Faster than Nyquist Signaling with Spatial Coupling

Haoran Liu and Lingwei Zhang
ha5051li-s@student.lu.se,li1568zh-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor: Michael Lentmaier, Mgeni Makambi Mashauri

Examiner: Ove Edfors

October 31, 2022

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

The emergence of the Faster-Than-Nyquist signaling (FTN) technology has improved the tight spectrum resources by compressing the transmission interval between adjacent pulses. In a natural environment, data transmission via a wireless medium and through multi-path, at a high data rate, the leakage of these paths among symbols can cause severe inter-symbol interference (ISI). In order to compensate for the ISI caused by FTN, turbo equalization technology is demonstrated in this thesis.

A challenge when implementing turbo-equalization is the required matching between channel codes and equalizer, leading to a fundamental trade-off between waterfall performance and error floor in different Signal-to-Noise (SNR) regions. Spatially coupled low-density parity-check codes (SC-LDPC codes) are demonstrated and optimized by using window decoder to improve the system performance.

In this thesis, we mainly investigate SC-LDPC codes with FTN signaling and turbo equalization. The simulation is written in MATLAB together with some MEX files. Firstly, FTN signaling with uncoupled convolutional codes and LDPC codes is simulated as a comparison to a previous article, then SC-LDPC codes are also implemented to evaluate the performance improvement according to changing the different parameters, such as code rate R and time-squeezing factor τ. Meanwhile, the spectral efficiency performance simulation with different code rates and time acceleration factors is also discussed. The results show that by implementing FTN signaling, turbo equalizer and spatial coupling together in the system, such a combination can achieve better performance than the traditional Nyquist scheme in bit error rate (BER) and spectral efficiency (SE) aspects.

Acknowledgement

First and foremost, we would like to acknowledge and extend our heartfelt gratitude to our supervisors at Lund University, Prof. Michael Lentmaier and Dr. Mgeni Makambi Mashauri. For their constant encouragement and patient guidance, generous assistance and invaluable advice, all of which have been of inestimable worth to the completion of our thesis.

Then, we would like to thank all the professors at Lund university for assisting us in different aspects of wireless communications and friends for their support throughout the master's education.

Finally, we are grateful to our families and friends for their constant companionship and support.

Haoran Liu
Lingwei Zhang

Lund, September 2022

Popular Science Summary

In the 21st century, the popularity of 4G networks has greatly improved people's quality of life. As a new generation of communication technology, 5G can bring better services than 4G to people's lives. With 5G technology, people can download a two-hour movie in tens of seconds and watch an HD live soccer match and demands for even higher transmission rates are increasing. The transmission rate, however, is limited by the available frequency resources. A range of frequencies used for signal transmission is called bandwidth. To achieve these higher transmission rates we, therefore need more bandwidth. However, the frequency resources are scarce and very expensive.

One solution is to pack more data to be transmitted in a limited bandwidth. If the data rate exceeds a certain limit, technically known as the Nyquist rate, we experience distortions in our received data. These distortions are caused by the overlap between adjacent signals, which is called intersymbol interference (ISI). We can, however, revert this interference at the receiver by a technique called equalization. We can therefore transmit faster than the Nyquist limit and use equalization to mitigate the resulting ISI. This kind of transmission is called faster than Nyquist (FTN) signaling.

However, because of noise when transmitting the signals, there will be some errors even after equalization. These errors can be corrected at the receiver if we use error-correcting codes, where controlled redundancy is added by the transmitter to enable the receiver to recover these errors. If the equalizer and error-correcting codes work separately the overall error-correction performance can be good, but even better performance can be obtained by the interaction between equalization techniques and error-correcting codes. This whole combination is called Turbo equalization.

Spatially-coupled codes are a kind of error correction codes that have shown good error correcting capability. In this thesis, we investigate the impact of using these spatially coupled codes for systems using FTN transmission. We show that, with spatially coupled codes, we can get higher data rates in limited bandwidth, using much lower energy at the transmitter compared to classical coding schemes.

Table of Contents

List of Figures

Chapter 1

Introduction

Communication systems form a tool for human society to exchange information. With the development of communications technologies in today's information age, more and more data should be transmitted at a relatively high data rate, such as in 5G mobile communications. However, in a natural environment, data transmission via a wireless medium and through multi-path with high data speed causes the inevitable ISI which will degrade the receiver performance.

In a digital base-band transmission system, conventionally, the Nyquist criterion shows that in order to obtain ISI-free communication, the symbol rate can not exceed the Nyquist rate. But in order to increase the spectral efficiency (SE) of single-carrier communications, alternative waveforms obtained by FTN signaling can be used [1]. FTN signaling increases the number of symbols transmitted within the same time interval which means implementing it can reach a higher data rate meanwhile, it results in the ISI problem inevitably. Equalization can mitigate the effect of the ISI problem [2]. Turbo equalization is motivated by normal turbo code, it refers to the iterative work of the soft output detector and the soft output decoder to obtain an approximate optimal decision for the transmitted bits. Unlike Viterbi equalizers, which usually produce hard outputs, BCJR equalizers have available soft outputs in the form of LLR values calculated by the BCJR algorithm and it is critical to the modern iteratively-decoded error-correcting codes such as LDPC codes and turbo codes.

In principle, there will be a trade-off between good performance at lower SNR called waterfall and good error floors at higher SNR as [3] said. By using spatially coupled LDPC codes which can be constructed from the protograph or followed by some random and special rules to construct the base matrix B, it will get a relatively good performance in the waterfall region compared with uncoupled LDPC codes.

1.1 Project goal

The main aim of this thesis is to investigate spatially coupled LDPC coded with FTN signaling, using the results without coupling from [4] as a reference. In order to compensate for the ISI caused by FTN, equalization is a good way to handle it.

Matching code parameters with the turbo equalizer, with FTN parameters as well as the QPSK modulation, is an important issue in such a scenario. As previous research [3] shown, the regular trade-off for turbo equalization can be avoided with spatial coupling, allowing to operate close to capacity with strong codes and low error floors. Therefore, the spatial coupling will be mainly investigated on the code level (DVB-S2 standard LDPC codes) with FTN signaling together to evaluate the performance with different parameters during this thesis.

1.2 Related work

How to use the limited bandwidth resources? The current high-speed and high-reliability data transmission has brought challenges to the communication physical layer technology. Several studies of FTN signaling have been published. Mazo [1975] firstly proposed that the transmission rate of symbols can exceed the Nyquist rate (FTN) with the minimum Euclidean distance not decreasing [5]. Later in 2013, Livens and Georghiades used raised cosine (RC) pulse instead of sinc pulse, meanwhile, they investigated the influence from the different roll-off factors to Mazo threshold [6]. Rusek et al. studied the capacity of the FTN system deeply and found that FTN transmission is higher than that achieved by traditional Nyquist transmission capacity [1]. Moreover, his group explored the higher order modulation in FTN signaling communication systems [4].

In order to improve the transmission speed, FTN destroys the orthogonality of the waveform and also brings ISI problems which can be alleviated by using the equalization technique. Normal techniques such as non-linear algorithm MLSE, are generally implemented by the Viterbi algorithm along with high complexity. Maximum likelihood (ML) estimation, which turns into a maximum a posteriori probability (MAP) estimation in presence of a priori information commonly be implemented with the BCJR algorithm [2]. In recent years, there has been an increasing amount of optimization algorithms aiming at balancing the detection performance and decreasing complexity but not been explored in this thesis.

Low-density-parity-check convolutional codes (LDPC-CC) as a variant of LDPC codes, is also called spatially coupled LDPC codes (SC-LDPC) in some literature [7] [8]. Lentmaier et al. demonstrated that using rate 1/2 SC-LDPC codes with a sum-product decoding algorithm can approach the Shanno limit. For the design and construction of SC-LDPC codes, it can be divided into prototyping-based construction and parity check matrix construction. Olmos et al. constructed SC-LDPC code based on protograph [9]. Tadayon et al. extended the method of randomly constructing a large girth matrix and constructed the SC-LDPC codes with shorter forward memory length [10]. There is still much progress to be made especially for the check matrix construction method. In terms of the decoding part, Iyengar et al. proposed window decoding algorithm which outperforms the normal BP algorithm in delay and complexity [11]. In the past two years, optimized algorithms such as improved window extension decoding and layered decoding improve the decoding performance further.

1.3 Thesis outline

The layout of this thesis is organized into five chapters. Chapter 1 introduces the project goal and related work. Chapter 2 provides the related basic concepts from the baseband transmission model to FTN system model, then such as turbo equalization, BCJR algorithm, protograph-based code construction and Tanner graph and protograph are introduced as well. Chapter 3 starts by implementing FTN with convolutional codes with different parameters, then we simulate LDPC codes with FTN and equalization with different parameters according to [4] and compare part of the results including that without FTN scheme. Meanwhile, we also study the SE performance by using different code rates and different FTN time accelerate parameters τ. Chapter 4 introduces the spatial coupling technique combined with LDPC codes with window decoding and shows the improvement compared with non-spatial coupled LDPC code. Finally, conclusions and future work are discussed in Chapter 5.

Chapter 2
Basic concepts

2.1 Introduction

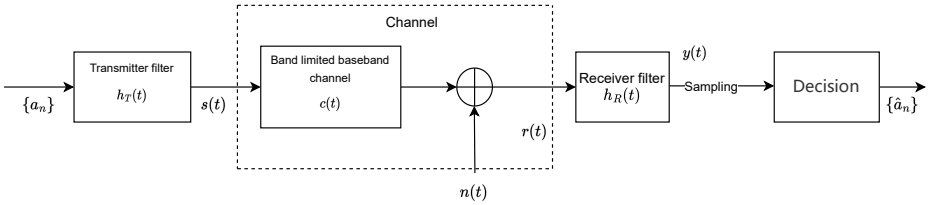

Figure 2.1: Baseband transmission model

The traditional digital baseband transmission system model is shown in Figure 2.1 [12]. The transmitted signal passes through the transmit filter and then passes through the band-limited baseband channel and reaches the receive filter. After making a sampling decision, the received signal will be obtained. As shown above, $\{a_n\}$ is a system input sequence that consists of the independent M-ary data symbols. Assume the unit symbol energy here, the pulsed superimposed transmitted signal $s(t)$:

$$s(t) = \sum_{n=-\infty}^{+\infty} a_n h_T(t - nT_s) \tag{2.1}$$

where $h_T(t)$, $c(t)$ and the below $h_R(t)$ are the impulse response and T_s is the transmission interval between symbols. Then it goes through the baseband channel with impulse response $c(t)$ and white Gaussian noise $n(t)$ with two-sided spectral density $N_0/2$ to get $r(t)$. Then $y(t)$ is the output of the receive filter (normally matched filter):

$$y(t) = \sum_{n=-\infty}^{+\infty} a_n x(t - nT_s) + \eta(t) \tag{2.2}$$

where $x(t) = h_T(t) * c(t) * h_R(t), \quad \eta(t) = h_R(t) * n(t)$.

Then $y(t)$ is sampled at $t = mT_s$ discrete points to get:

$$y_m = \sum_{n=-\infty}^{+\infty} a_n x_{m-n} + \eta_m = a_m x_0 + \sum_{n \neq m} a_n x_{m-n} + \eta_m \tag{2.3}$$

where the middle part represents the ISI component and the last part represents additive noise and these two parts influence the sample decision.

From (2.1), it is noticeable that the symbol interval T_s is important, if the transmission interval between two symbols does not meet nT_s, then the orthogonality is no longer satisfied, which will bring severe inter symbol interference, and then result in a higher error rate.

2.2 Faster than Nyquist signaling

In a digital communication system, validity and reliability are two indicators to measure the performance of the system. These two indicators are normally contradictory, therefore, how to achieve high-speed and reliable data transmission is our target. Does a band-limited digital communication system that introduces ISI necessarily perform worse than a transmission system that conforms to the Nyquist criterion? Traditionally it was thought true. But with further study of FTN technique, this view will be reformed.

FTN signaling technique traces back to the 1970s and attracted more and more interest as it can pack 30%–100% more data in the same bandwidth at the same energy per bit and error rate compared to traditional methods. The linear transmission with FTN method is widely used in data transmission which has the form of a sum of a sequence of data pulses, given by:

$$s(t) = \sqrt{E_s} \sum_{n=-\infty}^{+\infty} a_n h(t - n\tau T) \tag{2.4}$$

where E_s is the average symbol energy, $h(t)$ is the transmitter impulse response, τ is the time-squeezing factor which can be thought of as a time acceleration factor since now the pulses come too fast by a factor $1/\tau$ (in the FTN case, $\tau < 1$). Other parameters have the same meanings like in (2.1). Note that $s(t)$ here is not the same as that in (2.1) . For the Nyquist shaping pulse,

$$h(nT_s) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{2.5}$$

According to Nyquist first theorem, there will be no ISI over an AWGN channel when $\tau = 1$. And the Nyquist rate applied when $T_s = T$. In such a case, the frequency spacing, $1/T_s$, is called the Nyquist limit.

Mazo showed [5] that for ideal sinc pulses, the bandwidth can be reduced to 0.401 Hz/bit/s without any Euclidean distance loss. This corresponds to setting

$T\Delta = 0.802T$ instead of T (T refers to the transmission interval between symbol). This value is refered to the Mazo limit. With the decrease of τ, the orthogonality of symbols will be broken (due to being smaller than the Nyquist bandwidth), and then the symbol error probability (SER) will increase (due to over the Mazo limit). The Mazo limit theory states that at a certain bandwidth and modulation, more data can be transferred even in the presence of ISI.

In other words, according to Mazo's study, in the equal bandwidth case, FTN system can transmit more than up to 25% data without the loss in the average probability of error of the system. In this way, the system efficiency can be improved, which is the biggest highlight of FTN transmission technology.

In this thesis, we replace Mazo's sinc pulse with a more general family of raised-cosine pulses with roll-off factor $0 \leq \alpha \leq 1$ as [4] used, and assume the energy per symbol is 1, which is:

$$h(t) = \text{sinc}\left(\frac{t}{T}\right)\frac{\cos\left(\frac{\pi\alpha t}{T}\right)}{1-\left(\frac{2\alpha t}{T}\right)^2} \tag{2.6}$$

When the roll-off coefficient is smaller, the side lobe fluctuation on the time domain waveform will be higher, and the attenuation will be slower, which will also lead to a more severe ISI.

Assume in the non-ISI transmission scheme, by using the sinc pulse, the minimal transmission interval between symbols is $T_s = T = 1/(2W)$ and the corresponding non-ISI maximum symbol speed $2W$ baud is called Nyquist speed. Whatsmore, by using the raised-cosine pulse, the bandwidth of it is $(1+\alpha)/2T$. Based on the AWGN channel assumption, the received signal is expressed as:

$$r(t) = s(t) + n(t), \tag{2.7}$$

where $n(t)$ denotes the additive white Gaussian noise with two-sided spectral density $N_0/2$.

Figure 2.2 shows an example of a pulse $h(t)$ with orthogonal symbol time $T = 1$ and $\tau = 1$, the roll-off factor is 0.3 here, assume the $E_s = 1$ in the following two illustrations. The green line represents the sum of symbols curves, which is $s(t)$, except this, other colorful lines represent symbols +1, -1, +1, -1.

Figure 2.3 uses the same pulse as Figure 2.2 uses, the same T and E_s as well, but with $\tau = 0.8$. It can be seen that the five pulses (represented by the blue line, orange line, yellow line, purple line) are now advanced in time by 0.2, 0.4, etc. Meanwhile, at the corresponding sampling points, the ISI appears, which will influence the receiver detection performance and increase the complexity of the receiver part.

Significantly, for either Nyquist system or FTN system, the used bandwidth is determined by shaping pulse, thus, even though FTN system improves the sym-

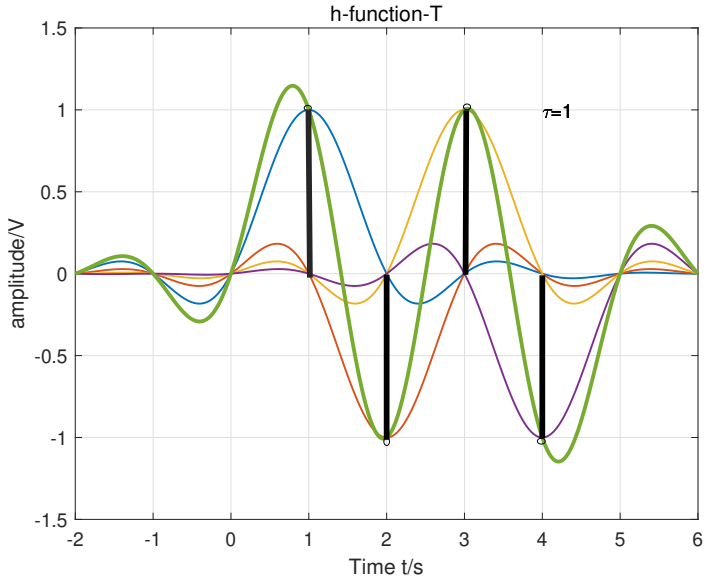

Figure 2.2: Nyquist signaling with unit-T h(t) pulses, $\tau = 1$

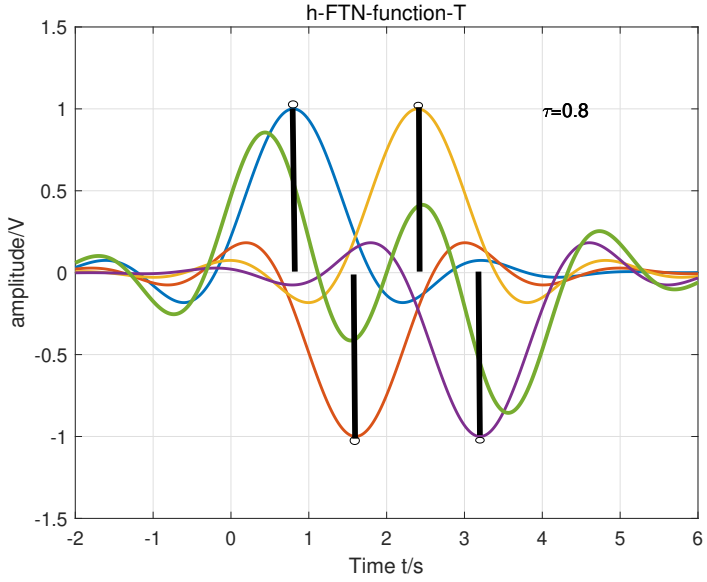

Figure 2.3: FTN signaling with unit-T h(t) pulses, $\tau = 0.8$

bol transmission speed, the used bandwidth remains constant, that is the FTN technique can increase the spectral efficiency (η).

2.2.1 FTN signaling model

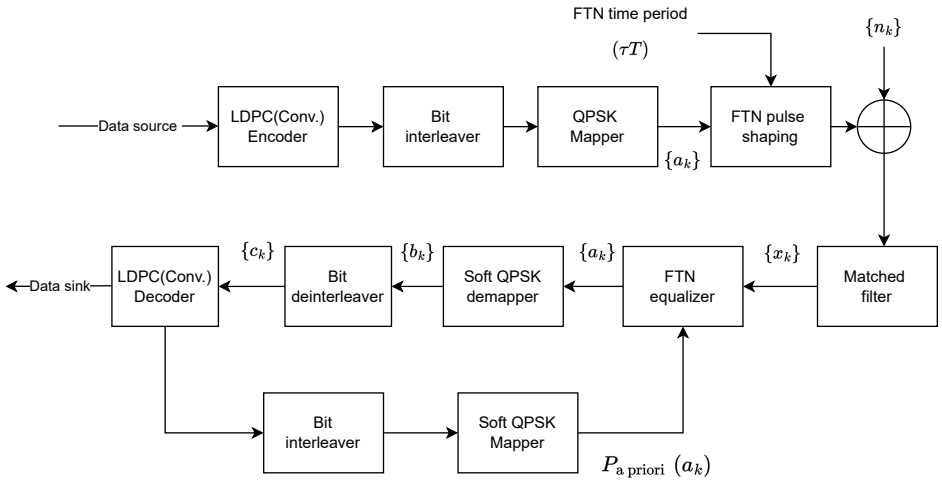

Figure 2.4: FTN transmitter and receiver model

Figure 2.4 is an overall model referring to [4] for QPSK modulation in FTN system which can be divided into transmitter part and receiver part. Next chapter will show the simulation result by using this model. A sequence of binary data bits comes into the LDPC encoder or convolutional encoder. Then the encoded codewords go through an interleaver before the QPSK mapper which will decrease the correlations. The outputs $\{a_k\}$ then pass through a pulse shaping filter $h(t)$ in (2.6). Normally, the spectrum of $h(t)$ is band limited, thus the corresponding waveform in time domain is infinite, so $h(t)$ needs to be cut. Let L be the truncation length (also called a suitable channel length), as [4] mentioned, for QPSK, $L = 4$ is sufficient for a wide range of τ that extends beyond the range that we are investigating in this paper, so we use $L = 4$ in the simulation.

And we use additive white Gaussian noise with two-sided spectral density $N_0/2$. In this model, the receiver first passes through a matched filter, and the output of the matched filter is:

$$x_k = \sum_{l=-L}^{+L} g_l a_{k-l} + n_k \tag{2.8}$$

where a_{k-l} and n_k are constellation points and noise samples defined in Figure 2.4, the suitable channel length (so-called channel memory) which equals to $L = 4$ here, and g_l is the sampled convolution of the transmitter shaping pulse h_t and the matched filter of the receiver. The sampled period is $\tau \cdot T$ from $g(t)$ (the right one in Figure 2.5), and g_l is shown as:

$$g_l = h(t) \star h^*(-t) \mid_{t=l\cdot\tau T} = h(t) \star h(t) \mid_{t=l\cdot\tau T} \tag{2.9}$$

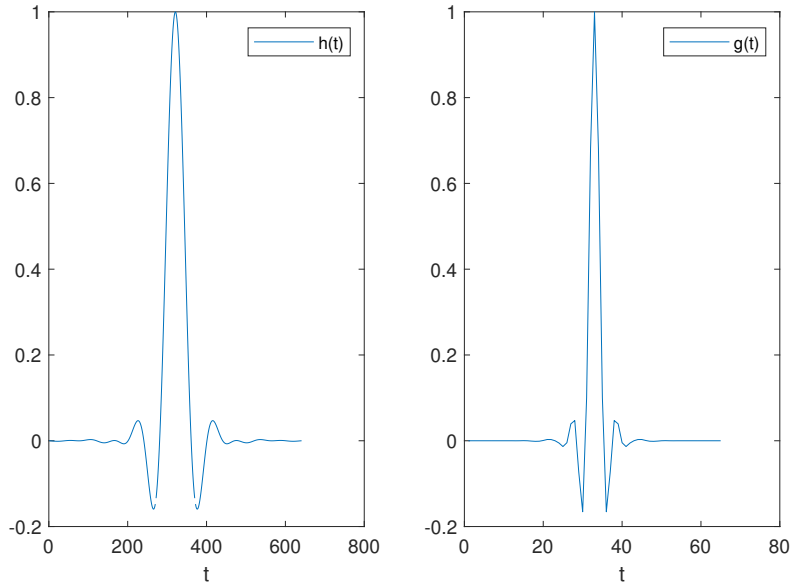

Figure 2.5: The raised cosine pulse h(t) and time domain convolution sequence g(t)

Figure 2.5 shows the partial raised cosine pulses $h(t)$ (from -8T to 8T, drawing inter is 0.037) and corresponding $g(t)$ which is the output of the matched filter in the simulation. And $g(t)$ is the convolution result of down-sampled $h(t)$ and that's why the X label is different. It is noteworthy that the parts of g_l that fall outside L are modeled as noise which is included in n_k. And g_l shown in Figure 2.6 is the discrete sampling of $g(t)$. The first five points shown as g0, g1, g2, g3, g4 are partial g_l for the equalizer and τ = 0.74 and the roll-off factor is 0.4 here.

Due to the symmetry of $g(t)$ and the statistical properties of the preceding and following symbols are the same, since, ISI given from [4] can be expressed by

$$isi(a_k, \sigma_k) = \frac{1}{2} a_k g_0 + \sum_{l=1}^{L} a_{k-l} g_l \tag{2.10}$$

where $\sigma_k = (a_{k-1}, a_{k-2}, ..., a_{k-l})$ is defined for convenience and here we just consider the interference between previous symbols to current symbols. The last but important part of the receiver is the iterative decoding of the turbo structure which consists of FTN equalization, interleaver, deinterleaver and LDPC decoder. By continuously feeding back prior information and extrinsic information to each other, the output sequence is finally obtained. In our simulation, since the observed matched-filter output x_k is corrupted by colored noise, rather than the white noise, we use the Ungerboeck model in [13]. And the branch label used in a BCJR-type algorithm from a particular state associated with σ_k for the input symbol a_k is given by:

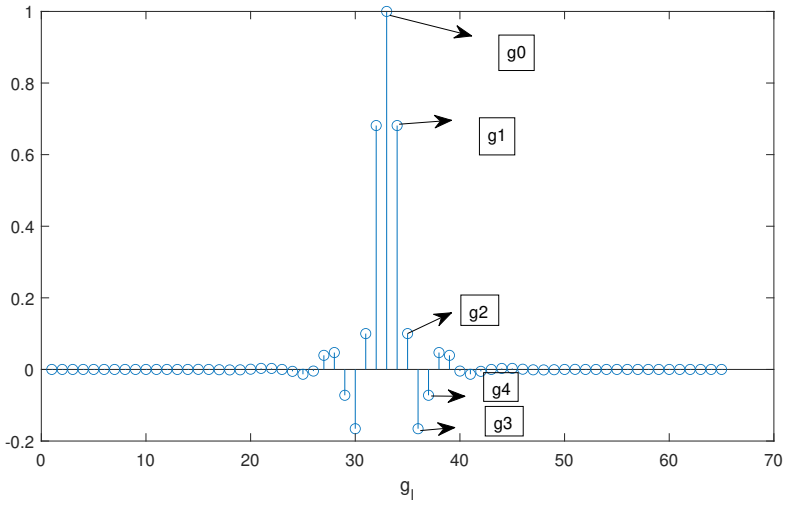

Figure 2.6: Discrete time convolution sequence g_l

$$\gamma\left(\sigma_k, a_k\right) = \exp\left(\frac{\mathcal{R}\left[2a_k^*\left(x_k - \text{isi}\left(a_k, \sigma_k\right)\right)\right]}{N_0}\right) P_{\text{a}}\left(a_k\right) \tag{2.11}$$

where $P_a(a_k)$ is the priori probability on a_k.

The purpose of interleaver and deinterleaver is to maintain relative independence between symbols. An illustration of the turbo iterative equalization structure and SC-LDPC code structure are the main tasks of this thesis and will be discussed in detail in the following chapter.

2.2.2 FTN system spectral efficiency

Comparing the SE by using the Nyquist system of raised cosine filter with ($B_{Nyquist} = (1+\alpha)/(2T_s),\ R = 1/T_s$):

$$\eta_{Nyquist} = \frac{R_{Nyquist}}{B_{Nyquist}} = \frac{2}{(1+\alpha)} \tag{2.12}$$

In an FTN system, the symbol rate changes to $R_{FTN} = 1/(\tau T_s)$, thus the SE becomes:

$$\eta_{FTN} = \frac{R_{FTN}}{B_{FTN}} = \frac{2}{\tau(1+\alpha)} \tag{2.13}$$

In order to express the improvement from Nyquist system to FTN system, we define the difference between those two systems over the Nyquist system, and obtain the ratio:

$$\triangle\eta = \frac{\eta_{FTN} - \eta_{Nyquist}}{\eta_{Nyquist}} = \frac{1}{\tau} - 1 = \frac{1-\tau}{\tau} \tag{2.14}$$

Figure 2.7 depicts the SE improvement of FTN system along with the changing τ: It can be seen that when τ is smaller, the SE of FTN system has great improve-

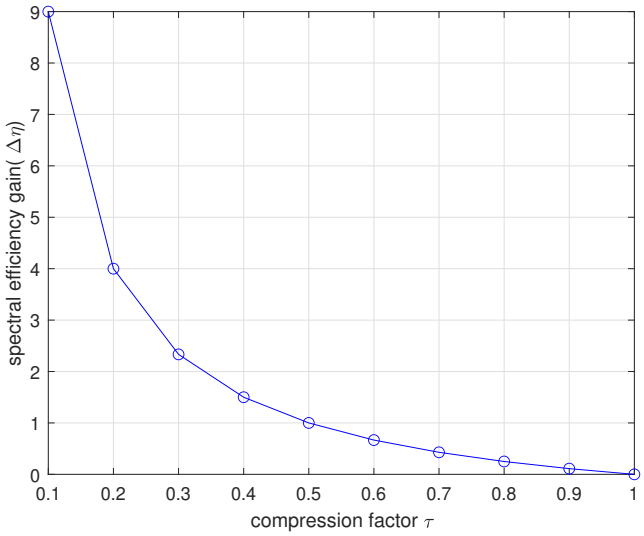

Figure 2.7: Spectral gain with τ

ment, this improvement gradually decreases along with the increasing τ, when $\tau = 1$, FTN system equals Nyquist system and without gain.

2.3 Turbo equalization

Turbo equalization technology combines the equalizer and the decoder, similar to turbo code processing. Through multiple iterations, the soft information between the equalizer and the decoder is fully exchanged to obtain iterative gain and improve system performance. It is assumed that at the transmitter, the information bits **b** go through the encoder, interleaver, and symbol mapping and then cross the channel, where the intersymbol interference and white Gaussian noise are introduced. Turbo equalization algorithms include the Maximum a Posteriori (MAP) algorithm and MMSE algorithm. The MAP algorithm has the best performance, and the Minimum mean square error (MMSE) equalization algorithm has a lower performance than the MAP algorithm, but the computational complexity is lower.

Figure 2.8 below shows the turbo equalization system model.

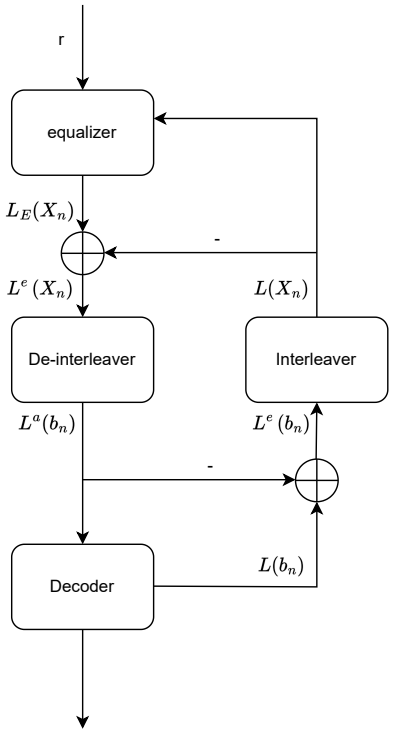

Figure 2.8: Block diagram of MAP equalizer

The FTN demodulation structure is shown in Figure 2.8. $L(x_n)$ and $L(b_n)$ are prior information of the equalizer and decoder [14]. $L_e^E(x_n)$ is the output of the sequence $\mathbf{r}$ after the MAP equalizer. Information is exchanged between the MAP equalizer and the decoder.

When the receiver receive the sequence $\mathbf{r} = [r_0, r_1, \cdots, r_{N-1}]$, according to MAP algorithm, the output of the MAP equalizer can be written as:

$$L_E(X_n) = \ln \frac{P(X_n = 0 \mid \mathbf{r}, L(\mathbf{X}))}{P(X_n = 1 \mid \mathbf{r}, L(\mathbf{X}))} = L^e(X_n) + L(X_n) \tag{2.15}$$

$L^e(X_n)$ and $L(X_n)$ are called the prior information and extrinsic information of the equalizer, respectively. For the decoder, the input sequence is $L(C_n)$. Then the output LLR is given by:

$$\begin{aligned} L(b_n) &= \ln \frac{P(b_n = 0 \mid L^a(\mathbf{b}))}{P(b_n = 1 \mid L^a(\mathbf{b}))} \\ &= \ln \frac{P(\mathbf{r} \mid b_n = 0)}{P(\mathbf{r} \mid b_n = 1)} + \ln \frac{P(b_n = 0)}{P(b_n = 1)} \\ &= L^e(b_n) + L^a(b_n) \end{aligned} \tag{2.16}$$

$L^e(b_n)$ and $L^a(b_n)$ are called the prior information and extrinsic information of the decoder, respectively.

After some iterations, the estimated sequence of data sent by the transmitter can be obtained, which is given by:

$$\hat{b}_n = \arg\max P(b_n = d \mid \mathbf{r}), d \in \{0,1\} \tag{2.17}$$

When the iteration begins, the MAP equalizer uses the MAP equalization algorithm to calculate the posterior probability based on the channel received sequence. The equalizer outputs the extrinsic information, which then passes through the de-interleaver and outputs the prior information of the decoder. The prior information is sent to the decoder, and after decoding, the decoder outputs the extrinsic information. Then the extrinsic information is sent to the interleaver and after interleaving, the information is sent back to the equalizer again. This entire iterative process is one iteration. When the maximum number of iterations is reached, the output of the decoder is judged to obtain the final information bits. It is called Turbo equalization because the feedback and iterative schemes are similar to the decoding process in Turbo codes.

2.4 Codes on graphs and BCJR algorithm

In 1974, Bahl et al. invented the BCJR algorithm, which is very important for iterative error correction decoding. At present, turbo decoding is based on the BCJR algorithm for iterative decoding. Silvio A and Abrantes explained the principle of the BCJR algorithm and gave examples in [19]. The following is a brief derivation and description of the steps of the BCJR algorithm. Since the decoding of turbo codes requires a component code decoder with soft input and soft output, the BCJR algorithm, or its simplified algorithm Log-MAP algorithm is usually used for the decoding scheme of turbo codes. The BCJR algorithm operates in the probability domain and is the optimal decoding algorithm for the trellis structure coding scheme.

Theoretically, the posterior probability of each information bit can be accurately calculated using the BCJR algorithm. But the algorithm is computationally expensive. Therefore, to simplify the calculation, the logarithmic operations can be used to calculate the maximum posterior probability, which is the Log-MAP algorithm. In the Log-MAP algorithm, the original complex multiplication and division operations in the probability domain become simple addition and subtraction operations in the logarithmic domain, thereby the complexity of the algorithm is reduced.

The figure below depicts the structure of the trellis.

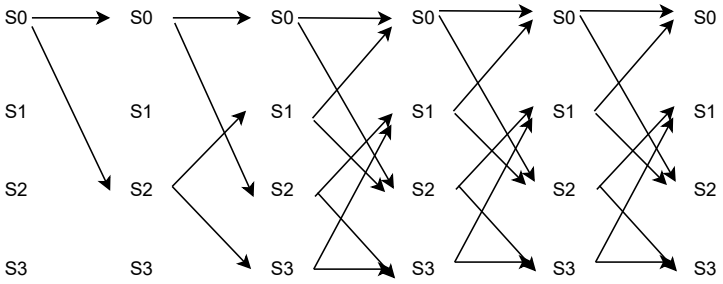

Figure 2.9: Graph of trellis

The BCJR algorithm calculates the posterior L value for each information bit, where U_L is the decoder output and r is the observation.

$$L\left(U_L\right) = \ln\left(\frac{P(U_L = +1|r)}{P(U_L = -1|r)}\right) \tag{2.18}$$

Then the decoder output is given by:

$$U_L = \begin{cases} +1, L\left(U_L\right) > 0 \\ -1, L\left(U_L\right) < 0 \end{cases} \tag{2.19}$$

In iterative decoding, the APP L value can be regarded as the decoder output. The forward recursion, backward recursion and branch metrics is defined as $\alpha_{l+1}(s)$, $\beta_l\left(s'\right)$ and $\gamma_l\left(s', s\right)$. The forward recursion can be calculated by:

$$\alpha_{l+1}^*(s) \equiv \ln \alpha_{l+1}(s) = \ln \sum_{s' \in \sigma_l} \gamma_l\left(s', s\right) \alpha_l\left(s'\right) = \underset{s' \in \sigma_l}{\overset{*}{\max}} \left[\gamma_l^*\left(s', s\right) + \alpha_l^*\left(s'\right)\right] \tag{2.20}$$

where $l = 0, 1, \cdots K-1$. The backward recursion can be calculated by:

$$\beta_l^*\left(s'\right) \equiv \ln \beta_l\left(s'\right) = \underset{s \in \sigma_{l+1}}{\overset{*}{\max}} \left[\gamma_l^*\left(s', s\right) + \beta_{l+1}^*(s)\right], l = K-1, K-2, \cdots 0 \tag{2.21}$$

The branch metrics is given by:

$$\gamma_1\left(\mathbf{s}', \mathbf{s}\right) \equiv \ln \gamma_1\left(\mathbf{s}', \mathbf{s}\right) = \begin{cases} \frac{u_l \cdot L_a(u_l)}{2} + \frac{L_c}{2} r_l \cdot v_l, l = 0, 1, \cdots K-1 \\ \frac{L_c}{2} r_l \cdot v_l, l = h, h+1, \cdots, K-1 \end{cases} \tag{2.22}$$

The steps of the BCJR algorithm:

1. Initialize forward and backward recursions $\alpha_0(s)$ and $\beta_N(s)$

2. Compute branch metrics $\alpha_{l+1}(s)$

3. Carry out forward recursion $\alpha_{l+1}(s)$ based on $\alpha_l(s)$

4. Carry out backward recursion $\{\beta_{l-1}(s)\}$ based on $\{\beta_l(s)\}$

5. Compute APP L-values

$$L(u_k) = \log \frac{\sum_{S_+} p(s_{k-1} = s', s_k = s \mid \mathbf{y})}{\sum_{S_-} p(s_{k-1} = s', s_k = s \mid \mathbf{y})} \tag{2.23}$$

2.5 LDPC codes introduction

Low-density parity check codes are defined by a sparse $M \times N$ parity check matrix by Gallager in [21]. An LDPC code v is an (N, K) linear block code with a code length of N and an information sequence length of K, which can be uniquely determined by its check matrix $\mathbf{H}$. The dimension of $\mathbf{H}$ is $M \times N$, and each row corresponds to a check equation, also called a check node. Each column corresponds to a bit of a codeword, also called a variable node. The number of non-zero elements in each row is called row weight, and the number of non-zero elements in each column is called column weight. The following is a 4×6 check matrix and its corresponding check equation:

$$\mathbf{H} = \begin{bmatrix} 110100 \\ 011010 \\ 100011 \\ 001101 \end{bmatrix}$$

$$\text{which means: } \begin{cases} v_0 + v_1 + v_3 = 0 \\ v_1 + v_2 + v_4 = 0 \\ v_0 + v_4 + v_5 = 0 \\ v_2 + v_3 + v_5 = 0 \end{cases}$$

Decoding algorithms are mainly divided into hard-decision decoding algorithms and soft-decision decoding algorithms. The main principle of the hard-decision decoding algorithm is to complete the decoding process by comparing the number 1 and 0 in the verification result through a series of orthogonal equations. Hard-decision decoding algorithm has a simple decoding structure and low complexity but has limited applications.

One typical soft-decision decoding algorithm is the BP algorithm. The transmission form of the information in the BP algorithm is the log-likelihood ratio (LLR), whose purpose is to calculate an approximation of the maximum posterior probability of each bit. When the iteration begins, each variable node will update the information of the node until the whole decoding process ends. The BP algorithm has excellent performance close to the Shannon limit but the message calculation complexity of the check node is complicated.

2.6 Protograph-based code construction

In order to derive the spatially coupled LDPC (SC-LDPC) codes constructed by protograph, we begin with a brief introduction to LDPC-BCs and LDPC-CCs and then illustrate the codeword structure based on protograph. In this thesis, we mainly discuss protograph based codes.

2.6.1 LDPC block codes

LDPC codes are initially proposed by Gallager in 1962, and Hamming codes and LDPC codes are both block codes (BC). A (J, K)-regular LDPC-BC is defined by a sparse binary parity-check matrix $\mathbf{H}$, where each row of $\mathbf{H}$ contains exactly K ones, each column of $\mathbf{H}$ contains exactly J ones, and both J and K are small compared with the number of rows in $\mathbf{H}$. Additionally, the irregular LDPC-BC means that J and K are not constant. Commonly, a parity-check matrix $\mathbf{H}$ is presented by Tanner graph [15]. Each column of $\mathbf{H}$ corresponds to a variable node, and each row represents a check node. If the position (i, j) of $\mathbf{H}$ equals 1, which means there is an edge connecting check node i with variable node j, the positions of entry 0 mean there is no edge connection. In addition to the requirement of the matrix $\mathbf{H}$ should be sparse, the most significant difference between LDPC codes and classic block codes is their decoding method, the classic block codes commonly use the maximum likelihood (ML) algorithm but LDPC codes can use a graphical representation of its parity check matrix for iterative decoding, so design with the properties of $\mathbf{H}$ as key.

2.6.2 LDPC convolutional codes

Felstrom et al.[16] presents that low-density parity-check convolutional codes can be described by a semi-infinite diagonal type parity-check matrix $\mathbf{H}$, for example, a code rate $R = n_b/n_c$ LDPC-CCs is:

$$\mathbf{H}_{[\infty]} = \begin{bmatrix} H_0(1) & & & & \\ H_1(1) & H_0(2) & & & \\ \vdots & H_1(2) & \ddots & & \\ H_{m_s}(1) & \vdots & \ddots & H_0(t) & \\ & H_{m_s}(2) & \ddots & H_1(t) & \ddots \\ & & \ddots & \vdots & \ddots \\ & & & H_{m_s}(t) & \ddots \\ & & & & \ddots \end{bmatrix} \tag{2.24}$$

where the entry $H_i(t), i = 0, 1, ...m_s$ is the binary matrix with dimension $(n_c - n_b) \times n_c$ and satisfy the following conditions:

1. $H_i(t) = 0, i < 0 \quad and \quad i > m_s, \forall t \geq 1$;

2. $\exists t > 0, H_{m_s} \neq 0$;

3. $H_0(t)$ is full rank, $\forall t \geq 1$

The parameter m_s is the memory length of codeword, $v_s = (m_s + 1)n_c$ is constraint length. The first two conditions make sure that codewords have memory length, and the third condition ensures that the parity-check matrix is full rank. The Hamming weights of each column must be very low here. If $H_i(t) = H_i(t + \tau), \forall i = 0, 1, ..., m_s, \forall t, \tau > 1$, then it is periodic LDPC-CCs. When $\tau = 1$, that is time-invariant; otherwise, it is time-varying.

The following parity check matrix of the truncated LDPC convolutional code is finite, that is, the convolutional code terminates instantaneously.

$$\mathbf{H}_{[L]} = \begin{bmatrix} H_0(1) & & & & \\ H_1(1) & H_0(2) & & & \\ \vdots & H_1(2) & \ddots & & \\ H_{m_s}(1) & \vdots & \ddots & H_0(L) & \\ & H_{m_s}(2) & \ddots & H_1(L) & \\ & & \ddots & & \\ & & & H_{m_s}(L) & \end{bmatrix} \tag{2.25}$$

If the number of 1 in each column is J, and the number of 1 in each row except the first row and the last $m_s(n_c - n_b)$ row is K, that ignores the terminated part, then these codewords are regular. In this case, L represents the coupling length.

2.6.3 Tanner graph and protograph

In addition to using the parity-check matrix to present LDPC code, Tanner graph and protograph can also describe code construction. And the Tanner graph is more intuitively represented in the description of the decoding algorithm.

Tanner graph is an undirected graph, and basic elements are two kinds of nodes (check nodes and variable nodes) and the edge which connects two kinds of nodes. In the Tanner graph, each variable node in the graph corresponds to each column of the **H** matrix, and the check node corresponds to a row. On each node, the number of connected edges is the degree of the corresponding node, that is row weight or column weight.

A simple check matrix and its Tanner graph (Figure 2.10) are given.

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{2.26}$$

Figure 2.10 is the Tanner graph expression of the **H** matrix.

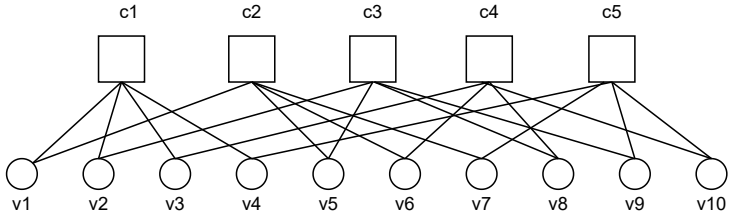

Figure 2.10: Tanner graph of check matrix $\mathbf{H}$

2.6.4 Protograph-based LDPC code construction

The longer the length of the LDPC code, the larger the size of the check matrix, the better the sparsity, and the better error correction performance. A protograph is a Tanner graph with relatively fewer nodes, in which parallel edges are also allowed. This means there may be several edges that connect a VN to a CN. By using 'copy and permutation' operation, the Tanner graph of a corresponding LDPC code can be obtained from a given protograph. And the resulting Tanner graph is called the derived graph.

The example in Figure 2.11 [17] is given to depicting the whole steps from a single protograph to the derived graph. The middle part in the Figure 2.10 represents the copy operation from a given protograph (the left part) and then is followed by the edge permutation operation which means the protograph is permuted among the three copies of the corresponding VNs and CNs. Here, $N = 9$, which means the first protograph is copied three times, and $M = 3$, which is called the lifting factor.

The construction process of the protograph-based SC-LCPC code is very similar to the protograph LDPC code, the difference is that the object of the 'copy and replacement' operation is not a single original pattern, but a whole piece of the coupled chain formed by the unrelated protograph, each protograph presents an LDPC-BC, thus the whole coupled chain is protograph based SC-LDPC. We will introduce details later for it.

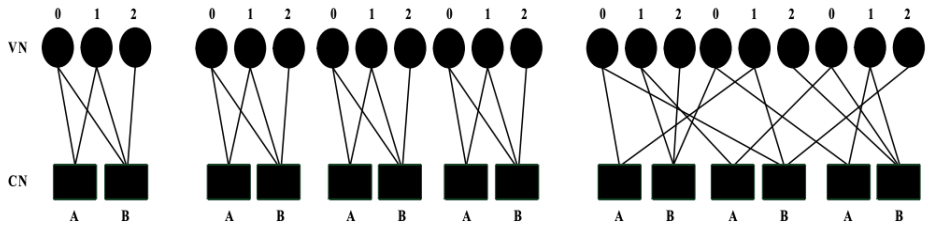

Figure 2.11: Protograph to the derived graph (Li Deng, 2021)

2.6.5 Loops in LDPC codes

A loop that exists in the LDPC codes is easily noticed from the Tanner graph. It is a closed loop that starts from any point and returns to the origin after going through multiple edges, and this includes variable nodes, check nodes, and end-to-end closed loop is so-called LDPC loop.

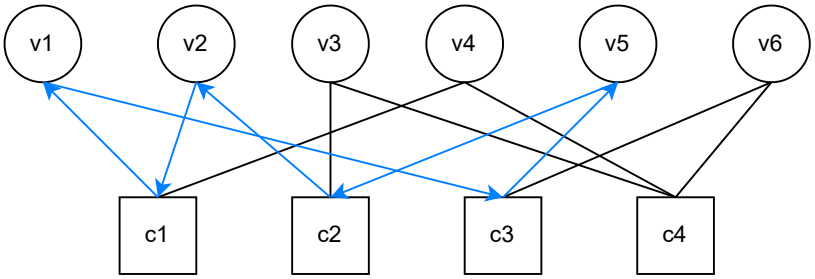

Figure 2.12: (6,3,2) regular LDPC code loop

Figure 2.12 represents a (6,3,2) regular LDPC code loop. Here the length of the codeword is 6 which is as same as the loop length (represented by the blue arrow line). Normally, the shortest loop is called girth for this Tanner graph. Due to the existing loop, LDPC codes can transfer information repeatedly in traditional Belief Propagation (BP) decoding case. If there is any error information, then it will add up which results in the degradation of decoding performance. Assumed that for the loop with length 4, the information returns back the origin after 2 iterations. For the current point, the received information is the output only from the last iteration, this alternate transmission is lack of independence, what's more, in this case, the linear correlation will appear after 2 iterations, meanwhile, if some error occurs in any iteration process, the error information will repeatedly add up which influence the decoding performance directly. Therefore, according to [18] mentions, the loop length should be no less than 6 when constructing the LDPC check matrix.

Chapter 3
FTN with turbo equalization

3.1 Communication systems with FTN signaling based on convolutional codes

In digital mobile communication, due to the multi-path transmission effect, the limited bandwidth, and the unsatisfactory characteristics of the communication channel, it is inevitable that inter-symbol interference will occur when the communication data is transmitted through the channels. The equalization technology can deal with the inter-symbol interference accordingly, and then compensate for the distortion of the communication channel. Generally, the processor that compensates for the corresponding inter-symbol interference is called an equalizer. Figure 3.1 shows the transmitting and receiving model of the system.

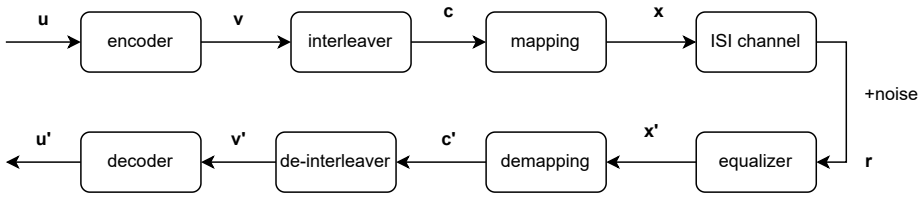

Figure 3.1: Transmitting and receiving structure

As shown in Figure 3.1, the binary data bit sequence **u** is sent to the encoder with code rate R at the transmitting end to generate encoded data **v**. After convolutional coding, the data **v** is sent to the interleaver, where data **v** is randomly scrambled by the interleaver to form a new data sequence **c**. Then the data is mapped into M-ary modulation data **x**.

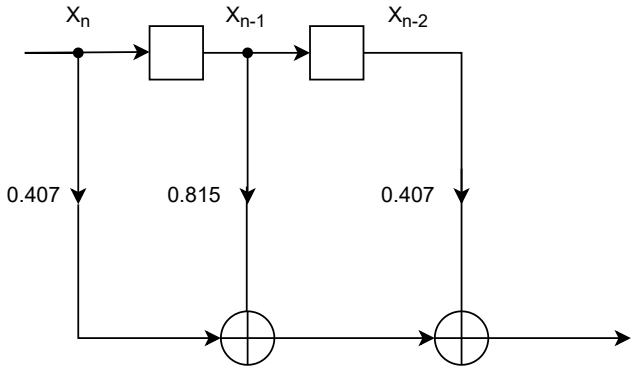

Figure 3.2: Tap delay line model of the channel

In [19], an ISI channel **h** = [0.407 0.815 0.407] is introduced, which can be seen as a tapped delay line shown in Figure 3.2 [20]. According to Figure 3.2, the state transition table can be obtained and together with the output results which will be described in the trellis. The trellis shown in Figure 3.3 can be constructed and the calculation of posterior probability can be done.

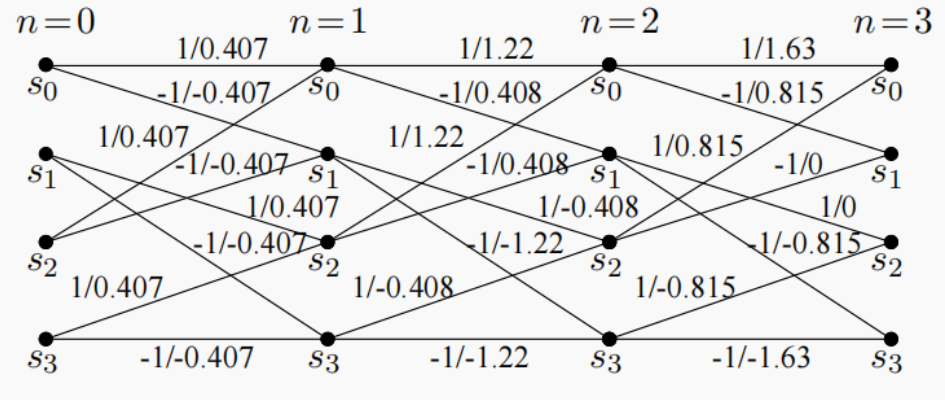

Figure 3.3: Trellis for a length-three ISI channel (Tuchler, 2011)

Since the FTN system increases the symbol rate at the transmitter and the orthogonality between symbols is influenced, which will introduce inter-symbol interference at the receiver. And as the FTN symbol speed increases, the degree of inter-symbol interference will further increase. The increase in the degree is not only reflected in the increase in the correlation coefficient between symbols but also in the memory length of the inter-symbol interference.

As shown in Figure 2.4 in Chapter 2, the transmitted binary bit stream is encoded, interleaved, and modulated to obtain the modulation symbol $\{a_k\}$, and then the FTN signal $s(t)$ is generated by the FTN module, and $\{x_k\}$ is obtained through a matched filter after introducing noise. The FTN mapping module is composed of an oversampling and a pulse shaping filter. The amount of oversampling can

be represented by τ, and by adjusting τ, different acceleration factors in the FTN system can be obtained. When τ =1, it is the Nyquist transmission system, and when $0 < \tau < 1$, it is the FTN transmission system.

Figure 3.4 shows the simulation results of turbo equalization with different numbers of iterations using convolutional codes with $\tau = 0.5$. In the simulation, we choose the codeword length $N = 64800$ with $R = 1/2$ convolutional codes, the memory length $m = 3$ with $g_0[D] = 1 + D^2 + D^3$ and $g_1[D] = 1 + D + D^3$. To compare the code performance with LDPC in the same parameter setting, we use different τ, the same roll-off factor of 0.4 and the same raised-cosine pulses as (2.6). It can be seen that performance improves with iteration. With 10 iterations, take BER equals 10^{-4}, there is a gain of about 1.2dB compared to 1 iteration. However, the gain decreases as the number of iterations increases. There is nearly no gain when increasing the number of iterations from 5 to 10.

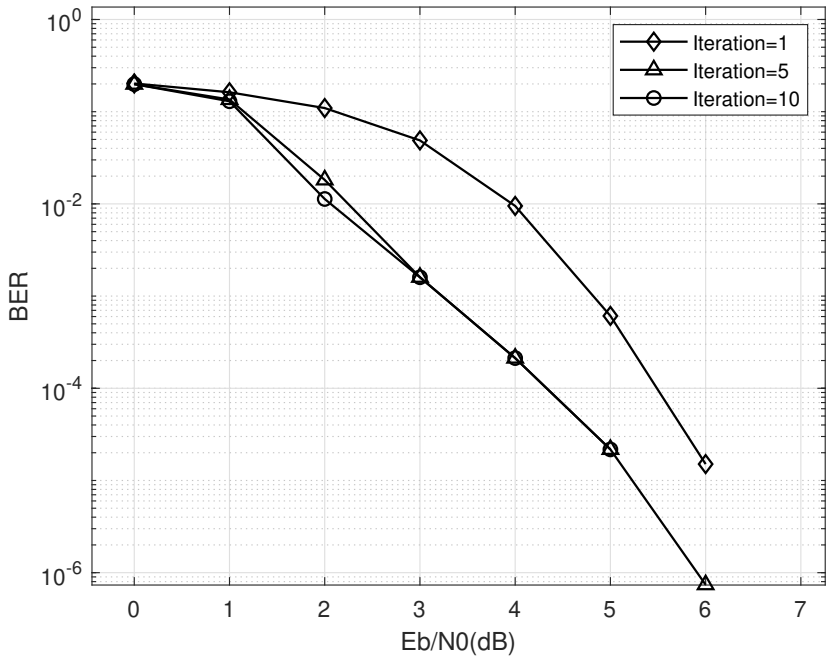

Figure 3.4: Results of turbo equalization with MAP equalizer and convolutional codes with $\tau = 0.5$

Figure 3.5 shows the simulation results of convolutional codes using different τ with an iteration number of 10. It can be seen that the bit error rate increases as τ decreases in a certain SNR. When the number of iterations is 10 and the BER is 10^{-4}, $\tau = 0.5$ has a 0.2dB loss compared to $\tau = 0.74$.

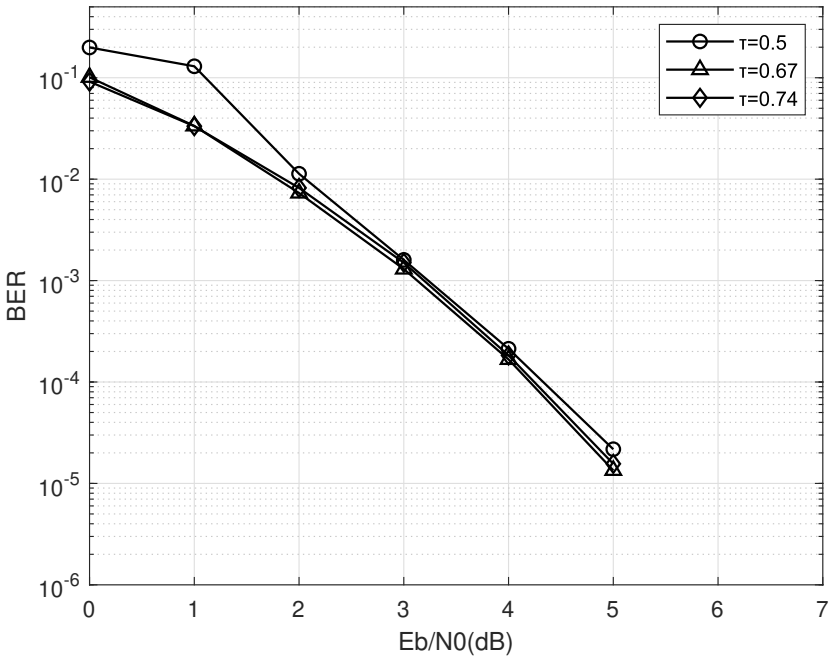

Figure 3.5: Results of turbo equalization with MAP equalizer and convolutional codes, iteration = 10

It can be seen from Figures 3.4 and 3.5 that for convolutional codes, the increase in the number of iterations can improve the system performance, but when the number of iterations increases to a certain extent, the performance improvement is less obvious. The change of τ is not very significant for the FTN system using convolutional codes along with the increasing SNR region. From [20], it can be explained that when the iterations increase, the cycles in the interleaver and the overall graph make the extrinsic LLRs from the code and channel become more and more correlated. These cycles may cause some variables to receive the information from themselves.

3.2 Communication systems with FTN signaling and turbo equalization based on LDPC codes

This subsection discusses the system performance with different τ when replacing convolutional codes with LDPC codes. At the same time, a curve without

FTN was added for comparison. In the case of not using FTN, the BP decoding is used. The principle of BP decoding will be briefly introduced in Section 4.2. Next, we analyze the impact on the system bit error rate performance of the FTN system when τ changes. Table 3.1 gives the simulation conditions of the system.

parameters	value
Channel	AWGN channel
roll off factor α	0.4
The maximum iterations	5
Decoding(equalization) algorithm	BP-decoding for No-FTN MAP Equalization for FTN
τ	0.5, 0.67, 0.74, 1
Code rate R	1/2
modulation type	QPSK

Table 3.1: Simulation parameters of convolutional and LDPC codes

The simulation results are shown in Figure 3.6. As can be seen from the graph, compared with LDPC codes, convolutional codes have lower bit error rates in the lower SNR region (below around 0.7dB). Along with the increasing of SNR, there will be a crossover point with corresponding curves using LDPC codes, which means after these points are within a certain range, LDPC codes outperform convolutional codes. When the BER is 10^{-3}, there are 0.5dB, 0.8dB, and 2.5dB gains respectively with τ = 0.74, 0.67, and 0.5 compared with non-FTN case. It can be analyzed from the data that when the compression factor τ = 0.74, the bit error rate performance of the FTN system using turbo equalization is only 0.5dB below the Nyquist system. When τ is reduced to 0.5, the performance of the system deteriorates significantly. This is because when τ decreases, the channel becomes worse and worse, and ISI increases, which makes the system require more energy to reach the same performance.

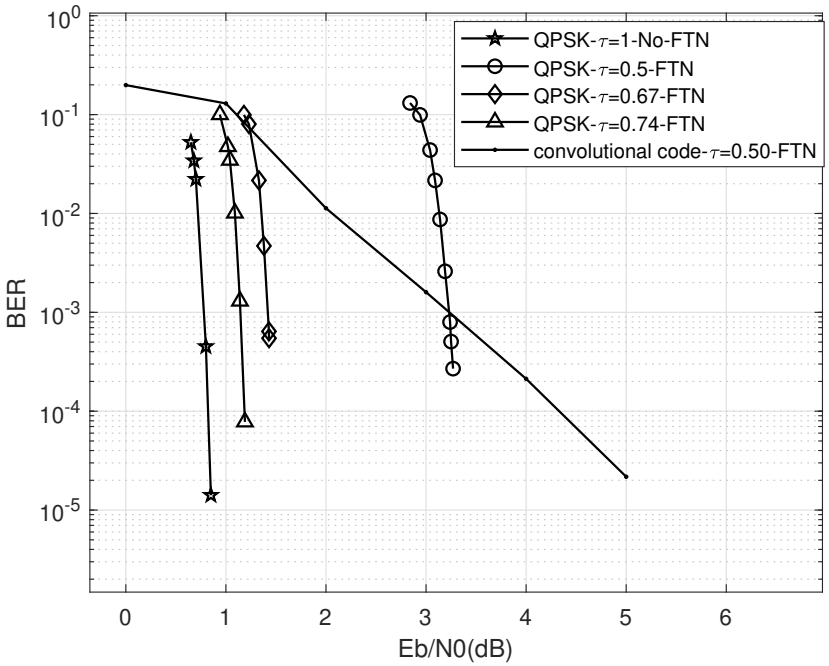

Figure 3.6: Results of turbo equalization with MAP equalizer and LDPC codes

3.3 Trade-off between τ and code rate

According to the introduction from Chapter 2, FTN introduces the time acceleration factor τ to make the pulses come faster thus improving the SE. What about the effect on SE by changing the code rate, Figure 3.7 gives the answer.

From Figure 3.7, three groups of different colors represent systems with different spectral efficiency η = 4/3, 3/2, and 2 bit/s/Hz respectively. It can be seen from the blue curves with $\eta = 4/3$ bit/s/Hz, at the BER equals 10^{-4}, using FTN can obtain around 0.472dB gain with a smaller code rate $R = 1/2$ compared with R = 2/3 without FTN. Here, the only variable compared with that in Figure 3.6 is the code rate.

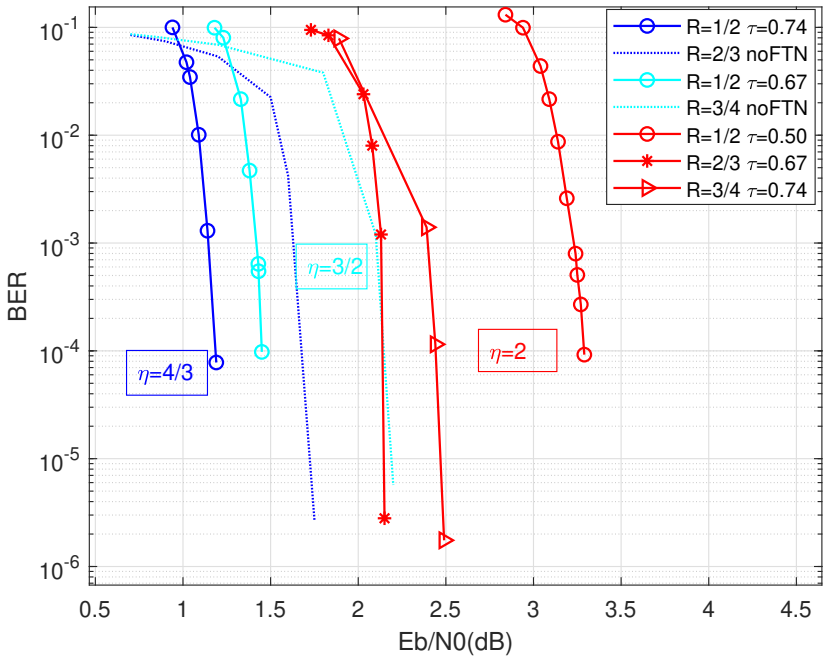

Figure 3.7: Results of LDPC codes with FTN in different cases

The similar results from the cyan group shows that under the condition of ensuring equal $\eta = 3/2$ bit/s/Hz, using FTN technique with rate $R = 1/2$ can gain around 0.68dB at the BER of 10^{-4} compare to no-FTN with code rate $R = 3/4$. The red group depicts using the different τ and different code rates to get the same $\eta = 2$ bit/s/Hz. Under the condition of BER equals 10^{-4}, using code rate $R = 2/3$with $\tau = 0.67$ FTN system can obtain 0.3dB energy gain than using code rate $R = 3/4$ with $\tau = 0.74$ FTN system, moreover, there is a significant gain (1.14dB) compared with that with code rate $R = 1/2$ and $\tau = 0.5$, so there is a trade-off when cooperating the code rate and FTN acceleration factor to get good performance.

What's more, the FTN system can also be analyzed from the perspective of efficiency. When the bit error rate of the system is fixed at 10^{-4}, the code rate R of different FTN systems is changed, according to $\eta = R\log_2 M/\tau$, the curves shown in Figure 3.8 can be obtained.

It can be seen that when we take the same SNR, the FTN system can obtain

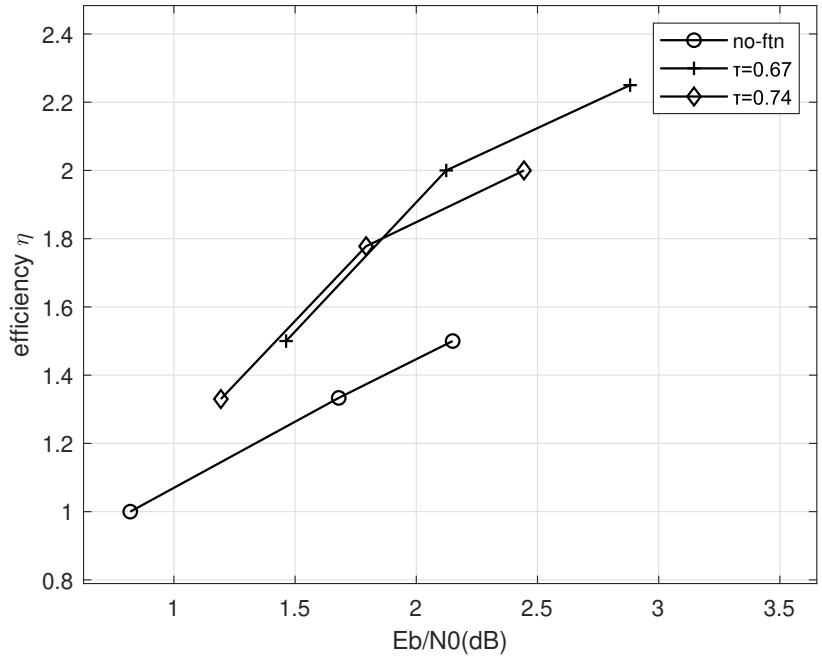

Figure 3.8: Efficiency performance of the system with different τ(BER=10^{-4})

higher efficiency than the system without FTN. When taking η = 1.5 as an example, the FTN system with τ = 0.67 has a gain of 0.7dB compared to the system without FTN. In all of three curves, the three points from left to right in each curve represent the code rates R =1/2, 2/3, 3/4 respectively.

Chapter 4

Spatial coupling

4.1 Protograph-based SC-LDPC construction

LDPC convolutional codes (LDPC-CCs) were proposed in [16] by introducing memory into the encoding procedure of LDPC codes. These codes are also known as spatially coupled LDPC codes, which can be obtained by the edge spreading operation from LDPC codes over the graph representation. Generally, a regular protograph can be expressed as (J, K), J is the number of edges connected with variable nodes and K represents the number of edges connected with check nodes. Figure 4.1(a) [21] shows an example protograph with (3,6), then, make L copies of this graph, where L is called coupling length, and each of these copies is indexed by t, $t = 1, ..., L$. Defining w as the coupling memory, then the edge of the variable nodes (VNs) at t is connected to the checking nodes (CNs) at $t, t+1, \ldots, t+w$, here $w = 2$. We define M as a lifting factor, which means each edge in the protograph becomes a bundle of M edges, connecting M copies of a variable node to M copies of a check node.

The protograph of a (3,6) SC-LDPC is shown in Figure 4.1 (d). There are extra w check nodes added to truncate the edge spreading. As [22] mentions, since the left and right boundary do not have right neighbors the overall chain graph is slightly irregular at the boundary. This structure makes the codes more powerful, therefore on the decoder side, the original bits can be recovered with a lower error rate at the boundaries.

The protograph can be represented by its $n_c \times n_v$ (here, $n_c = 1$ and $n_v = 2$) base biadjacency matrix $\mathbf{B}$, where $B_{i,j}$ is taken to be the number of edges connecting variable node v_j to check node c_i. Generally, there are multiple edges connecting two nodes, which corresponds to entries in $\mathbf{B}$ greater than 1. The corresponding matrix $\mathbf{B}$ in Figure 4.1(a) [21] is $\mathbf{B}$ = [3 3], due to the edge spreading operation, $\mathbf{B}$ can be divided to a set of sub-matrix B_i, i = { 0,1,...,w }, $B = B_0 + ...B_w$. In this ensemble, B_i=[1 1], $i = \{0, 1, 2\}$. Furthermore, the $\mathbf{B_{sc}}$ matrix of the corre-

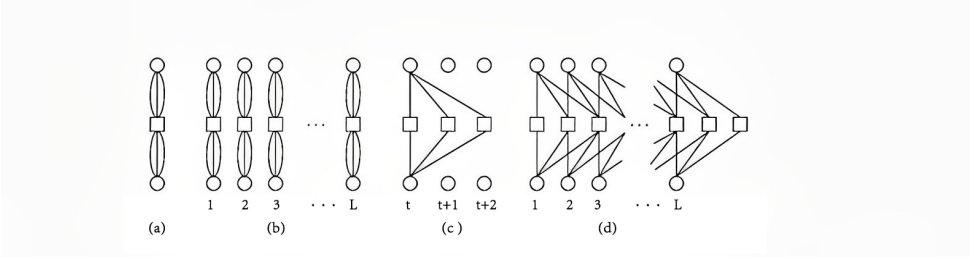

Figure 4.1: (a) A (3, 6)-regular LDPC protograph, (b) L replicas of (3, 6)-regular LDPC protographs, (c) process of edge spreading with w = 2, and (d) protograph of the (3, 6) SC-LDPC codes with w = 2 (Yamei Zhang, 2020)

sponding SC-LDPC ensemble is:

$$\boldsymbol{B}_{\text{sc}} = \begin{bmatrix} \boldsymbol{B}_0 & \cdots & \cdots & \cdots \\ \boldsymbol{B}_1 & \boldsymbol{B}_0 & \cdots & \cdots \\ \vdots & \boldsymbol{B}_1 & \cdots & \cdots \\ \boldsymbol{B}_w & \vdots & \cdots & \boldsymbol{B}_0 \\ \vdots & \boldsymbol{B}_w & \cdots & \boldsymbol{B}_1 \\ \vdots & \vdots & & \vdots \\ \cdots & \cdots & \cdots & \boldsymbol{B}_w \end{bmatrix}_{[(L+w)n_c \times n_v L]} \tag{4.1}$$

To obtain the sparse parity-check matrix of an SC-LDPC code, graph lifting should be used by M, which is replacing each non-zero entry in $\mathbf{B}$ by a sum of $\mathbf{B_i}$ permutation matrices of size $M \times M$ and each zero entry by the $M \times M$ all-zero matrix. For example, assume the base matrix $\mathbf{B}$ = [3 3], the corresponding parity-check matrix corresponding to an M-lifting of this base matrix is

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{\Pi}_{1,1} & \boldsymbol{\Pi}_{1,2} \end{bmatrix} \tag{4.2}$$

The dimension of $\mathbf{H}$ is $(L+w)M$ by LM. Similarly, the $\mathbf{H}$ matrix shows the diagonal band structure with non-zero entries, the maximum length of this diagonal band is $L_c = (w+1)Mn_v$, which is called the constrain length of an SC-LDPC code. And the design rate is:

$$R = 1 - \frac{n_c(L+w)M}{n_v LM} = 1 - \frac{n_c(L+w)}{n_v L} \tag{4.3}$$

Obviously, the extra w check nodes result in a rate loss of the SC-LDPC codes, which required an improved coupling structure to figure out.

4.2 Window decoding

SC-LDPC codes have similar structures to LDPC codes. In principle, SC-LDPC codes can use BP decoding as well, but considering the traditional BP decod-

ing starts to decode until it receives the whole codewords sequence information which will result in higher latency and complexity in longer SC-LDPC sequences ensemble due to greater coupling length L and lifting factor M. However, this can be improved by using Window Decoding (WD). Figure 4.2 illustrates the WD process.

In Figure 4.2 [23], the sliding window decoder of size W operates on the matrix

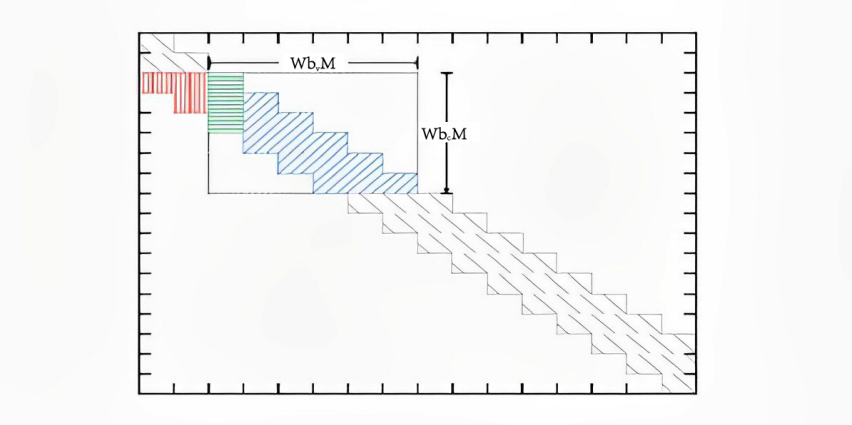

Figure 4.2: Window Decoding process for SC-LDPC codes (Hua Zhou, 2019)

$\boldsymbol{H}$ with dimension $Wb_cM \times Wb_vM$, where b_c and b_v refer to the numbers of rows and columns of the component base matrix B_i and this size determines the number of (check nodes) CN, (variable nodes) VN relatively. The window size W is much less than coupling length L which means the complexity of WD does not depend on L. In principle, $m + 1 \leq W \leq L$.

The principle of window decoding is that the window runs along the diagonal and slides to the right corner, meanwhile, using the traditional BP decoding algorithm. In Figure 4.2, at every window position t, the first Mb_v symbols at time instant t are decoded thus termed as target symbols (the green part in the above graph). Moreover, the red part in the graph is recorded LLR values for parity check in the current window. When finishing decoding at t instant, the window slides Mb_c rows down and Mb_v columns right towards $t + 1$ instant for preparing the next Mb_v check nodes. Moreover, saving the LLR values from previous symbols decoding, rather than initializing the target symbol again by using received LLR from the channel. Continue like this until the window slides out the full **H** matrix. It is noticeable that error transmission might occur in this decoding process. For example, as Figure 4.2 shows, assuming the coupling memory is w =2 here, and this is the $(p + w)$th window, and the red parts come from pth window and $(p + 1)$th window regarding as the related information of target symbols and join the $(p + w)$th window decoding process. In this case, if the red parts carry the wrong output log-likelihood ratio which will cause a direct impact on the target symbol. What's more, in the decoding of the multiple target symbols output, the

error propagation will be more serious due to the lack of iterations.

Considering there is some performance loss by using window decoding compared with using BP decoding directly on the whole chain construction, and the decoding performance in a certain window depends on the output from the previous window. Therefore, in order to achieve the ideal decoding performance, when the decoding results are not optimal, partial data information should be re-transmitted so that provide more apriori information for decoding.

The structural latency for the window decoder T_{wd} in terms of the number of information bits can be expressed as [24].

$$T_{wd} = T_w + DT_{dec}(W) \tag{4.4}$$

where T_w refers to the needed waiting time for codewords information before the first window starts decoding, T_{dec} refers to the iterative decoding time within each window, D represents the sum of overall window numbers. In general, the slide step is 1, which satisfies the relation, $D = L - W + 1$.

4.2.1 Belief propagation algorithm

Assume the codewords sequence $\mathbf{x} = [\mathbf{x_0}, ..., \mathbf{x_{n-1}}]$ with QPSK modulation transmitted in Additive White Gaussian Noise (AWGN) channel, $n = LMb_v$ and $\mathbf{y} = [\mathbf{y_0}, ..., \mathbf{y_{n-1}}]$ represent codeword length and received codeword. The brief process will be illustrated below.

Step 1: Initialization. Make some transform of variable nodes at time $t \quad (0 \leq t \leq L-1)$ and save as initialized information of variable nodes.

$$\gamma_i = \ln \frac{P(x_{i=0} \mid y_i)}{P(x_{i=1} \mid y_i)} \tag{4.5}$$

Step 2: Iteration. Only exchange and update information for variable nodes and check nodes in the window. The following are updating formulas of the check nodes and variable nodes:

$$R_{ij}^l = 2 \operatorname{artanh}^{-1} \left(\prod_{k \in N(i) \backslash j} \tanh \left(Q_{ik}^{l-1} / 2 \right) \right) \tag{4.6}$$

$$Q_{ij}^l = \begin{cases} \gamma_j, & l = 0 \\ \gamma_j + \sum_{k \in M(j) \backslash i} R_{kj}^l, & l > 0 \end{cases} \tag{4.7}$$

where l denotes the iteration times, R_{ij}^l denotes a posterior probability log-likelihood ratio from check nodes i to variable nodes j at lth iteration. $N(i) \backslash j$ represents the set of all variable nodes connected to check nodes i except j, Q_{ij}^l denotes a posterior probability log-likelihood ratio from variable nodes j to check nodes i at

lth iteration, and $M(j)\backslash i$ denotes the set of all check nodes connected to variable nodes j, except i.

Step 3: Hard decision and terminate the iteration.

$$Q_j^l = \gamma_j + \sum_{k \in M(j)} R_{kj}^l \tag{4.8}$$

When $Q_j^l \geq 0$, $x_j = 0$, otherwise $x_j = 1$. If the bit error rate of the current window target symbol is zero or the number of iterations reaches the maximum, the window stops iterating and outputs the judgment result; otherwise, return to Step 2 to continue the iteration.

Step 4: Window sliding. After the decoding window finishes decoding the current target symbol, save it and prepare for the decoding of the next window. When the window slides to the position of $t + 1$, return to Step 1 to decode the next target symbol.

4.3 Simulation results for SC-LDPC codes with FTN

In this section, we simulate the SC-LDPC code with FTN with different τ compared with the LDPC code without spatial coupling shown in Chapter 3. What's more, we change various code iterations (in the LDPC decoder part) to see the influence of the different number of iterations.

The simulated BER performance with QPSK modulation with/without spatial coupling is given in Figure 4.3. Two groups of curves mainly show the waterfall region performance with different τ, at the constant rate of $R = 1/2$. According to [4], the BER performance is presented as a function of $\mathrm{E}_b/N_0 = Es/(N_0\eta)$, where Es is the received signal energy per received signal, and N_0 is two-side noise power spectral density and

$$\eta = R \times \log_2 M/\tau \tag{4.9}$$

denotes the spectral efficiency in bits/s/Hz, R is the error correcting code rate, M is the constellation size, i.e., M = 4 for QPSK and $\tau = T_\triangle/T$ ($T_\triangle$ is the sinc pulses sent period) also so-called time-squeezing factor or the relative time period packing. In the simulations, we simulate the BER performance of Nyquist signaling ($\tau = 1$) as a comparison.

To control variables, we use the rate 1/2 LDPC codes with QPSK modulation of length N = 64800 bits from the DVB-S2 standard under the Nyquist system. The number of iterations in the decoder is 80 and the number of simulations is 10000, it is shown in Figure 4.3 as the black plus line. The spectral efficiency is η = 1 in this case. And the BER has a sharp decrease in E_b/N_0 belongs to [0.65, 0.9]. Then three dotted curves use the same parameters as a comparison group, where different τ is chosen to provide different spectral efficiency. With code rate R = 1/2, τ = 0.74, 0.67, 0.5 corresponding to η = 4/3, 3/2 and 2 bit/s/Hz respectively.

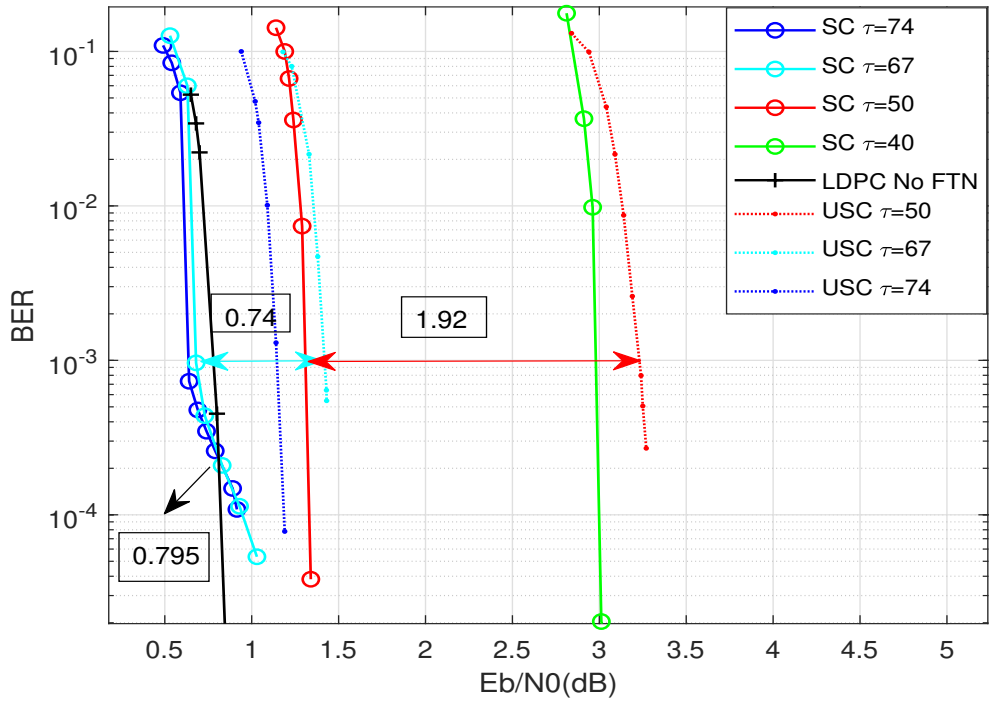

Figure 4.3: Simulation results for LDPC codes with or without SC

Furthermore, from Figure 4.3, the four solid circle curves show the performance of spatially coupled LDPC codes with FTN signaling and turbo equalization. And the iterative equalization decoding process is followed the Figure 4.4 [25]. Here, $La(x)$ is the prior information of the transmitter source. In the first iteration, the prior information is initialized first. For the regular system, since the input signal is equally distributed, the corresponding prior information LLR is set to LLR = 0, $Lc(x)$ is the inherent information output by the FTN equalizer, and $Le(x)$ is the external information output by the FTN equalizer, $Lh(x)$ is the external information output by the channel decoder, and $L(I)$ is the LLR of the final received bit after the decision.

In a complete iteration, the FTN equalizer uses the BCJR algorithm to calculate the posterior probability according to the channel reception sequence and then outputs the external information of the equalizer. Then the external information is deinterleaved as a priori information of the SC-LDPC decoder followed by the decoding and outputs of the inherent information and external information of the channel decoder. The last step is the external information output by the SC-LDPC decoder which is interleaved and sent to the equalizer.

The simulation results for SC-LDPC codes with QPSK modulation and FTN signaling ensembles are shown in Figure 4.3. We use the bit error rate (BER) to

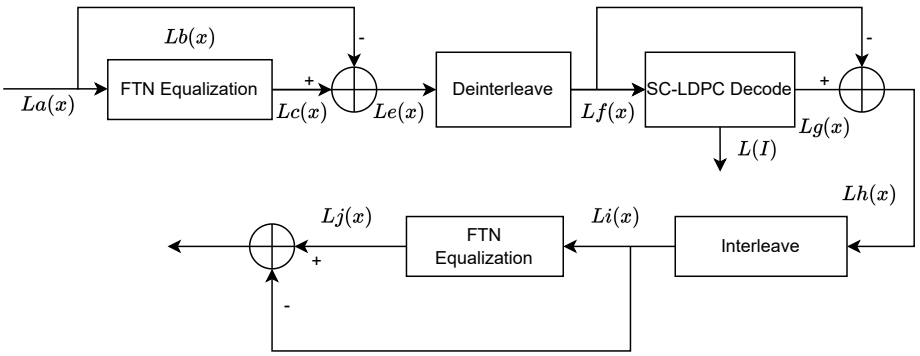

Figure 4.4: Iterative equalization decoding

compare the results. Signal to noise ratio (SNR) is given by E_b/N_0, where E_b is the average energy per bit. The DVB-S2 standard LDPC codes have code length $N = 1944000$ with a total code rate of $R = 0.495$ in the simulation.

For the FTN system part, the raised-cosine pulses h(t) in (2.6) with roll-off factor $\alpha = 0.4$ is used. Different time acceleration factor $\tau = 0.74, 0.67, 0.5, 0.4$ are simulated separately. We choose a suitable channel length 4 followed that used in [4] and employ the AWGN channel in the simulation as Chapter 3 introduced. The number of iteration equalization is set to 10. The infinitely extended parity check matrix $\mathbf{H}$ is terminated by the zero termination method, that is $\mathbf{H}$ is truncated at $t = L$, where L is the coupling length. We choose the partial coupling length $L_1 = 25$ of the whole length $L = 100$. Spatially coupled LDPC code can also be classified into strong coupling and weak coupling. Generally, weak coupling refers to coupling memory $w = 1, 2$, here we use $w = 1$, and the base matrix B is divided into B_0 and B_1, which are shown in (4.10) and (4.11). The lifting factor M is set to 972. The sliding window decoding algorithm with window size $W = 5$ is used in SC-LDPC decoding.

$$\mathbf{B_0} = \begin{bmatrix} 0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0 \\ 0&1&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0 \\ 1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1 \end{bmatrix} \tag{4.10}$$

$$\mathbf{B_1} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.11}$$

As shown in Figure 4.3, some significant gains can be obtained after employing the SC-LDPC code with turbo equalization and utilizing window decoding than that without spatial coupling in Chapter 3.

From the overall trend, we can find significant gains after using SC-LDPC in their waterfall regions compared with those with the same τ but without spatial coupling. Compare the SC-LDPC codes with $\tau = 0.74$ with that in no spatial coupling ensemble, when the BER is equal to 10^{-3}, it is clear that there is a 0.51dB gain with spatial coupling, but along with the slightly increasing SNR, at around 0.645dB, the slope begins to slow down which depicts the relatively worse error floor compared with that without spatial coupling (which means it hit the error floor at a higher error rate than non-SC codes) and $\tau = 0.74$. It is because the window size is just 5, not big enough, as [26] mentions, within the appropriate range, the bigger window size will bring better decoding performance. Similarly, it can be seen from other comparison groups, such as at BER = 10^{-3} when $\tau = 0.67$, the performance improves by about 0.74dB, and when $\tau = 0.5$, there is an apparent gain of about 1.92dB. Even using SC-LDPC codes with $\tau = 0.4$ can get around 0.26dB compared with that without spatial coupling at BER = 10^{-3} and τ = 0.5.

What's more, it is noticeable that at the same BER level, while the τ decreases from 0.74 to 0.5, the spatial coupling gain will be more and more obvious from the lower SNR region to the higher SNR region. Meanwhile, the error floor performance also improves. It can be explained that by using the regarded ISI channel, the smaller τ refers to more severe ISI. The spatial coupling has a property of threshold saturation, which can improve the BP threshold by shifting it close to the MAP threshold. Therefore, more severe ISI can obtain better gain by using spatial coupling.

Some other interesting finding occurs. Taking account of the SC-LDPC codes with $\tau = 0.67$, the BER performance is almost as good as that with $\tau = 0.74$ when SNR changes from 0.5dB to 0.74dB, and the small gain keeps almost constant (0.042 dB), but when SNR exceeds 0.74dB, these two curves almost coincide. On the other hand, compared with the curve using a regular Nyquist system without FTN, SC-LDPC codes with $\tau = 0.74$ and $\tau = 0.67$ implemented with FTN signaling and turbo equalization has a better performance gain before SNR = 0.795dB,

but be surpassed after that point. It can be concluded that when keeping other parameters constant, implementing the SC-LDPC codes with window decoding has more improvement than that without spatial coupling. And the superiority of SC-LDPC codes gets more obvious with lower τ case.

From another perspective, considering the system spectral efficiency as equation (4.9) introduced. In this report ensemble, $\tau = 0.74, 0.67, 0.5$ corresponds to system spectral efficiency $\eta = 4/3,\ 3/2,\ 2$ respectively. From Figure 4.3, with the same SE versus E_b/N_0, the SC-LDPC code can achieve the same SE with less energy consumption than using the same τ without using spatial coupling. What's more, considering the two curves with SNR above 3dB, for QPSK-based systems by applying FTN signaling with the DVB-S2 SC-LDPC codes we can achieve an even higher SE than that without SC.

Like what we did in Figure 3.8, the curves of efficiency and SNR in the case of BER $= 10^{-4}$ when using coupled code can be drawn. For all the uncoupled curves, the points starting from the left side to the left correspond to $\tau = 1$, 0.74, and 0.67 respectively and for coupled curve, the points from the left to the right correspond to $\tau = 0.74$, 0.67, 0.5 and 0.4 respectively. From Figure 4.5, it can be seen that all the points move to the left side compared with the uncoupled case. That means when the system operates at the same efficiency, a significant gain can be obtained by using coupled codes. When we take $R = 1/2$ and $\tau = 0.74$ for coupled and uncoupled points,it can be seen that there's 0.7dB gain. Especially when spectral efficiency $\eta = 2$, there is 0.9dB and 2.2dB gain for $R = 2/3$ and $3/4$ respectively by using coupled codes.

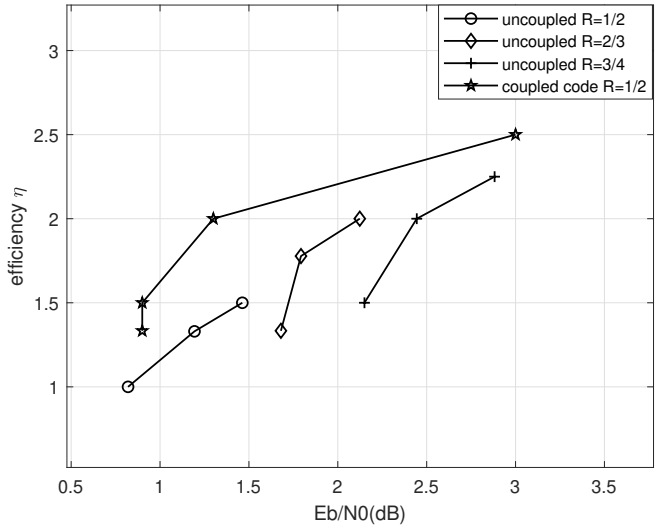

Figure 4.5: Comparison of efficiency performance for uncouple and coupled codes

Chapter 5
Conclusion and future work

5.1 Conclusion

The main work of this thesis can be divided into two parts. Firstly, we take [4] as a reference, the simulation based on the QPSK format with convolutional codes and LDPC codes separately is introduced under the FTN system when the spectrum efficiency (η) changes, and the Nyquist system without FTN is also simulated as a comparison. Secondly, spatially coupled LDPC codes were used with a window decoding algorithm to combat the trade-off between the error floor and waterfall performance caused by system iteration.

The simulation results prove that under the condition of equal simulation parameters, the convolutional code outperforms the LDPC code in the lower SNR region, but along with the BER increasing, the stronger LDPC codes have better waterfall performance even though in the high SNR region, the convolutional codes have better error floor than the LDPC code. And higher spectral efficiency can be obtained by using smaller τ or increasing R, which also needs more power to reach the same BER level.

The combination of spatial coupling can effectively eliminate performance gaps, which means we can get the higher SE with the lower SNR requested. Take $\tau = 0.74$ and BER = 10^{-3} as an example, using spatial coupling codes has a total gain of about 0.51dB compared with that without a spatial coupling scheme, moreover, this gain will be more significant within higher SE system. And these findings are suitable for future communications with high rates and high spectral efficiency required.

5.2 Future work

This thesis investigates spatially coupled FTN signaling based on LDPC codes and achieves performance gains in QPSK-modulated FTN systems. With the deepening of research, the further steps can be carried out in the following aspects:

1. In this article, the single-carrier FTN receiver is designed and simulated with QPSK modulation. When a higher-order mapping method is used, which is very sensitive to channel noise, the design of the receiver faces greater challenges. How to reduce the influence of noise when using higher modulation? What methods can be taken at the transmitter and receiver?

2. This paper adopts the Turbo equalization of the MAP algorithm, while there are some other equalization techniques. From the perspective of algorithm complexity, the MAP algorithm has a high implementation complexity. Other equalization algorithms with lower complexity, such as MMSE equalizer, can be studied.

3. This paper mainly studies the single-carrier FTN transmission system. The multi-carrier FTN transmission system can be further studied. For example, the MIMO multi-carrier FTN transmission system can be investigated [27].

4. Further spatial coupling methods such as by introducing memory between the output of a code and the channel, and by introducing memory between the outer and inner code then acting a componet code in turbo equalization combined with FTN can be investigated.

5. Some other improvements can be done in the decoding part. Firstly, the Residual Belief-Propagation (RBP) algorithm and the Node-Wise RBP (NWRBP) algorithm can be combined effectively into the simulation. Whatsmore, simulation can be done by optimizing the regular window decoding algorithm, such as using dynamic and adaptive window size decoding.

References

[1] J. B. Anderson, F. Rusek, and V. Öwall, "Faster-than-Nyquist signaling," *Proceedings of the IEEE*, vol. 101, no. 8, pp. 1817–1830, 2013.

[2] M. Tüchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 920–952, 2011.

[3] M. M. Mashauri and M. Lentmaier, "Spatial coupling in turbo equalization," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[4] J. Yu, J. Park, F. Rusek, B. Kudryashov, and I. Bocharova, "High order modulation in faster-than-Nyquist signaling communication systems," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. IEEE, 2014, pp. 1–5.

[5] J. E. Mazo, "Faster-than-Nyquist signaling," *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1451–1462, 1975.

[6] A. D. Liveris and C. N. Georghiades, "Exploiting faster-than-Nyquist signaling," *IEEE Transactions on Communications*, vol. 51, no. 9, pp. 1502–1511, 2003.

[7] N. Ul Hassan, M. Lentmaier, and G. P. Fettweis, "Comparison of LDPC block and LDPC convolutional codes based on their decoding latency," in *2012 7th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*. IEEE, 2012, pp. 225–229.

[8] S. Wang, "Research on strongly coupled LDPC convolutional codes based on FTN system," 2021.

[9] P. M. Olmos, D. G. Mitchell, D. Truhachev, and D. J. Costello, "Improving the finite-length performance of long SC-LDPC code chains by connecting consecutive chains," in *2014 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*. IEEE, 2014, pp. 72–76.

[10] M. H. Tadayon, A. Tasdighi, M. Battaglioni, M. Baldi, and F. Chiaraluce, "Efficient search of compact QC-LDPC and SC-LDPC convolutional codes with large girth," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1156–1159, 2018.

[11] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2303–2320, 2011.

[12] H. Y. J. J. Q. W. Xin SU, Sen WANG, "Faster-than-nyquist signaling: value and challenges of 6G-oriented applications," *Telecommunications Science*, vol. 37, no. 9, pp. 38–47, 2021.

[13] F. Rusek, M. Loncar, and A. Prlja, "A comparison of Ungerboeck and Forney models for reduced-complexity ISI equalization," in *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*. IEEE, 2007, pp. 1431–1436.

[14] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: principles and new results," *IEEE transactions on communications*, vol. 50, no. 5, pp. 754–767, 2002.

[15] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on information theory*, vol. 27, no. 5, pp. 533–547, 1981.

[16] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 2181–2191, 1999.

[17] L. Deng, "Design and application of LDPC encoding and decoding algorithms," Ph.D. dissertation, University of Electronic Science and Technology of China, 2021.

[18] K. Deka, A. Rajesh, and P. Bora, "An improved technique to find the trapping sets of the irregular LDPC codes," in *2013 National Conference on Communications (NCC)*. IEEE, 2013, pp. 1–5.

[19] M. Tüchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 920–952, 2011.

[20] M. M. Mashauri, "Spatially coupled codes in turbo equalization," 2019.

[21] Y. Z. et al., "Performance of spatially coupled LDPC codes in AWGN channel," 2020.

[22] S. Moloudi, M. Lentmaier, and A. G. i Amat, "Spatially coupled turbo-like codes," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6199–6215, 2017.

[23] H. Z. et al., "Size adaptive windowed decoding for spatially coupled LDPC codes," 2019.

[24] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2303–2320, 2011.

[25] J. Xie, S. Wu, and Y. Jia, "Probabilistic shaping combined with spatially-coupled LDPC code in FTN system," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2020, pp. 1005–1009.

[26] I. Ali, J.-H. Kim, S.-H. Kim, H. Kwak, and J.-S. No, "Improving windowed decoding of SC LDPC codes by effective decoding termination, message reuse, and amplification," *IEEE access*, vol. 6, pp. 9336–9346, 2017.

[27] M. Yuhas, Y. Feng, and J. Bajcsy, "On the capacity of faster-than-Nyquist MIMO transmission with CSI at the receiver," in *2015 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2015, pp. 1–6.