CRC AIDED DYNAMIC LIST SCL DECODER FOR POLAR AND PAC CODES

by

Hossein Sharifzadeh Behzadi

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
August 2023

To the Unsung Heroes, this article is dedicated to all those remarkable souls whose selfless endeavors shape the world's fate. Often unseen, yet bearing the burden of countless lives, you make existence easier for millions. The silent champions, eclipsed by shadows, whom no one ever dedicated anything to them. We humbly offer our heartfelt gratitude for your invaluable contributions. You, truly make the world a better place, and for that, we honor you

Table of Contents

List of Figures

Abstract

Error-correction codes are indispensable to ensure reliable data transmission in various communication systems. Among these, Polar codes and PAC (Polarized Adjustable Convolutional) codes have gained significant attention due to their capacity-achieving properties and low encoding complexity. This thesis presents a novel decoding technique that combines the Targeted Parity Protection (TPP) algorithm and Dynamic Successive Cancellation List (SCL) decoding to achieve substantial performance improvements while maintaining much lower complexity compared to the conventional Fano decoder.

The proposed technique focuses on leveraging Cyclic Redundancy Check (CRC) bits to protect critical information bits, leading to enhanced error correction capabilities. By applying TPP, the decoding process is accelerated by effectively reducing latency and saving computational resources by eliminating wrong codeword candidates at an earlier stage before the decoder favors them.

The integration of Dynamic SCL introduces adaptability to the decoder, allowing it to adjust the list size based on the decoding requirements for different code rates and lengths. This adaptability enables efficient use of hardware resources by optimizing the utilization of the decoder storage but also tailors the decoder's performance to each specific scenario, yielding substantial error correction performance gains in both Polar and PAC codes.

Extensive simulations and performance evaluations demonstrate the effectiveness of the proposed technique. The results indicate a gain of up to 0.4 dB for Polar codes and up to 0.6 dB for PAC codes compared to the standard SCL decoding approach. It is worth mentioning that such gains are considered substantial for fiber optical communications which is one of the target applications of the new technique. Furthermore, the method showcases a reduction in decoding complexity, outperforming the traditional Fano decoder.

Acknowledgements

A million thanks to my supervisor Dr. Dmitry Trukhachev who has always helped me with his insightful supervision.

Chapter 1

Introduction

In an era where information is a non-separable part of human daily life, optical fiber channels have emerged as a necessary tool, making us capable of better, faster, and more reliable connections.

Optical fiber, composed of strands of ultra-thin, transparent glass or plastic, has proven to be a game-changer in telecommunications. Its ability to transmit vast amounts of information over long distances at the speed of light has opened up new frontiers of possibility, transforming the world into a global village where barriers of time and space dissolve. One of the most tangible benefits of optical fiber lies in its exceptional speed. Traditional copper-based transmission systems are unable to compete with the speed of light. As a result, the improvement of optical fiber channel speeds holds immense potential for improving the quality of our lives. [7]

By expanding the bandwidth and optimizing the speed of optical channels, we can benefit from faster high-quality video conferencing, enabling virtual meetings that bridge gaps and foster collaboration across continents.

Over the past few years, significant advancements have been made in detecting and mitigating the effects of noise in fiber channels. Noise, stemming from various sources such as environmental factors, signal interference, and imperfections in the fiber itself, can degrade the quality of data transmission and limit the potential of optical fiber networks. However, through sophisticated techniques and innovations, researchers have developed remarkable solutions to overcome these challenges.

Recent developments in modulation schemes have significantly contributed to the increased capacity and efficiency of optical fiber channels. Traditional modulation schemes, such as on-off keying (OOK), have given way to more advanced formats like quadrature amplitude modulation (QAM) and phase-shift keying (PSK). These modulation techniques allow for the transmission of multiple bits per symbol, effectively increasing the data rate without requiring additional bandwidth. By exploiting the amplitude, phase, and polarization of light waves, these sophisticated modulation schemes enable higher transmission speeds and better spectral efficiency. [8]

One notable improvement lies in the development of advanced noise detection mechanisms. By employing error detection algorithms and signal processing techniques, optical fiber networks can identify and isolate noise disturbances more accurately. These detection systems, often integrated into the optical transceivers, can monitor the signal quality in real-time and dynamically adjust the transmission parameters to optimize performance. With enhanced noise detection capabilities, fiber channels can maintain higher data integrity, minimizing errors and maximizing the efficiency of data transmission.

In addition to noise detection, advancements in encoding and decoding systems have played a crucial role in improving the overall performance of optical fiber channels. Encoding techniques such as forward error correction (FEC) enable the detection and correction of errors that occur during transmission. FEC algorithms introduce redundancy into the transmitted data, allowing the receiver to reconstruct the original information even if some bits are corrupted or lost along the way. These error-correcting codes, combined with powerful decoding algorithms, enhance the reliability and robustness of optical fiber communication, ensuring the delivery of accurate and intact data.[9]

In recent years, remarkable breakthroughs in coding theory have further propelled the capabilities of optical fiber channels, effectively bridging the gap between channel performance and the dispersion bound. Among these groundbreaking schemes, polar codes have emerged as a game-changing technology, laying the foundation for the seamless transfer of data in the era of 5G connectivity.

Polar codes, pioneered by Erdal Arıkan, have revolutionized the field of error correction coding. These codes exhibit remarkable properties that allow for achieving the capacity of a communication channel with low-complexity encoding and decoding algorithms. By exploiting the polarizing effect of carefully chosen channel transformations, polar codes can achieve near-optimal error correction performance, surpassing previous coding schemes.[3]

The article [10] presents a comprehensive Polar coding solution that addresses the industry's concerns regarding low-complexity implementation and superior performance across a wide range of code lengths and rates. The proposed design achieves better performance compared to Arikan's scheme while requiring only a fraction of the implementation cost. It incorporates reliability calculation, rate matching, and parity-check coding, providing a complete solution that is both hardware-friendly and flexible. The design applies to arbitrary code lengths and rates, allowing for fine-grained simulation results for numerous cases. The released results can serve as a baseline for future optimizations of Polar codes. So this novel Polar construction exhibits superior and stable error correction performance, making it suitable for 5G and future networks. The design surpasses previous approaches, such as CA-Polar, by integrating hardware-friendly techniques and providing a full solution for arbitrary code lengths and rates. Zhang and his team emphasize that their solution can be reproduced and serve as a benchmark for further optimizations in Polar coding.

In another work[11], Vardy and his team discuss polar coding and its performance limitations at short blocklengths with standard successive cancellation decoding. They propose polarization-adjusted convolutional (PAC) codes as a new scheme to improve the performance of polar codes in such scenarios. PAC codes utilize convolutional precoding and sequential decoding instead of CRC codes and list decoding used in conventional polar codes. The primary goal of the article is to explore whether sequential decoding is essential for the superior performance of PAC codes and whether list decoding can achieve similar results. They demonstrate that list decoding, with a moderately large list size ($L > 128$), can achieve comparable performance to sequential decoding. List decoding offers advantages in low-SNR regimes and situations where worst-case complexity or latency is a primary concern. Insights into the remarkable performance of PAC codes are provided by observing the similarity of sequential and list decoding to maximum likelihood (ML) decoding. The estimation of low-weight codewords and the approximation of the union bound on performance further highlight the superiority of PAC codes over polar codes and Reed-Muller codes. [11]

There have been some advancements to develop systematic Polar and PAC in [12] and [13]. These articles discuss fast methods for systematic encoding of polar subcodes, focusing on their potential benefits for improving bit error rate (BER), decoding complexity, and latency in 5G communication systems. Systematic encoding refers to a method where the original payload data is directly included in the transmitted codeword, making it easier to retrieve the payload during decoding. The study demonstrates that systematic encoding of polar subcodes leads to lower BER compared to both non-systematic encoding and systematic polar codes. This finding highlights the importance of employing systematic encoding techniques for achieving better error performance in polar subcodes. The proposed algorithms in the article enable efficient systematic encoding for polar subcodes, providing a means to simplify the decoding process by directly obtaining payload data from the restored codeword. This has practical implications for reducing decoding complexity and latency in 5G polar subcodes, which are essential for enhancing the overall performance of 5G communication systems. Overall, the article presents an efficient approach to systematic encoding for polar subcodes, showing its potential benefits in terms of improved BER, decoding simplicity, and applicability to the polar subcodes defined in the 5G standard.

Also, another one of the best decoder algorithms for Polar codes available was explained in detail in [14]. It is the log-likelihood ratio (LLR) based successive cancellation list decoding for polar codes, which offers numerical stability and simplifies the sorting step in decoding. A hardware architecture for the LLR-based decoder is proposed, demonstrating improved efficiency compared to existing architectures. Stimming also investigates the performance of CRC-aided successive cancellation list decoding with different list sizes and CRC lengths, highlighting the importance of adapting the CRC length to the list size for achieving the best error-rate performance. Additionally, they synthesize conventional successive cancellation decoders and compare them to the proposed CRC-aided successive cancellation list decoders in terms of throughput, area, and decoding latency. While the conventional decoders have higher throughput and smaller area, the proposed decoders exhibit significantly lower decoding latency, making them more suitable for low-latency applications.

Furthermore, [14] discusses simplified successive cancellation (SSC) decoders aimed at reducing decoding latency, but notes that the family of SSC decoders is not directly applicable to LLR-based successive cancellation list decoding. However, promising pruning approaches introduced for LLR-based SCL decoding can be utilized in the

LLR-based SCL decoder architecture. So the LLR-based path metric for SCL decoding of polar codes enables numerically stable decoding and simplified sorting. The proposed LLR-based SCL decoder exhibits higher throughput and lower area compared to existing decoders, leading to increased hardware efficiency. The inclusion of a CRC unit and the use of CRC-aided SCL decoding offer improved hardware efficiency at a given block-error probability. The results suggest that the proposed LLR-based SCL decoder is well-suited for low-latency applications and contributes to the advancement of efficient decoding techniques for polar codes.

Moreover, [15] proposes an improvement to the standard successive cancellation (SC) decoding of polar codes by incorporating the Fano sequential decoding technique. This approach, known as SC-Fano decoding, addresses the limitation of SC decoding by allowing backward movement when the reliability of an ongoing path is insufficient. SC-Fano decoding is shown to offer a better performance-complexity tradeoff compared to existing decoding methods such as SC-List (SCL), SC-Stack (SCS), and SC-Flip (SCF) decodings. Furthermore, [15] extends the SC-Fano decoding technique to cyclic redundancy check (CRC) concatenated polar codes by incorporating the bit-flipping idea from SCF decoding. Simulation results demonstrate the improved performance and complexity tradeoff of the proposed SC-Fano decoding method compared to existing decoding methods.

To address the complexity issue, [6] proposes a method for bounding the computational complexity of sequential decoding for PAC codes. The proposed bound limits the total number of node visits during decoding, potentially reducing the computational complexity without significantly compromising the frame error rate (FER) performance. [6] also discusses the choice of the data index set in PAC codes, based on the Reed-Muller (RM) design rule, which maximizes the minimum distance d_{min} and is crucial for achieving better FER performance at high signal-to-noise ratio (SNR). However, using the RM rule for A selection may not be fully compatible with the channel polarization created by the polar transform, resulting in higher computational complexity. Also, they studied new research aspects such as including exploring alternative design rules for a selection and their impact on performance and complexity. The study highlights the significance of connection sequence c, which affects the performance of PAC codes by influencing the multiplicity of minimum-distance codewords (A_{dmin}). The choice of a specific generator sequence can significantly impact the FER performance, indicating the importance of finding good generator sequences for PAC codes.

Also [16] presents a reduced complexity sequential decoding algorithm for polar codes and their variations. The proposed approach decomposes the polar code into multiple outer codes and constructs codewords of these codes on-demand in descending order of their probability. This is facilitated by fast decoding algorithms available for the decomposed codes. The algorithm also makes hard decisions on intermediate log-likelihood ratios (LLRs) and avoids decoding certain outer codes, further reducing complexity. The article highlights the data structures used for the sequential decoding of polar codes. The introduced algorithm applies to polar codes and polar codes with CRC and short-extended BCH codes. It offers lower average decoding complexity compared to existing decoding algorithms for these codes. [16] highlights that the proposed block sequential decoding algorithm achieves comparable performance to sequential, stack, and list decoding algorithms while having lower complexity. Particularly, in high SNR scenarios, the proposed algorithm exhibits higher throughput than the fast list decoder with a smaller list size. This implies that the proposed algorithm provides better performance and lower decoding complexity than the list decoding algorithm. Overall, the algorithm can be used effectively for decoding various types of polar codes and offers improved performance with reduced complexity.

In general, Polar codes, with their superior error correction capabilities, have been widely adopted as the channel coding scheme for 5G data transfer. They provide the necessary reliability to counteract the adverse effects of channel noise, allowing for robust and error-free transmission even in challenging wireless environments. As a result, 5G networks can achieve higher data rates, improved spectral efficiency, and lower latency, empowering an array of transformative applications such as autonomous vehicles, smart cities, and the Internet of Things (IoT).

Furthermore, polar codes have played an important role in closing the gap between the channel performance and the dispersion bound in optical fiber communication. Dispersion, which refers to the spreading of optical signals as they travel through the fiber, limits the achievable data rates over long distances. By employing powerful forward error correction through polar codes, the impact of dispersion can be mitigated, allowing for higher transmission speeds. For example, Figure 1 demonstrates

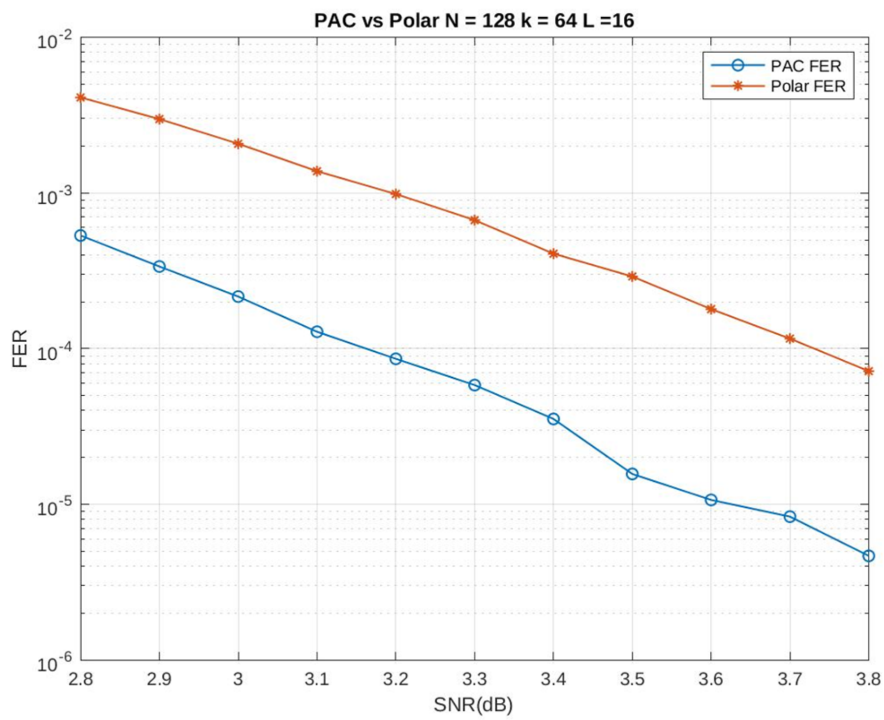

Figure 1.1: Polar vs PAC performance curves over the BIAWGN channel[1]

the performance of a polar code with a length of 128 and a rate of 1/2 when subjected to binary input and AWGN channel, utilizing standard successive cancellation (SC) decoding. This figure closely replicates the simulation outcomes originally presented by Arıkan in [1]. Throughout Arıkan's recent paper [1], codes with a length of 128 and a rate of 1/2 are utilized as the ongoing example, and we will adopt this approach as well. Figure 1 also includes the BIAWGN dispersion bound approximation for such codes, which can be seen as an estimation of the performance of random codes under ML decoding. As indicated in [17] and other studies, the gap between the two is attributed to two factors: the inherent weakness of polar codes at shorter lengths and the comparative weakness of SC decoding compared to ML decoding. To address both issues, a well-known technique involves CRC precoding followed by successive cancellation list (SCL) decoding. By Arikan's study, Figure 1 displays the performance of CRC-aided polar codes (with an 8-bit CRC) of rate 1/2 under SCL decoding with a list size of 32. During his Shannon Lecture at the ISIT in 2019, Erdal Arıkan introduced a significant advancement in polar coding, which significantly enhances the performance of polar codes at shorter lengths. Specifically, in Arıkan's work [1], he proposed a novel polar coding scheme known as polarization-adjusted convolutional (PAC) codes. Impressively, when subjected to sequential decoding, PAC codes exhibit performance very close to the BIAWGN dispersion bound approximation.

In the [18] researchers again focus on the impact of pre-transformations using upper-triangular matrices on the minimum distance and performance of different types of codes, such as cyclic redundancy check (CRC), parity-check (PC), and convolution codes (CC). "Bin Li" establishes that pre-transformations with upper-triangular matrices, including convolution matrices, do not reduce the code's minimum distance. However, they also demonstrate that a well-designed pre-transformation can reduce the number of codewords with the minimum distance. Consequently, the pre-transformed polar codes (CRC/PC/PAC) exhibit superior performance compared to the non-pre-transformed Polar/RM codes, specifically under maximum likelihood (ML)-type decoding. The article concludes by noting that optimizing the number of minimum distances through the design of T (the transformation matrix) remains an unresolved challenge.

The article [19] focuses on a comparative study of different polar code constructions in an additive white Gaussian noise (AWGN) channel. The objective is to identify the construction algorithm that produces the best polar codes at different design signal-to-noise ratios (design-SNR) in terms of bit error rate (BER). The article begins by conducting a comprehensive survey of well-known polar code constructions, including their full implementations. Since optimal polar code construction is challenging, the paper explores various suboptimal algorithms proposed at different computational complexities. It is noted that polar codes are non-universal, meaning their performance significantly varies with the design-SNR. To determine the optimal design-SNR for constructing the best possible polar codes, "Vangala" and "Viterbo" introduce a heuristic algorithm that searches among different design-SNR values. Using this algorithm, they conduct extensive simulations to compare the performance of various construction algorithms. The results of the study demonstrate that all polar code construction algorithms generate equally good polar codes in an AWGN channel when the design-SNR is optimized. The conclusion highlights the presentation of a comprehensive survey of well-known construction algorithms, the proposal of a simple discrete search to identify the optimal design-SNR for any given construction algorithm, and the finding that all algorithms perform equally well in AWGN when the design-SNR is optimized. As a result, they suggest the potential use of simple algorithms in the future. Overall, the [19] provides insights into the performance characteristics of different polar code constructions and emphasizes the importance of optimizing the design-SNR for achieving optimal results in an AWGN channel.

Recently there have been some developments to use deep learning algorithms for polar coding. [20] presents a deep learning-based algorithm for constructing polar codes. The algorithm represents the information/frozen bit indices of the polar code as a binary vector, which is interpreted as trainable weights of a neural network (NN). By relaxing the binary vector to a soft-valued vector, gradient descent is employed for efficient code construction. The approach considers different polar code design constraints, such as code rate, through careful conversions between binary and soft values, along with rate adjustment during each learning iteration. The integration of the decoder in the learning process allows for improved code design. Simulation results demonstrate significant performance gains over state-of-the-art construction schemes when using belief propagation (BP) decoding over both AWGN and Rayleigh fading channels. The framework showcases the potential of deep learning techniques and more specifically, standard stochastic gradient descent(SGD)-based training in communication system design, particularly the selection of frozen bit positions in polar codes. The relaxation of the binary vector representation to a soft-valued vector, along with appropriate loss functions and weights, enables satisfying various design constraints while tailoring the code to the decoder and channel. The results demonstrate improved performance compared to existing construction schemes. The framework can be extended to different setups, and its applicability to other codes, such as LDPC codes, is discussed. Several future research aspects, such as exploring different rate adjustment schemes and optimizing intermediate loss functions, are suggested. They also discussed the capability of their SGD learning method for decoding the CRC-aided Polar codes and also using it as a kind of belief propagation list (BPL) decoder with forward and backward propagations. Moreover, "Ebada" and his team discuss the optimum "Loss function" needed for training based on different code rates.

Finally, In [21] a selectively precoded polar (SPP) code is proposed, which is an enhancement of Arikan's capacity-achieving polar codes. The encoding and decoding schemes for the SPP code are provided. Simulation results demonstrate that the (128, 64) SPP code achieves a frame erasure rate (FER) performance that is only 0.23 dB away from the information theoretic limit at this block length, with better distance properties compared to other polar code variants. In conclusion, the SPP code outperforms other polar code variants in terms of FER performance and performs slightly better than Arikan's PAC codes at high signal-to-noise ratios (SNRs). The SPP code also exhibits favorable distance properties similar to PAC codes. However, there are still open problems to be addressed, such as exploring rate profilers for code rates other than 0.5 and determining the optimal combination of precoding indices and vectors for achieving the best FER performance. These areas are identified as future work for further investigation.

1.1 Motivation

In the past years, polar codes could show good advantages in lots of applications including inner code for optical channels. they are implemented next to a forward error correction (FEC) to improve the channel's performance. [2]

Polarized adjustable convolutional code(PAC) is an improved version of polar code that adds some convolutional features to the polar code and using that, the gap between channel performance and dispersion bound would be decreased. Both Polar code and PAC code could be decoded using two main methods called successive cancellation and Fando decoding. both methods are a kind of tree search that searches the codeword tree for the best possible codeword that matches the soft information in the decoder. SCL is a variant of the SC decoder that keeps a list of the best candidates and this way it can identify the best codeword easier than the SC decoder. Fano decoder currently has the best performance among different methods of decoding polar codes because there's a possibility to search all of the possible codewords for this method which increases the reliability of decoding with a huge cost of adding

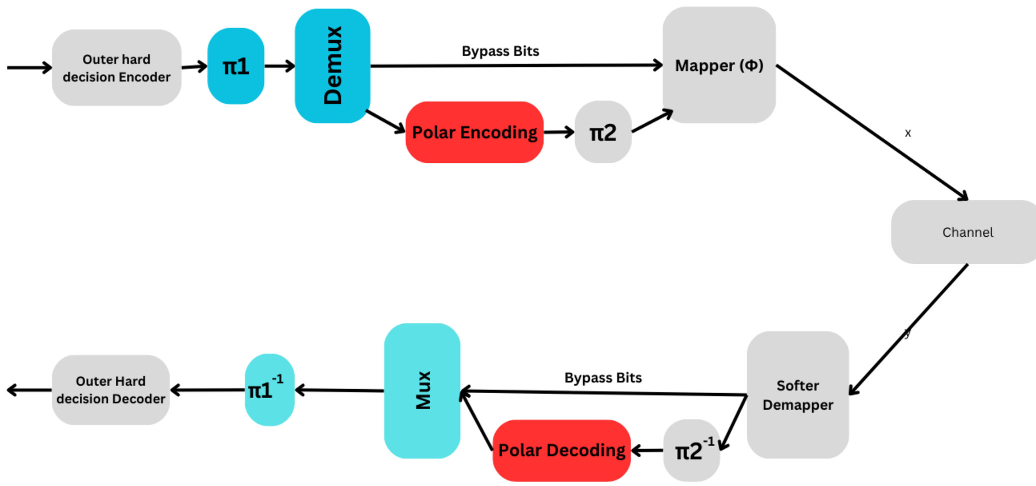

Figure 1.2: Multi-layer optical system using Polar code as inner code [2]

much more complexity.

SCL has much less complexity compared to the fano decoder which makes the decoding faster and improves the data throughput in optical communication. The goal of this research is to study the weaknesses of SCL decoding and try to improve the performance of the SCL decoding while keeping the complexity low so it becomes more suitable and appealing for applications in optical channels.

This thesis commences its exploration by delving into the historical underpinnings of fiber optics, its recent improvements, and multi-level coding in Chapter 2. Subsequently, Chapter 3 provides an in-depth examination of channel polarization and the various components constituting a polar code system. Building upon this foundational knowledge, Chapter 4 shifts its focus to an enhanced iteration of the Polar code, termed the Polarized Adjustable Convolutional Code (PAC). This chapter serves as the culmination of the background information. Moving forward, Chapter 5 introduces the concept of Targeted Parity Protection (TPP) coupled with Dynamic List Successive Cancellation List (SCL) decoding techniques for both Polar and PAC systems. Finally, Chapter 6 elucidates the remarkable potential of this novel approach, demonstrating a noteworthy enhancement in performance, specifically a 0.4 dB improvement for Polar code systems and a 0.6 dB enhancement for PAC systems in the context of AWGN Channel simulations.

Chapter 2

inner/outer optical coding schemes

2.1 Fiber Optics channel

Fiber optic channels have revolutionized the way we transmit data over long distances. With their high data transfer rates and immunity to electromagnetic interference, fiber optic channels have become the preferred choice for many applications, ranging from telecommunications to data centers.

Fiber optic channels are a type of communication channel that uses light to transmit data over long distances. They consist of a core made of high-quality glass or plastic fibers, surrounded by a cladding layer that reflects light back into the core. A protective coating, usually made of polymer, covers the cladding layer to protect it from damage.

The core of a fiber optic channel is typically just a few microns in diameter, allowing for the transmission of data at extremely high speeds. The light that travels through the core of the fiber is known as the mode, and the mode is guided along the length of the fiber by total internal reflection.

Fiber optic channels are used for a variety of applications, including telecommunications, data centers, and medical equipment. They are also used in inter-continent network connections using big cables at the bottom of the oceans.

2.1.1 Improvements in fiber optic channels

Forward error correction (FEC) is a technique commonly used in fiber optic communication to enhance the reliability of data transmission. One of the most promising FEC codes used in fiber optics is the low-density parity-check (LDPC) code. LDPC codes are a type of FEC code that has been shown to provide high coding gain and low decoding complexity. They work by adding redundant information to the transmitted data, allowing the receiver to detect and correct errors that may have occurred during transmission.

The optimal performance of LDPC codes is often realized in long-range communication scenarios. These codes excel in scenarios where the signal-to-noise ratio is relatively low, making them a preferred choice for deep-space communication, satellite links, and terrestrial wireless communication over extended distances. However, their efficiency decreases as the channel becomes less noisy or for shorter-range communications, where simpler codes like Reed-Solomon codes may suffice with lower computational requirements.

To enhance their performance, LDPC codes are frequently combined with Convolutional codes [22]. This amalgamation creates a powerful error-correction system known as LDPC convolutional codes. The synergy between LDPC and Convolutional codes capitalizes on the strengths of both codes, providing exceptional error-correction performance across a wide range of communication scenarios. This combined approach leverages the structured and systematic nature of Convolutional codes with LDPC's ability to correct errors efficiently, resulting in an error-correction system that adapts well to different channel conditions. Moreover, LDPC codes are often paired with probabilistic shaping modules, which optimize the distribution of symbol amplitudes to further enhance their performance. However, the downside of this sophistication is the substantial computational complexity, making LDPC codes power-hungry solutions that may not be suitable for energy-constrained applications such as IoT devices or mobile phones with limited battery life.

LDPC codes have been applied in various fiber optic applications, including long-haul communication, where transmission distances can exceed several thousand kilometers. In these applications, LDPC codes have been shown to improve data transmission rates while maintaining high reliability.[23] It is because of both the parity check effects of the LDPC code and the probabilistic shaping functionality of these codes. Probabilistic shaping tries to reduce the occurrence of signals that are vulnerable to noise and mostly uses the signals that are less affected by noise.

Another significant advancement in decoding techniques for fiber optic communication is the use of "Soft decoding" instead of "Hard decoding". Hard decoding refers to the process of making a definitive decision about the transmitted information symbol or bit without considering the reliability of received signals. In other words, it involves making a straightforward "hard" decision, typically binary (0 or 1), based on the received signal's amplitude or phase, without taking into account the probability of error or the quality of the received signal. This approach can lead to errors, especially in cases of noisy or poor-quality channels.

Soft decoding, on the other hand, takes into consideration the confidence or reliability of the received signal when deciding on the transmitted symbol or bit. When a received signal is processed during soft decoding, Log-likelihood-ratios (LLRs) are calculated for each possible bit value (typically 0 and 1) of the transmitted signal. The LLR provides a measure of the probability that each bit value was sent, taking into account the characteristics of the received signal and the influence of noise. The LLR is usually computed as the logarithm of the ratio of the probabilities that the given bit was a 1 to the probability that it was a 0, or vice versa.

Mathematically, the LLR for a bit 'b' is calculated as:

$$LLR(b) = log(\frac{P(y|b=0)}{P(y|b=1)})$$

Where:

'$P(y|b = 0)$' is the probability of receiving the observed signal 'y' assuming that bit 'b' was a 0.

'$P(y|b = 1)$' is the probability of receiving the observed signal 'y' assuming that bit 'b' was a 1.

This allows for more accurate decoding decisions, especially in scenarios where the received signal is subject to noise and other impairments. Soft decoding has been shown to be particularly effective in fiber optic communication systems that operate at high data rates.

One of the key advantages of using LDPC codes in fiber optic communication is that they can operate at a high signal-to-noise ratio (SNR), which is important for fiber optic systems where the signal can be degraded due to various factors, such as dispersion and attenuation. LDPC codes offer high coding gain, which means they can achieve a high SNR even in the presence of noise and other impairments. This makes them ideal for use in high-speed fiber optic communication systems.

In fiber optic communication, the dispersion bound is a fundamental limit to the data transmission rate over a communication channel. Dispersion is the broadening of light pulses as they propagate through an optical fiber, which can limit the maximum data transmission rate over a fiber optic link. The dispersion bound takes into account the fundamental tradeoff between bandwidth and noise and provides a benchmark for the maximum achievable data transmission rate over a given channel.

To improve channel performance closer to the dispersion bound in fiber optic communication, various techniques have been proposed, including the use of advanced modulation techniques, such as quadrature amplitude modulation (QAM) and pulse amplitude modulation (PAM), which enable the transmission of multiple data streams simultaneously over the same frequency band.

In addition to LDPC codes, concatenated inner codes and outer codes have been proposed as promising techniques for improving the reliability of data transmission in fiber optic communication systems. The inner codes are used to protect the data at a finer granularity, while outer codes are used to protect the data at a coarser granularity. This approach allows for more efficient use of the available redundancy, resulting in higher coding gain and improved error correction performance.

Concatenated codes have been shown to be effective in reducing the bit error rate (BER) in fiber optic communication systems. By protecting the data at a finer granularity, they can correct more errors in the received signal, resulting in a lower BER. Outer codes, on the other hand, are used to protect the data at a coarser granularity, providing an additional layer of protection against errors that may not be corrected by the inner codes.

In fiber optic communication, the use of concatenated inner codes and outer codes, in combination with LDPC codes, has been shown to provide significant improvements in data transmission rates while maintaining high reliability.

2.1.2 Benefits of fiber optic channels

The improvements in the final performance and speed of fiber optic channels have led to a range of benefits for users. One of the most significant benefits is the ability to transmit data over long distances without loss of signal quality. This makes fiber optic channels ideal for applications such as telecommunications, where signals must be transmitted over long distances.

Fiber optic channels also offer higher data transfer rates than traditional copper cables, making them ideal for applications where large amounts of data must be transmitted quickly. This is particularly important in data center applications, where high-speed data transfer is essential for the efficient operation of modern computing systems.

Another benefit of fiber optic channels is their immunity to electromagnetic interference. Unlike traditional copper cables, fiber optic channels are not affected by electromagnetic interference, making them ideal for use in industrial environments where electromagnetic interference can be a problem.

2.2 Concatenated Codes

Concatenated Code is a powerful error-correction technique that frequently employs a combination of one outer hard decoder and one inner soft decoder to achieve remarkable Bit Error Rate (BER) reduction. In this approach, the inner soft decoder plays a pivotal role by significantly reducing the BER to a manageable level, typically around 10^{-3}. This inner decoder uses the probabilistic information associated with received symbols to make informed decisions about the transmitted data, which is especially effective in scenarios with relatively high noise or interference. Once the BER is reduced to this level, the outer hard decoder takes over, making a final decision on the decoded data and dropping the error rate to an extremely low level, typically on the order of 10^{-15}. This level of error correction is crucial for demanding telecom applications where data integrity is paramount, ensuring that the transmitted information is highly reliable and can be accurately reconstructed at the receiving end.

Importantly, Concatenated Code is versatile and can be adapted to various scenarios, including lower-power applications. It's not limited to just two layers; it can potentially incorporate more than two different concatenated codes, depending on the specific communication requirements. This flexibility allows designers to tailor concatenated code schemes to the unique needs of different applications, optimizing error-correction performance using different parity bits while managing power consumption effectively. Consequently, Concatenated Code is a valuable tool in modern telecommunications, striking a balance between robust error correction and efficient use of resources, making it suitable for a wide range of practical communication systems.

Fiber optic channels are used to transmit digital data over long distances and are susceptible to various forms of noise and interference that can degrade the quality of the transmitted signal. Concatenated Codes are powerful tool that can be used to mitigate the effects of these disturbances.

One of the key advantages of Concatenated Codes with soft inner code and hard outer code is its ability to provide a high level of error correction without requiring a large amount of redundancy in the transmitted data. This is achieved by dividing the data into multiple levels, each with a different level of redundancy. The redundancy is adjusted based on the level of error correction required for that level, so that the overall amount of redundancy is minimized while still providing a high level of error correction.

In addition to its effectiveness in mitigating the effects of noise and other disturbances in the channel, multilevel coding with soft inner code and hard outer code is also well-suited to the high-speed, high-bandwidth requirements of modern fiber optic channels. The use of multiple levels and adjustable redundancy allows for efficient use of available bandwidth, while the use of error-correcting and error-detecting codes ensures reliable transmission of data.

In conclusion, Concatenated coding with soft inner code and hard outer code is a powerful technique that can be used to improve the performance of fiber optic channels. By providing a robust and efficient mechanism for error correction and detection, multilevel coding can mitigate the effects of noise and other disturbances in the channel, while also providing a high level of reliability and efficiency.

Chapter 3

Polar Codes

It has always been argued whether a more reliable sub-channel can be extracted from the heart of a channel. IN 2008 Arikan introduced this coding scheme for channel polarization that can split a channel into sub-channels with different capacities, some extremely noisy and some almost noiseless channels. As a result, Polar Code is the first proven way to close the capacity gap for discrete memory-less channels and Binary inputs [3]. Polar codes are based on the concept of channel polarization. The idea behind channel polarization is that when a set of binary symmetric channels (BSCs) are combined in a specific way, some of the resulting channels will have a high probability of error (called "frozen" channels). In contrast, others will have a low probability of error (called "reliable" channels). This can be achieved by using a polarizing transform, which is a linear transformation that takes a set of input bits and maps them to a set of output bits. The output bits are then transmitted over the channel and the receiver uses a decoding algorithm to recover the original input bits.

One of the key features of polar codes is that they allow for a significant reduction in the number of error-correcting bits required compared to traditional codes. This is because the polarizing transform can be used to identify the most reliable channels and only those channels need to be used for error correction. Additionally, polar codes have the property that the error-correction performance can be improved by increasing the length of the code. In other words, when the code length approaches infinity, the noisy sub-channels will be extremely noisy and unusable, and the reliable sub-channels will be almost clean and noiseless. the main idea behind polar coding is to use the XOR operation to replicate some of the bits and make them stronger by exposing some other bits to more noise, therefore, making those bits less reliable. For example, as shown in Fig.3.1, the block "G2" puts a replica of U_2 inside X_1 channel, and then the X_1 and X_2 signals are being transmitted through a channel with

AWGN. Now on the decoder side, the signal that originally belonged to U_2 had a dedicated sub-channel plus it was replicated in the other sub-channel as well, On the other hand, U_1 only had part of a sub-channel which makes U_2 channel more robust to noise compared to U_1.

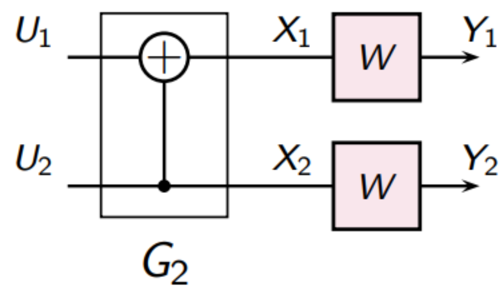

Figure 3.1: Polarizing two channels using XOR blocks. U2 is a more reliable channel that gets replicated and U1 is more exposed to the noise and less reliable [3]

To expand the channel polarization to larger codewords systems, some bits need to be replicated into more than 2 channels. for example in Fig. 3.2 a conventional 4bit polar encoder is demonstrated. As expected u_4 is replicated in 3 out of 4 sub-channels which makes it the strongest bit while u_1 only has a part of one of the sub-channels, sharing it with 3 other bits making u_1 the weakest sub-channel. block "R" is a permutation block that connects the first half of the codeword to the second half to ensure that some bits are spread in all parts of the codeword in different sub-channels, This scheme can be applied recursively and build up different encoders for longer codewords as seen in Fig.3.3. It is worth mentioning that in the conventional method for channel polarization, the weakest channels are usually among the first sub-channels and the last channels are the strongest and most replicated sub-channels. This matter will be discussed later in detail.

3.1 Polar transform

The primary encoding technique utilized in this system revolves around the application of the polar transform. The polar transform is represented by an N*N matrix, which is basically a Kronecker power of a 2*2 triangular matrix. In this particular

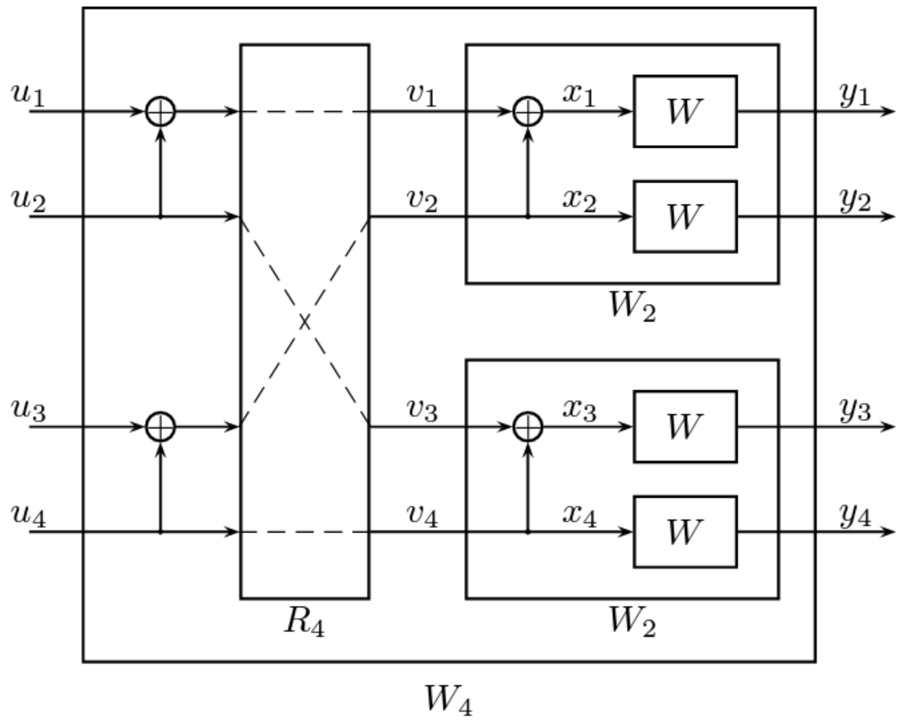

Figure 3.2: Polarizing 4 channels and adding permutations and making bigger block to polarize a channel with 4 sub-channels [3]

step, the goal is to generate a suitable polar transform matrix for the system and subsequently multiply the output of the rate-matcher by this transform matrix. Alternatively, another method of implementing the encoder involves using the original "XOR" circuit to transform the frames, with the inclusion of frozen bits, into polar codewords. Usually, polar coding is done by polar block transform, a 2^N to 2^N matrix that is a Kronecker power of matrix $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. and Kronecker power is defined based on the Kronecker product concept;

$$Kroneckerpower(A, 2) = \begin{pmatrix} A_{11} * A & A_{12} * A \\ A_{21} * A & A_{22} * A \end{pmatrix}$$

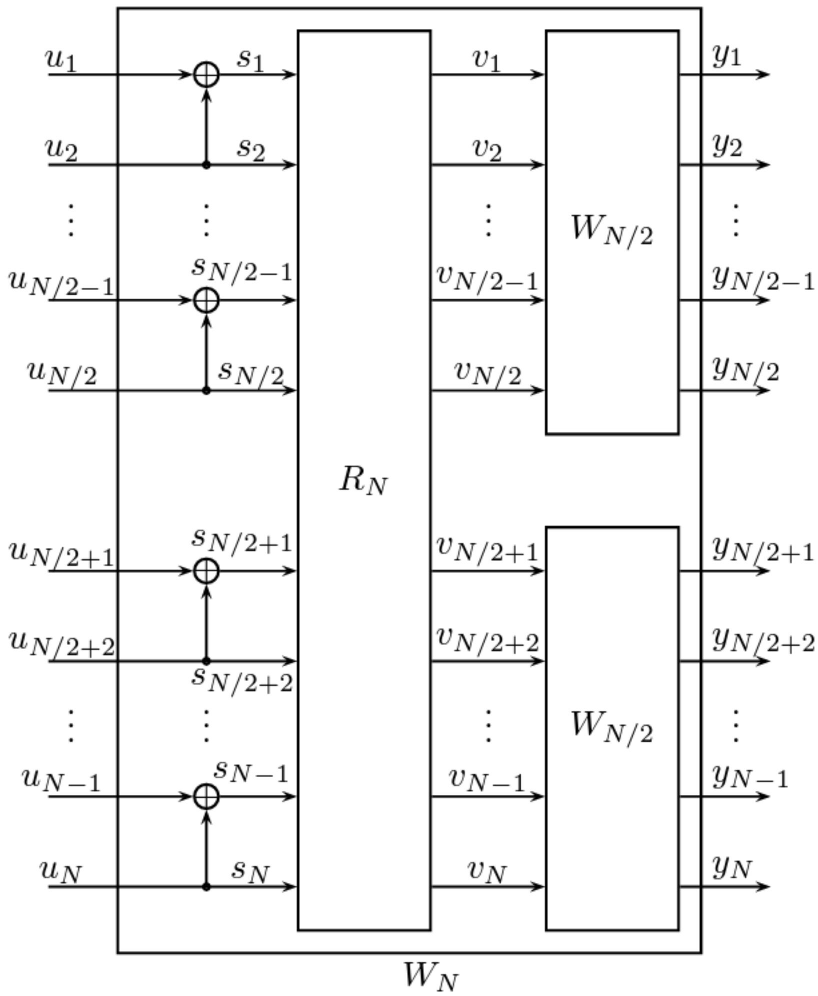

Figure 3.3: recursively scaling up the encoder for polar code to create systems with bigger code length[3]

So based on that the polar transform matrix for the code length of 4 looks is as follows;

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

For simulation purposes, polar transform is usually used for encoding in polar systems. A transform matrix with proper dimensions should be created and multiplied by the code word before transmitting through the channel.

3.2 AWGN Channel

The AWGN (Additive White Gaussian Noise) channel is a commonly used model in communication systems to represent the noise that affects the transmitted signal. It is a theoretical model that assumes the noise added to the signal is white Gaussian noise, meaning it has a flat power spectral density across all frequencies and follows a Gaussian (normal) distribution. Mathematically, the AWGN channel can be represented as follows:

Received Signal (y) = Transmitted Signal (x) + Noise (n)

The noise (n) is assumed to be a zero-mean Gaussian random variable, and its probability density function (PDF) is given by:

$$f(n) = (1/\sqrt{(2\pi\sigma^2)}) * exp(-(n^2)/(2\sigma^2))$$

where f(n) is the probability density function of the noise. σ^2 is the variance of the noise, which represents the power of the noise. The variance (σ^2) is related to the noise power (N0) by the formula:

$$\sigma^2 = N_0 * T$$

where N_0 is the noise power spectral density (power per unit bandwidth). T is the signal duration or symbol time. In practice, the noise power (N_0) is often specified in terms of the noise power per bit ($N_0/2$), assuming a binary transmission system where each symbol represents one bit. In this case, the variance becomes:

$$\sigma^2 = (N_0/2) * T$$

It's important to note that the AWGN channel assumes the noise is independent and identically distributed (i.i.d.) across different time instants and frequency components.

Usually, AWGN is used to study the general improvements that a new encoding or decoding scheme has before it gets tested on more precise channel simulations with specific physical details. In this paper, all of the channels in the simulations are assumed to be AWGN channels for simplicity and we can expect the method that improves transmission reliability in AWGN channels can be beneficial for other channels as well.

3.3 Polar Decoding

So far we established that Polar codes are a class of channel codes that have been shown to have the capacity to achieve the maximum performance for symmetric binary-input discrete memoryless channels. This means that they have the ability to approach the theoretical limits of data transmission over a given channel. In this essay, we will explore the concept of polar codes and discuss the various decoding techniques that have been developed for them.

Several decoding techniques have been developed for polar codes. The most basic decoding algorithm is the successive cancellation (SC) decoding algorithm. This algorithm is based on the idea of using the channel likelihoods to make a decision about each bit in the code. The algorithm starts at the most significant bit and proceeds to the least significant bit. At each step, the algorithm uses the channel likelihoods to decide on the current bit based on the decisions made about the previous bits.

Another popular decoding technique is the belief propagation (BP) decoding algorithm. This algorithm is based on the idea of using channel likelihoods to calculate the probability of each bit in the code. The algorithm starts with an initial probability for each bit and then iteratively updates the probabilities based on the channel likelihoods and the probabilities of the other bits.

There are also several variations of these decoding algorithms that have been developed to improve the performance of polar codes. For example, the list successive cancellation (LSC) decoding algorithm is a variation of the SC decoding algorithm that allows for multiple decoding paths to be considered. Additionally, the successive cancellation list (SCL) decoding algorithm is a variation of the BP decoding algorithm that allows for multiple decoding paths to be considered.

3.4 Rate Profiler

The key feature of polar codes is that they can be designed for different rate profiles to match the specific requirements of the communication system. A rate profile is a characterization of the number of information bits and frozen bits in a polar code. Different rate profiles are designed to match different requirements, such as error correction capability, overhead, and computational complexity. In general, the rate profiler is part of the system that injects zero bits into the data vector in specific positions that later on, ”less reliable” sub-channels are going to transfer those bits. The first job of the rate profiler is to sort the subchannels from the least reliable subchannels to the most reliable ones. Then it chooses the least reliable subchannels to carry the frozen bits and more reliable subchannels to carry the information bits. the ratio between the two groups is depending on what rate we need for our system. based on static or dynamic ratios we can have 3 different types of rate profilers;

The first type of rate profiler for polar codes is the fixed-rate profiler. The fixed-rate profiler is designed to have a fixed number of information bits and frozen bits, regardless of the channel conditions. The advantage of fixed-rate profilers is that they are simple and straightforward to implement, and they do not require channel state information at the encoder. However, the drawback of fixed-rate profilers is that they may not be suitable for all channel conditions and may lead to a large overhead or poor error correction performance.

The second type of rate profiler for polar codes is the adaptive-rate profiler. The adaptive-rate profiler adjusts the number of information bits and frozen bits based on the channel conditions. The advantage of adaptive-rate profilers is that they can achieve better error correction performance by using more frozen bits in poor channel conditions. The drawback of adaptive-rate profilers is that they require channel state information at the encoder, which increases the complexity of the encoding and decoding algorithms.

The third type of rate profiler for polar codes is the variable-rate profiler. The variable-rate profiler allows the number of information bits and frozen bits to vary dynamically during the transmission. The advantage of variable-rate profilers is that they can achieve the best error correction performance by using more frozen bits in poor channel conditions and fewer frozen bits in good channel conditions. The drawback of variable-rate profilers is that they require channel state information at both the encoder and the decoder, which significantly increases the complexity of the encoding and decoding algorithms.[24]

3.4.1 RM vs GA rate profiler

There are two well-known rate profilers for polar codes, the Reed-Muller (RM) rate profiler and the Gaussian Approximation (GA) rate profiler. Both rate profilers have their strengths and weaknesses, and they have been the subject of much research in recent years. Here, we will discuss these rate profilers in detail and compare their performance.

The RM rate profiler is based on a hamming weight of the generator matrix. This approach is used to calculate the mutual information between each bit-channel and the output of the channel. The calculation is performed by finding the channels that carry less reliable information because of mixture with lots of other sub-channels and then freezing them. The rate of each sub-code is determined based on the mutual information between the input and output of the channel. [25]

The RM rate profiler has several advantages over the GA rate profiler. First, it is a simple and straightforward approach that can be easily implemented. Second, it provides an accurate estimate of the channel polarization for each bit-channel. Finally, the RM rate profiler is a deterministic approach, which means that the same rate profile will be obtained each time the calculation is performed.

However, the RM rate profiler also has a disadvantage. It is computationally intensive and requires a large amount of processing power to calculate the mutual information for each sub-channel. Also, it is not customized based on the channel distribution so It has sub-optimum efficiency.

The GA rate profiler is based on the Gaussian approximation of the channel distribution. The GA rate profiler uses a Gaussian approximation to estimate the distribution of the channel. This approximation is used to determine the rate of each bit-channel. The GA rate profiler is a flexible approach that can be applied to a wide range of channels. [26]

The GA rate profiler has several advantages over the RM rate profiler. First, it is a fast and efficient approach that requires less computational power than the RM rate profiler. Second, it is a flexible approach that can be applied to a wide range of channels. Finally, the GA rate profiler provides an accurate estimate of the channel polarization for each bit-channel.

However, the GA rate profiler also has some disadvantages. First, it is an approximation and is not as accurate as the RM rate profiler. Second, the accuracy of the GA rate profiler depends on the accuracy of the Gaussian approximation used in the calculation. Finally, the GA rate profiler is not a deterministic approach, as the result may vary each time the calculation is performed.

In conclusion, both the RM rate profiler and the GA rate profiler have their own strengths and weaknesses. The RM rate profiler provides an accurate estimate of the channel polarization for each bit-channel, but it is computationally intensive and not very flexible. The GA rate profiler is a fast and efficient approach that can be applied to a wide range of channels, but it is an approximation and is not as accurate as the RM. In this research we use RM rate profiler as it is more straightforward and is independent of channel and information situation.

3.5 Successive Cancellation (SC) decoding

Successive cancellation sequential decoding, also known as SC decoding, is a method of decoding error-correcting codes that is used to correct errors in Polar systems. The basic principle behind this method is that it uses a series of algorithms to iteratively decode the received code word until the correct code is found. This method is particularly useful in situations where the error rate is high and traditional decoding methods are not effective.

The first step in successive cancellation sequential decoding is to initialize the decoder with a list of possible code words. This list is typically created by using a generator matrix that is associated with the code being used. The generator matrix is a matrix that is used to generate the code words that are transmitted over the communication channel. The generator matrix is typically chosen such that the code words that are generated are as far apart as possible, making it easier to correct errors. for simplicity, we can imagine this list of code words as an irregular tree, and the process of decoding is basically a search in this tree. Every leaf in this tree is also associated with a possible code word in this system. [figure 3.4]

Once the list of possible code words has been initialized, the decoder begins the process of iteratively decoding the received code word. This is done by comparing the received code word to each code word in the list, and using a metric to determine

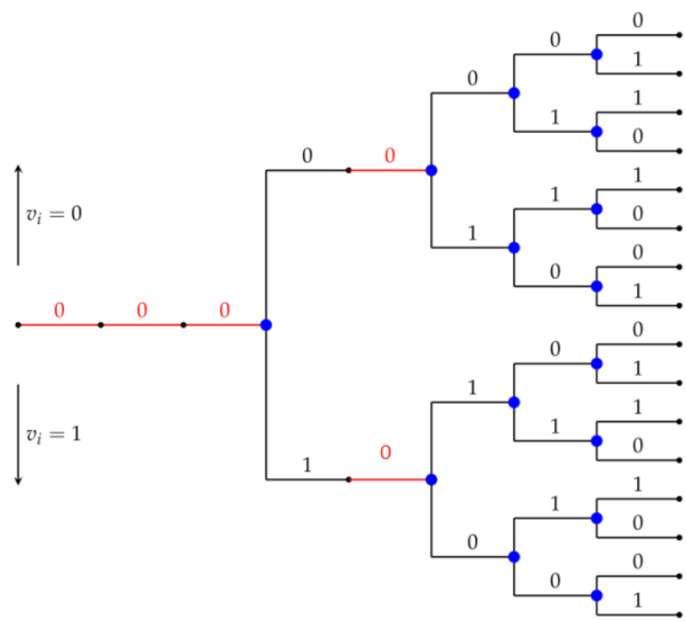

Figure 3.4: CodeWord Tree

the similarity between the two. The code word that is most similar to the received code word is then chosen as the most likely candidate for the correct code word.

Once a candidate code word has been chosen, the decoder then uses an algorithm to iteratively refine the candidate code word. This is done by using a series of mathematical operations to adjust the values of the candidate code word in order to make it more similar to the received code word. The process is repeated until the candidate code word is an exact match to the received code word, or until a certain number of iterations has been reached. In simpler terms, the decoder takes into account how different bits were duplicated across the sub-channels. When a bit is decoded, the soft information of other sub-channels containing that bit is updated, and the influence of that bit is removed from the remaining sub-channels. This allows for the extraction of the remaining bits from those sub-channels at a later stage. The successive cancellation method often utilizes a corresponding tree diagram, accompanied by the relevant formulas, to facilitate the decoding process.

the process of SC decoding is based on two main mathematical functions α and β. α function works with soft information as input and output but β function takes the soft information created by α and generates hard information $\in 0, 1$ which we will learn later that can be used as code words. The application of these two functions is shown in figure 3.5.

It starts from the top of the tree when soft non-binary signals are fed to the tree. then at each layer, the α function in the most left column generates the input needed for the α function on the next level. This iterative procedure continues in each layer. Also, the frame breaks into two sub-frames with the length of half of the previous frame. This continues until we reach a single soft symbol decoding. That's when the β function takes that value, decodes the first bit, and sends it back to the previous level. The new decoded bit and the soft information of the rest of the frame are what α function uses to generate new soft information representing the symbols and this new soft information is later decoded to binary data by the β function.

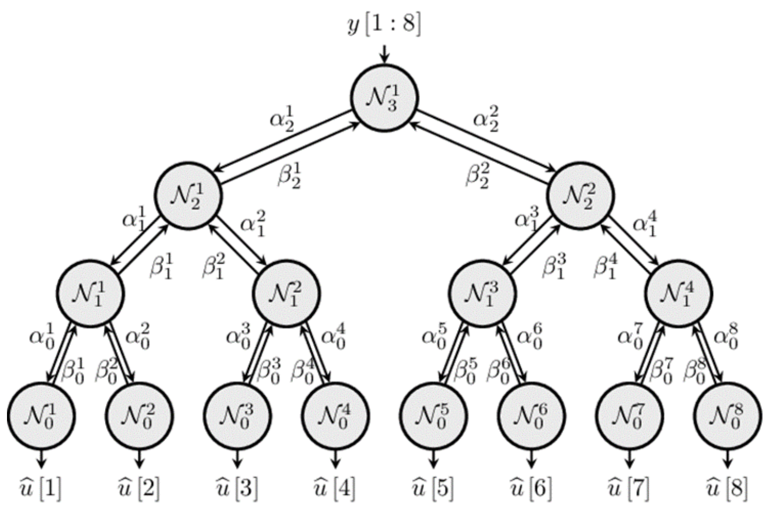

Figure 3.5: successive cancellation Tree

$$\alpha_i^l = \text{sgn}(\alpha_i) \cdot \text{sgn}\left(\alpha_{i+\frac{N_0}{2}}\right) \cdot \min\left(|\alpha_i|, \left|\alpha_{i+\frac{N_0}{2}}\right|\right)$$
$$\alpha_i^r = \left(1 - 2\beta_i^l\right) \cdot \alpha_i + \alpha_{i+\frac{N_0}{2}}$$

where

$$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

$$\beta_i = \begin{cases} \beta_i^l \oplus \beta_i^r, & \text{if } i < \frac{N_o}{2} \\ \beta_{i-\frac{N_o}{2}}^r, & \text{otherwise} \end{cases}$$

One of the key advantages of successive cancellation sequential decoding is its ability to correct errors in situations where traditional decoding methods are not effective. This is because the method is able to correct errors that are distributed throughout the code word, rather than just at the edges. Additionally, the method is able to correct errors that are caused by noise, interference, or other sources of distortion.

Another advantage of this method is that it is relatively simple to implement and requires minimal computational resources. This makes it an attractive option for use in low-power devices such as wireless sensors, where energy consumption is a critical factor.

However, there are also some limitations to successive cancellation sequential decoding. Since the SC decoder keeps only one candidate at a time and if it decodes only one bit wrong, the error spreads and causes the next bits to be decoded wrong, the performance of SC decoders is so limited. there are two ways to prevent that, we can either keep more candidate code words at a time so if we make a mistake in close calls, the other code word with the different bit is also saved and has a chance to be the decoded code word. Another way to improve Polar code decoding is to activate moving back in the code word tree which is studied in the Fano decoder, We will discuss both of these ideas in later parts.

3.6 Successive Cancellation List (SCL) Decoding

Successive cancellation list decoding for polar codes is a widely used decoding algorithm that provides high-performance error correction for channel coding in communication systems. This method of decoding is considered to be one of the most effective methods for correcting errors in polar codes, which are a popular form of error correction codes used in many communication systems.

The process of successive cancellation list decoding is a two-step process, where the first step involves the detection of errors in the data stream, and the second step involves the correction of those errors. In the first step, the decoder identifies the locations of the errors in the data stream by using a metric called the likelihood function. This function is used to estimate the probability of each bit being in error, and the decoder uses this information to determine which bits are in error and which are not.

In the second step, the decoder corrects the errors by using the SC decoding which was discussed in the previous section. This involves the use of a list of potential code words that the decoder tries to match to the data stream. The decoder then selects the code word that provides the best match and uses this pattern to correct the errors in the data stream.

In other words, The SCL decoder creates a list of different potential code words and progressively eliminates less probable code words as more bits are decoded. It prioritizes decoding the more likely code words from the list. The size of the list is finite and can vary depending on the specific system design and application. In essence, if we envision all the possible code words before the polar transform as a tree, the structure would be like the tree shown in Figure 3.4:

3.6.1 Tree search

In the given irregular tree representation, the stages where no branching occurs correspond to the frozen sub-channels. These stages have only a single possible value of 0. On the other hand, the stages where branching occurs correspond to the information bits, which can have either zero or one value. This tree represents all the potential code words after rate profiles and before the polar transform at the encoder. These code words serve as the candidates for the decoder to examine and select one as the output of the polar decoder.

The SCL decoder initiates from the first node on the left side of the tree. It decodes each bit, progresses through the tree, and maintains the best L paths up to that stage in memory. If a stage does not involve branching, there is no elimination of paths. The SCL decoder simply updates the code words in memory and continues forward. However, if branching occurs, only the top L paths are retained in memory, while the remaining paths up to that stage are eliminated.

3.6.2 Path Metric

The method utilized to determine the best paths and rank them according to their proximity to the soft received information is known as the path metric method. This method employs a straightforward formula:

$$\mathrm{PM}_{i_l} = \begin{cases} \mathrm{PM}_{i-1_l}, & \text{if } \hat{u}_{i_l} = \frac{1}{2}\left(1 - \operatorname{sgn}\left(\alpha_{i_l}\right)\right) \\ \mathrm{PM}_{i-1_l} + \left|\alpha_{i_l}\right|, & \text{otherwise} \end{cases}$$

At each stage, the path metric (PM) for the new bits in each codeword is calculated, and this PM is then added to the path metric of the preceding stage. The path metric represents the cumulative metric for the entire path up to that stage.
After calculating the path metrics for all different candidate codewords, they are ranked based on their respective path metrics. The lower the path metric, the better the candidate codeword's alignment with the received information. Subsequently, only the top L candidates with the best path metrics are retained in the list, while the rest are eliminated. This process allows the decoder to focus on the most promising codewords as it progresses through the decoding stages.
One of the key advantages of successive cancellation list decoding is its ability to provide high-performance error correction even in the presence of high noise levels in the channel. This is due to the use of list decoding, which allows the decoder to consider multiple possible error patterns and select the one that provides the best match to the data stream. This results in a much higher error correction rate compared to other decoding algorithms that only consider a single error pattern at a time.

In addition to its high-performance error correction capabilities, successive cancellation list decoding is also highly efficient in terms of its computational requirements. This is because the decoder only needs to consider a small number of error patterns in each iteration, making it much faster and more efficient than other decoding algorithms that require the consideration of a larger number of error patterns.

Another key advantage of successive cancellation list decoding is its scalability.

This means that the decoder can be easily adapted to different communication systems, regardless of the size and complexity of the data stream. This makes it a highly flexible decoding algorithm that can be used in a variety of different applications and communication systems.

Chapter 4

Polarized adjustable Convolutional Codes (PACC)

Polarized adjustable convolutional codes (PACC) are a type of error-correcting code used in digital communication systems. They are a class of convolutional codes that can be adjusted to achieve a desired trade-off between code rate and error correction performance.

The main advantage of adding a convolutional code on top of the polar code is to squeeze the lost capacity because of the frozen sub-channels. By freezing some of the su-channels in polar codes we accepted that there is no data to be transferred through them but convolutional pre-code mixes the frozen channels with adjacent non-frozen channels and tries to send as much information as possible in those less-reliable channels.

PACC uses a polarizing transform to convert a binary input sequence into a sequence of complex-valued symbols, after encoding binary input by a convolutional encoder. The polarizing transform allows for a higher degree of flexibility in the code design, as it can be adjusted to optimize the code rate and error correction performance for a given channel.

In general, in PACC, the Convolutional code as the outer code, and the Polar code as the inner code are used. Or we can consider it a kind of Convolutional code that uses channel polarization advantages to send data through a polarized more reliable channel. the convolutional code used in PACC mostly has the data rate = 1 which means that it doesn't add any data overhead on top of the polar transform.

The decoding of PACC is typically done using a soft-decision Viterbi and SC algorithm, which utilizes the phase information of the complex-valued symbols to improve the error-correction performance. It is worth noting that the error-correction performance of PACC is highly dependent on the choice of the polarizing transform and the convolutional encoder used. Therefore, it is important to carefully design the polarizing transform and convolutional encoder to achieve the desired trade-off between code rate and error correction performance.

Overall, PACC has shown to be effective in improving the error-correction performance in digital communication systems while maintaining a relatively low code rate. However, further research and development are needed to fully realize the potential of these codes in practical applications.

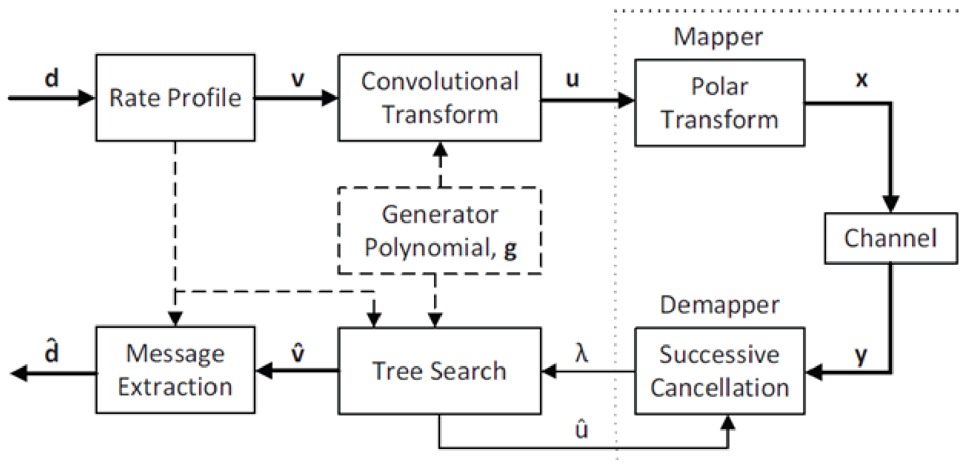

Figure 4.1: Polarized adjustable convolutional code (PAC) system Block diagram

the schematic of a PACC system in both Encoding and decoding is shown in figure 4.3. The main technical details of PACCs include:

Rate profiling: Same as Polar codes in PACC, we need a rate profiler to identify the less reliable channels and freeze them. The rate profiler used in PACC systems is Usually RM and that's what is used in our simulations. In simulation; the rate profiler receives the binary information bits in a frame with the length K (system input) and injects 0s in the less reliable sub-channels positions, and sends the frame with length N (code word length) to the next block.

Convolutional code: The convolution encoding is done to the frame after rate profiling. This block needs a generator that slides through the frame. The output of this block is the result of xor between the generator matrix and the frame bits. we will discuss convolutional codes in a later section.

Polar transform: Same as explained for polar code, we need to polarize the channel and send the convoluted frame through it. The polar transform uses the generator matrix with"$N * N$" dimension. All parts including the polar transform, AWGN channel and Successive Cancellation could be considered as a "polar channel".

Decoding: There are different methods that can decode a PACC. All of which use a decoding algorithm called belief propagation, which is based on the principle of passing information between different sub-streams in order to correct errors. This is the same idea that SC applies to decode the frames. The difference between the main decoding techniques of decoding PAC is the way they search the code word tree. Some of them keep a list of the potential candidates (SCL), some others can move backward on the tree (Fano) and some others break the frame into different parts and decode each part with different methods.

Error Correction Capability: PACCs have a higher error correction capability than traditional convolutional codes, as they are able to correct a larger number of errors in a given data stream.

Implementation: PACCs can be implemented in hardware or software, and are commonly used in wireless communication systems, such as cellular networks and satellite communication systems and now we are studying their application in Fiber optics channels.

Flexibility: PACCs are flexible in terms of their implementation, as they can be adjusted to suit different channel conditions and data rates, making them suitable for a wide range of applications. the convolutional part usually is a rate 1 code but the polar code is totally customizable. The rate in PACC is defined by choosing the best K sub-channels from the total N channels. So it can be easily modified even during the transmission based on the situation of the channel. the code length in polar codes is flexible too, as RM approach can calculate the mutual information for each sub-channel independent from the length of the frame.

Complexity: PACCs are more complex than traditional convolutional codes and it is also more complex than simple polar codes as they require more calculations and memory to polarize the channel and also pass it through a convolutional encoder. Also, different decoding methods have varying levels of complexity. based on how many nodes of the code word tree are re-visited by the decoder, the time and computation complexity can be higher or lower.

Efficiency: PACCs are highly efficient in terms of their error correction capability. They improved the performance of Polar codes, especially for the smaller frame sizes.

4.1 Convolutional codes

Convolutional codes are a type of error-correcting code that is used to transmit digital data over a noisy communication channel. These codes are used in a wide range of applications, including wireless communication, satellite communication, and digital television transmission. The main advantage of convolutional codes is that they can provide a high level of error correction without increasing the data rate or bandwidth required for transmission.

The basic principle of convolutional coding is to add redundancy to the data being transmitted. This redundancy is added in the form of binary multiplication of bits in a pattern that is also known as the generator vector. The parity (redundancy) bits are then added to the data before it is transmitted. When the data is received, the parity bits are used to check for errors and correct them if necessary.

Convolutional codes are characterized by their constraint length and code rate. The constraint length is the number of bits used to calculate the parity bits, and the code rate is the ratio of data bits to parity bits. A higher constraint length and a lower code rate result in a higher level of error correction.

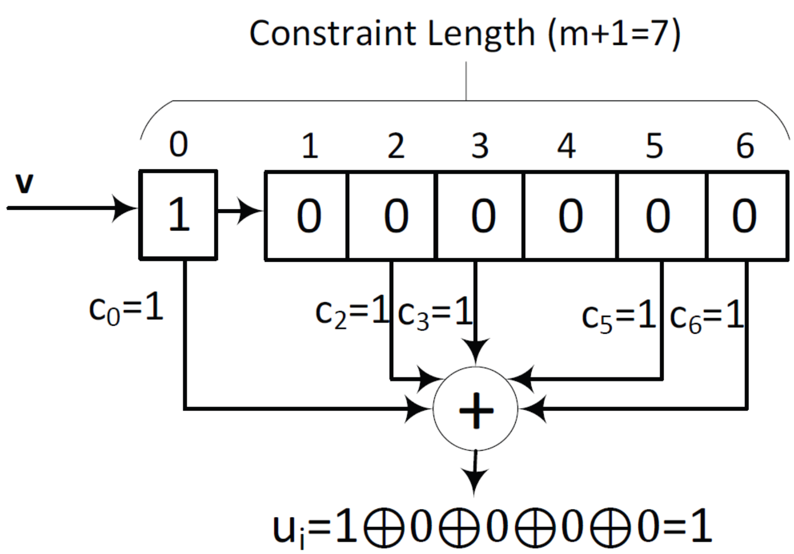

Figure 4.2: A schema of a convolutional encoder logical circuit[4]. The generator vector for this encoder would be [1,0,1,1,0,1,1]

There are several different types of convolutional codes, including systematic codes, non-systematic codes, and recursive systematic codes. Systematic codes are the most common type of convolutional code and are characterized by the fact that the original data bits are included in the transmitted data. Non-systematic codes, on the other hand, do not include the original data bits in the transmitted data. Recursive systematic codes are a combination of systematic and non-systematic codes, and are characterized by the fact that the original data bits are included in the transmitted data, but are also used to calculate the parity bits. [13]

Convolutional codes can be decoded using a variety of techniques, including maximum likelihood decoding, probability decoding, and Viterbi decoding. Maximum likelihood decoding is the most common decoding technique and is based on the principle of finding the most likely sequence of data bits given the received data. Probability decoding is similar to maximum likelihood decoding but takes into account the probability of each possible data sequence. Viterbi decoding is a more efficient decoding technique and is based on the principle of finding the most likely sequence of data bits given the received data and the constraint length of the code.

4.2 Viterbi Codes

PAC Viterbi decoding, also known as the Viterbi algorithm, is a widely used method in digital communications and signal processing for decoding convolutionally encoded data. This algorithm is particularly useful in situations where the received signal is corrupted by noise or interference, as it is able to effectively correct errors and recover the original transmitted data.

The Viterbi algorithm is based on the principle of maximum likelihood decoding, which aims to find the most likely sequence of symbols that was transmitted given the received signal. This is done by considering all possible sequences of symbols and selecting the one that is most likely to have produced the received signal.

The first step in the Viterbi decoding process is to calculate the path metric for each possible sequence of symbols. The path metric is a measure of the similarity between the received signal and the corresponding sequence of symbols. The path metric is calculated using the Euclidean distance between the received signal and the corresponding sequence of symbols, with a small weighting factor applied to correct the effects of noise and interference.

Once the path metric has been calculated for all possible sequences of symbols, the Viterbi algorithm then selects the sequence with the highest path metric as the most likely sequence of symbols that were transmitted. This sequence is then used to decode the original data.

In other words, the Viterbi decoder for convolutional codes, tries to generate all possible code words in the form of a trellis and then finds the codeword with minimum path metric in this trellis[27].

One of the key advantages of the Viterbi algorithm is its ability to correct errors and recover the original transmitted data even in the presence of high levels of noise and interference. This is achieved by considering all possible sequences of symbols and selecting the one that is most likely to have produced the received signal, rather than relying on a fixed threshold or decision rule.

Another advantage of the Viterbi algorithm is its efficiency, as it is able to make use of the inherent structure of the convolutional code to reduce the number of calculations required to decode the data. This makes it well suited to real-time applications, where low latency and high throughput are critical.

4.3 Fano Decoding

PAC Fano decoding is a powerful error-correcting technique used in digital communications systems to correct errors in data transmission. This technique is really dependent on how the parameters in the decoder are defined. by defining them poorly the time and process complexity of this method can be worse than ML decoder and the performance of the decoder in case of Bit error rate or frame error rate can have no advantages over a simple SC decoder. But on the other hand, if the parameters are optimized this decoding method can have the efficiency of the best known decoders (as ML) while keeping the efficiency reasonably low.

The basic principle of the Fano algorithm is to use two thresholds to decide when it is a good time to leave the candidate that is being inspected and move back and correct when the error potentially has happened. It is a sequential decoder just like SC and SCL decoders. Same as previous techniques it looks up into an irregular code word tree, including all possible code words. starting from the root node corresponding to the first bit, it updates soft information about other bits and moves forward.

Two main threshold parameters make sure the decoder is optimized, one of them sets a cap for the path metric corresponding to each bit that is being decoded, and the other one sets a cap on the sum of the path metrics from the beginning of the frame up to each bit (cumulative path metric).

the move-back functionality in the Fano decoder is basically when each of the path metrics exceeds their threshold while decoding a bit. Then the decoder moves back to the last node that satisfied the threshold situation before that final node, but it was not chosen by the decoder as the candidate codeword because it had a higher path metric compared to the other candidate.

The Fano decoder needs to keep track of every node that it visits from the root node moving forward. these records include the path metric of each child of each node that is calculated, even the children that are not chosen at first. Movebacks happen until all the nodes have been visited and the decoder moves back to the root, the node referring to the first bit. then the decoder re-evaluates the thresholds, increases them, and makes the tree search go further.

As was explained before, if the thresholds are too low, the decoder constantly has to move back all the way to the root node, re-adjust the thresholds, and start the decoding from the beginning. This causes to visit nodes on the code word tree multiple times over and over again causing the complexity of the decoder to get marginally greater than SC or SCL decoder.

4.4 SCL for PAC

The successive cancellation list decoder for PAC codes is very similar to the SCL we discussed for the polar decoding system. The only difference is a convolutional decoder placed after the tree search. The PAC encoder and the SCL decoder are explained in detail in figure 4.3.

the components of the Fig. 4.3 block diagram includes;

Rate profiler: As discussed before, the RM rate profiler is the method we use for our PAC encoder. RM rate profiler calculates the mutual information for each sub-channel. This block will get the data as binary bits and inject the "0"s in the weakest sub-channel positions.

Convolutional generator: The generator vector is used to create a generator

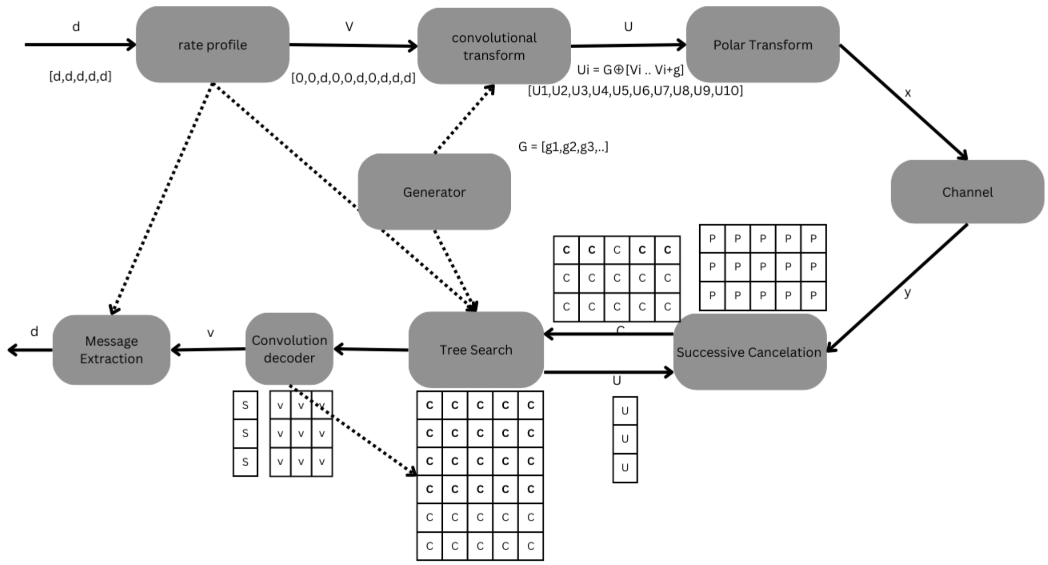

Figure 4.3: PACC System Block Diagram

matrix for the convolutional encoder. For this, the generator vector will be copied on every row of the generator matrix but every row just moves one number forward, creating an all-zero matrix with the generator vector on its diagonal. This kind of matrix is also known as Toeplitz.

Convolutional transform: The convolutional code used in PACC is usually rate one encoder. the length of this block's input and output are the same.

Polar transform: The polar transform here is the same as the polar transform in a regular polar encoder. the polar transform matrix is a Kronecker power or a lower triangular base matrix.

Channel: The Channel would be a normal AWGN channel as was used in polar simulations as well.

Successive Cancellation: In this step, the decoder keeps a record of the soft information that initially was received from the channel. lso this block updates all the soft information for all the candidates according to the new decoded bits that it receives back from the tree search.

Tree Search: As discussed in Polar coding, the tree search process involves navigating through an irregular tree structure to determine the potential children for each candidate code word. When a frozen bit is populated, there is only one child for each candidate. However, in the case of a non-frozen or information bit, the number of candidates under consideration is doubled. Subsequently, the decoder performs a selection step wherein it eliminates half of the candidates based on their path metric, which is determined by their Euclidean distance from the received soft information. The decoder retains only the best-performing candidates while discarding the rest. The final bit of each successful candidate is considered a decoded bit and is then utilized in the subsequent decoding process. These decoded bits are fed back to the successive cancellation list (SCL) decoder, enabling the decoder to decode the next set of bits. This iterative process continues until all the necessary bits have been decoded. By employing the tree search algorithm and feeding back the decoded bits, the Polar decoder effectively explores the irregular tree structure, selects the most favorable candidates, and incrementally decodes the transmitted code word.

Convolutional decoder: In the PAC decoder framework, the convolution decoder performs two crucial tasks. Firstly, it is responsible for maintaining and updating the convolutional state of all candidate code words as new bits are decoded. Secondly, based on the state of each candidate and the freezing status of the bit, the convolution decoder populates the children for the code word and determines the new value of the bits in the code word, and transmits this information back to the tree search algorithm. For example, when a bit is frozen, the convolutional decoder functions like a convolutional encoder by utilizing the existing code state and the new bit (zero), thereby generating the subsequent bit for each code word. Similarly, for non-frozen bits, the decoder considers both possibilities of the bit being zero and one, resulting in the creation of two child code words with distinct states and outputs. the new states of these candidates are stored in memory and The outputs are subsequently sent back to the tree search. The tree search algorithm treats these bits as the next set of candidate bits on the tree and selects the most promising candidate while discarding the others. It is worth noting that when a candidate is eliminated by the tree search, the corresponding convolutional state is also removed from the decoder. Additionally, the convolutional decoder also keeps track of the decoded form of the original data. For instance, if two child code words (representing zero and one) are generated for a particular code word, the decoder maintains information on which child corresponds to which bit, ultimately yielding the output of the decoder as these data bits are generated by the convolutional decoder.

Message extraction: The message extraction step is just extracting information bits and ignoring the frozen bits among decoded bits in the code word.

4.5 Fast SCL Decoding

Fast SCL decoding for polarized adjustable convolutional codes (PACCs) is an important area of research in the field of coding theory. PACCs are a class of convolutional codes that have been shown to have excellent performance in terms of error correction and decoding speed. The goal of this section is to provide an overview of the key concepts and techniques involved in fast SCL decoding for PACCs, and to discuss some of the recent advances in this area.

This Section discusses the development of fast list decoders for polarization-adjusted convolutional (PAC) codes to improve the decoding speed while maintaining error-correction performance. "Hongfei Zhu" in [5] highlights that PAC codes can achieve near-bound performance for a specific code (128, 64) under list decoding, but the large list size required results in excessive latency. To address this issue, the paper proposes fast list decoding algorithms for four types of constituent nodes in PAC codes.

The presented simulation results demonstrate that fast list decoding with three types of constituent nodes can achieve the same error-correction performance as list decoding while reducing more than 50% of the decoding time steps for the code (128, 64). Furthermore, the use of four types of constituent nodes in fast list decoding can further decrease decoding latency with minimal performance degradation.

The section then focuses on defining the four types of constituent nodes for PAC codes based on the message vector v, namely Rate-0 nodes, Rate-1 nodes, Rev nodes, and SPC nodes. For each node type, a corresponding fast list decoding algorithm is provided, along with necessary proofs. 4.4

Rate-0 nodes represent nodes where all bits that are included in them are frozen bits, and the decoding process does not require path splits. Rate-1 nodes correspond to nodes with all information bits in the frame, and path splits follow specific SCL rules. Rev nodes have the first sub-node bits in the frame as frozen bits and the last bit as an information bit. SPC nodes have the first bit as a frozen bit and all the bits after that as information bits.

The paper proposes fast list decoders based on a tree pruning technique that decodes constituent nodes on the top of the subtree without traversing the binary tree. The fast decoding algorithms for Rate-0, Rate-1, and Rev nodes are equivalent to list decoding, while the SPC node decoding is an approximation.

some experimental results indicate that fast list decoding with three types of constituent nodes does not degrade performance, while the inclusion of SPC nodes only causes negligible performance loss. The proposed decoders significantly reduce the number of time steps required for decoding, thereby speeding up the process.

Potential future directions include the exploration of additional constituent nodes to further reduce decoding latency.[5]

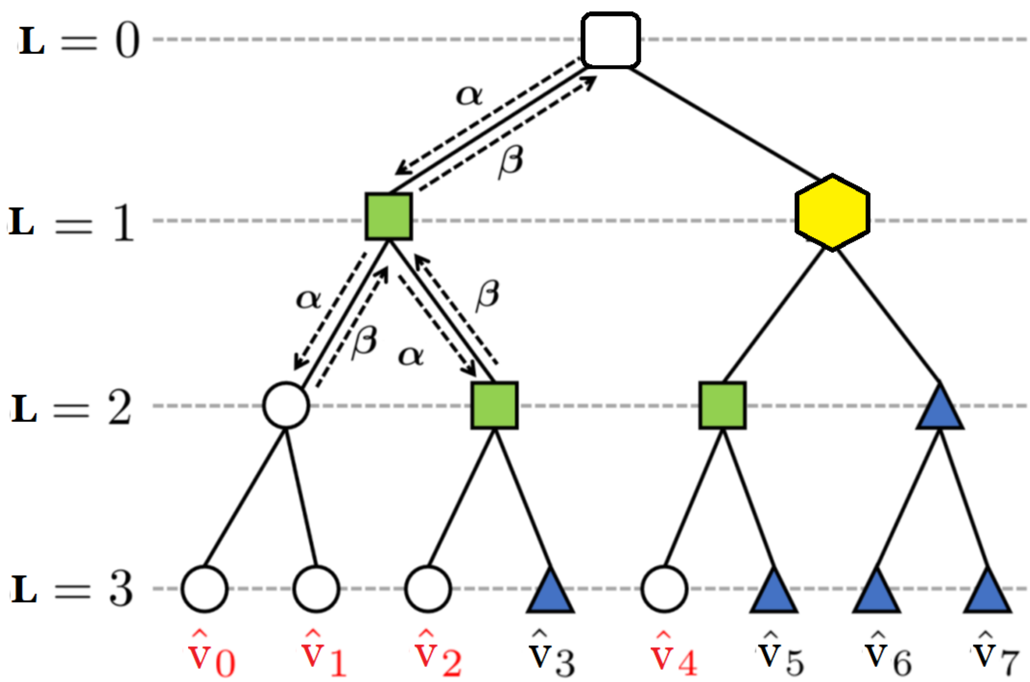

Figure 4.4: Binary tree of the polar code inside the PAC code. White circles are Rate-0 nodes, blue triangles are Rate-1 nodes, green squares are Rev nodes, yellow Octagonal are SPC nodes and white squares are general nodes. [5]

4.6 CRC Aided PAC

CRC codes are sometimes effective in improving the performance of Polar codes and PAC codes. [28] discusses the design and comparison of codes for cyclic redundancy check (CRC)-aided list decoding of tail-biting convolutional codes (TBCCs) and polar codes in the context of short control message communication in 5G and beyond. The paper aims to provide a fair comparison of error performance and decoding complexity between polar and TBCC techniques for a specific case. A TBCC is designed to match the rate of a (512, 43) polar code, and optimal 11-bit CRCs are designed for both codes. The paper analyzes the distance spectra of polar and TBCC codes, highlighting their different distance structures. Adaptive and non-adaptive CRC-aided list decoding schemes are considered.

Simulation results demonstrate that the optimized CRC-TBCC design achieves a lower total failure rate and higher throughput than the optimized CRC-polar design while requiring less computational complexity. "Jacob King" presents design methods for low-rate CRC-polar and CRC-TBCC codes compatible with the 5G PBCH coding standard. Direct comparisons are made between the two codes, analyzing their distance spectra and error rates under simulation. The CRC-TBCC is found to have a superior distance spectrum, faster decoder, and better total failure rate performance compared to CRC-polar codes. [28]

It is noted that the block length of 512 used in this paper matches the block length of the polar encoder in the 5G PBCH code. However, the 5G PBCH polar code has a block length of 864 with a repetition of the first 352 bits from the 512-bit polar code. This repetition is ignored in this paper as it does not improve total failure rate performance. Nevertheless, a well-designed rate-1/20 CRC-TBCC with a 32-bit message and 11-bit CRC could potentially outperform the rate-1/12 CRC-TBCC presented in this paper and better align with the 5G standard.

The problem is that this method and most similar methods that use CRC to improve Polar codes and PAC, use the CRC encoder and decoder as the outer code and Polar and PAC code as the inner code. While the CRC codes can be used in the decoding step to on tree search to guide the tree search.

4.7 PAC SCL limitations and improvements

The minimum Hamming distance is a measure of the error-correcting capability of a code. A code with a greater minimum Hamming distance can correct more errors than a code with a smaller minimum Hamming distance. The convolutional pre-coder is a technique that can be used to increase the minimum Hamming distance of polar codes.

This is done by using a convolutional encoder to pre-code the data before it is passed to the polar encoder. The convolutional encoder can be used to add redundancy to the data, which can help to increase the minimum Hamming distance. As the code rate increases, the minimum distance between code words decreases. This is because the polar encoder is more likely to map the same data to different codewords when the code rate is higher. However, the convolutional pre-coder can help mitigate this loss of minimum distance by making sure that the hamming distance is almost equal for all code words. The length of the convolutional generator can also affect the minimum distance of a PAC code. A longer convolutional generator will produce a code with a greater minimum distance. However, a longer convolutional generator will also increase the complexity of the decoder. In [4] they used an extra convolutional pre-coder to increase the minimum distance, the convolutional pre-coder can also improve the error-correcting capability of PAC codes by protecting the less reliable channels.

Our experiments have shown that both Polar and PAC SCL decoding are more likely to make errors in the early stages of the tree search. This is because the early stages of the tree search are the ones that encounter less reliable channels. One of the weaknesses of PAC is its weakness in protecting less reliable channels. This is because the PAC encoder does not take into account the reliability of the channels when it maps data to code words. As a result, the PAC decoder is more likely to map data to the wrong code words that are transmitted over less reliable channels. As also mentioned in [4] the least reliable sub-channels are the ones that are usually in the first bits of the frame. Improving the reliability of the first bits is important because it can help to reduce the number of errors that are made in the early stages of the tree search. "M. Rowshan" suggested an extra second convolutional pre-coder that can be used to improve the reliability of the first bits. This pre-coder involves more reliable channels, which helps to protect the less reliable information-carrying channels. The second convolutional pre-coder is a promising technique that can be used to improve the error-correcting performance of PAC codes. However, more research is needed to evaluate the performance of this technique in a variety of communication scenarios. figure 4.5

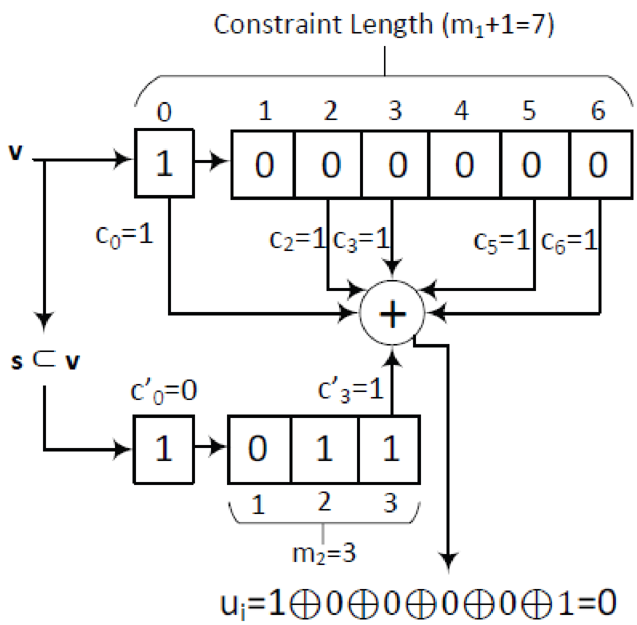

Figure 4.5: Instead of one convolution generator, [4] suggests two different vectors that one of them specifically just protects less reliable channels

Chapter 5

Targeted Parity Protected (TPP) Polar and PAC codes

In this section, we will propose a new PAC system that targets the less reliable information bits that are in the early stages of tree search and their validity is important to decode the whole frame correctly. Then we will add a parity bit for every few of these bits. At the receiver side, knowing the position of the transmitted parity bit, we do the tree search accordingly. In the list of candidate code words if the parity check for the frames shows an error, we can certainly say that there has been a problem in decoding this frame. we might still consider this frame to continue with, but since the CRC bits are different than what they should have been, the path metric for these candidates is greater than other candidates therefore their children have less chance of surviving in the next iterations.

5.1 Polar and PAC code weakness

In the context of polar codes, the objective is to create a balance between weaker and stronger channels. In other words, by increasing the rate of the polar codes, the weak channels become too weak, and errors in decoding these weaker bits will lead to errors in decoding the whole frame in sequential decoders. The conventional approach, as devised by Arikan, typically places the weaker sub-channels towards the beginning of the frame. These weaker sub-channels are more susceptible to errors during the decoding process, and thus, they represent a critical part of the overall system performance. While it is common to assume that these weaker bits are located at the beginning of the frame, their placement can vary without significant impact.

In [4], "M. Rowshan" delves into an approach to enhance the protection and reliability of these initial weaker bits, ultimately leading to a reduction in both the bit error rate and frame error rate. To accomplish this, the researchers introduce an additional component known as a convolutional precoder. This precoder is essentially an extra convolutional code with a large length and a rate of one. The purpose of this precoder is to utilize the more reliable channels to reinforce and strengthen the weaker sub-channels.

By incorporating the convolutional precoder, the overall performance of the polar code, specifically the polarization-adjusted convolutional (PAC) code, is significantly improved. The additional convolutional code acts as a safeguard and aids in mitigating errors in the vulnerable sections of the frame. This approach effectively enhances the reliability of the weaker bits, reducing the occurrence of errors and thereby improving the overall error performance of the system.

5.2 A study on errors in SCL decoder

We conducted an investigation into the common errors encountered by the SCL decoder and discovered that the initial incorrect detection often occurs during the decoding of the first few bits. In other words, when we commence the tree search from the root node, there is a higher likelihood of errors appearing in the early stages rather than in the subsequent stages. And the reason for that is as explained before, the location of the less reliable sub-channels in the first sub-channels.

In various decoding schemes, such as Successive Cancellation List (SCL) and Fano decoding, a common error occurs when the path metric for one of the candidate code words in the code word tree is lower than the path metric for the correct codeword. In the context of SCL, this can be particularly problematic since there is no option to backtrack or move back on the code word tree. As a consequence, if the correct codeword is eliminated from the list of candidate codewords during decoding, the entire frame may be decoded incorrectly. This situation arises because, during decoding, the decoder often encounters code words with better path metrics than the true codeword, especially in the initial part of the code word. This is primarily due to the fact that the first sub-channels (corresponding to the early part of the code word) tend to be weaker, leading to a higher likelihood of finding code words with favorable path metrics compared to the true codeword. To address this issue, decoding algorithms need to carefully manage the list of candidate codewords and ensure that the correct codeword remains a viable candidate throughout the decoding process. Strategies like path metric reordering or pruning can be employed to enhance the decoding performance and reduce the probability of excluding the true codeword from the list.

Additionally, techniques that exploit the polar code's channel polarization properties can be utilized to improve the performance of decoding schemes. These methods aim to prioritize the most reliable channels (sub-channels) early in the decoding process, thus increasing the chances of correctly identifying the true codeword.

To illustrate this, let's consider a real example from our simulations with a list size of 16. After decoding just 32 bits of the frame, we would have 16 different codewords, each with a length of 32, and assigned different path metrics. During our observation, we noticed that the presence of noise in the signal, similar to other encoding schemes, causes the codeword that has traversed the channel to resemble at least one other wrong codeword instead of the truly transmitted codeword. Consequently, the decoder makes an error and incorrectly identifies the codeword. In this decoding method, it is the responsibility of the path metrics to indicate the proximity of a candidate codeword to the expected soft information.
As you can see the original true transmitted codeword "U" on top, some possible candidate codewords considered in the decoder in the middle (C_1, C_2, C_3) and the PM assigned to these candidate codewords on the bottom;

$$U = \{1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, ...\}$$

$$C_1 = \{1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, ...\}$$

$$C_2 = \{1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, ...\}$$

$$C_3 = \{0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, ...\}$$

$$.$$

$$.$$

$$.$$

$$C_L = \{...\}$$

$$PM = \{2.0618, 3.9538, 0.8555, ...\}$$

Obviously, the PM for candidate C_3 is much less than candidate C_1 (the true codeword) so It the decoder keeps choosing the children populated by candidate C_3 over candidate C_1. These samples were taken from a PAC code system which has a convolutional outer code extra compared to Polar code system. That's why the codewords look so different but in general, for the polar codes the correct codeword and incorrect codeword with low PM are similar to each other.

5.3 New SCL approach using CRC

To prevent the occurrence of the mentioned situation, we can employ CRC (Cyclic Redundancy Check) bits to filter out codewords that have only a few incorrect bits but possess a lower PM than the correct codeword. To implement this, the frame is divided into two parts: a shorter, more vulnerable first part where errors in decoding commonly occur, and a secondary part consisting of more reliable sub-channels with more certain decoded bits. In the first part, a few of the frozen bits are replaced with CRC bits instead of all being zeros. To achieve this, a slight modification is made to the rate profiler component. Instead of selecting K sub-channels out of N sub-channels for information bits, it chooses the most reliable "K + (length of the CRC)" bits out of the N bits. The addition of CRC bits can be done before the rate profiler, and the CRC bits are injected into the middle of the information bits, where they will later be positioned at the final bits of the first part of the frame. For example,

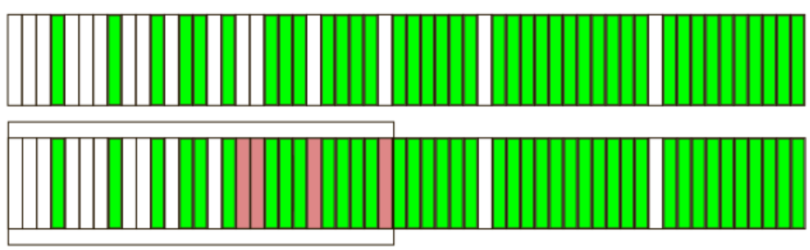

Figure 5.1: Frame diagram for bits in transmitted polar codeword, green bits are information bits, white bits are frozen bits and red bits are CRC bits.

let's consider a polar system with N = 128, K = 100, and CRC length = 4 bits, where the length of the first part of the frame is 32. The rate profiler is instructed to choose the best 104 sub-channels out of 128 for the information bits. Then, within the first 32 sub-channels (including frozen and information bits), the number of frozen and information bits is calculated. Assuming there are 18 information channels and 14 frozen bits among them, it is determined that the CRC bits should be placed in the last 4 sub-channels of the information channels (15-18). Thus, the location for injecting the CRC bits is determined, and these CRC bits reference all the information bits that come before them. Moreover, by increasing the number of information bits in the rate profiler for CRC bits, Basically, some there are some frozen bits that are replaced with CRC bits, without affecting the code rate of the entire system.

On the decoder side, the decoder consists of two parts. The first part decodes the CRC-aided section of the frame and then passes the decoded information bits, including the CRC bits, for evaluation. Only the candidates with the least amount of errors are allowed to continue in the second decoder, while the rest are filtered out. This way, numerous candidates that mistakenly have a low path metric are excluded from the list, and incorrect candidates that happen to possess the correct CRC bits will have a sufficiently high PM. As a result, their children will be at a disadvantage compared to the correct candidate and will be filtered out at a later stage. Each of the first and second decoders follows the Successive cancellation procedure independently. The formulas for SC decoding have been explained in section 3.4 in detail. The procedure for selecting the best candidates between the first and second decoders plays a crucial role in improving the overall performance of the decoding process. When considering both Frame Error Rate (FER) and Bit Error Rate (BER), it becomes essential to strike a balance between maximizing the likelihood of correctly decoding the entire frame and achieving a low bit error rate.

To achieve this, the decoder may retain a few of the candidate codewords with errors in their Cyclic Redundancy Check (CRC) information. The decoder knows that these candidate codewords are not the correct ones, but they have better path metrics than other candidate codewords, making them promising candidates for delivering a good bit error rate during decoding.

The decoder follows a specific procedure to select the best candidates from the outputs of first Successive Cancellation (SC) decoder. First, the candidates are sorted based on the number of errors in their CRC. This step helps prioritize codewords with fewer CRC errors, as they are more likely to be the correct ones. Next, within each group of candidates with the same number of CRC errors, they are further sorted based on their path metrics. Sorting based on the path metric allows the decoder to prioritize codewords that have better overall likelihood of being correct, considering both the CRC information and the channel conditions. Once the candidates are appropriately sorted, the decoder selects the "L" best candidates from the sorted list. The value of "L" depends on the specific requirements of the decoding scenario and can be adjusted to balance FER and BER performance also hardware limitations. By implementing this candidate selection procedure, the decoder increases the probability of identifying the correct codeword while still maintaining a competitive bit error rate. This approach takes advantage of the path metrics to choose promising candidates and employs the CRC information to further refine the selection process, resulting in improved decoding accuracy. Overall, this selection strategy contributes to enhancing the performance of decoding schemes, making them more robust and efficient in achieving both low FER and BER, which are essential factors in reliable and high-throughput communication systems.

5.4 Dynamic Length SCL decoder

The storage requirements of the Successive Cancellation List (SCL) decoder are influenced by both the length of the codewords and the size of the list used in the decoding process. The list size, denoted as "L," represents the number of candidate codewords that are retained at each decoding stage. While increasing the list size can potentially enhance decoding performance, it also leads to a proportional increase in memory requirements. In the SCL decoder, to handle a list size of L and codeword length of N, two main types of memory are needed. The first memory stores the soft information for the codewords, and the second memory stores the codewords themselves. Each of these memories has a size of N*L.

Throughout the decoding process, both memories are preserved and utilized. Initially, when decoding the first bit of a codeword, the remaining N-1 memory spaces for each codeword in the list remain empty and unused. However, it is crucial to retain these empty spaces because they will be needed in subsequent decoding steps when decoding the subsequent bits of the codewords. As the decoder proceeds, it generates additional candidate codewords through the process of successive cancellation. These new candidate codewords are then added to the list, and their soft information and codewords are stored in the respective memories. The list is gradually expanded with each decoding step, and the decoder maintains all the information needed for each candidate codeword until the decoding process is complete.

The Idea behind Dynamic Length SCL is that since we have more memory than needed at the beginning of the decoding, that memory could be used to increase the number of candidates that are being investigated in the SCL decoder. In other words, we can increase the value of L at the beginning of the decoding to use all the storage for performance purposes and as we go forward again we switch it back to the value of design. For example, if we have a system with N = 128 and L = 16, the storages with

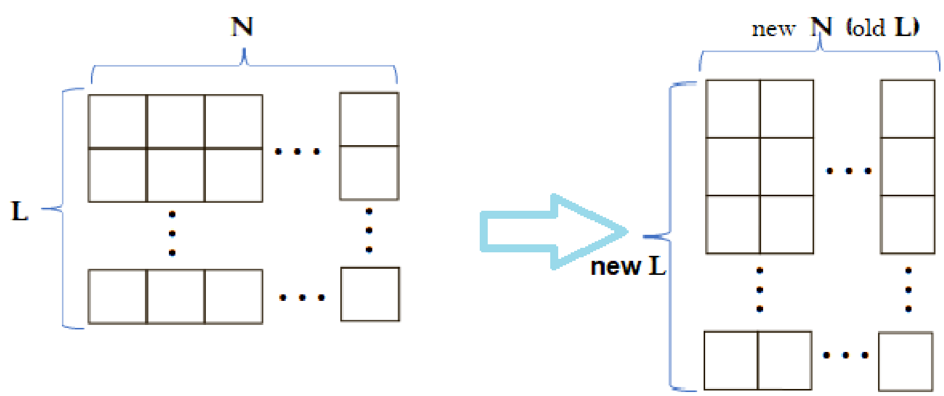

Figure 5.2: Dynamic list successive cancellation Storage allocation

the size of 16*128 are designed but what we can do is to increase the L = 128 for the first 16 bits that are being decoded to use all the storage and leave none of it unused at any time. Sure there are some storages that will expand more when we increase the list size since they are only depending on the size of the list. An example of these memories is the memory that keeps the state of the convolutional code in the PAC code. But still, by increasing the code length, these memories would be negligible.

5.5 TPP and Dynamic SCL together

The proposed method, termed "CRC-aided dynamic SCL for polar codes," combines the two previously explained techniques. In this approach, the decoding process starts by increasing the number of codewords being investigated in the initial part of the frame. Instead of limiting the list size as in traditional SCL, a larger number of candidate codewords is considered in the first decoder. As the decoding progresses, the decoder employs the CRC information to identify the best candidate codewords that either have no errors or fewer errors in their CRC. These promising codewords are then carried forward to the next decoder, while the others are discarded, resulting in a dynamic reduction of the candidate list size. By utilizing CRC information to select the most reliable candidate codewords, the method efficiently narrows down the list of candidates as the decoding proceeds. This dynamic selection process ensures that only the most promising codewords are retained, reducing the computational burden associated with maintaining a large list throughout the entire decoding process.

The advantage of this CRC-aided dynamic SCL approach lies in its ability to strike a balance between decoding performance and computational complexity. It starts with an expansive exploration of candidate codewords, allowing the decoder to take advantage of the inherent diversity provided by a larger list size initially. As the decoder acquires more information through CRC checks, it narrows down the list to focus on the most likely codewords, improving the efficiency of the decoding process. By combining these two methods, CRC-aided dynamic SCL for polar codes aims to achieve enhanced error-correction capabilities while maintaining manageable computational complexity. This approach is especially relevant for polar codes, where SCL decoding can benefit significantly from leveraging CRC information to guide the selection of candidate codewords and optimize the decoding process.

An example of this system could be a system that has N=128, K=100, L=16. For the first part of the frame that is more protected, the number of protected bits is 32, L (list size) for the protected part is 64 codewords, CRC bits = 4 bits. Now in the encoder same as in the previous example, the CRC bits are injected in the middle of the information bits. but on the decoder side, first, an SCL algorithm with L = 64 looks for the best possible candidates in the first 32 bits of the codeword. Of course, by the best candidates, we use the same criteria as TPP as we discussed in section

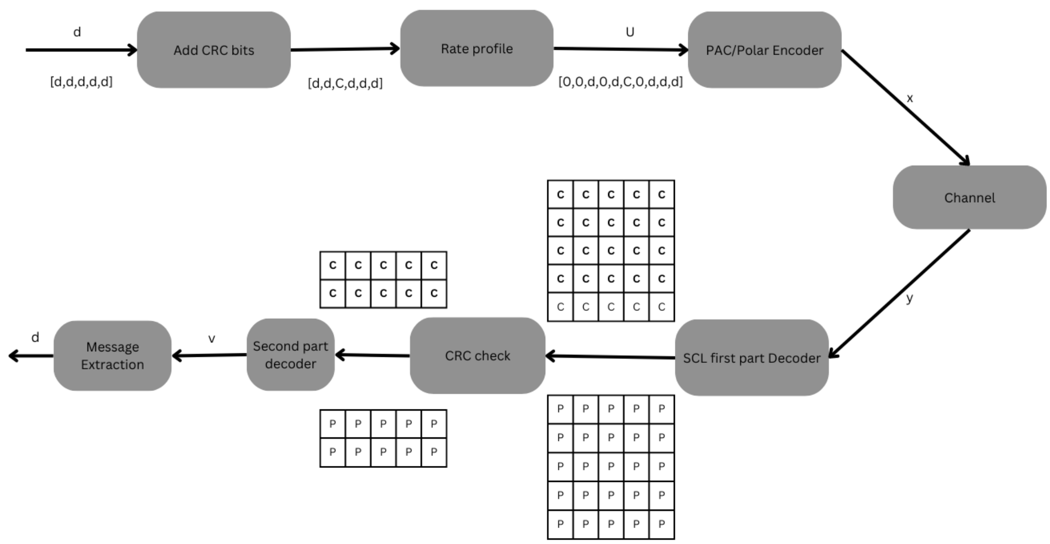

Figure 5.3: CRC Aided Dynamic List size SCL decoder

5.3. After that, the CRC bits which are the last 4 bits of the information bits in the first part of the frame are checked. The candidates that have the least error will be passed into the next decoder and from there the next nodes of the tree are being searched but this time with List size = 16. The second SCL decoder continues the decoding with the new list size and finishes the decoding until only one codeword is left and that's the chosen codeword.

5.6 TPP for PAC

The convolutional code in the encoder is added between the Rate profile and Polar transform and the decoding of convolutional code is a part of the tree search in both first and second decoders where we keep the convolutional status for each candidate and based on the status and the candidate's codeword, we can decode what that candidate is referring to. The CRC-Aided Dynamic SCL method, which was initially introduced for polar codes, is also applicable to Polarization-Adjusted Convolutional (PAC) codes. In fact, this new SCL approach can potentially yield even greater improvements for PAC codes compared to polar codes.

In the context of PAC codes, the convolutional code has a significant impact on the CRC-Aided Dynamic SCL method, enabling further enhancements. When a codeword is transmitted through a noisy channel, some bits in the polar codeword may be incorrectly interpreted by the decoder. This typically occurs when one codeword is mistaken for its adjacent codewords, causing potential errors. However, the convolutional code in PAC codes is designed to increase the Hamming distance between information bit frames associated with adjacent codewords. By doing so, when the CRC is checked for an information bit frame, the convolutional code creates a greater separation between these frames, leading to a higher probability of eliminating the most likely errors.

The convolutional code's ability to increase the Hamming distance between adjacent codewords contributes to the robustness of the CRC-Aided Dynamic SCL method for PAC codes. This added separation reduces the chances of misinterpretation of codewords, enhancing the accuracy of error detection through CRC checks. By leveraging the CRC-Aided Dynamic SCL method along with the advantageous properties of convolutional codes in PAC codes, the decoding process becomes more efficient and reliable. The method capitalizes on the error-correction capabilities of the convolutional code, combined with the dynamic selection of candidate codewords based on CRC information, resulting in improved decoding performance. In general, the CRC-Aided Dynamic SCL method is not limited to polar codes and is well-suited for PAC codes as well. The convolutional code's impact on PAC codes allows for increased Hamming distance between adjacent codewords, leading to more effective error detection through CRC checks. By exploiting these features, the CRC-Aided Dynamic SCL approach can optimize the decoding process and enhance the error-correction capabilities of PAC codes in noisy communication channels.

5.7 Complexity of TPP Dynamic SCL decoder VS Similar decoding methods

The Best practice state of the art decoding scheme for PAC codes is known as the Fano decoder. But although the fano decoder might have advantages in terms of lower FER and BER, it is generally considered to be less efficient than the Successive Cancellation List (SCL) decoder for PAC in terms of computational complexity. One of the main reasons for this is the Fano decoder's exponential time complexity as it searches for the optimal decoding path through the entire tree. The Fano algorithm explores multiple paths, but a single path at a time, and uses backtracking, which can lead to the re-visit of some nodes multiple times, significantly increasing the decoding time. On the other hand, the SCL decoder visits each node at most once and only moves forward, reducing the time complexity and improving the overall decoding efficiency.

For instance, consider a PAC(128,99) code at a signal-to-noise ratio (SNR) of 4. In this scenario, the average number of re-visited nodes encountered by the Fano decoder is remarkably high at 6569 nodes. This large number of re-visited nodes is a clear indication of the Fano decoder's inefficiency and its tendency to backtrack and reevaluate previously explored paths [6].On the contrary, the SCL decoder with a list size of L=16 exhibits significantly better performance in terms of computational complexity for the same PAC(128,99) code at SNR = 4. The SCL decoder, with its forward-only traversing strategy, visits approximately 1600 nodes only once during the decoding process. This reduced number of visited nodes highlights the SCL decoder's ability to efficiently explore the decoding tree and make informed decisions without resorting to extensive backtracking, ultimately resulting in a more computationally efficient decoding process. In the Fano Algorithm, reducing the code-rate decreases the number of visits during decoding due to a simpler tree, but it also limits design options. Despite the reduction in visits, the decoding process can still revisit thousands of nodes, which may lead to increased complexity. Fig.5.4

Another factor that contributes to the Fano decoder's higher complexity is its sensitivity to the parameter Δ. The choice of Δ affects the search strategy of the Fano decoder, and while selecting a smaller Δ can potentially improve error correction performance, it leads to increased computational complexity. In contrast, the SCL decoder does not face such sensitivity to a specific parameter, making it more robust and easier to optimize for a given coding scheme.[29]

Moreover, as synthetic channels become more reliable at high Eb/N_0 regions, the average computational complexities of both decoders tend to converge to the conventional SC decoding complexity. However, the Fano decoder does not provide a significant advantage in such regions, and its computational complexity remains high due to its inherent search-based approach. On the other hand, the SCL decoder's low

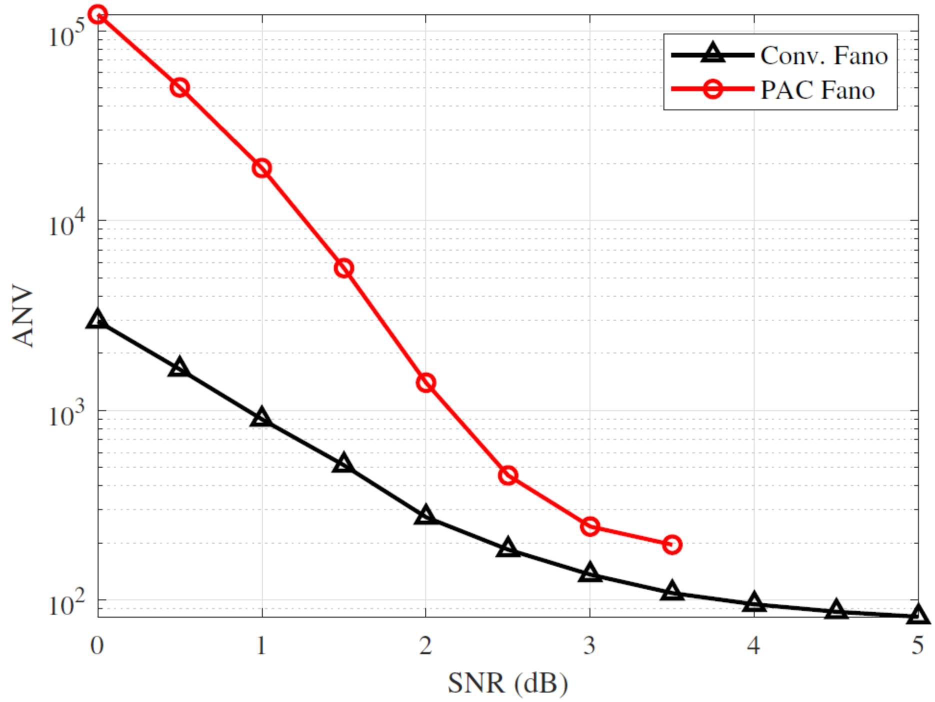

Figure 5.4: Average number of visits for PAC(128,59) and convolutional Fano decoder for the same rate.[6]

level of complexity persists across various Eb/N_0 regions, making it a more favorable choice for PAC decoding when considering complexity.

In practical scenarios, the difference in computational complexity becomes even more pronounced when comparing the two decoders for larger codes and higher SNR values. The Fano decoder's exponential growth in complexity becomes a major drawback, making it less suitable for real-time applications or hardware implementations where decoding speed is critical. In contrast, the SCL decoder's linear complexity with respect to the code length makes it more scalable and efficient for larger codes, allowing for easier adaptation to various communication systems with diverse requirements. Overall, due to its exponential time complexity, sensitivity to parameter Δ, and inefficiency in handling large codes, the Fano decoder is generally considered inferior to the SCL decoder for PAC decoding in terms of complexity.

TPP Dynamic SCL decoding is designed to improve error correction performance for Polar codes by incorporating a CRC and Dynamic list size to assist in making early decision pruning. Despite the added benefit of the CRC, the computational complexity of this method remains comparable to that of the Conventional SCL decoding method i.e., $O(NlogN)$[29]. The reason for this similarity lies in the fundamental nature of both methods, where no backtracking or revisiting of nodes is required.

Chapter 6

Results

6.1 Polar codes with TPP

Arikan's polar code is highly effective with replicating bits to reduce errors. So, if we replace only some frozen bits with cyclic redundancy check (CRC) bits without incorporating dynamic successive cancellation list (SCL) decoding, the system's performance is expected to deteriorate. fig 6.1

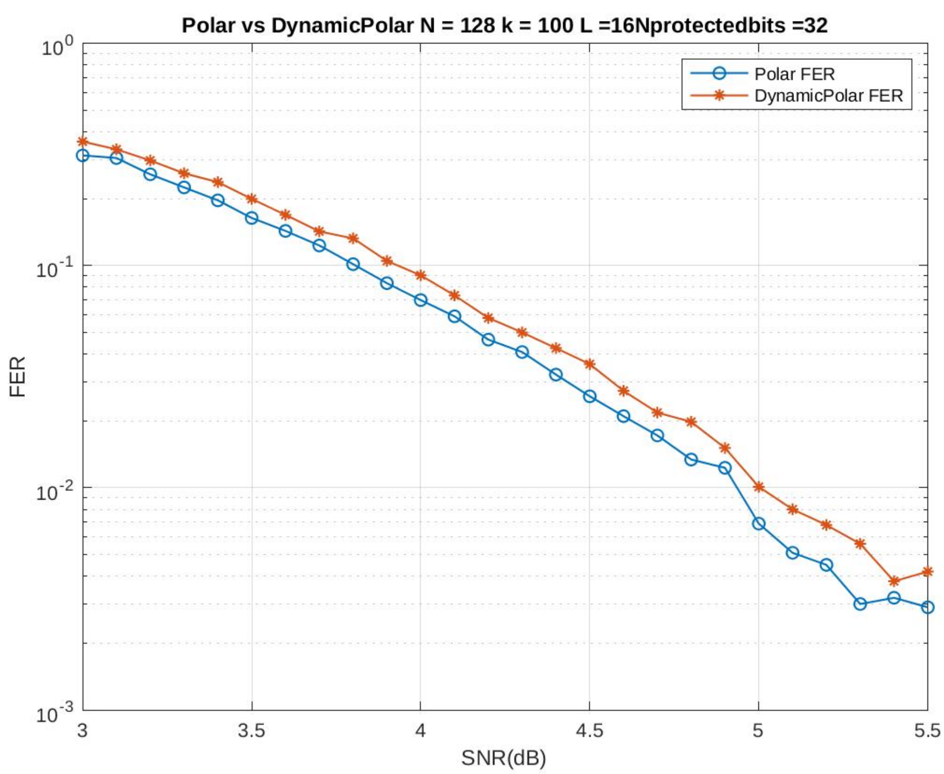

Figure 6.1: Frame error rate for Polar code vs CRC Aided Polar code without applying Dynamic SCL, N=128, k=100, L=16, N for protected bits = 32, CRC bits = 4 bits

6.2 Polar codes with TPP and Dynamic SCL

By combining the Targeted parity protection (TPP) algorithm with Dynamic Successive Cancellation List (SCL) decoding, a significant enhancement in the performance of Polar Successive Cancellation List (SCL) codes can be achieved. For instance, when considering a high-rate polar code with a specific configuration, such as a polar(128,100) code, this joint approach has demonstrated an improvement of up to 0.4 dB in terms of signal-to-noise ratio (SNR). The improvement in performance is achieved through a replacement of only four frozen bits with Cyclic Redundancy Check (CRC) bits and with no reduction in the rate of the code. fig.6.2

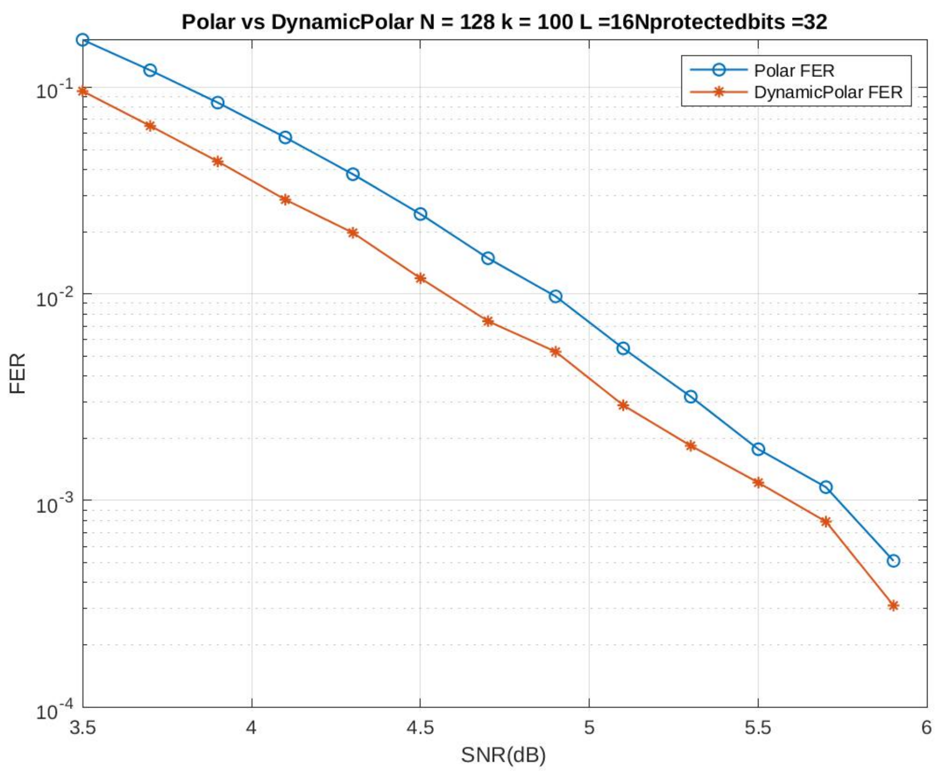

Figure 6.2: Frame error rate for Polar code vs CRC Aided Dynamic SCL for Polar code with N=128, k=100, L=16, L protected part=64, N for protected bits = 32, CRC bits = 4 bits

The Targeted parity protection (TPP) method with Dynamic SCL is effective for higher codeword sizes as well. as you can see in fig. 6.3 the performance of SCL for

Polar codes is improved by almost 0.5 dB when increasing the codeword size to 256 bits instead of 128.

In the case of the polar(256,180) system, the number of cyclic redundancy check (CRC) bits remains constant, as increasing the CRC bits may not necessarily lead to a further improvement in the system's performance. The primary focus for enhancing the system's error correction capabilities lies in appropriately adjusting the number of bits in the first block, especially as the code length increases. As the code length increases, the number of frozen bits in the first block also increases. To protect a larger number of information bits in the first block, it becomes essential to adjust the number of bits in the first block accordingly. This adjustment involves strategically increasing the number of bits in the first block to ensure more information bits can be effectively protected and accurately decoded during transmission.

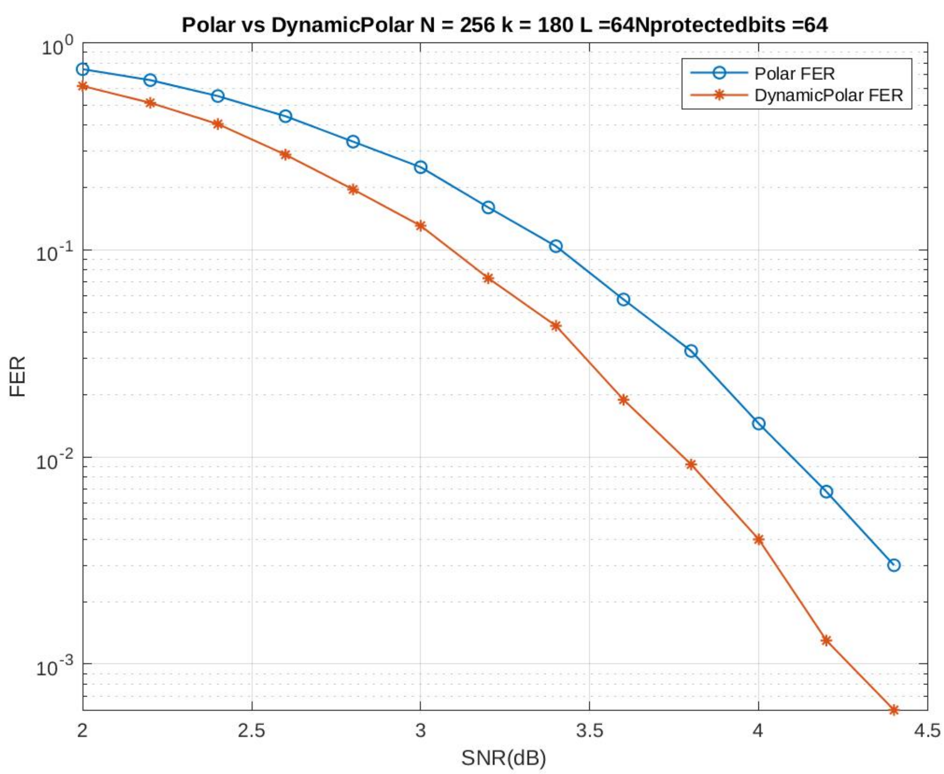

Figure 6.3: Frame error rate for Polar code vs CRC Aided Dynamic SCL for Polar code with N=256, k=180, L=64, L protected part=256, N for protected bits = 64, CRC bits = 4 bits

The improvement in the performance of SCL is for both Bit error rate and Frame error rate. As it is demonstrated on fig.6.4 the polar decoder with TPP and dynamic SCL has almost 0.4 dB improvement over polar code with same rate and list size but the normal SCL decoder.

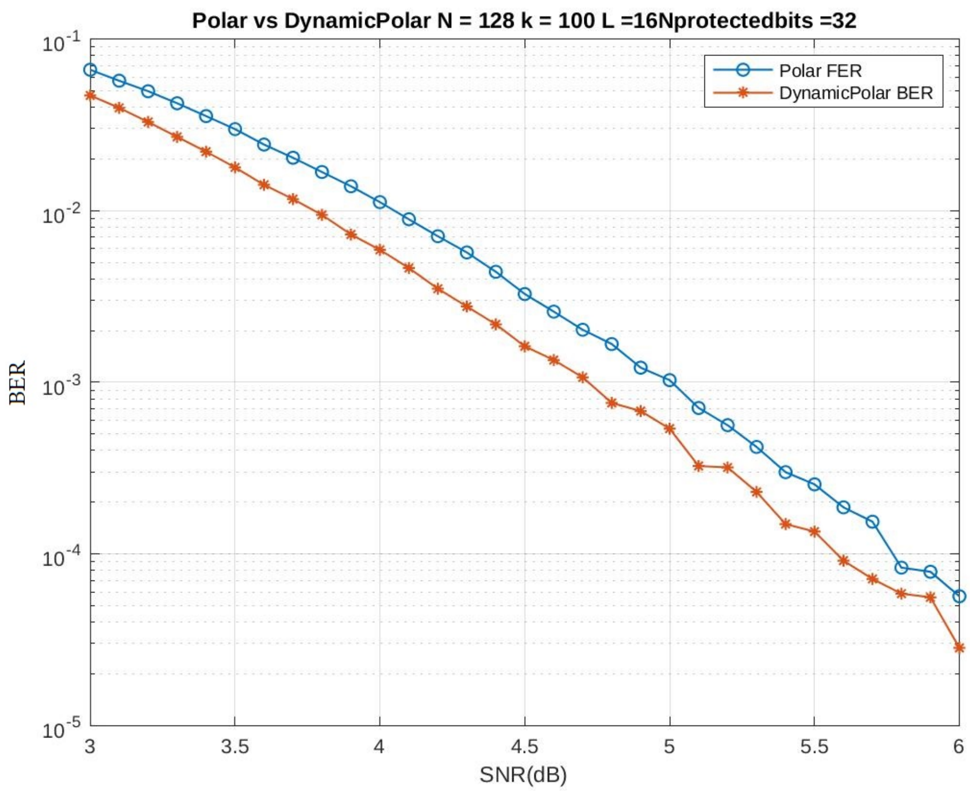

Figure 6.4: Bit error rate for Polar code vs CRC Aided Dynamic SCL for Polar code with N=128, k=100, L=64, L protected part=32, N for protected bits = 64, CRC bits = 4 bits

6.3 PAC with TPP

Same as Polar codes TPP without Dynamic SCL cannot improve the performance in the case of SNR, as the SCL is already in best practice. the Fig. 6.5 demonstrates that the TPP has a little worse performance than conventional SCL for the PAC(128,100) with the list size = 16 and number of the CRC bits = 4.

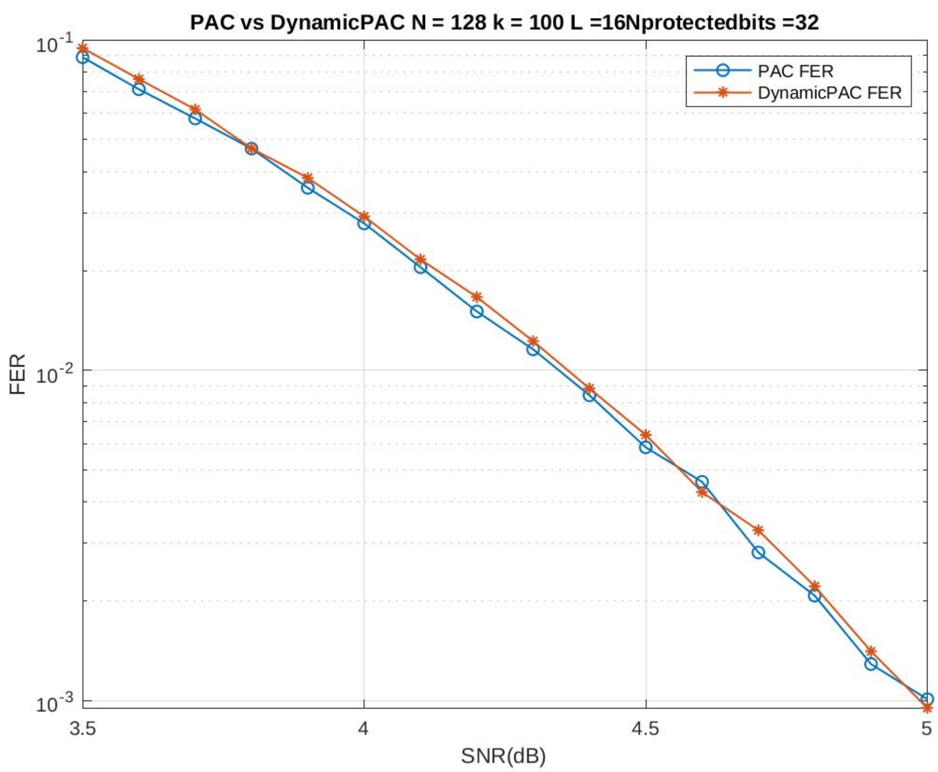

Figure 6.5: Frame error rate for PAC vs CRC Aided PAC without applying Dynamic SCL, N=128, k=100, L=16, CRC bits = 4 bits

6.4 PAC with TPP and Dynamic SCL

Targeted parity protection concatenated with Dynamic SCL works best for PAC. it's because convolutional code in PAC mixes most of the bits in the bits in first block and the CRC bits can protect more bits compared to the Polar code. this leads to the improvement of performance up to 0.6 dB for PAC codes compared to the conventional SCL fig. 6.6

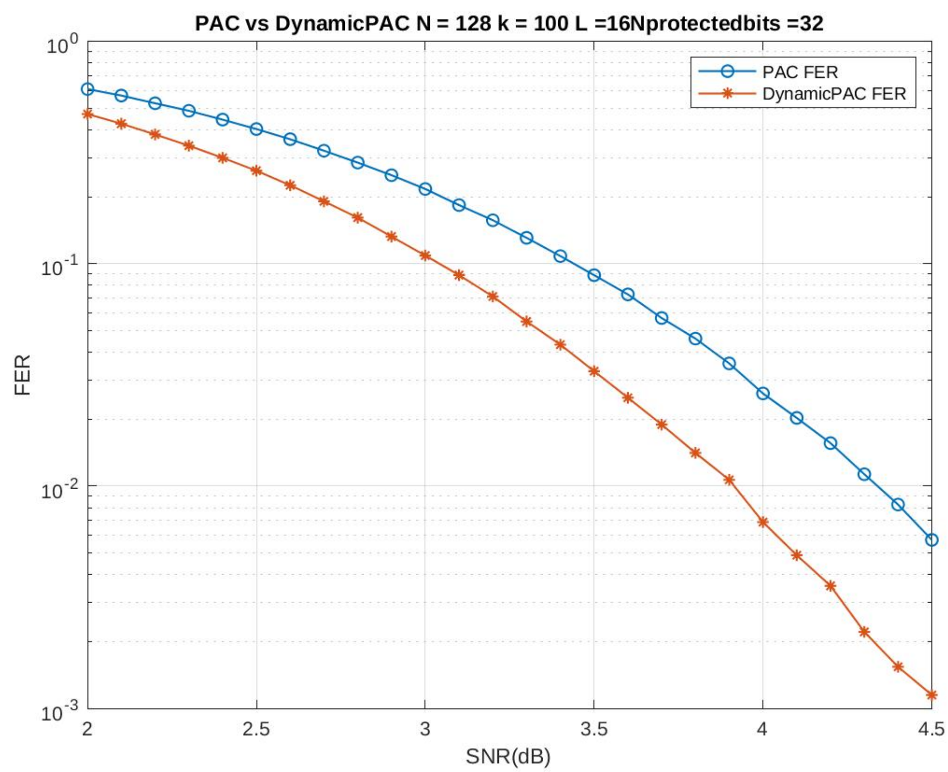

Figure 6.6: Frame error rate for PAC vs CRC Aided SCL for PAC with N=128, k=100, L=16, L protected part=64, N for protected bits = 32, CRC bits = 4 bits

Indeed, the combination of the Targeted Parity Protection (TPP) method and Dynamic Successive Cancellation List (SCL) decoding can yield substantial performance improvements in PAC with larger codewords. Similarly to the gains observed in Polar codes, when considering a specific configuration like PAC(256,180) with a list size of 16 and a protected block size of 64 bits, this joint approach has demonstrated an improvement of up to 0.6 dB in Signal-to-Noise Ratio (SNR). (fig. 6.7) The significance of these findings lies in the potential application of PAC in various high-rate and high-capacity communication systems.

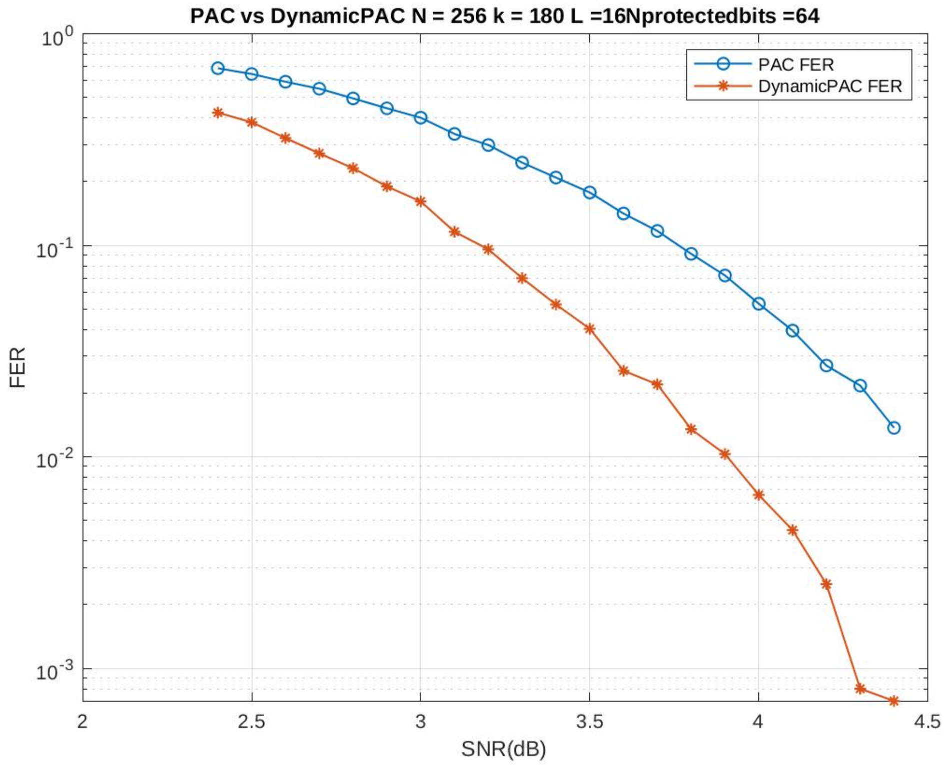

Figure 6.7: Frame error rate for PAC vs CRC Aided PAC SCL with N=256, k=180, L=16, L protected part=128, N for protected bits = 64, CRC bits = 4 bits

And finally, the bit error rate performance for PAC Vs TPP Dynamic SCL PAC is almost the same as the Frame error rate, Fig. 6.8

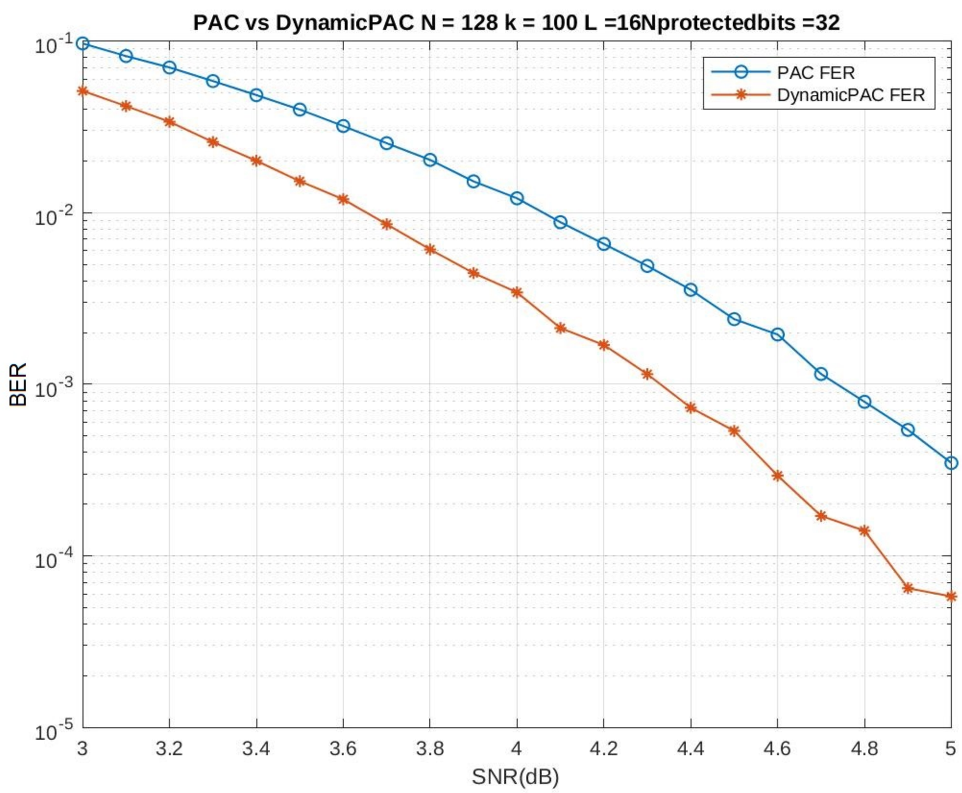

Figure 6.8: Bit error rate for PAC vs CRC Aided SCL for PAC with N=128, k=100, L=16, L protected part=64, N for protected bits = 32, CRC bits = 4 bits

Chapter 7

Conclusion

In this thesis, we have presented a novel decoding technique that combines Targeted Parity Protection (TPP) and Dynamic Successive Cancellation List (SCL) algorithms to significantly enhance the error-correction performance of Polar and PAC codes. The integration of TPP enables more efficient exploration of multiple paths in the tree search part of the decoder by incorporating the Cyclic Redundancy Check (CRC) bits. The Dynamic SCL ensures adaptability to varying code rates and lengths to maximize TPP capability, leading to improved performance and reduced complexity.

The results obtained from extensive simulations demonstrate substantial gains of up to 0.4 dB and 0.6 dB for Polar and PAC codes, respectively, compared to conventional SCL decoding techniques. Moreover, the proposed method exhibits a considerable reduction in computational complexity, surpassing the performance of the traditional Fano decoder, which opens up new possibilities for practical implementation in communication systems.

Future Works:

Since the decoding technique presented in this thesis is applied specifically to the tree search part of the decoder and is independent of other decoder components, several avenues for future research and improvements are evident.

Integration with Fast SCL and Systematic SCL: One promising direction is to investigate the combination of the proposed TPP and Dynamic SCL method with other state-of-the-art decoding algorithms, such as Fast SCL and Systematic SCL. By leveraging the strengths of each technique, it is expected that even higher performance gains can be achieved, making the decoding process more efficient and reliable.

Hybrid Decoding Architectures: The second SCL decoder employed in this method could be replaced by a Fano decoder, potentially creating a hybrid decoding architecture. This hybrid approach could exploit the advantages of both SCL and Fano decoders, further reducing complexity while maintaining competitive error correction capabilities.

Practical Implementation and Hardware Realization: As with any novel decoding method, a crucial future step involves practical implementation and hardware realization. Investigating the feasibility of integrating the proposed technique into real-world communication systems and optimizing its performance for specific hardware platforms will be an essential aspect of future research.

Performance Analysis for Large Block Sizes: The current work focused on evaluating the proposed technique for specific block sizes and code rates. Extending the analysis to larger block sizes and various code configurations will provide a comprehensive understanding of the technique's scalability and generalizability.

Exploration of Other Code Families: While the current study focused on Polar and PAC codes, similar techniques may prove valuable for other error-correction code families as well. Exploring the applicability of TPP and Dynamic SCL in conjunction with alternative code families could unlock new possibilities for enhanced performance and reduced decoding complexity.

In conclusion, the novel decoding technique introduced in this thesis showcases the remarkable potential for improving the error-correction performance of Polar and PAC codes. By independently targeting the tree search part of the decoder, the method is versatile and can be integrated with other decoding algorithms to further optimize performance. These findings open up exciting opportunities for advancing error-correction strategies in modern communication systems, fostering more reliable and efficient data transmission in the future.

Bibliography

[1] Erdal Arikan. From sequential decoding to channel polarization and back again. *CoRR*, abs/1908.09594, 2019.

[2] Tayyab Mehmood, Metodi P. Yankov, and Søren Forchhammer. Rate-adaptive concatenated multilevel coding with fixed decoding complexity. In *2021 11th International Symposium on Topics in Coding (ISTC)*, pages 1–5, 2021.

[3] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, 2009.

[4] Mohammad Rowshan and Emanuele Viterbo. On convolutional precoding in pac codes. In *2021 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2021.

[5] Hongfei Zhu, Zhiwei Cao, Yuping Zhao, Dou Li, and Yanjun Yang. Fast list decoders for polarization-adjusted convolutional (pac) codes. *IET Communications*, 17(7):842–851, 2023.

[6] Mohsen Moradi, Amir Mozammel, Kangjian Qin, and Erdal Arikan. Performance and complexity of sequential decoding of pac codes, 2020.

[7] Satish Addanki, I.S. Amiri, and P. Yupapin. Review of optical fibers-introduction and applications in fiber lasers. *Results in Physics*, 10:743–750, 2018.

[8] Yang Yue, Qiang Wang, and Jon Anderson. Experimental investigation of 400 gb/s data center interconnect using unamplified high-baud-rate and high-order qam single-carrier signal. *Applied Sciences*, 9(12), 2019.

[9] I. S. Amiri, Ahmed Nabih Zaki Rashed, A.H.M Shahariar Parvez, Bikash Kumar Paul, and Kawsar Ahmed. Performance enhancement of fiber optic and optical wireless communication channels by using forward error correction codes. *Journal of Optical Communications*, 2019.

[10] Huazi Zhang, Rong Li, Jian Wang, Shengchen Dai, Gongzheng Zhang, Ying Chen, Hejia Luo, and Jun Wang. Parity-check polar coding for 5g and beyond. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7, 2018.

[11] Hanwen Yao, Arman Fazeli, and Alexander Vardy. List decoding of arıkan's pac codes. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 443–448, 2020.

[12] Ruslan Morozov and Peter Trifonov. Successive and two-stage systematic encoding of polar subcodes. *IEEE Wireless Communications Letters*, 8(3):877–880, 2019.

[13] Thibaud Tonnellier and Warren J. Gross. On systematic polarization-adjusted convolutional (pac) codes. *IEEE Communications Letters*, 25(7):2128–2132, 2021.

[14] Alexios Balatsoukas-Stimming, Mani Bastani Parizi, and Andreas Burg. Llr-based successive cancellation list decoding of polar codes. *IEEE Transactions on Signal Processing*, 63(19):5165–5179, 2015.

[15] Min-Oh Jeong and Song-Nam Hong. Sc-fano decoding of polar codes, 2019.

[16] Grigorii Trofimiuk, Nikolai Iakuba, Stanislav Rets, Kirill Ivanov, and Peter Trifonov. Fast block sequential decoding of polar codes. *IEEE Transactions on Vehicular Technology*, 69(10):10988–10999, 2020.

[17] Ido Tal and Alexander Vardy. List decoding of polar codes. *IEEE Transactions on Information Theory*, 61(5):2213–2226, 2015.

[18] Bin Li, Huazi Zhang, and Jiaqi Gu. On pre-transformed polar codes, 2019.

[19] Harish Vangala, Emanuele Viterbo, and Yi Hong. A comparative study of polar code constructions for the awgn channel, 2015.

[20] Moustafa Ebada, Sebastian Cammerer, Ahmed Elkelesh, and Stephan ten Brink. Deep learning-based polar code design, 2019.

[21] Samir Kumar Mishra and KwangChul Kim. Selectively precoded polar codes, 2020.

[22] Ali Emre Pusane, Alberto Jimenez Feltstrom, Arvind Sridharan, Michael Lentmaier, Kamil Sh. Zigangirov, and Daniel J. Costello. Implementation aspects of ldpc convolutional codes. *IEEE Transactions on Communications*, 56(7):1060–1069, 2008.

[23] Tobias Fehenberger, Georg Böcherer, Alex Alvarado, and Norbert Hanik. Ldpc coded modulation with probabilistic shaping for optical fiber systems. In *2015 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, 2015.

[24] Mohammad Rowshan, Andreas Burg, and Emanuele Viterbo. Polarization-adjusted convolutional (pac) codes: Sequential decoding vs list decoding. *IEEE Transactions on Vehicular Technology*, 70(2):1434–1447, 2021.

[25] Bin Li, Hui Shen, and David Tse. A rm-polar codes, 2014.

[26] He Sun, Emanuele Viterbo, and Rongke Liu. Optimized rate-profiling for pac codes, 2021.

[27] Kilavo Hassan, Kisangiri Michael, and Salehe I. Mrutu. Design of soft viterbi algorithm decoder enhanced with non-transmittable codewords for storage media. 2017.

[28] Jacob King, Hanwen Yao, William Ryan, and Richard D. Wesel. Design, performance, and complexity of crc-aided list decoding of convolutional and polar codes for short messages, 2023.

[29] Qasim Jan, Shahid Hussain, Muhammad Furqan, Zhiwen Pan, Nan Liu, and Xiaohu You. Parity-check-crc concatenated polar codes sscflip decoder. *Electronics*, 11(23), 2022.