channel (in Hertz), SNR is the Signal-to-Noise Ratio (numerical value).  Entropy (Average information per message interval): In Information Theory, entropy quantifies the uncertainty or randomness of information content. Information entropy is a mathematical definition used to describe the average uncertainty of all the information that an information source can generate. The higher the entropy of information, the greater the amount of information it contains, and the stronger its ability to reduce uncertainty.  where, H represents the entropy (in bits per message), Pk is probability of occurrence of message mk.  Max. Average information per message interval = **Hmax =**  Max. Channel Capacity = Cmax  Cmax = (Max. Average information per message interval) X (Rate of message in message per seconds)  **Max. Channel Capacity = Cmax = Hmax X *r*** |

|  |
| --- |
| **Stepwise-Procedure:** |
| **Activity 1:** Mathematical Analysis:   * Step 1: To evaluate Channel Capacity of various communication channels (known bandwidths). * Step 2: To calculate the Entropy for probabilistic statistics of messages is given. * Step 3: To calculate the Maximum Channel Capacity for probabilistic statistics of messages is given. * Step 4: To Plot Entropy vs probability curve.   **Activity 2:** Analysis using the MATLAB/Python/C++:   * Step 1: Write a code to calculate Channel Capacity of various communication channels (known bandwidths). * Step 2: Write a code to calculate the Entropy for probabilistic statistics of messages is given. * Step 3: Write a code to calculate the Maximum Channel Capacity for probabilistic statistics of messages is given. * Step 4: Write a code to Plot Entropy vs probability curve.   Attach the mathematical calculation (handwritten) done and the MATLAB/Python/C++: code with its output.  **Code:**  import math  import matplotlib.pyplot as plt  def convert\_dB\_to\_numerical(snr\_dB):      return math.pow(10, snr\_dB / 10)  def calculate\_individual\_entropies(probabilities):      return [pk \* math.log2(1/pk) if pk > 0 else 0 for pk in probabilities]  def calculate\_total\_entropy(probabilities):      return sum(pk \* math.log2(1/pk) for pk in probabilities if pk > 0)  def calculate\_max\_entropy(M):      return math.log2(M)  def calculate\_max\_channel\_capacity\_by\_snr(snr\_linear):      return 1.44 \* math.log2(snr\_linear)  def calculate\_max\_channel\_capacity\_by\_rate(r, Hmax):      return r \* Hmax  def calculate\_channel\_capacity(B, snr\_dB):      snr\_linear = convert\_dB\_to\_numerical(snr\_dB)      return B \* math.log2(1 + snr\_linear)  def plot\_entropy(probabilities, entropies):      plt.figure(figsize=(10, 6))      plt.plot(probabilities, entropies, marker='o', linestyle='-', color='b')      plt.xlabel('Probability (pk)')      plt.ylabel('Entropy (H)')      plt.title('Entropy vs Probability')      plt.grid(True)      plt.show()  def main():      M = int(input("\nEnter the number of messages: "))        print("\n")        probabilities = []      for message in range(M):          pk = input(f"Enter the probability of message {message + 1}: ").strip()          probabilities.append(float(pk))        individual\_entropies = calculate\_individual\_entropies(probabilities)      total\_entropy = calculate\_total\_entropy(probabilities)      Hmax = calculate\_max\_entropy(M)        r\_input = input("\nEnter the rate of message, r (or 'na' if not given): ").strip()      if r\_input.lower() == 'na':          r = None      else:          r = float(r\_input)          rate\_type = input("Is this the Nyquist rate? (Y/N): ").strip().upper()          if rate\_type == 'Y':              r \*= 2        B\_input = input("\nEnter the bandwidth of message, B in Hertz (Hz) (or 'na' if not given): ").strip()      B = None if B\_input.lower() == 'na' else float(B\_input)        snr\_dB\_input = input("\nEnter the signal to noise ratio, S/N in decibels (dB) (or 'na' if not given): ").strip()      snr\_dB = None if snr\_dB\_input.lower() == 'na' else float(snr\_dB\_input)        Cmax\_by\_snr = Cmax\_by\_rate = 'na'      if snr\_dB is not None:          snr\_linear = convert\_dB\_to\_numerical(snr\_dB)          C = calculate\_channel\_capacity(B, snr\_dB) if B is not None else 'na'          Cmax\_by\_snr = calculate\_max\_channel\_capacity\_by\_snr(snr\_linear)      else:          C = 'na'        if r is not None:          Cmax\_by\_rate = calculate\_max\_channel\_capacity\_by\_rate(r, Hmax)        print("\nOptions to display:")      print("1. Entropy H")      print("2. Maximum entropy, Hmax")      print("3. Channel capacity, C")      print("4. Maximum channel capacity, Cmax")      print("5. Display entropy graph")      chosen\_options = input("\nEnter the numbers of the values you want to print, separated by commas: ").strip().split(',')      chosen\_options = [int(option) for option in chosen\_options]        print("\n-------------------------------------------------------------------------------------------")        if 1 in chosen\_options:          print(f"nEntropy H = {round(total\_entropy, 3)} bits/message")        if 2 in chosen\_options:          print(f"\nMaximum entropy, Hmax = {round(Hmax, 3)} bits/message")        if 3 in chosen\_options:          if C == 'na':              print("\nChannel capacity, C = insufficient data")          else:              print(f"\nChannel capacity, C = {round(C, 3)} bps")        if 4 in chosen\_options:          if Cmax\_by\_snr == 'na' and Cmax\_by\_rate == 'na':              print("\nMaximum channel capacity, Cmax = insufficient data")          else:              if Cmax\_by\_snr != 'na':                  print(f"\nMaximum channel capacity by SNR, Cmax by SNR = {round(Cmax\_by\_snr, 3)} bps")              if Cmax\_by\_rate != 'na':                  print(f"\nMaximum channel capacity by rate, Cmax by rate = {round(Cmax\_by\_rate, 3)} bps")        if 5 in chosen\_options:          plot\_entropy(probabilities, individual\_entropies)  if \_\_name\_\_ == "\_\_main\_\_":      main()  **Question 1:**  **A discrete memoryless channel has 6 messages with probabilities 0.3, 0.12, 0.12, 0.16, 0.15, 0.15 respectively attached to M1, M2, M3, M4, M5, M6.**   1. **Plot entropy vs probability graph.** 2. **Determine Hmax if message rate follows Nyquist criteria for band limited signal at 4Khz.** 3. **Determine maximum channel capacity.**       **Question 2:**  **A 2kHz channel has a signal to noise ratio 24dB. Calculate channel in bps.** |

|  |
| --- |
| **Observations:** |
| Note down the results obtained using mathematical analysis:   * Channel Capacity of various communication channels * Entropy * Maximum Channel Capacity * Plot of Entropy vs probability   Validate your results by values obtained from the program. |

|  |
| --- |
| **Post Lab Subjective / Objective type Questions:** |
| 1. **Which wired communication channel offers highest channel capacity? Write its applications.**   Channel Capacity:  Optical fiber communication channels offer the highest channel capacity among wired communication channels. This is due to their ability to transmit large amounts of data at very high speeds with minimal loss and interference. The channel capacity can reach up to terabits per second (Tbps).  Applications:   * Internet Backbone: Optical fibers form the backbone of the global internet infrastructure, enabling high-speed data transmission across continents. * Telecommunications: Used in long-distance and local telephone networks for transmitting voice and data. * Cable Television: Delivering high-definition TV channels and on-demand content to households. * Data Centers: Facilitating fast data transfer between servers and storage systems. * Medical Imaging: Used in endoscopy and other imaging techniques. * Military and Aerospace: Secure, high-speed communication for defense and space applications.  1. **Explain ‘Nyquist Rate’ used for generation of messages at the transmitter.**   The Nyquist Rate is a critical concept in digital signal processing and communication theory. It is defined as twice the maximum frequency present in the signal. For a signal with a maximum frequency fmax, and the Nyquist rate is 2fmax.  Sampling Theorem: According to the Nyquist-Shannon sampling theorem, to accurately reconstruct a continuous-time signal from its samples, it must be sampled at a rate at least equal to the Nyquist Rate. This ensures that there is no aliasing, which is the distortion that occurs when high-frequency components of the signal are misinterpreted as lower frequencies.  Generation of Messages: At the transmitter, the signal is sampled at the Nyquist Rate to convert the analog message into a digital form. This ensures that the original message can be perfectly reconstructed at the receiver without any loss of information.  In summary, the Nyquist Rate is essential for ensuring accurate signal sampling and reconstruction in digital communication systems. |

|  |
| --- |
| **Conclusion:** |
| In conclusion, the experiment successfully implemented Channel Capacity Analysis across various communication channels and effectively demonstrated the variation in Entropy vs. probability through detailed plots using Python. The results indicate that channel capacity increases with optimized probability distributions, highlighting the importance of efficient communication channel design. |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |