CT216: End of the Semester Project

Sunday 24th March, 2024

Contents

Chapter 1

Introduction

In the end-of-semester project, you will develop one of the following three topics.

1. LDPC decoding for 5G NR (New Radio)

2. Polar Codes

3. Convolution coding, its Viterbi decoding and a mathematical performance analysis of the decoder

Each of these topics involves the following components. You will be evaluated on the basis of how well you deliver on these components.

1. Self-study: you have to understand the conceptual framework of your topic. Some of this may be covered in the class, but you should read alternate textbooks, research papers and other references. You will be judged on the basis of the depth and the breadth of your conceptual understanding of the topic.

2. Main deliverable: this is a simulation program that you have to develop and demonstrate that it is operating correctly. You have to obtain experimental results using this program and show these results during the final presentation.

3. Analysis and Presentation: see Appendix D.

Chapter 2

LDPC Decoding for 5G NR

A starter Matlab code will be provided to you. See Appendix B.1. You are asked to perform the following tasks.

1. Pass the output of the encoder to a BPSK modulator, and the transmitted BPSK symbols to an AWGN channel which introduces a per-symbol SNR of E_S/N_0. See Appendix A.
2. Send the output of this AWGN channel to the LDPC message passing hard decision and soft decision decoders. See Appendix B.2.
3. Performance demonstration: evaluate performance metrics in Appendix C.3.2 (you should do C.1 and C.2, but you do not need to show this).

   The following are the suggested deliverables of this project.

   (a) Self-study: you should do a study of the definitions of the matrices BG1 and BG2 in 5G NR. Understand how these matrices are selected, how is an appropriate value of z (the lifting size) determined, and how the rate matching (puncturing of $\mathbf{H}$ matrix to obtain the desired code rate) is performed in 5G NR.

   (b) Main deliverable: hard and soft decision decoder and demonstration of their performance (bullets 2 and 3 above). Vary the code rate in the range $r \in \{1/4, 1/3, 1/2, 3/5\}$ and the SNR $E_b/N_0 \in \{0, 10\}$ in steps of 0.5 dB and produce the curves showing the probability of detection errors as a function of E_b/N_0 for each value of r.

   (c) Analysis: Compare the simulation results with the Shannon channel capacity bound.

Chapter 3

Polar Codes

Polar Codes are the most recent of all currently well-established coding schemes. They have been proven to be a very good/efficient coding scheme as evidenced by its adoption by the 3GPP standardization body for the 5G NR control channels.

You are asked to perform the following tasks.

1. Do a self-study of the Polar Coding scheme.[1] Understand the main idea[2] behind the Polar Codes, their encoding scheme and the decoding algorithm.

2. Write a Matlab code that performs the Polar encoding.

3. Pass the output of the encoder to a BPSK modulator, and the transmitted BPSK symbols to an AWGN channel which introduces a per-symbol SNR of E_S/N_0. See Appendix A.

4. Send the output of this AWGN channel to the successive cancellation list decoding algorithm for the Polar codes.

5. Performance demonstration: evaluate performance metrics in Appendix C.3.2 (you should do C.1 and C.2, but you do not need to show this).

The following are the suggested deliverables of this project.

1. Self-study: this is a major component for this project. Through a self-study, you have to understand and explain the principle and the concept underlying the Polar Codes.

2. Main deliverable: development of the Polar Successive Cancellation List Decoder, a conceptual explanation of this decoder, and its performance characterization (see bullets 4 and 5 above).

3. Analysis: explain how the Polar Codes can be theoretically proven to achieve Shannon's Channel Capacity bound (this is a continuation of the self-study component).

[1]An ISIT (IEEE International Symposium on Information Theory) lecture [2] by Prof. Emre Telatar provides a good conceptual description of the polar codes. You may also refer to an Event Helix article [3] for a brief summary. A detailed description is in an IEEE Communications Magazine article [7], which is part of a broader list of references available in [5]. Prof. Andrew Thangaraj at IIT Madras has given a nice series of lectures on (the LDPC and) the Polar Codes [9].

[2]The main idea makes use of multiple uses of BEC(p) that we have studied in the class and it is derived from an Information Theory standpoint. The concept of the Information Transfer $I(X;Y)$ that we have studied in the class is used in the derivation of the Polar Codes. You will see why are they called 'polar' codes, and how they exemplify the "Matthew's effect".

Chapter 4

Convolution Coding

The following are the suggested deliverables of this project.

1. Self-study:
   - ▷ Understand the transfer function of a convolution code and how it can be used to evaluate the minimum Hamming distance $d_H^{\min}$ and the pairwise Hamming distances d. You may refer to [8, Section 8-2-1].
   - ▷ Generalization of the hard decision Viterbi decoding explained in [10] to the soft decision Viterbi decoding.
2. Main deliverable:
   - Convolutional encoder for the following rates and constraint lengths $[r, K_c]$:
     - ▷ $\{r = 1/2, K_c = 3\}$ specified in the first row of [8, Table 8-2-1],
     - ▷ $\{r = 1/3, K_c = 4\}$ specified in the second row of [8, Table 8-2-2],
     - ▷ $\{r = 1/3, K_c = 6\}$ specified in the fourth row of [8, Table 8-2-2]
   - Pass the output of the encoder to a BPSK modulator, and the transmitted BPSK symbols to an AWGN channel which introduces a per-symbol SNR of E_S/N_0. See Appendix A.
   - Perform soft decision and hard decision Viterbi decoding and demonstrate the performance by varying the SNR $E_b/N_0 \in \{0, 10\}$ in steps of 0.5 dB. Produce the curves showing the probability of detection errors as a function of E_b/N_0 for each value of r.
3. Analysis:
   - ▷ Study and explain the derivations of the performance of the soft decision decoder of convolution code [8, Section 8.2.3], specifically, Equations (8-2-20), (8-2-21), (8-2-26)
   - ▷ Study and explain the derivations of the performance of the hard decision decoder of convolution code [8, Section 8.2.4].
   - ▷ Compare the simulation results of SDD and HDD with their respective analysis results for all the three convolution codes mentioned above

Appendix A

Simulation of BPSK Modulation and AWGN

1. The output of the channel encoder or the information source is a vector, whose each element is a bit $b \in \{0, 1\}$.

2. Each of these bits is sent to the BPSK modulator, which is simply a mapping of b to s, where $s = +1$ (volts) if $b = 0$, else $s = -1$ volts. This mapping can be implemented simply as $s = 1 - 2b$.

3. Let the AWGN introduce a per-symbol SNR $\gamma = E_S/N_0$ in the linear scale. Since the BPSK symbols $s \in \{-1, +1\}$, the signal power/energy is $E_S = 1$ Joule (in fact, it is proportional to 1 Joule, but we can ignore this constant of proportionality since it is the same for both E_S and N_0 and so it cancels out in the ratio). Therefore, the noise power is simply $\sigma_n^2 = 1/\gamma$.

4. The AWGN is simulated as $u = \sigma_n \times u_s$, where $u_s \sim N(0, 1)$ is the standard Normal variate with zero mean and variance of unity.

Appendix B

LDPC Scheme

B.1 A Starter Code

Following is a brief description of the LDPC starter code.

- The starter code generates the parity check matrix $\mathbf{H}$ for 5G New Radio (NR) LDPC codes [9]. In the 5G NR standard, the LDPC codes are specified using nonbinary symbolic representations called Base Graphs (BGs) [9]. There are two BGs in 5G NR – BG1 and BG2. The starter code uses BG2, a matrix with a size of 42×52.
- The non-binary BG matrices are converted to binary LDPC $\mathbf{H}$ matrices using a lifting variable called z. The starter code has all the necessary functions that perform this conversion.
- The transmitted information codeword $\mathbf{m}$, which is the vector `b` on line 39 of the starter code, is encoded (not using the generator matrix $\mathbf{G}$ that we have studied in the class) but directly using the BG matrix. This encoding function is part of the starter code (see line 40).
- Depending on the code rate r you specify, the starter code performs puncturing, i.e., the size of the matrix $\mathbf{H}$ is reduced to match with the specified r – see lines 18 to 20.
- The first $2z$ information bits of `b` are always punctured, see line 44.

B.2 Decoding Algorithm

▷ Your code should automatically convert the pattern of ones and zeros in the LDPC parity check matrix $\mathbf{H}$ into a Tanner Graph model. The connections between the VNs and the CNs of this bipartite graph are based on the locations of ones in $\mathbf{H}$. Make your code generic enough so that it works with any matrix $\mathbf{H}$ given as the input.

▷ Your model of Tanner Graph should perform the hard and the soft decision message passing from the VNs to the CNs and vice versa. This is described in [1].

  → Recall that in the bipartite Tanner Graph, each VN is connected to a total of d_v CNs, and each CN is connected to a total of d_c VNs. The value of d_v and d_c may vary across the VNs and the CNs, respectively.

  → Each CN is an SPC code. It enforces the even parity among all the VNs that are connected to it. Thus the message passed from the CN to the VN is based on the (hard decision or soft decision) SPC decoding.

→ Similarly, each VN is essentially a repetition code. It requires that all the messages are the same (either 1 or 0). Thus, the message passed from the VN to the CN is based on the (hard decision or soft decision) decoding of the repetition codes.

→ Refer to [1, 6, 4] to understand the message computations at the VNs and the CNs for the soft decision and hard decision decoding. These references are provided to you on Google Classroom.

Appendix C

Performance Metrics

C.1 Functionality Test

- ▷ In the functionality test, you want to confirm that the basic functions implemented you are working correctly in benign ideal conditions.
- ▷ You perform this test by disabling the channel impairments. For the channel coding projects and the MIMO project, you turn off the AWGN. For the OFDM project, you additionally make the LSI channel model to be the Dirac Delta function, which is the simplest type of the channel.
- ▷ You want to make sure that the receiver is correctly able to recover the transmitted bits or symbols at least in this ideal situation. If your receiver generates estimates that do not match with the transmitted symbols during the functionality test, you have some debugging to do!

C.2 Spot Checks

Here, you will deliberately introduce errors in the transmitted codeword or symbols, and check whether the decoder is able to recover the original message bits. One way to do so is to enable the AWGN but make the SNR high.

C.3 Monte Carlo Simulations

This is the real performance verification test. In this case, you enable the AWGN and perform N_{sim} Monte Carlo experiments for each value of SNR. Keep N_{sim} to be some large number such as 10000. The purpose of the Monte Carlo simulations is to evaluate the following performance metrics (here, n denotes the simulation experimental index; $n = 1, 2, \ldots, N_{sim}$).

C.3.1 Definition of SNR

For the channel coding projects, the definition of SNR is E_b/N_0. Perform the Monte-Carlo simulation for each $E_b/N_0 \in \{0, 0.5, 1, 1.5, \ldots, 10\}$ dB. Recall, from an earlier lecture, that SNR per symbol $E_S/N_0 = r \times E_b/N_0$, where r is the code rate and E_S/N_0 and E_b/N_0 are both in the linear scale (not in the decibel scale). Refer to Appendix A for a description on how to simulate the given value of E_S/N_0.

C.3.2 Probability of Decoding Success

For each value of E_b/N_0 in the range mentioned above, you are asked to evaluate the performance of your algorithms in terms of the probability of decoding success. To evaluate this probability, you will need to implement the following logic:

- ▷ At the n^{th} simulation experiment, at the ith decoding iteration (where $i \in 1, \ldots, i_{\max}$, and $i_{\max}$ is the maximum number of iterations), set a flag $F_{n,i}$ to 1 if the decoding is successful (decoding is successful if the decoded message bits match the transmitted message bits). Otherwise, $F_{n,i}$ is to be set to 0.
- ▷ *Algorithm Convergence:* Probability of successful decoding as a function of iteration index i is computed as $P_c(i) = \frac{1}{N_{sim}} \sum_{n=1}^{N_{sim}} F_{n,i}$.
- ▷ *Algorithm Performance:* Probability of Successful Decoding is simply $P_c i_{\max}$. Note that, given $i_{\max}$, this metric is not a function of the iteration index i.

C.3.3 Probability of Symbol Detection Error

For the OFDM and the MIMO projects, you are asked to evaluate the performance of your algorithms in terms of the probability of symbol detection error. Denote N_b as the number of symbols per block of transmission (N_b is equal to the number of subcarriers for the OFDM and number of transmit antennas for the MIMO). Let $0 \leq N_e(n) \leq N_b$ be the number of symbols that are decoded in error at the receiver at the nth simulation trial. The symbol error probability is evaluated as $P_s = \frac{1}{N_{sim}} \sum_{n=1}^{N_{sim}} N_e(n)/N_b$.

Appendix D

Modalities

D.1 Project Groups and Evaluations

- ▷ For the final project, you will be working as a team. There will typically be ten to twelve members per team. The composition of the project teams (i.e., assignment of the students to the teams) falls under the purview of your mentor TA. Each mentor TA will supervise ~ 3 teams.
  - → Although you will work as a member of a team, the project grades will be individually awarded. It is in your best interests, both from the learning point of view as well as getting a good evaluation, that each one of you does not sit on the sidelines or be satisfied with some secondary tasks (preparing the presentation, understanding your team-mate's code/logic, etc.). Instead you should insist on engaging in the actual program development and working as if this is your own individual project. Do not hope or expect that your team-mate will do a thorough study of the algorithm performance, instead be proactive and take up any such task yourself.
- ▷ You will make a demo and a presentation of your project work about a week prior to the final exams. At the time of evaluation, you should come prepared with a presentation (e.g., in Powerpoint) of your work (one presentation per project team), and should have a working code. You will be asked to demonstrate the operation of the program.
  - → Your work will be evaluated on the basis of how well you are able to deliver the technical task. You will be judged on the basis of whether (i) you fall short of meeting the deliverable, (ii) you have met the deliverable, or (iii) you have exceeded the expectations and done an outstanding job. To ensure that your evaluations are in category (ii), you should ensure that you have a working code whose performance results are aligned with the expectations. To be in category (iii), you should additionally perform detailed studies required for the technical deliverable, and do a deep analysis and thinking on *why* does your code behave in the manner that you have observed.
    - ▷ A good[1] way to show the results generated by your computer program is as follows: "these are the results that my program produces, and I think I know why they are coming out in this manner. Let me explain …". Thus, you would show that you have not only solved the problem of developing the simulation but you have examined its

[1] A bad way to show the results by your program is to say "this is what came out of my simulation, don't ask me why!".

inner workings and you have figured out why it works the way it works. Without such an explanation, it is not possible for you to make your results credible.[2]

   - ▷ It is crucial that your project presentation is professional and formal; it is crisp, concise and clear, and it flows[3] well (like a well-written story, a nice musical note or a good movie). This relates to developing and honing the technical communication skills. Those who master this art produce a presentation that is harmonious. Others generate a presentation that feels like discordant musical notes.

- ▷ You can use either Matlab, Python or C++ for this project.

D.2 Honor Code

You will be required to submit the following pledge at the time of the interim and final evaluations of the project work. This should be the opening page of your project presentation slides.

- We declare that
  - → The work that we are presenting is our own work.
  - → We have not copied the work (the code, the results, etc.) that someone else has done.
  - → Concepts, understanding and insights we will be describing are our own.
  - → We make this pledge truthfully. We know that violation of this solemn pledge can carry grave[4] consequences.
- Signed by: all the members of the project group.

[2]This would naturally encompass the process of sanity checking – ensuring that your program performance matches the expectations and that some bugs are not making your program perform either too good or too bad. A program with bugs is at the best useless and at the worst harmful.

[3]The presentation should begin with Objectives and Problem Statement, followed by Theoretical Concepts, followed by the performance results and their conceptual/insightful description. There should be conclusions and summary section and a page summarizing all the references. The detailed mathematics can be included as Appendices at the end of the presentation.

[4]E.g., you will be awarded 0 marks even if there is a slightest hint of plagiarism. It will be an extremely *foolish* mistake to engage in the plagiarism.

Bibliography

[1] CT216. *Introduction to Communication Systems: Lecture 3 Channel Coding.* DA-IICT, Winter 2024.

[2] Telatar Emre. The flesh of polar codes, 2017. Accessed on March 24, 2024.

[3] EventHelix. Polar codes: Develop an intuitive understanding, 2019. Accessed on March 24, 2024.

[4] R. Gallager. Low density parity check codes. *IRE Trans. Information Theory*, 1962.

[5] IEEE. Best readings in polar codes, 2019. Accessed on March 24, 2024.

[6] Bernhard Leiner. LDPC codes – a brief tutorial. *Self Study Project*, 2005.

[7] Kai Niu, Kai Chen, Jiaru Lin, and Q. T. Zhang. Polar codes: Primary concepts and practical decoding algorithms. *IEEE Communications Magazine*, 52(7):192–203, 2014.

[8] J. G. Proakis. *Digital Communications.* McGraw-Hill, 4th edition, 1995.

[9] Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019. Accessed on March 24, 2024.

[10] Matthew Valenti and Bibin Baby John. *Introduction to Communication Systems: A Lecture on Convolution Coding and Viterbi Decoding.* DA-IICT, Winter 2024.