Developing a Compression Algorithm for Deep Space Image Transmission

Trey Whitehead
School of Electrical and
Applied Sciences
Harvard University
Cambridge, Massachusetts 02138
Email: treywhitehead@college.harvard.edu

***Abstract*—NASA's Galileo probe mission the Jovian system faced a major setback when the spacecraft's high-gain antenna, designed to transmit high-quality images of Jupiter back to Earth, failed to open properly. Fortunately, the probe still had a low-rate omnidirectional antenna intact, but its transmission rate was far below what was required for mission success. This paper explores potential solutions for deep space image transmission, and how the combination of image compression and channel coding could be tantamount to the propagation of information through space, as it was to the success of the Galileo mission.**

I. Introduction

At his 1961 "Address at Rice University on the Nation's Space Effort," president John F Kennedy remarked that "the opening vistas of space promise high costs and hardships, as well as high reward." What president Kennedy, nor anyone else could not have anticipated, was the impact spacefaring would have on Earth's own internal communication networks.[1]

Perhaps no undertaking has proven more important in this regard than the Galileo Mission to Jupiter and Its Moons launched by NASA in 1989. The mission was carried out by the Space Shuttle Atlantis, which deployed the Galileo probe to investigate Jupiter and its four largest moons.[2]

During the mission's first flyby around Earth after launch, the Galileo probe received a remote command to open its high-gain antenna. The antenna was designed to transmit high-quality images of Jupiter back to Earth, but unfortunately, it failed to fully open, leaving the spacecraft with only a low-rate, omnidirectional antenna. This reduced the expected transmission rate of approximately 130 kbps to a meager 8-16 bps, which was far below what NASA had originally planned on having.

To save the Galileo probe, NASA's engineers had to contrive an innovative software approach. In 1991, they implemented a custom-made image compression algorithm, based on the discrete cosine transform (DCT) and channel coding remotely on the probe. The combination of image compression and channel coding was critical to maintaining a reliable transmission rate and salvaged the mission. This event was significant, as it showed that image compression algorithms could be used in deep space image transmission despite the concerns of potential loss of resolution.[3]

This paper begins with an overview of digital communication systems and coding, explaining Binary Phase-Shift Keying (BPSK), Quadrature Amplitude Modulation (QAM), and the concept of fading at a fundamental level. Then, it investigates the notion of reliability and added redundancy in reducing error before exploring the challenges of developing compression algorithms for deep space image transmission, taking the Galileo mission as a case study. Finally, it discusses the importance of compression algorithms in communication systems, particularly in situations where there are limitations placed upon bandwidth and transmission rates.

II. An Overview of Digital Communications and Coding

We will start our discussion on digital communication systems and coding by formally defining the components of a single-user channel, which will provide pertinent background information for the Galileo probe mission.

1) Terminology: The *input* to this channel is denoted by $\mathbf{m} = (m_1, m_2, \ldots, m_d) \in 0, 1^d$, where m constitutes a sequence of *d* bits. It is worth noting that messages, or random bit sequences, are uniformly distributed, which means that each message can occur with probability $1/2^d$.

Now that we have defined our input, we can move on to channel encoding. A *channel code* is a mapping $c : 0, 1^d \to 0, 1^k$ that adds redundancy to a transmitted message, where $k \geq d$, and is denoted by $m_c := c(m)$. Redundancy, which will be further explicated in *Section 3*, refers to the intentional replication of information in a signal or data; redundancy is critical because it allows us to detect and correct errors that might otherwise cause significant distortions or loss of information in the signal.

[1] Kennedy, John F. "Address at Rice University on the Nation's Space Effort." Address at Rice University on the Nation's Space Effort — JFK Library, 12 Sept. 1962, www.jfklibrary.org/learn/about-jfk/historic-speeches/address-at-rice-university-on-the-nations-space-effort.

[2] Taylor, Jim, et al. "Galileo Telecommunications." NASA, July n.d., 2002, descanso.jpl.nasa.gov/DPSummary/Descanso5–Galileo$_{new.pdf}$.

[3] Cheung, Kar-Ming, and Kevin Tong. "Proposed Data Compression Schemes for Galileo S-Band Contingency Mission ." NASA, 1 Apr. 1993, ntrs.nasa.gov/api/citations/19930015365/downloads/19930015365.pdf.

Next, we map the binary input sequence m_c to a sequence of n symbols $\mathbf{x(m)} = x_1(m), \ldots, x_n(m) \in R^n$. Although we produce x directly from m_c, we denote it as a function of m for simplicity. The value $x(m)$ is usually referred to as the channel input, and the set of all channel inputs make up a codebook. The defining feature of a codebook is its average power constraint, which can be given by:

$$\frac{1}{2^d} \sum_{m \in \{0,1\}^d} \|x(m)\|_2^2 \leq P,$$

where $||x||_2 = \sqrt{x_1^2 + \cdots + x_n^2}$ denotes the Euclidean norm.

Finally, in order to capture the effect of noise and fading on the received signal, we adopt a model of the form

$$\mathbf{y(t)} = H(t)x(t) + N(t),$$

where $N(t)$ and $H(t)$ are stochastic processes representing the additive noise and fading. Upon processing the received signal $y(t)$, we obtain a demodulated vector $y = Hx + N$ with $H \in R^{n \times n}$ and $N \in R^n$.

2) Testing BSPK: First, we will examine Binary Phase-Shift Keying, a modulation scheme involving two different phases of the reference signal. The primary goal of this section is to design a receiver which maps **y** to **m** so that we might demodulate our transmitted bit in a way that minimizes interference from noise. In order to do so, we will produce a graph that shows the effect of energy-per-bit over noise ratio, defined as $E_b := \frac{P}{d}$, on average bit error probability (BER). [4]

To find BER, we set our keys to be $x(0) = -\sqrt{P}$ and $x(1) = \sqrt{P}$. Our *threshold detector*, which determines where a bit maps to at the boundary condition $y = 0$, will take $y \geq 0 \rightarrow 1$ and $y < 0 \rightarrow 0$.

In addition, let's simplify H and N to be $H = 1$, and $N \sim \mathcal{N}(0, 1)$, respectively. We represent noise by the following probability distribution function (PDF):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right).$$

From these conditions, it follows that if a bit is equal to 0, then $x(0) = -\sqrt{P}$, which implies that $y = -\sqrt{P} + n$. More more importantly, however, we notice that an input bit of 0 leads to a negative output because of our aforementioned keying scheme and threshold detector, but only so long as $y \geq 0$ (or rather, $n \geq P$). If $n \leq P$, we would deduce that a bit had flipped and an error had occurred. [5]

To find error probability, we craft the following closed-form expression for when the variance is equal to 1:

[4]"Lecture 06: Bit Error Rate (BER) Performance." YouTube, YouTube, 29 June 2015, www.youtube.com/watch?v=v5jZSm6lMwU.

[5]"Lecture 07: Bit Error Rate (BER) of AWGN Channels." YouTube, YouTube, 3 July 2015, www.youtube.com/watch?v=-qUQrImVZog.

$$P\left(n \geq \sqrt{\frac{P}{\sigma^2 = 1}}\right) = \int_{\sqrt{\frac{P}{\sigma^2=1}}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \, dx.$$

When P ranges from 0 to 4, we can plot our expression for different variance levels (see *Figure 1*). [6]

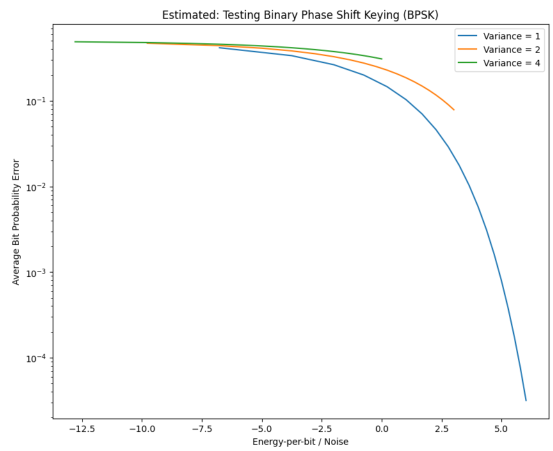

Fig. 1. Estimated BSPK with no Channel Coding when $n = 1$ and $d = 1$

In testing this equation against a Monte-Carlo simulation of the actual event, we verify the validity of our closed-expression. Assuming 500 simulations and the same variance levels as our predictive plot, we generate a parallel scatter plot (see *Figure 2*), which roughly mirrors our expected results in *Figure 1*.

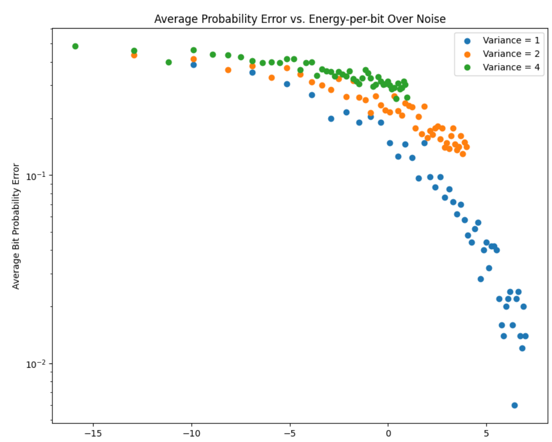

Fig. 2. Actual BSPK with no Channel Coding when $n = 1$ and $d = 1$

Hence, BER plummets as the energy-per-bit over noise ratio increases, which indicates that a higher average power constraint eliminates error.

[6]See Reference [1].

3) Testing 4-QAM: We can abstract this idea to Quadrature Amplitude Modulation (4-QAM) as well. Instead of sending 1 symbol and 1 bit at a time, we will 2 symbols and 2 bits. When the number of bits is equivalent to the number of symbols, we can complete a $1:1$ mapping which enables us to use the same receiver as in *Section 2.2*. The only adjustment we will make is that instead of $x(m) = \pm\sqrt{P}$, we will now use $x(m) = \pm\sqrt{\frac{P}{n}}$ to ensure that we comply with our average power constraint. H and N will remain the same as they were previously, and we can run another Monte Carlo simulation with 500 iterations (see *Figure 3*.

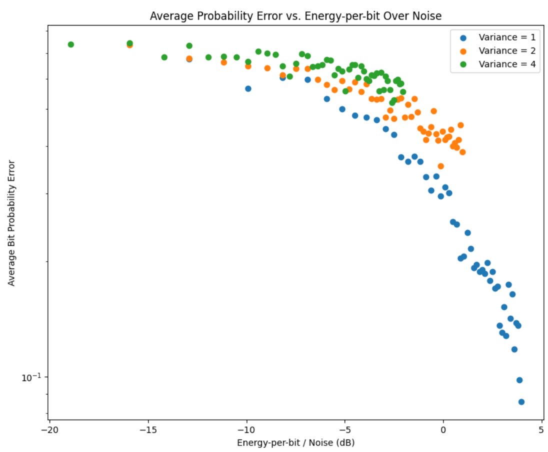

Fig. 3. Actual BSPK with no Channel Coding when $n = 2$ and $d = 2$

This new plot has a higher BER than our first by a factor of *almost* 10. As we progress through with different variations of d and n, it will be reasonable for us to expect that an increased amount of *bits* will be positively correlated with a higher BER.

4) More Complex QAM: As we begin to examine more complex modulation systems, specifically those with fewer symbols than bits, we start to lose information, which can be evidenced by a steep increase in BER. Systems with a different number of bits and symbols also beg the question: How do we simplify from d bits to n symbols. Although there are several different maps, we opted for the *mode method*, which involved dividing the initial bit sequence into substrings and simplifying each substring to its most common digit. From there, we used the previous bits-to-symbols architecture: $x(m) = \pm\sqrt{\frac{P}{n}}$, which ensured that we would remain beneath the average power constraint or our simulated P values. The results for three characteristically unique systems can be seen in *Figure 4*, *Figure 5* and *Figure 6*.

From these graphs, it is evident that performance quality progressively worsens as d grows. We will continue to investigate this relationship as we select more complex structures for H.

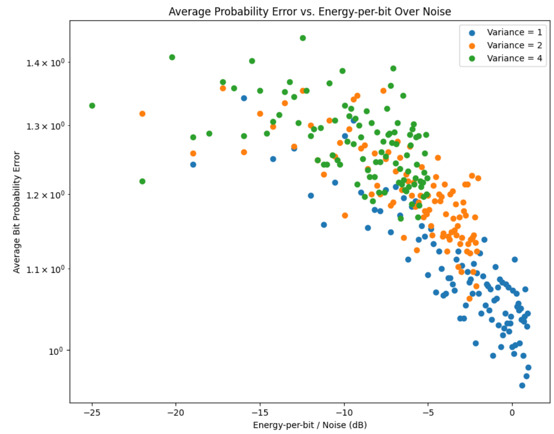

Fig. 4. 4-QAM when $n = 2$ and $d = 4$

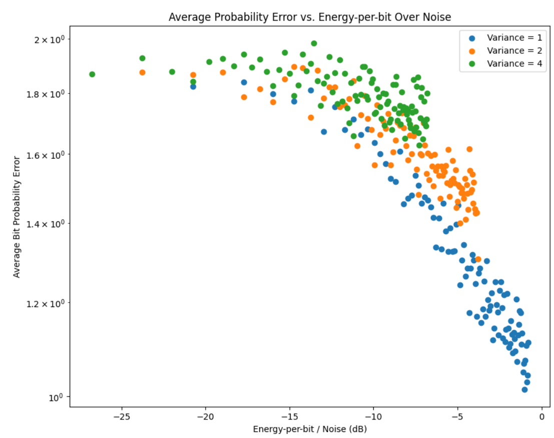

Fig. 5. 4-QAM when $n = 2$ and $d = 6$

5) The Impact of Fading: Fading, one of the main challenges in wireless communications, can be loosely defined as the attenuation of a signal as it travels from the transmitter to the receiver. It comes as a consequence of noise or other interference.

To model the impact of fading on our system, we will let $H \in R^{2\times 2}$. All diagonal entries of H, denoted as $[h_1, h_2]$ will be distributed as follows: $h_{1,2} \sim \mathcal{N}(0,1)$ When fading is unknown, BER will skyrocket because of the increased variation of the system.

$$y_{fade} = H \times x + N$$

However, if H is known, then fading can be mitigated at the receiver with linear algebra. By transforming our received signal by H^{-1}, we can recover more of what was lost due to noise:

$$y_{antifade} = H^{-1} \times (H \times x + N).$$

For differing levels of variance, the results from a system with $H \in R^{2\times 2}$ fading have been captured in *Figure 7*, *Figure 8*, and *Figure 9*. The subsequent anti-fading plots may be seen in *Figure 10*, *Figure 11*, and *Figure 12*.

Next, we will examine channel coding

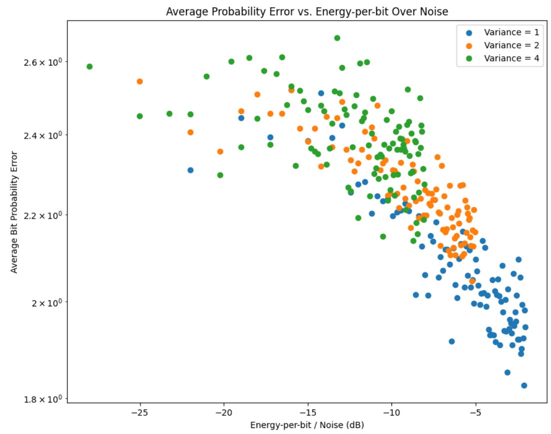

Fig. 6. 4-QAM when $n = 2$ and $d = 8$

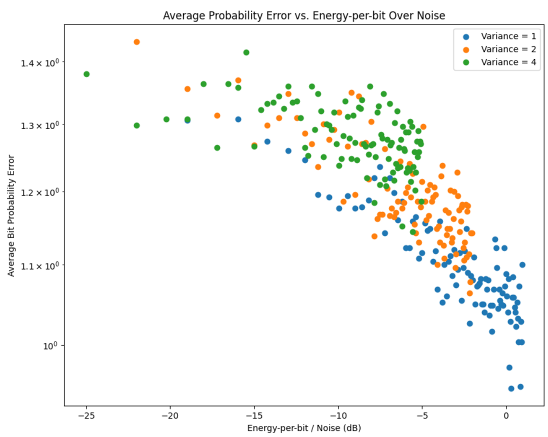

Fig. 7. Fading when $n = 2$ and $d = 4$

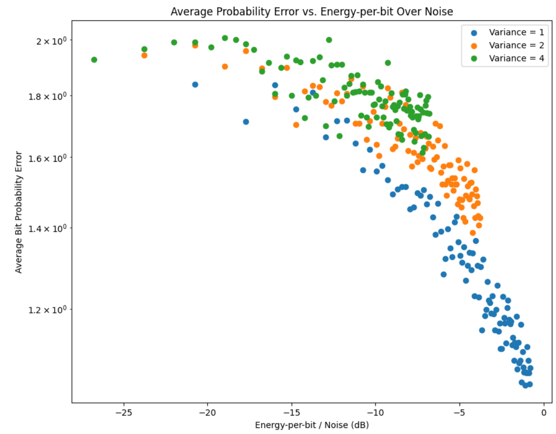

Fig. 8. Fading when $n = 2$ and $d = 6$

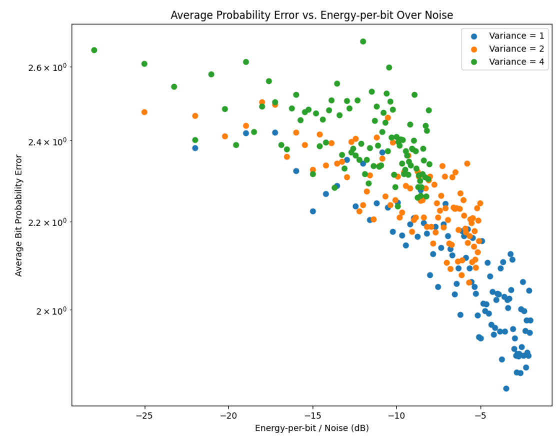

Fig. 9. Fading $n = 2$ and $d = 8$

III. The Impact of Coding in Reliable Communications

Channel coding is an important technique used in wireless communication systems to improve the accuracy and reliability of transmitted signals. By introducing redundancies and parity bits, it can enable us to detect and correct errors caused by fading, noise, and interference, leading to more robust and efficient communication systems.

Although graphical representations for the phenomena in the following section are currently out of the author's computer science skill zone and time allocation, they will be discussed from a conceptual level.

1) Repetition Code: A repetition code is a simple error-correcting code used in digital communication systems. The primary objective a repetition code is to transmit the same bit multiple times in a new singular message to improve the reliability of the transmission. A generalized repetition code could be defined as follows $\mathbf{c(m)} = m_1 m_2 ... m_d m_1 ... m_d m_1 ... m_d$. For example, using a 3-repetition code and BPSK when transmitting a 1-bit sequence where $m = 0$, we would generate $c(0) = 000$ and $x(0) = \sqrt{P/3}, \sqrt{P/3}, \sqrt{P/3}$.

If we were to graph BSPK and QAM as we did in Secction 1 with repetition codes, we suspect that BER would be significantly reduced as $\frac{E_b}{\sigma^2}$ increases. If this were to happen, it would be a direct effect of the added redundancy.

2) Other Codes: On that note, there are many other codes, aside from repetition codes, that are capable of increasing transmission reliability and efficiency. We will briefly dive into one of these codes, which involves S-band communication and lossless compression in the next section.

IV. Disaster Strikes the Galileo Probe

To solve their transmission conundrum, the engineers at NASA opted for two strategies on the Galileo Mission: *lossless compression* and *lossy compression*. The key distinction between the two involves how they preserve information. Lossless compression algorithms reduce the size of a digital file without losing any information, whereas lossy compression algorithms intentionally discard some information to achieve

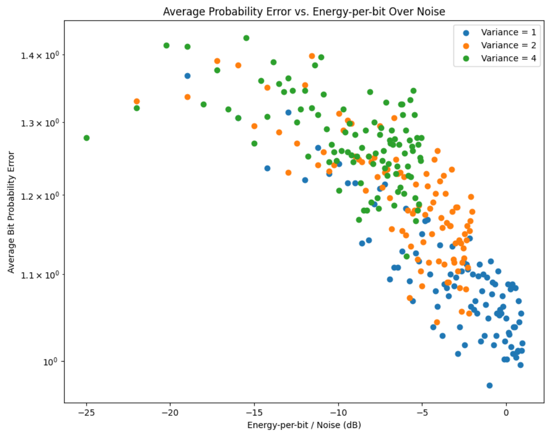

Fig. 10. Anti-fading when $n = 2$ and $d = 4$

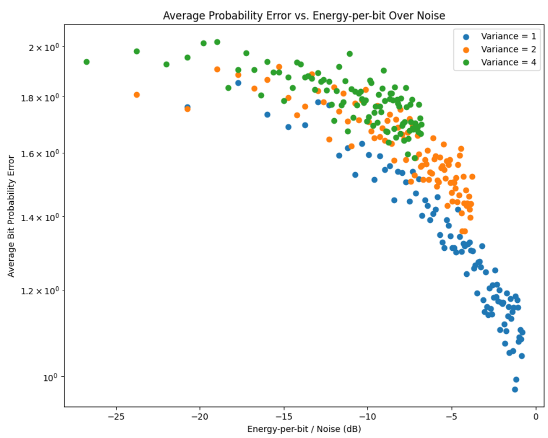

Fig. 11. Anti-fading when $n = 2$ and $d = 6$

greater compression.[7]

I speculate that NASA might have used arithmetic coding, a lossless compression algorithm, to increase their operational capabilities on the probe. Arithmetic coding employs fractional values to represent the probabilities of different symbols in the data. Put differently, this allows the algorithm to achieve high compression ratios because data can be encoded as a single fraction between 0 and 1, as the numerator and denominator are iteratively updated as each symbol is processed. At the end of the encoding, the resulting fraction gets output as the compressed data.[8]

What distinguishes arithmetic coding is its ability to handle variable-length codes. Arithmetic coding works exceptionally well with highly skewed frequency distributions, since it can assign shorter codes to more frequently occurring symbols and longer codes to less frequent symbols. As a result, the

[7]Taylor, Jim, et al. "Galileo Telecommunications." NASA, July n.d., 2002, descanso.jpl.nasa.gov/DPSummary/Descanso5–Galileo$_{n}ew.pdf$.

[8]Gad, Ahmed. "Lossless Data Compression Using Arithmetic Encoding in Python and Its Applications in Deep Learning." Neptune.ai, 21 Apr. 2023, neptune.ai/blog/lossless-data-compression-using-arithmetic-encoding-in-python-and-its-applications-in-deep-learning.

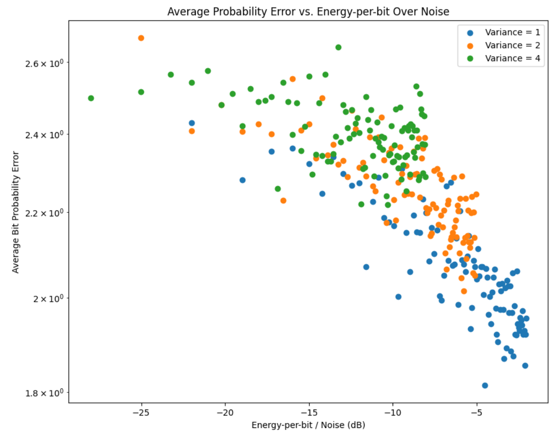

Fig. 12. Anti-fading $n = 2$ and $d = 8$

algorithm to take advantage of the statistical redundancy in the data, resulting in more efficient compression.[9]

V. Conclusion

In a humble attempt to develop a compression algorithm for deep space image transmission, we explored digital communications and coding, the impact of coding on reliable communications, and a potential solution to the Galileo probe's data compression problem. Areas for further research include more rigorous error-correcting codes and their graphical representations.

That said, we also discussed some of the same basic digital communications systems that Shannon touched upon in his 1948 paper "A Mathematical Theory of Communications." His paper would go on to change the world, as would the United States's continued pursuit of space.

In attempting these *giant leaps for mankind*, Earth takes a step towards a brighter future. Through creativity and innovation, the scientists at NASA were able to salvage the Galileo probe and discover vital information about Jupiter's satellites.

Acknowledgment

I would like to thank Flavio Calmon, Assistant Professor of Electrical Engineering at Harvard's John A. Paulson School of Engineering and Applied Sciences, Joel Rakhamimov, and Lucas Monteiro Paes for their continued guidance throughout this project and the ES-156: Signals and Communications course. The *1.1* section was pulled directly from the final project description that they wrote, and it should be noted that Open Ai's ChatGPT was used to (1) recreate the exact LaTeX that they used in that document and (2) help me define "lossless compression" in *Section 3*. I would also like to thank Aryan Naveen, who was kind enough to help debug my code for the plots in *Section 1.3*.

[9]Howard, P. G., and J. S. Vitter. "Arithmetic Coding for Data Compression." Proceedings of the IEEE, vol. 82, no. 6, 1994, pp. 857–65, https://doi.org/10.1109/5.286189.

This has been a fantastic semester, and I felt extremely grateful to be a member in this class. Thank you for having me on board, and I look forward to working with you all in the future! Enjoy the summer.

REFERENCES

[1] See this link for the code for all notebook plots: https://colab.research .google.com/drive/18lRCNGd2qvBRQkUGRbp$_{4a}rMQL8jsEd?usp = sharing$.

[2] Cheung, Kar-Ming, and Kevin Tong. "Proposed Data Compression Schemes for Galileo S-Band Contingency Mission ." NASA, 1 Apr. 1993, ntrs.nasa.gov/api/citations/19930015365/downloads/19930015365.pdf.

[3] Dunbar, Brian. "Galileo's New Telecommunications Strategy." NASA, 23 Mar. 2024, www.nasa.gov/audience/formedia/factsheet/ Galileo$_{c}ommunication_{f}actsheet.html$.

[4] Gad, Ahmed. "Lossless Data Compression Using Arithmetic Encoding in Python and Its Applications in Deep Learning." Neptune.ai, 21 Apr. 2023, neptune.ai/blog/lossless-data-compression-using-arithmetic-encoding-in-python-and-its-applications-in-deep-learning.

[5] Howard, P. G., and J. S. Vitter. "Arithmetic Coding for Data Compression." Proceedings of the IEEE, vol. 82, no. 6, 1994, pp. 857–65, https://doi.org/10.1109/5.286189.

[6] "Galileo Antenna Deployment Problem Under Analysis." NASA, 19 Apr. 1991, www.jpl.nasa.gov/news/galileo-antenna-deployment-problem-under-analysis.

[7] "Galileo." NASA, 19 July 2021, solarsystem.nasa.gov/missions/galileo/in-depth/.

[8] Kennedy, John F. "Address at Rice University on the Nation's Space Effort." Address at Rice University on the Nation's Space Effort — JFK Library, 12 Sept. 1962, www.jfklibrary.org/learn/about-jfk/historic-speeches/address-at-rice-university-on-the-nations-space-effort.

[9] "Lecture 06: Bit Error Rate (BER) Performance." YouTube, YouTube, 29 June 2015, www.youtube.com/watch?v=v5jZSm6lMwU.

[10] "Lecture 07: Bit Error Rate (BER) of AWGN Channels." YouTube, YouTube, 3 July 2015, www.youtube.com/watch?v=-qUQrImVZog.

[11] Taylor, Jim, et al. "Galileo Telecommunications." NASA, July n.d., 2002, descanso.jpl.nasa.gov/DPSummary/Descanso5–Galileo$_{n}ew.pdf$.