Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ30140

A Comparative Study of Channel Coding Schemes in Wireless Communication: Turbo, LDPC, and Polar Codes

Zhengxu Li

International School, Beijing University of Posts and Telecommunications, Beijing, China
lizhengxu@bupt.edu.cn

Abstract. The fast growth of wireless communication means we need to send data reliably, even with noise and interference, which therefore makes channel coding indispensable. This paper looks at three landmark schemes that shaped modern wireless communications: Turbo codes, Low-Density Parity-Check (LDPC) codes, and Polar codes. Specifically, it explains the basic ideas of channel coding, and it discusses methods to approach the Shannon limit. After that, this paper describes the structure and decoding steps for each code. Turbo codes use parallel concatenation and iterative decoding. LDPC codes need sparse parity-check matrices and belief propagation. Also, Polar codes get reliability using channel polarization. Moreover, the codes are also compared based on their error performance, how complex they are to decode, the delay they introduce, and whether they are used in standards. The results reveal that Turbo codes work best for medium to long blocks but have problems with delay and error floors. LDPC codes perform well with long blocks and high throughput. Polar codes are useful for short blocks, even though they are harder to decode. This comparison shows that the codes complement each other and suggests that adaptive, AI-assisted coding could be a promising approach for 6G.

Keywords: Channel Coding, Turbo Codes, Low-Density Parity-Check (LDPC) Codes, Polar Codes, Wireless Communication

1. Introduction

The rise of data-driven applications means wireless communication must be reliable, and signals always face noise, fading, and interference, which makes service quality worse. As a result, channel coding helps solve these problems by detecting and correcting errors to keep communication stable. Over the past decades, advanced coding methods have been central to the development of wireless systems from 3G to 5G, and they are expected to remain important for 6G. Among the many coding techniques, three stand out in modern wireless systems: Turbo codes, Low-Density Parity-Check (LDPC) codes, and Polar codes. Turbo codes came out in the 1990s. They got performance close to the Shannon limit and were still easy to decode, making them a main part of 3G and 4G. Then, LDPC codes came back in the late 1990s. These codes gave strong error correction and were simple to decode in parallel. This helped them get used in Wi-Fi, DVB-S2, and 5G data channels. Polar codes were the first codes that were proven to reach channel capacity, and were later chosen for 5G control channels. Many studies explore these three codes alone, but few compare their performance,

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.TJ30140

complexity, and standardization side-by-side. This paper fills that gap by reviewing and comparing Turbo, LDPC, and Polar codes, highlighting their characteristics, limitations, and potential future directions, including the integration of artificial intelligence.

2. Basic principles and performance requirements of channel coding

In wireless communication, channel coding is important because it adds extra information to the transmitted data. And this allows the receiver to find and fix errors, keeping communication reliable even when the channel is poor. When wireless signals travel, they are affected by noise, fading, and interference from other channels. Without proper error control, the received signal can be distorted, causing bit errors or even communication failure [1]. Channel coding adds extra bits to the data in a systematic way, which helps the system handle noise and interference better and improves overall communication performance

To achieve this goal, the basic principle of channel coding is to expand the original information bit sequence by adding redundant bits to form a codeword. Let the number of information bits be k and the number of redundant bits be r , then the codeword length is given by Equation (1).

$$n = k + r \tag{1}$$

The code rate R is defined as shown in Equation (2).

$$R = \frac{k}{n} \tag{2}$$

The code rate affects the balance between efficiency and reliability. A high code rate allows more data to be sent but gives less error protection because there are fewer extra bits. A low code rate improves error resistance but uses more bandwidth. How the extra bits are designed determines the trade-off between error correction, decoding complexity, and system delay, so practical systems need to find the right balance.

Shannon introduced the concept of channel capacity in 1948 [1]. It represents the highest rate at which information can be transmitted reliably over a channel. The channel coding theorem says that if the code rate is lower than the capacity, then codes exist that can make the error chance almost zero. Therefore, the theorem makes channel capacity the main goal for coding performance. This is why channel coding research works to design codes that get close to this limit but still work easily and quickly in real life. Following this theory, channel coding methods have kept changing. Early codes such as convolutional codes, BCH codes, and Reed-Solomon codes started error control for satellite, storage, and digital communication. In the 1990s, Turbo codes showed up and significantly boosted coding performance, allowing practical systems to get close to the Shannon limit [2]. Later, LDPC codes were brought back into use, and Polar codes were invented, further pushing the limits of channel coding [3,4]. These codes fix errors well, which helps keep high speeds and low errors in today's wireless systems. Besides, they lead the way for 6G designs, such as high frequencies, big antenna arrays, and fast links.

DOI: 10.54254/2755-2721/2026.TJ30140

3. Mainstream channel coding schemes and their characteristics

3.1. Turbo code structure and performance

Turbo codes were first suggested by Berrou et al. in 1993. They adopted a PCCC structure (Parallel Concatenated Convolutional Code) to reach near the Shannon limit while still being easy enough to decode [2]. A typical Turbo encoder has two identical Recursive Systematic Convolutional (RSC) encoders linked by a pseudo-random interleaver. The input sequence is sent directly as systematic bits and is also encoded by the first RSC encoder to create parity bits. After the interleaver shuffles the input sequence, the second RSC encoder produces another set of parity bits. The final codeword combines the systematic bits with both sets of parity bits. The interleaver breaks input correlations, which prevents low-weight codewords from dominating and lowers the overall error probability [5].

Turbo codes are decoded through an iterative Soft-Input Soft-Output (SISO) algorithm, usually with the BCJR approach [6]. And these two decoders share extra information by interleaving and deinterleaving. This slowly makes the bit estimates more reliable. The system makes final decisions on the LLRs (Log-Likelihood Ratios) after a specific number of tries or when a stop rule is met. At medium to high signal-to-noise ratios (SNR), Turbo codes perform very well and were among the first practical codes to approach channel capacity. However, they do have problems. For example, an “error floor” occurs when the bit error rate is very low. This stops them from working for things that need super-high reliability. Besides, the way the decoding must run in order makes the delay and power use higher. Still, Turbo codes are now a main technology in mobile broadband. They are used a lot in 3G (UMTS/HSPA) and 4G (LTE) standards [7,8]. They helped advance near-capacity coding and influenced the design of later codes.

3.2. Low-Density Parity-Check (LDPC) code and features

Low-Density Parity-Check (LDPC) codes were first proposed by Gallager in 1962 [3]. At that time, the required iterative decoding algorithms were too demanding for available hardware, so the idea was mostly unused for decades. In the late 1990s, MacKay and others revisited LDPC codes, taking advantage of better computing power and insights from Turbo code research, and showed that they could achieve performance close to the Shannon limit [9].

In this context, an LDPC code is a linear block code defined by a sparse parity-check matrix, and a vector qualifies as a valid codeword only if it satisfies the parity-check conditions. The sparsity of the matrix, where the number of ones is much smaller than the number of zeros, is key to efficient decoding. The structure is often shown with a Tanner graph. In this graph, code bits are variable nodes, parity constraints are check nodes, and the edges show the connections set by the matrix. Based on this structure, decoding is usually done using the Belief Propagation (BP) or Sum-Product Algorithm (SPA) [10]. The process starts when the variable nodes get their first values from the channel output. These values are often LLRs. In each iteration, variable and check nodes exchange reliability messages, gradually improving confidence in the estimated codeword. Iterations proceed until the parity constraints are met or the maximum number is reached, after which hard decisions are taken. The way LDPC codes decode, which involves iteration, plus their sparse structure, means they can do a lot of parallel computing. This makes them great for hardware on FPGAs and ASICs [9]. When block lengths are long, LDPC codes perform better than Turbo codes and do not have an error floor. However, they don't work as well at short and medium block lengths. Still, their strong error fixing and hardware efficiency make them very useful. Thus, LDPC codes are widely used in

DOI: 10.54254/2755-2721/2026.TJ30140

standards such as Wi-Fi (IEEE 802.11n/ac/ax), DVB-S2 (Digital Video Broadcasting), and the data channels for 5G New Radio eMBB services [11, 12].

3.3. Polar code construction and applications

Arıkan first suggested Polar codes in 2009. This was a big step in coding theory [4]. These codes were the first ones with a clear design that could reach the Shannon capacity for any binary-input channel that does not “remember” its past. This answered a long-running question and gave a strong theoretical base for making codes that get close to the channel limit.

The main idea behind Polar codes is channel polarization. And this works by repeatedly putting copies of a channel together and then splitting them apart. As the number of sub-channels goes up, some become almost free of error, but the rest get very noisy. The reliable sub-channels carry the information bits, and the unreliable ones are set to fixed values, usually zeros, called frozen bits. The method is efficient, and the encoding process has relatively low complexity [4]. Decoding starts with the Successive Cancellation (SC) algorithm. This algorithm is simple and quick, but it does not perform well when block lengths are only medium-sized. The Successive Cancellation List (SCL) algorithm improves decoding by keeping multiple possible paths, which helps reduce errors [13]. When a Cyclic Redundancy Check is added to SCL, forming CRC-aided SCL (CA-SCL), accuracy increases and the decoder can choose the correct path more reliably [15]. The improvements helped Polar codes move from theory to practice. The 3GPP later chose them for control channels in 5G New Radio, showing their strength for short block lengths and important signaling tasks. In today’s communication systems, Polar codes work alongside LDPC codes [12].

4. Performance evaluation of main channel coding schemes

4.1. Coding performance and BER-SNR

"When comparing Turbo, LDPC, and Polar codes, it helps to see how close they get to the Shannon limit and how they perform under different signal-to-noise ratios (SNRs) and block lengths. Each type has its own strengths and weaknesses depending on the situation. Turbo codes perform well in the waterfall region, where a small increase in SNR leads to a sharp drop in bit error rate (BER). This makes them effective for medium to long block lengths. However, at very low BERs, their performance levels off because of the error floor, which limits their use in ultra-reliable systems [2,5]. LDPC codes have smoother error curves and almost no error floor, especially with long block lengths. Their performance stays close to the Shannon limit, making them suitable for systems that need high throughput and very low error rates. Their stable performance across a wide SNR range is one reason they are used in Wi-Fi, DVB-S2, and 5G standards [3,9,15]. Polar codes depend more on the decoding method. The basic Successive Cancellation (SC) decoder gives moderate results, but the Successive Cancellation List (SCL) or CRC-aided SCL (CA-SCL) decoders greatly improve reliability for short to medium block lengths, often outperforming Turbo codes and coming close to LDPC performance [13,14].

4.2. Decoding complexity and efficiency

Decoding complexity is an important factor in how practical a code is because it affects delay, power use, and memory needs. Although Turbo, LDPC, and Polar codes all provide strong error correction, their different decoding methods bring different challenges for implementation. Turbo codes use iterative SISO decoding, often with the BCJR algorithm. Each iteration makes the bit estimates

DOI: 10.54254/2755-2721/2026.TJ30140

more reliable, but the serial process adds delay and increases power use. Techniques such as windowing and partial parallelism can help reduce these costs, but repeated serial steps still limit overall efficiency [5]. LDPC codes use graph-based decoding, usually BP or the SPA, on Tanner graphs. Since these graphs are sparse, messages between variable and check nodes can be processed in parallel. This makes LDPC codes well suited for FPGA or ASIC hardware, hence reaching high throughput with moderate delay and relatively low power use [9,15]. Furthermore, polar codes work differently. The basic Successive Cancellation (SC) decoder has low computational complexity and works well for short blocks. More advanced methods, like Successive Cancellation List (SCL) and CRC-aided SCL (CA-SCL), are more complex because they must track and sort multiple decoding paths, which increases memory use and delay. Still, for short blocks such as those in 5G control channels, the performance gains make the extra cost worthwhile, giving Polar codes a good balance between reliability and efficiency [16].

4.3. Standardization applications and limitations

Turbo, LDPC, and Polar codes have all been key in building wireless standards, but their real-world use shows different limits. Turbo codes helped 3G networks like UMTS and HSPA and were later a part of 4G/LTE. They made mobile broadband possible by fixing errors well. However, the “error floor” and the high delay from their iterative decoding meant they were not good for the data channels of 5G, which need low delay and high reliability [7,8]. LDPC codes became the top choice for moving large amounts of data. Their near-capacity performance with long data blocks and their highly parallel decoding make them fit well with hardware. Hence, they were chosen for standards such as DVB-S2/S2X, Wi-Fi (IEEE 802.11n/ac/ax), and the coding scheme for the 5G NR data channels in eMBB (enhanced mobile broadband) [18,19]. These codes perform reliably over a wide range of SNR. A problem is that their performance drops with short blocks, and it's still hard to design codes that do not have damaging structures like “trapping sets.”

Polar codes are mainly used in 5G NR control channels, where reliable short-block transmission is critical [17]. CRC-aided SCL decoding improves performance but increases memory use, delay, and limits flexibility because block lengths must be powers of two [16,17]. Table 1 summarizes the main features of Turbo, LDPC, and Polar codes, including their theoretical value, performance at different block lengths, decoding methods, parallelism, delay, error floor, and use in standards.

Table 1. Performance evaluation of Turbo, LDPC, and polar codes in various dimensions

Dimension	Turbo Codes	LDPC Codes	Polar Codes
Theoretical Significance	First practical approach to Shannon limit [2,5]	Near-capacity, key to modern coding theory [3,9]	First capacity-achieving construction for B-DMC [4]
Short/Medium Block Performance	Excellent waterfall, suffers error floor at low BER [5]	Depends on design; often weaker than CA-SCL Polar [9,15]	CA-SCL improves performance, competitive for control [13,14]
Long Block Performance	Limited by error floor [5]	Near-capacity, smooth error curves [9,15]	Weaker than LDPC with certain constructions [4]
Decoding Mechanism	Iterative SISO (BCJR), serial dependency [5]	BP/SPA on Tanner graph, parallelizable [9]	SC (low complexity), SCL (better performance at extra cost) [13,14]
Parallelism / Throughput	Moderate (iteration-dependent)	Strong (naturally parallel, high throughput)	SC: moderate, SCL: limited by list management [16]

DOI: 10.54254/2755-2721/2026.TJ30140

Table 1. (continued)

Latency & Power Consumption	High (multiple iterations required)	Low to medium (scalable parallelism reduces delay)	SC: low, SCL: higher latency/memory [13,14,16]
Error Floor	Pronounced	Rare or negligible	Depends on construction and block length
Standardization	3G/4G (UMTS/HSPA, LTE) [7,8]	DVB-S2/S2X, Wi-Fi, 5G NR data channels [11,12]	5G NR control channels [12,17]
Typical Limitations	Error floor, decoding delay and power	Short-block performance, iterative convergence	Higher SCL complexity, sensitivity to construction [14,16]

5. Challenges and future directions

5.1. Current challenges and limitations

Even though these three codes helped a lot, they still struggle with the tough demands of 5G and 6G applications. They fall short on ultra-low latency, high efficiency, and the extreme reliability needed for URLLC (ultra-reliable low-latency communication) [5].

Turbo codes, though historically significant, are limited by their iterative SISO decoding, which causes high latency and power consumption. Furthermore, their inherent error floor at very low bit error rates restricts their use in applications requiring near-zero residual errors. LDPC codes excel in long-block transmissions but are less effective in short-block scenarios such as URLLC. Their deployment is also challenged by complex design requirements, such as avoiding harmful structures like trapping sets, to ensure high reliability [15,20]. Polar codes' performance is highly dependent on the decoding algorithm. Advanced decoders like SCL and CRC-aided SCL (CA-SCL) improve reliability but at the cost of increased latency, memory usage, and complexity. Additionally, their requirement for block lengths to be powers of two limits their adaptability [16,17].

5.2. Future directions and trends

The limits of current codes call for new strategies in 6G to meet the goals of ultra-reliability, low delay, and energy efficiency. A promising approach is hybrid and adaptive coding, which combines the strengths of different schemes to keep performance stable under changing conditions. This can involve linking codes such as Polar and LDPC, or adjusting coding parameters in real time based on channel conditions and service needs to improve spectral and energy efficiency [21].

Another direction is to bring artificial intelligence into coding and decoding. Deep learning can support end-to-end code design, creating new structures or replacing traditional decoders to boost performance, especially when channels do not follow standard models [22,23]. And AI can also help automate design tasks like choosing frozen sets in Polar codes, allowing ongoing optimization. In addition, semantic-oriented coding marks a shift from bit-level accuracy to conveying meaning. By keeping only information essential to the application, this method aligns well with 6G goals in areas like virtual reality and smart networks [24,25].

6. Conclusion

This paper reviews three key channel coding techniques in modern communications: Turbo codes, LDPC codes, and Polar codes. Their historical development, core principles, and trade-offs show that no single code suits all applications. Turbo codes enabled near-capacity performance with

DOI: 10.54254/2755-2721/2026.TJ30140

practical complexity, LDPC codes excel in high-throughput long-block scenarios, and Polar codes provide reliable short-block performance for control signaling. The 5G NR standard reflects this division of roles. This review focuses on theoretical comparison rather than simulations. Future work should explore performance in realistic channels and the integration of artificial intelligence to support advanced coding for 6G systems.

References

[1] Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal, 27, 379-423.
[2] Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). Near Shannon limit error-correcting coding and decoding: Turbo codes. In Proceedings of IEEE ICC.
[3] Gallager, R. G. (1962). Low-density parity-check codes. IRE Transactions on Information Theory, 8(1), 21-28.
[4] Arıkan, E. (2009). Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Transactions on Information Theory, 55(7), 3051-3073.
[5] Berrou, C., & Glavieux, A. (1996). Near optimum error correcting coding and decoding: Turbo codes. IEEE Transactions on Communications, 44(10), 1261-1271.
[6] Bahl, L. R., Cocke, J., Jelinek, F., & Raviv, J. (1974). Optimal decoding of linear codes for minimizing symbol error rate. IEEE Transactions on Information Theory, 20(2), 284-287.
[7] 3GPP TS 25.212. (2007). Multiplexing and channel coding (FDD) (Release 7).
[8] 3GPP TS 36.212. (2011). Multiplexing and channel coding (E-UTRA) (Release 10).
[9] MacKay, D. J. C. (1999). Good error-correcting codes based on very sparse matrices. IEEE Transactions on Information Theory, 45(2), 399–431.
[10] Kschischang, F. R., Frey, B. J., & Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory, 47(2), 498–519.
[11] ETSI EN 302 307-1 V1.4.1. (2014). Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2).
[12] 3GPP TS 38.212. (2018). NR; Multiplexing and channel coding (Release 15).
[13] Tal, I., & Vardy, A. (2015). List decoding of polar codes. IEEE Transactions on Information Theory, 61(5), 2213–2226.
[14] Niu, K., & Chen, K. (2012). CRC-aided decoding of polar codes. IEEE Communications Letters, 16(10), 1668–1671.
[15] Richardson, T., & Urbanke, R. (2008). Modern coding theory. Cambridge University Press.
[16] IEEE Std 802.11-2016. IEEE standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements.
[17] ETSI EN 302 307-2. (2015). Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications; Part 2: DVB-S2 Extensions (S2X).
[18] Balatsoukas-Stimming, A., Parizi, M. B., & Burg, A. (2015). LLR-based successive cancellation list decoding of polar codes. IEEE Transactions on Signal Processing, 63(19), 5165–5179.
[19] 3GPP TR 38.802. (2017). NR; Study on new radio access technology; Physical layer aspects (v14.2.0).
[20] Han, Y., & Ryan, W. E. (2009). Low-floor decoders for LDPC codes. IEEE Transactions on Communications, 57(6), 1663–1673.
[21] Zhang, Z., Zhang, L., & Chen, D. (2018). Design of hybrid concatenated coding schemes for next-generation wireless systems. IEEE Communications Magazine, 56(3), 124–130.
[22] Nachmani, E., Be'ery, Y., & Burshtein, D. (2016). Learning to decode linear codes using deep learning. In 54th Annual Allerton Conference on Communication, Control, and Computing (pp. 341-346).
[23] Gruber, T., Cammerer, S., Hoydis, J., & ten Brink, S. (2017). On deep learning-based channel decoding. In 51st Annual Conference on Information Sciences and Systems (CISS) (pp. 1–6).
[24] Xie, H., Qin, Z., Li, G. Y., & Zhao, B. (2021). A lite distributed semantic communication system for Internet of Things. IEEE Journal on Selected Areas in Communications, 39(1), 142–153.
[25] Qin, Z., Tao, X., Lu, J., & Li, G. Y. (2021). Semantic communications: Principles and challenges. IEEE Communications Surveys & Tutorials, 23(3), 1575–1600.